

*Thesis submitted for the degree of
Doctor of Philosophy*

**The Joan Oró Telescope at the
Montsec Astronomical Observatory
Solutions for Unattended Operation**

by

Josep Colomé Ferrer

Advisor

Dr. Ignasi Ribas Canudas

Universitat de Barcelona

Institut de Ciències de l'Espai (CSIC-IEEC)

**Universitat de Barcelona,
Departament d'Astronomia i Meteorologia
Institut de Ciències de l'Espai (CSIC – IEEC)**

Biennium 2000-2002

**The Joan Oró Telescope at the Montsec
Astronomical Observatory
Solutions for Unattended Operation**

by

Josep Colomé Ferrer

A Thesis submitted for the degree of
Doctor of Philosophy

Advisor

Dr. Ignasi Ribas Canudas

Tutor

Dr. Jordi Torra Roca

Barcelona, September 2011

IEEC 



 UNIVERSITAT DE BARCELONA



CONSEJO SUPERIOR
DE INVESTIGACIONES
CIENTÍFICAS

The background image of the front and back covers shows a night view of the Joan Oró Telescope at the Montsec Astronomical Observatory, courtesy of E. Herrero. The back cover also depicts an anonymous drawing that illustrates the human discovery of the Universe.

Dedico aquesta Tesi
a l'Aïnhua i als nostres fills, l'Aniol i l'Artald

Thesis panel

The evaluation panel for this Thesis is composed by:

Prof. **Klaus Strassmeier** (Astrophysical Institut of Potsdam, Germany), panel chairman

Dr. **David Galadí Enríquez** (Centro Astronómico Hispano Alemán AIE, Spain), panel secretary

Prof. **Alberto J. Castro Tirado** (Instituto de Astrofísica de Andalucía – CSIC, Spain)

Agraïments

La Serra del Montsec és una serralada formada per tres grans massissos originats pel xoc entre plaques tectòniques i l'erosió de l'aigua de tres rius (Segre, Noguera Pallaresa i Noguera Ribagorçana) que el creuen de nord a sud. L'Observatori Astronòmic del Montsec s'ubica en el massís del Montsec d'Ares, situat al mig de les parts oriental (Montsec de Rubies) i occidental (Montsec d'Estall) de la serralada. Milions d'anys d'evolució han donat la forma ideal al Montsec per esdevenir un lloc privilegiat per l'activitat astronòmica a Catalunya. En aquest indret és habitual trobar fòssils de crustacis de més de 250 milions d'anys d'antiguitat (era del Mesozoic) entre d'altres restes geològiques, produint-se un cas paradoxal: l'observació del cel i del sòl des del mateix lloc ens permet veure el nostre passat i conèixer l'origen de tot allò que ens envolta. És per aquest motiu que el Montsec ha estat objecte d'estudi per part de persones de diferents disciplines científiques durant les últimes dècades. Una d'aquestes disciplines, l'astronomia, és la que em va portar a mi a mitjans de l'any 2001 fins aquesta terra, just després d'acabar la llicenciatura de Ciències Físiques i en motiu d'un estudi de la contaminació lumínica a Catalunya que realitzava el Departament d'Astronomia i Meteorologia de la Universitat de Barcelona. Aquest treball formava part d'un esforç conjunt per trobar un lloc adequat per construir un observatori i que va coincidir amb la iniciativa que diverses persones van emprendre per tal de fer palès al món la riquesa astronòmica del cel del Montsec, alhora que impulsaven la recuperació socioeconòmica d'aquest territori. En el pròleg d'aquesta tesi es pot llegir una nota històrica que descriu perfectament aquest context i que molt amablement va acceptar escriure el Sr. Xavier Palau, aleshores president d'ASTER (agrupació astronòmica de Barcelona) i representant de la Fundació Joan Oró en la comissió que ha assessorat el projecte en tota la seva fase de construcció.

El perfil del Montsec ha canviat des del moment en què es van començar a aixecar les parets de l'Observatori. Jo n'he pogut ser testimoni gràcies al fet d'haver-hi treballat des de l'any 2002 fins a l'actualitat vinculat a l'Institut d'Estudis Espacials de Catalunya i a l'Institut de Ciències de l'Espai - CSIC. A diferència de les altres ciències experimentals que han fruit d'aquest indret (geologia, paleontologia, arqueologia, etc.), l'observació dels astres requereix la instal·lació permanent d'instruments de precisió que destorben l'entorn i modifiquen els hàbits d'aquells que en gaudien. Com en d'altres ocasions, s'ha posat a prova l'equilibri entre el desenvolupament i la conservació del medi ambient per tal que es pogués dur a terme la recerca astronòmica. En aquest sentit, espero que la feina que hem fet en la seva construcció i, en particular, el treball realitzat en aquesta Tesi, hagi contribuït a donar valor a aquest esforç.

La construcció de l'Observatori ha significat un pas endavant per l'astronomia a Catalunya. En el moment en què es va plantejar el projecte, la construcció d'un telescopi robòtic representava un repte tecnològic, accentuat per les característiques ambientals extremes de la Serra del Montsec. Prova d'això n'és l'existència de pocs observatoris similars a tot el món i que es repassen al capítol introductor de la Tesi. És amb aquests observatoris que ens hem emmirallat sovint per saber quin era el nostre horitzó. I hem tingut la sort de poder col·laborar amb alguns d'ells: els grups de l'Institut d'Astrofísica de Potsdam i el de la Universitat de Queens a Belfast, responsables de la construcció dels telescopis STELLA i SuperWASP, respectivament, en són l'exemple més clar. De la seva mà hem conegut solucions innovadores i maneres de treballar que són l'herència de dècades d'aprenentatge en l'aplicació de la tecnologia al servei de l'astronomia; i les hem intentat aplicar al Montsec per tal d'assolir els objectius que ens havíem proposat, entre els quals hi havia la voluntat de tenir un instrument d'una alta qualitat, tan pel que fa a la seva operació sense intervenció humana com per la qualitat de les dades recollides. El treball d'aquesta Tesi ha pretès contribuir-hi de la forma més acurada i ambiciosa de la qual hem estat capaços. L'intercanvi d'experiències i

coneixements amb grups d'arreu del món perseguia també la projecció internacional del projecte amb el desig de poder participar en la construcció de nous telescopis, augmentant així la nostra experiència i el retorn científic cap a les institucions que hi havien participat. L'assoliment d'aquestes fites ja és una realitat i és mèrit de totes les persones que han participat directament o indirecta en la construcció del Telescopi Joan Oró.

El Montsec ha estat sempre un punt de referència en el meu recorregut professional, que comença amb el primer contacte amb el món de la recerca en el camp de l'astronomia i continua fins a la consecució d'aquesta tesi. En tot aquest temps han estat moltes les persones que han contribuït d'una manera o una altra a que jo assolís aquest objectiu. No puc anomenar-les a totes per no fer d'aquests agraïments una llista inacabable (i segurament incompleta) de noms, però si que els voldria donar les gràcies, encara que sigui amb la simplicitat d'una agrupació per ubicació geogràfica. Voldria esmentar, en primer lloc, a la mateixa gent del territori del Montsec, a qui no agrairé mai prou la generositat amb la què m'han deixat entrar a casa seva per conèixer la seva manera de fer, per treballar junts en la modificació del seu paisatge natural i social, i per gaudir de la seva terra (i de la majestuositat de la seva serralada!). Voldria agrair, també, als companys del Departament d'Astronomia de la Universitat de Barcelona que m'ajudessin en els meus primers passos en l'astronomia observacional, malgrat la curta estada al departament. El contacte amb grups d'altres països també ha estat de gran importància per mi, tant per dissenyar les solucions tecnològiques pel Montsec com per conèixer de primera mà la manera de treballar d'indrets on la recerca científica es considera una font de riquesa i coneixement ineludibles. I, finalment, vull destacar tota la gent de l'IEEC que he conegut en els gairebé deu anys que fa que hi treballa. D'ells he après el valor de la passió per aquest món i el privilegi que suposa fer una feina com la nostra. Durant aquests anys he compartit molt bones experiències i he fet bones amistats, la qual cosa em fa sentir una persona afortunada.

En aquest breu repàs, però, si que voldria fer esment explícitament de tres persones que han representat un punt d'inflexió en la meua carrera professional. El Jordi Torra, que fou qui em va donar la primera empenta per poder entrar a treballar a la Universitat de Barcelona. Allà vaig conèixer els vessants instrumental i medi ambiental de l'astronomia en un projecte que tenia per objectiu la reducció de la contaminació lumínica a Catalunya i que es duia a terme des del Departament d'Astronomia. El Jordi Isern, que m'ha empès en l'aprenentatge de les tècniques instrumentals des del moment en què vaig entrar a treballar a l'IEEC provinent de la Universitat. En molts moments, ha dipositat en mi una confiança que li agraeixo i que m'ha permès participar en projectes que, més enllà de la repercussió per l'Institut, han suposat un enriquiment per mi que difícilment hagués pogut tenir en altres àmbits. I, finalment, el Ignasi Ribas, que és el màxim responsable que aquesta Tesi sigui una realitat. Més enllà del seu estricte sentit crític que el porta a buscar l'excel·lència en tot allò que fa, té la capacitat d'empènyer a tothom que treballa amb ell per tal d'aconseguir el millor de cadascú alhora que tothom en tregui el màxim profit. Aquesta generositat ha omplert de coneixements el treball presentat en aquesta Tesi i ha significat, també, un gran benefici pel projecte del Montsec. La seva capacitat ha fet que actualment puguem exportar tot l'aprenentatge assolit i que puguem participar en nous projectes d'instrumentació astronòmica en llocs privilegiats d'arreu del món i col·laborant amb grups de renom internacional. Serveixin aquestes línies per expressar-los a tots tres el meu agraïment més sincer.

Acknowledgments

The Montsec is formed by three mountain ranges arising from the clash of tectonic plates and water erosion by three rivers (Segre, Noguera Pallaresa and Noguera Ribagorçana) that run from North to South. The Montsec Astronomical Observatory is located on the Montsec Ares mountaintop, between the Eastern (Montsec de Rubies) and Western (Montsec d'Estall) ranges. Millions of years of evolution have converted the Montsec into a privileged place for astronomical observation in Catalonia. In this region, it is common to find fossils of crustaceans over 250 million years old (Mesozoic era) and other geological remains, which give rise to a paradoxical situation: the observation of the sky as well as the ground from the same site allows us to view our past and discover the origin of all that surrounds us. For this reason, Montsec has been a place of study for researchers from different scientific disciplines in recent decades. One of these disciplines, astronomy, is what led me to this land in the midst of 2001, just after finishing a degree in Physics and while I was carrying out research on light pollution in Catalonia by the Department of Astronomy and Meteorology of University of Barcelona. This research was part of a comprehensive effort to find a suitable location for the construction of an observatory. This was being done through an initiative to show the great astronomical potential of the Montsec site as well as to promote the social-economical growth of the region.

The Thesis preface is written by Mr. Xavier Palau, at that time president of the Astronomical Association of Barcelona (ASTER), who has perfectly described the historical context of the project during those years. Mr. Palau is also member of the Fundació Joan Oró and part of the consulting committee that supervised the project throughout its execution.

Montsec's silhouette has changed since the walls of the Observatory began to rise. I have been a witness of this transformation since I have been working in the project from the 2002 as member of the Institute of Space Studies of Catalonia and the Institute of Space Sciences - CSIC. Unlike other experimental sciences that have taken advantage of this site (geology, paleontology, archeology, etc.), the observation of the cosmos requires the permanent installation of instruments that inevitably disturb the environment and change the habits of those who inhabit and enjoy it. As in other occasions, a balance between development and conservation of the environment has been a challenge to make astronomical research feasible in this region. In this respect, I hope the work that has been done to build the observatory, and in particular the work carried out in this Thesis, has made this effort worthy.

The construction of the Observatory has meant a step forward for Astronomy in Catalonia. When the project to build a robotic telescope was presented, its construction posed a technological challenge, emphasized even more by the extreme environmental characteristics of the Montsec range. The existence of very few observatories worldwide, mentioned in the introduction of the Thesis, provides a clear proof. We have used these observatories as a guide and have been extremely fortunate to collaborate with several groups such as those of the Astrophysical Institute of Potsdam and the Queens University in Belfast, responsible for the construction of the STELLA and SuperWASP telescopes, respectively. Thanks to their help, we have implemented innovative solutions and working procedures that are a heritage of decades of learning, searching for the right ways of having technology serve Astronomy. We have implemented these at the Montsec to fulfill our objectives, mainly aiming at having a high-quality instrument capable of operating without human intervention and capable of collecting high-quality data.

The work carried out in this Thesis was meant to be as thorough and ambitious as possible. The exchange of experiences and knowledge with groups around the world was also achieved with the aim of having projection and recognition worldwide. Such recognition would permit us to participate in the construction of new telescopes, thus increasing our scientific expertise and providing a scientific return to the participating institutions. Achieving these goals is a reality and credit should be given to all of those who have participated, directly or indirectly, in bringing the Joan Oró telescope to reality.

Montsec has always been a landmark in my professional career, which begins with my first contact with research in the field of astronomy and continues until the completion of this Thesis. During all these years there have been many people who have contributed in one way or another to help me reach this goal. I cannot name them all since the list would be endless (and probably incomplete), but I would like to thank them all, even by grouping people in terms of geographical location. First, I would like to mention the people from Montsec, with whom I cannot be grateful enough for their infinite generosity by inviting me to their community and allowing me to appreciate their way of doing, to work together in modifying their natural and social landscape, and to enjoy their land (and the majesty of their mountains!).

I would also like to thank all the colleagues of the Department of Astronomy and Meteorology, of the University of Barcelona, who helped me enter the world of observational astronomy, despite my short stay in the department. Contact with groups from other countries has also been very important for me,

both to acquire technological solutions for the Montsec facility and to learn at firsthand how to work in places where scientific research is considered a source of wealth and unavoidable knowledge. Finally, I mention all the people of the Institute that I have met in the nearly ten years I have worked there. From them I have acquired the passion for this field and felt privileged to have a job like this. Over the years I have shared very good experiences and I made good friends, for which I feel a very lucky person.

In this brief review, however, I would explicitly like to mention three people who have had a strong impact on my career. Jordi Torra, who gave me the first opportunity to work at the University of Barcelona. There, I was able to acquire valuable knowledge on the instrumental and environmental aspects of astronomy in a project aimed at reducing light pollution in Catalonia. Jordi Isern, who pushed me to learn instrumental techniques right after I began working at the IEEC. On many occasions, he has placed his trust in me, which I am thankful for, and which has allowed me to participate in projects that, beyond the Institute's scientific enrichment, have been an unforgettable and nurturing experience for me. Finally, Ignasi Ribas, who has brought this Thesis to reality. Beyond his strict and critical sense of seeking excellence in everything he does, he has the ability to push everyone who works with him to get the best out of themselves while obtaining the maximum return. This generosity has filled of knowledge the work presented in this Thesis and has also been of great benefit for the Montsec project. His capacity has allowed us to export all this acquired knowledge and to participate in new astronomical instrumentation projects that are being developed in privileged sites around the world as well as working with renowned international groups. These lines are written to express to all three my warmest and most sincere gratitude.

(English translation by A.Hirschmann)

Pròleg

El 23 d'octubre del 2008 fou inaugurat per l'honorable Conseller d'Innovació Universitats i Empresa, Josep Huguet, l'Observatori Astronòmic del Montsec que aixopluga un telescopi de 80 cm de diàmetre, amb el nom de Joan Oró i que l'amic Josep Colomé ha contribuït decididament a convertir-lo en un dels telescopis robòtics més avançats del territori europeu, tal i com queda palès en la tesi que presenta.

L'origen del telescopi és la síntesi de dues fascinacions, la del professor Joan Oró i la de dos aficionats a l'astronomia, Jordi Sentenach i la meva personal.

Esdevingut un dels grans científics catalans als Estats Units, investigador de bioquímica, més concretament en l'origen de la vida, col·laborador de la NASA, professor emèrit de la Universitat de Houston i autor de més de 30 llibres i uns 350 articles científics, Joan Oró (1923-2004) ha estat sempre un convençut lluitador per apropar la recerca catalana a l'òrbita americana.

Per la meua part, m'interesso pels astres arran de la sèrie televisiva "Cosmos" de Carl Sagan, emesa cap als anys 80 del segle passat. L'any 1993, essent president de l'Agrupació Astronòmica ASTER de Barcelona i amb el suport del bon amic Jordi Sentenach, vàrem pensar amb la possibilitat d'instal·lar un telescopi a la serra del Montsec amb l'objectiu de tenir un lloc d'observació permanent en un indret amb baixa contaminació lumínica i excel·lent transparència atmosfèrica. Així l'any 1994 vaig trucar a Joan Oró per buscar un suport sòlid a la nostra idea.

Joan Oró va manifestar el seu suport incondicional i ens incorpora a mi i a la idea en el sí de la Fundació que porta el seu nom, Fundació Joan Oró, per tal de convertir-la en l'objectiu principal de la mateixa.

Amb la persistència que el caracteritzava, Joan Oró traslladà la idea convertida en projecte alaleshores president de la Generalitat de Catalunya, l'honorable Jordi Pujol. El projecte contemplava la construcció d'una infraestructura amb 3 eixos, la recerca, l'educació i la divulgació de l'astronomia i volia, a més, contribuir decididament al desenvolupament socioeconòmic d'un territori que, malgrat el seu patrimoni científic i cultural, estava lluny del pensament de les autoritats polítiques nacionals potenciar-lo.

No exempt de moltes dificultats, finalment l'any 2000 la Generalitat inclou el projecte en el sí del programa "Montsec Sostenible" i s'aconsegueix el finançament per la creació del que és avui el Parc Astronòmic del Montsec, format per l'Observatori Astronòmic del Montsec (l'eix de recerca), i el Centre d'Observació de l'Univers (els eixos educatiu i divulgatiu), inaugurats els anys 2008 i 2009, respectivament.

Desgraciadament en Joan Oró no va veure el projecte fet realitat, ja que va morir el setembre de l'any 2004.

Reconeixent la seva decisiva contribució a fer realitat la idea inicial, la Generalitat de Catalunya va decidir posar el nom de Joan Oró al telescopi de l'observatori del Montsec, sens dubte ben merescudament.

Estic convençut que el telescopi Joan Oró i la resta de infraestructures del Parc Astronòmic del Montsec contribuiran decisivament al desenvolupament de l'astronomia a casa nostra, tal i com el professor desitjava i, a més, ajudaran a la potenciació de les comarques del territori.

Vull agrair al científic Josep Colomé que hagi permès posar un toc d'història humana en la seva tesi, indiscutiblement de gran alçada tècnica, i exposar, de forma breu, l'origen d'un somni fet realitat.

Xavier Palau Molins

Fundació Joan Oró

Barcelona, 14 de setembre de 2011

Prologue

On October 23rd of 2008 the maximum authority of the Catalan government for Innovation, Universities and Industries, Josep Huguet, inaugurated the Montsec Astronomical Observatory. This observatory houses a telescope of 80 cm in diameter, named after the famous biochemist Joan Oró. The work and effort of my friend Josep Colomé has contributed decisively to convert it in one of the most advanced robotic telescopes of the European territory, as it is evidenced in this research Thesis.

The construction of the telescope and its facility are actually the fulfillment of the dreams of Professor Joan Oró and two amateur astronomers, Jordi Sentenach and myself, Xavier Palau.

Joan Oró (1923-2004) became one of the greatest Catalan scientists ever known. As a biochemist researcher mainly devoted to understanding the origin of life, he resided in the USA, collaborated with NASA, was named professor emeritus by the University of Houston, and also wrote more than 30 books and 350 scientific articles. Throughout his life he fought convincingly to convey and bring the Catalan research into the American orbit.

I became interested in astronomy through the television show "Cosmos" by Carl Sagan broadcasted in the 80's. In 1993, as president of the Astronomical Association of Barcelona (ASTER) and with the support of my friend Jordi Sentenach, we thought about the possibility of installing an observatory at the Montsec range. The objective of this project was to have a permanent astronomical observational location with very little light pollution and excellent atmospheric and weather conditions.

Therefore, in 1994 we contacted Joan Oró to see whether or not he could also support our intentions. He expressed his unconditional support to this initiative and converted this project into one of the main objectives of the Joan Oró Foundation.

With the persistency that characterized Joan Oró, he presented this project to the president of the Generalitat de Catalunya at that time, Jordi Pujol. The project contemplated the construction of a facility with 3 aims: research, education and astronomy public outreach. He also wanted this facility to contribute to the socioeconomic development of a territory that, despite its magnificent scientific and cultural heritage, was far from being included in national political party agendas for its promotion.

In spite of all the difficulties encountered, in the year 2000 the Generalitat finally approved the project "Sustainable Montsec" and obtained the funding for the construction of Parc Astronòmic del Montsec. This park is formed by the Astronomical Observatory of Montsec (the research branch) and the Center of Observation of the Universe (the educational and outreach branch). These facilities were inaugurated in 2008 and 2009 respectively.

Unfortunately, Joan Oró was not able to see the project materialize since he died in September of 2004. In recognition to his unmeasurable contribution in making this initiative a reality, the Generalitat decided to name the telescope of the observatory after him.

I am more than convinced that the telescope Joan Oró and the rest of facilities of the Parc Astronòmic del Montsec will definitely contribute to the development and promotion of Astronomy in our territory, as the professor desired. Evenmore, it will also help promote and strengthen the surrounding regions of the province.

I would like to thank the scientist Josep Colomé for letting me include a touch of human history to his Thesis, of advanced technical level, and allow me to explain, briefly, the origin of a dream that has become reality.

Xavier Palau Molins

Fundació Joan Oró

Barcelona, 14th of September, 2011

(English translation by A.Hirschmann)

Table of Contents

Resum de la Tesi	15
Chapter 1 Introduction	27
1.1 Motivation of the Thesis	27
1.2 TJO – OAdM General Description	31
1.3 Thesis structure	33
Chapter 2 Qualified Robotic Control at the TJO-OAdM	35
2.1 Historical perspective	35
2.2 TJO-OAdM operational concept	38
2.3 Robotic control requirements	40
Chapter 3 System Architecture Design	47
3.1 Operational design	47
3.2 Architecture conceptual design	58
3.3 Logical architecture	59
3.4 Physical architecture	61
Chapter 4 Physical Architecture Implementation	65
4.1 Site areas	65
4.2 Telescope - Joan Oró Telescope (TJO)	66
4.3 Dome	68
4.4 MEIA Instrument	74
4.5 Housekeeping	78
4.6 Communications	83
4.7 Subsystem computers	85
Chapter 5 Robotic Observatory Control System	87
5.1 Main control layers	87
5.2 Housekeeping	88
5.3 End-to-end data flow	92
5.4 OpenROCS	96
Chapter 6 OpenROCS	121
6.1 OpenROCS1.0, the first prototype	121
6.2 OpenROCS2.0	126
Chapter 7 DDROCS Database	137
7.1 Database analysis	137
7.2 Database design	138
7.3 Database implementation	142

Chapter 8 Observation Preparation	149
8.1 Tasks considered and insertion process	149
8.2 Science task preparation	149
8.3 Calibration tasks	155
Chapter 9 ISROCS, the Observatory Time Scheduler	159
9.1 Scheduling task overview	159
9.2 Scheduler analysis	160
9.3 Scheduler design	163
9.4 Scheduler detailed design	166
9.5 Scheduler implementation	182
Chapter 10 Task Execution Control	187
10.1 Analysis and design	187
10.2 Implementation	189
Chapter 11 ICAT for Data Processing	193
11.1 Data processing analysis	193
11.2 Data processing design	196
11.3 ICAT detailed design	200
11.4 ICAT implementation details	222
Chapter 12 Observatory User Interface	243
12.1 User interface analysis and general design	243
12.2 User interface detailed design and implementation	247
Chapter 13 Quality Assurance	257
13.1 Quality assurance and qualified robotic operation	257
13.2 Risk management	258
13.3 Verification and validation	262
13.4 Testing	267
13.5 Maintenance	274
13.6 Configuration management	275
Chapter 14 Conclusions	277
14.1 The TJO-OAdM in the context of the robotic telescopes	277
14.2 Contributions of the Thesis to the TJO-OAdM robotization	278
14.3 On-going and future work	281
14.4 Gained expertise and new projects	282
References	285
Lists of Figures, Equations and Tables	289
List of Acronyms and Abbreviations	295
Annex 1: Test Procedure, Running Step by Step	299

Resum de la Tesi

(Summary of the Thesis in Catalan)

Aquesta Tesi descriu el treball realitzat per l'autor en la robotització del Telescopi Joan Oró de l'Observatori Astronòmic del Montsec (TJO- OAdM¹). Es presenta per optar al grau de Doctor.

Motivació de la Tesi

El projecte de construcció del telescopi TJO-OAdM es va iniciar a mitjans dels anys 90, quan el Govern català va aprovar el projecte Montsec Sostenible, iniciativa presentada per promoure la zona del Montsec a Catalunya. Aquest preveia la construcció d'una nova instal·lació de divulgació científica (el Centre d'Observació de l'Univers) i un observatori per a la recerca astronòmica (l'Observatori Astronòmic del Montsec, OAdM), equipat amb el telescopi més gran de Catalunya i preparat per a ser operat de forma totalment desatasa.

La definició inicial del projecte TJO-OAdM incloïa una especificació genèrica de requeriments i basava la construcció de l'observatori, principalment, en l'adquisició d'un telescopi que suposadament tenia un funcionament totalment robòtic. No obstant això, múltiples deficiències van impedir un funcionament completament desatès i robust de l'observatori un cop va finalitzar la instal·lació del telescopi el mes de març de 2004. El projecte va haver de ser modificat posteriorment i se'n va elaborar un de nou (Colomé et al. 2006) per completar la robotització de l'observatori. Aquest projecte és referit en tot el text com a *projecte de robotització* i va ser aprovat l'any 2007 per la Generalitat de Catalunya sota la direcció del Dr Ignasi Ribas, i es va estendre fins el 2010. El treball presentat en aquesta tesi s'ha dut a terme en el marc d'aquest projecte i ha contribuït a la consecució dels seus objectius.

Descripció general del TJO – OAdM

Generalitats

L'Observatori Astronòmic del Montsec (OAdM) és un projecte finançat per la Generalitat de Catalunya i és propietat del seu ens local oficial, el Consorci del Montsec (CdM). L'Institut d'Estudis Espacials de Catalunya (IEEC) és el responsable de la seva gestió científico-tècnica. Diversos instruments de recerca estan instal·lats en el recinte d'aquest observatori: el telescopi TJO, gestionat per l'IEEC; el Telescopi Fabra-ROA al Montsec (TFRM) de la Reial Acadèmia de Ciències i Arts de Barcelona i el Reial Observatori de l'Armada; una càmera d'observació de tot el cel, també de l'IEEC, dedicada a la detecció de meteors i bòlids; una estació de monitoreig de la concentració de partícules en l'aire (contaminants) propietat del govern català (Departament de Medi ambient); i una estació meteorològica del Servei Meteorològic de Catalunya (SMC, govern català) que és un dels nodes de la xarxa d'estacions existent a Catalunya destinada a contribuir en l'estudi i previsió meteorològiques.

Diverses institucions i universitats han contribuït activament amb la supervisió científico-tècnica i amb personal especialitzat al desenvolupament del TJO-OAdM: l'IEEC², la Universitat de Barcelona (UB³),

¹ <http://www.oadm.cat>

² <http://www.ieec.cat>

³ <http://www.am.ub.es>

la Universitat Politècnica de Catalunya (UPC¹), el Consell Superior d'Investigacions Científiques (CSIC²) i la Fundació Joan Oró (FJO³).

Descripció de l'emplaçament

L'OAdM es troba situat a una alçada de 1570 m per sobre del nivell del mar a la Serra del Montsec, a 50 km al sud dels Pirineus centrals i a 50 km al nord de la ciutat de Lleida (Catalunya, Espanya). Durant els anys 1997-1999 es van dur a terme campanyes de mesura de la qualitat del cel, amb la mesura de l'estabilitat de l'aire, la transparència atmosfèrica i les condicions meteorològiques. Aquestes proves van mostrar que el cel nocturn de la Serra del Montsec és adequat per a realitzar observacions astronòmiques d'alta qualitat. La contaminació lumínica a l'OAdM també es va trobar que era molt baixa, amb una brillantor del cel nocturn en el zenit de $22.0 \text{ arcsec}\cdot\text{mag}^{-2}$ a la banda visual (V) de Johnson.

Les dades meteorològiques i d'estabilitat atmosfèrica es monitoren de forma continua des de l'any 2004 (meteorologia) i des del 2009 (estabilitat) i la seva avaluació confirma la permanència de les bones condicions per a l'observació. Les mesures de l'estabilitat atmosfèrica (seeing, en anglès) es van dur a terme l'any 2009 mitjançant un RoboDIMM (RoboDIMM, Aceituno 2004), aparell dissenyat especialment per aquesta finalitat, i van mostrar un valor de 0.86 segons d'arc de mediana per cinc mesos de mesures i que es reduïa fins a 0.7 segons d'arc durant un 32% del temps. Actualment, s'està preparant una nova avaluació de les dades meteorològiques en base a sis anys de mesures (2004-2010) proporcionades pel SMC.

Instrumentació

El TJO-OAdM està equipat amb el Telescopi Joan Oró (TJO) de 0.8 m de diàmetre subministrat per Optical Mechanics Inc. (OMI) i instal·lat el mes de març de 2004. La cúpula té 6.15 m de diàmetre va ser fabricada per Baader Planetarium GmbH i disposa d'un funcionament completament automàtic. La instrumentació es compon d'una càmera CCD ProLine fabricada per Finger Lakes Inc., amb un xip Marconi CCD42-40 il·luminat per darrera i format per una matriu de 2048×2048 píxels de 13.5 micres. El camp de visió del sistema és de 12.4×12.4 minuts d'arc. El telescopi està equipat amb una roda de filtres de 12 posicions i control automatitzat i que conté un joc de 5 filtres del sistema fotomètric de Johnson-Cousins (U, V, B, R, I). Diversos instruments recullen dades per conèixer les condicions d'entorn i de la base de temps en tot moment: dues estacions meteorològiques, una antena GPS, un detector de tempestes, etc. Un ràdio enllaç amb un ample de banda de 10 Mbps permet la comunicació externa necessària per al control i monitoreig remot de la instal·lació. Aquesta connexió serà substituïda per un enllaç de fibra en els propers mesos, cosa que augmentarà significativament l'ample de banda disponible.

Una arquitectura complexa de programari gestiona totes les operacions de l'observatori. El programa TALON és l'encarregat de dur a terme el control dels moviments bàsics del telescopi i la cúpula.

Com a instrumentació complementària, es disposa d'una càmera CCD per a l'observació de tot el cel per detectar bòlids i meteors. Com a objectiu a mig termini, les imatges recollides amb aquesta càmera es faran servir per generar un mapa de núvols que permetran la selecció d'àrees de claredat en el cel.

Mode d'operació

L'operació robòtica és un requeriment fort per l'ús rutinari de l'observatori. Això és degut, principalment, als objectius científics i a l'aïllament del seu emplaçament. La interacció amb persones només està prevista en l'enviament de noves propostes d'observació i en la recollida de les imatges i les dades per part de l'usuari. L'objectiu principal del projecte durant les fases de disseny i desenvolupament ha sigut garantir aquest mode d'operació durant la realització d'observacions astronòmiques de gran qualitat.

Cas científic

El TJO-OAdM és una instal·lació d'ús general i, com a tal, porta a terme una sèrie d'observacions relacionades amb diversos casos científics (Isern et al. 2000). Donades les seves dimensions, el seu

¹ <http://www.upc.edu>

² <http://www.ice.cat>

³ <http://www.fjo.es>

principal nínxol és el domini temporal, en el qual el requisit primordial són les observacions contínues d'alta cadència. És obvi que el TJO-OAdM no pot competir amb telescopis més grans, que tenen una àrea de captació de fotos 170 vegades més gran. No obstant això, el seu avantatge és una operació flexible que permet el seguiment de les fonts durant un període prolongat, així com també la possibilitat d'un ràpid temps de reacció, possiblement de menys d'un minut. Tenint en compte aquestes característiques, els casos científics del TJO-OAdM inclouen:

- Recerca d'exoplanetes, seguiment de planetes amb trànsit coneguts o recerca individual d'objectes.
- Binàries eclipsants, per entendre les propietats de les estrelles i la seva estructura.
- Variables polsants. per explorar l'interior de les estrelles.
- Estrelles variables evolucionades (p.e., supergegants).
- Activitat estel·lar, per entendre la dinamo magnètica i per calibrar el temps de caiguda d'aquesta activitat.
- Variabilitat de nuclis de galàxia actius relacionats amb el procés estocàstic d'acreció.
- Objectes del Sistema Solar (seguiment d'asteroides, objectes propers a la Terra, cometes).
- Supernoves, amb el valor afegit de poder observar fotometria primerenca.
- Binàries de raigs X, fenòmens de variabilitat rotacionals i d'acreció.
- Noves, també amb la possibilitat d'obtenir dades primerenques.
- Contrapartides òptiques de Fulguracions de Raigs Gamma (GRBs).
- Qualsevol fenomen transitori en general.

Els casos científics mencionats requereixen una alta flexibilitat en la programació de la nit, per tal de permetre que el sistema reaccionï ràpidament a les alertes d'observació en relació, per exemple, amb GRBs, noves i supernoves, i esdeveniments similars crítics en el temps. També es considera la possibilitat de participar en les xarxes d'observatoris robòtics per poder dur a terme, entre d'altres, observacions que requereixen la cobertura de temps continu. De la mateixa manera, el TJO-OAdM es pot utilitzar com un servei de suport a les missions espacials per recollir dades fotomètriques i astromètriques. Un exemple obvi és la missió Gaia (amb la participació destacada de l'IEEC), en la qual es podria contribuir de manera important donant suport des de Terra (pre- i post-lançament). El TJO-OAdM oferirà temps a la comunitat astronòmica a través de propostes competitives revisades per un Comitè d'Assignació de Temps (CAT).

Contribució de la Tesi a la robotització del TJO-OAdM

Amb aquesta Tesi hem contribuït al desenvolupament del telescopi TJO-OAdM per tal que pugui operar de forma robòtica qualificada. Afegim el terme *qualificat* per destacar l'esforç realitzat per tenir una operació desatesa robusta, fiable i eficient, alhora que les dades recollides són d'una qualitat òptima. Aquestes característiques són importants per aquest telescopi donades les condicions en les quals opera (ubicació, entorn meteorològic i cas científic) i donen un grau de singularitat a aquest projecte. L'assoliment de les mateixes ha estat basat en una sèrie de punts clau, que es descriuen en les següents seccions.

Operació robòtica

L'operació d'un observatori astronòmic mitjançant el control robòtic té nombrosos avantatges en comparació amb el control atès. Aquestes han estat extensament descrites en l'última dècada en diverses publicacions (Strassmeier et al. 2004). Aquest mode d'operació implica l'ús de tecnologies complexes utilitzades habitualment en observatoris espacials o en certes aplicacions industrials, però que no són comunes en observatoris terrestres. El seu ús, però, s'ha estès a aquests observatoris en les dues últimes dècades gràcies a l'evolució de les capacitats del maquinari i el programari i ha estat motivat pels clars beneficis que ofereix en termes de l'optimització del temps per maximitzar el retorn científic. Malgrat tot, l'assoliment d'una operació robusta i fiable és encara un repte per la majoria d'aquestes instal·lacions, les quals han d'estar preparades per treballar en llocs remots i, sovint, en condicions d'entorn extremes.

La definició del control robòtic aplicat a l'operació de telescopis s'ha formulat de manera diferent des de la seva aparició. Aquest fet està estretament relacionat amb l'evolució de les tecnologies utilitzades per la seva implementació, que han permès un augment gradual de l'autonomia d'aquests instruments. Es poden identificar quatre línies principals d'acord amb el nivell tecnològic que requereixen (Baruch et al. 2004, Baruch 1993):

- *Telescopis automàtics*: Aquests instruments es mouen manualment seguint un patró definit per tal que quedi registrat i, llavors, tenen la capacitat de repetir aquest moviment de forma

automàtica i indefinida. En el cas dels telescopis, l'operador obté les posicions d'un conjunt d'estrelles i es deixa que el sistema segueixi el grup d'objectes mentre creua el cel. Aquest tipus de sistema acostuma a disposar de sensors que li permeten conèixer l'estat de l'entorn i que aturen el sistema quan es produeix la sortida del Sol o quan empitjoren les condicions meteorològiques.

- *Telescopis automatitzats*: Aquests telescopis són capaços de trobar estrelles i de centrar els detectors (p.e. fotòmetres) en els objectes que es vol estudiar. Poden seguir una seqüència d'observació predefinida i moure's a diferents punts del cel. Aquests telescopis imiten la flexibilitat dels robots industrials, els quals són capaços de realitzar un nombre de tasques i respondre a requeriments complexos gràcies al programari que utilitzen.
- *Telescopis remots*: Els telescopis operats telemàticament realitzen les tasques seguint les instruccions de control definides per un operador.
- *Telescopis robòtics*: Els telescopis robòtics són els únics operats de forma autònoma. No requereixen la interacció amb persones per tal de ser operats, la qual cosa proporciona un augment de la seva eficiència, malgrat impliquen el desenvolupament de sistemes d'enginyeria complexos.

La definició de telescopi robòtic que apareix en aquesta classificació és la que hem considerat en aquesta Tesi i és a aquesta que ens referim sempre que mencionem aquest mode d'operació. El concepte de telescopi robòtic utilitzat arreu, però, ha cobert històricament un ampli espectre de nivells de control: des de telescopis operats remotament fins a telescopis completament desatesos. Aquest malentès encara continua existint i és força habitual trobar articles en la literatura on es descriu telescopis robòtics que tenen un cert nivell de supervisió o control atès. Es pot veure, per exemple, la descripció del sistema de control de la cúpula del telescopi *robòtic* NMSU d'1 m de diàmetre publicat a Holtzman et al. (2010), l'obertura de la qual està sincronitzada amb l'estat de la cúpula del telescopi ARC de 3.5 m situat al mateix observatori (Apache Point Observatory) i controlat per operadors in situ. O el cas del projecte Physics Innovations Robotic Astronomical Telescope Explorer (PIRATE), un telescopi operat remotament (veure Holmes et al. 2011), el nom del qual el defineix clarament com un telescopi robòtic. Aquests són només dos exemples que il·lustren l'incorrecte ús del terme robòtic per descriure instal·lacions que funcionen de forma supervisada.

Aquest ús generalitzat del terme *robòtic* pels observatoris astronòmics ha portat a una certa confusió alhora de descriure les característiques d'aquest tipus d'instal·lacions i una subestimació de la enginyeria complexa que impliquen. Aquest fet ha sigut força comú en aquells projectes que s'han basat en la compra de telescopis comercials que el fabricant descrivia com a instruments *robòtics*. Les solucions industrials estan dissenyades normalment per l'adquisició d'imatges d'objectes, però l'assoliment d'un alt grau d'autonomia amb aquests aparells no s'aconsegueix sense una gran inversió de temps i de recursos. A més, els resultats acostumen a ser força insatisfactoris degut a qüestions de disseny dels propis telescopis. Per exemple, la majoria d'aquests telescopis no comprenen un sistema complet que inclogui el recinte (cúpula, sostre) i el control de les condicions de l'entorn. Cal afegir que la majoria de solucions es basen en l'ús de tecnologia propietària que no utilitza solucions estandarditzades, la qual cosa implica que no poden ser modificades per integrar-les en el sistema de control d'un observatori. Aquests motius han fet que molt projectes en aquesta situació optessin per reduir els seus objectius i decidissin operar el sistema de forma remota o amb un control atès.

Restriccions per a una operació robòtica

Un observatori terrestre té diverses restriccions per poder operar sense la interacció amb cap persona. Aquestes restriccions es poden agrupar en intrínseques o vinculades a l'emplaçament.

Els observatoris desatesos han d'emular la resposta que donaria una persona a totes aquelles situacions que puguin pertorbar el flux normal de les tasques. Qualsevol decisió ha de ser planificada en la fase de disseny. El sistema ha de poder detectar el mal funcionament de qualsevol element del maquinari o del programari i ha de disposar d'una resposta adequada per assegurar la seva fiabilitat i seguretat. Cal dedicar un esforç especial en aquells punts que poden causar danys en parts crítiques del sistema. Totes aquestes qüestions són força comunes a tots els observatoris i es poden definir com a restriccions intrínseques, ja que depenen de la configuració d'aquest tipus de sistema.

L'entorn específic de cada observatori introdueix requeriments particulars pel seu desenvolupament. S'ha de planificar una resposta adequada a cada una de les variables meteorològiques que, per exemple, pugui tenir efecte en el correcte funcionament del sistema o en la qualitat de les dades recollides. La seguretat de l'instrument i la seva eficiència depenen d'una resposta correcta a les condicions d'entorn.

El nivell d'autonomia d'un observatori astronòmic s'estableix, finalment, a partir de la confiança en la resposta del sistema davant de canvis en les condicions d'entorn o de possibles alarmes. Aquest punt determina el risc que a una instal·lació s'hi puguin produir danys en qualsevol moment. L'operació robòtica s'aconsegueix quan la supervisió humana no és necessària per mantenir el risc per sota un nivell acceptable. Aquest nivell serà més o menys alt depenent, també, de l'aïllament de l'emplaçament on es trobi ubicat.

Nivells de control robòtic

Podem distingir diferents nivells de control desatès entre els projectes desenvolupats fins ara. Els podem agrupar tenint en compte les tasques que duen a terme sense interacció humana i utilitzant dos criteris diferents: la completesa de l'operació desatès pel control del flux de dades i pel control de la seguretat del sistema, i el nivell de dinamisme del sistema per respondre a condicions canviants.

En relació al nombre de processos executats sense interaccionar amb cap persona, els telescopis robòtics s'han dedicat històricament a l'automatització del sistema d'execució de tasques i de les operacions per garantir la seguretat del sistema (p.e., seguiment de les condicions d'entorn). La majoria executen una llista predefinida de tasques seguint una operació rutinària que s'inicia amb l'engegada del sistema, continua amb la recollida de dades dels objectes a observar i acaba amb l'aturada completa del sistema. Els altres processos vinculats al flux de dades es realitzen manualment i/o en un moment posterior. L'eficiència d'una instal·lació astronòmica es pot veure clarament incrementada amb una completa automatització d'aquests processos. Aquests haurien d'incloure tots els passos compresos entre l'enviament d'una proposta fins a la descarrega de les dades per part dels usuaris, i s'haurien de centrar principalment en la planificació de les tasques, la seva execució i en les rutines per processar les dades recollides. Això és especialment important pels observatoris multi propòsit (o de cas científic no restringit) i multi usuari, els quals han de gestionar diferents tipus de programes observacionals i requereixen l'ús d'eines intel·ligents de gestió.

El nivell de dinamisme d'un sistema per respondre a canvis en les condicions d'operació és el segon criteri de classificació que tenim en compte. Aquest es defineix pel tipus de retorn que el sistema de control té de tots els processos que executa i de com utilitza aquest retorn per reaccionar automàticament amb accions correctives o reactives.

Els observatoris estàtics són aquells que no reaccionen davant del canvi de les condicions d'operació, exceptuant les accions executades per minimitzar el risc en condicions de mal temps (p.e., tancament de la cúpula quan plou). Els observatoris dinàmics, per contra, utilitzen les dades del monitor que supervisa la salut del sistema o la qualitat de les dades recollides per decidir quines accions executar per corregir el mal funcionament d'un component o per modificar la planificació de les tasques a realitzar. Les instal·lacions amb funcionament dinàmic estan preparades per reaccionar d'una manera eficient davant d'esdeveniments rutinaris i no rutinaris.

La majoria dels telescopis robòtics estan dissenyats per dur a terme un control simple del flux de dades, enfocat principalment a l'execució de les observacions, i es basen en el model de control estàtic. La seva autonomia és força baixa, però són bona una solució donat que impliquen un desenvolupament més simple. Són adequats per telescopis situats en emplaçaments no aïllats, a on la interacció amb persones és possible per donar una resposta immediata a alarmes crítiques o per canviar la planificació d'una nit.

Un petit nombre de telescopis robòtics utilitzen planificadors intel·ligents (o dinàmics) i/o rutines de processat en línia com a eines per completar el control desatès del flux de dades. Aquests són temes encara poc explorats per aquest tipus d'observatoris terrestres, encara que s'ha demostrat que incrementen de manera substancial l'eficiència del sistema en termes del nombre de dades recollides i del retorn científic. També podem trobar alguns telescopis que han estat construïts per operar de manera dinàmica amb l'objectiu d'assolir un alt nivell d'autonomia. Alguns exemples de telescopis basats en aquestes tecnologies són: els telescopis RAPTOR (RAPid Telescopes for Optical Response, Vestrand et al. 2002), el telescopi BRT (Bradford Robotic Telescope, Baruch 2007), el projecte STELLA (STELLar Activity project, Granzer et al. 2010) o el telescopi LT (Liverpool Telescope, Steele et al. 2004).

El següent pas amb aquest camp tecnològic és la construcció de xarxes de telescopis robòtics de resposta altament dinàmica. Actualment s'estan desenvolupant xarxes de dos tipus. En el primer grup trobem xarxes liderades per un únic consorci o institució basades en tecnologia escollida ad hoc i que no tenen la necessitat d'utilitzar solucions estandarditzades pel que fa als protocols o a les interfícies entre els diferents telescopis. Dos dels projectes inclosos en aquest grup serien: MONET (Bischoff et al. 2006) o Las Cumbres Observatory Global Telescope Network (LCOGT, Pickles et al. 2010). En el segon grup trobem diverses iniciatives el treball de les quals es destina a definir un protocol estàndard per a la comunicació d'esdeveniments, per la planificació de tasques, per l'emmagatzematge massiu

de dades, etc. Aquestes busquen les eines que permetin la col·laboració entre telescopis heterogenis situats en diferents punts del món. Entre els diferents projectes que es troben actualment en desenvolupament, destaquem l'eSTAR (Allan et al. 2004), l'HTN (Heterogeneous Telescope Network, White 2007), o el Virtual Observatory Event (White et al. 2006). Tots aquests projectes *obriran noves finestres de descoberta científica* gràcies a l'operació coordinada de múltiples telescopis robòtics.

Objectius de la tesi, estructura i continguts

El telescopi robòtic TJO-OAdM proposat a mitjans dels anys 90 representava un repte tecnològic pel camp de la instrumentació astronòmica. Es va concebre com a un telescopi multi propòsit amb una operació completament desatès. En aquells moments, només existien alguns telescopis en el món que funcionessin de forma desatès (p.e., BRT – Baruch 2007, Fairborn Observatory¹ - Henry 1995), la majoria dels quals tenien un funcionament estàtic destinat a gestionar només el procés d'execució de tasques. A més, es trobaven ubicats en llocs on la supervisió humana era habitual i el seu cas científic era molt específic i dedicat a una reduïda comunitat d'usuaris, la qual cosa no suposava un requeriment important pel que fa a la necessitat de gestionar diferents modes d'operació ni programes d'execució (programes de monitoreig, d'observacions amb efemèrides, etc.).

El desenvolupament inicial del TJO-OAdM no va complir amb les expectatives de funcionament. Tenia diverses deficiències causades principalment per un desenvolupament basat en una instrumentació comercial que no era adequada per un control desatès i fiable. I també presentava una incompleta definició de l'arquitectura del sistema per garantir el control de tota la instal·lació i de tots els processos inclosos en el flux de dades.

Tenint en compte aquestes obstacles previs, els objectius principals d'aquesta Tesi han sigut el desenvolupament d'un telescopi robòtic amb un alt nivell d'autonomia i eficiència basat en el projecte original.

El treball s'ha focalitzat en aplicar els canvis necessaris al disseny inicial per tenir un sistema capaç de donar respostes intel·ligents a qualsevol situació (seguint el model de control dinàmic) així com proporcionar un control desatès del flux complet de dades. Les especificacions inclouen des de la preparació i introducció de les propostes al sistema fins al processat de les dades i la posterior entrega als usuaris. Aquest concepte d'operació imposa un control intel·ligent d'alt nivell. L'aïllament del lloc i les condicions meteorològiques extremes afegeixen un requeriment de robustesa al desenvolupament del sistema. Es considera fonamental, doncs, assolir una alta fiabilitat, robustesa i qualitat de les dades recollides per tal de maximitzar l'eficiència i el retorn científic.

Aquesta Tesi ha evolucionat en paral·lel amb altres telescopis robòtics arreu del món basats en aproximacions similars (p.e., LT – Steele et al. 2004, STELLA – Granzer et al. 2010, Faulkes – Smith et al. 2004), els quals es troben en diferents graus de desenvolupament en el moment d'escriure aquest document. El telescopi TJO-OAdM és un projecte singular i pioner en alguns aspectes, malgrat presentar diverses similituds amb els projectes esmentats, en els quals s'han aplicat tecnologies i dissenys comparables. Les qüestions més innovadores aportades tenen a veure amb les eines desenvolupades per la gestió de les propostes d'observació, la planificació intel·ligent de les tasques a realitzar, el processat de les dades recollides i la determinació d'un índex de la qualitat per cada imatge, o la infraestructura informàtica que implementa el control general del sistema. Aquestes solucions es complementen amb d'altres destinades a reforçar la robustesa (p.e., sistemes redundants) i a implementar accions de contingència que incrementen la independència i autonomia del sistema, condicions necessàries donades les característiques de l'emplaçament de l'observatori. L'aparició de noves implementacions tecnològiques durant el desenvolupament del projecte també ha ajudat a trobar solucions diferents en el cas d'aquest telescopi.

Les característiques del TJO-OAdM i les seves dimensions fan que aquest telescopi sigui força singular i únic en comparació amb d'altres telescopis robòtics que es troben actualment en funcionament (consultar, per referència, la extensa compilació produïda per F. V. Hessman²).

La Tesi s'estructura en tres blocs: anàlisi i disseny del sistema, fase de desenvolupament i control de qualitat.

Anàlisi i disseny del sistema

En la fase d'anàlisi del sistema, vam dur a terme una avaluació exhaustiva dels requeriments que havia de complir per tal de construir un disseny adequat per l'operació del telescopi. Alguns d'aquests requeriments es van especificar per adaptar la configuració inicial, heretada de la fase de

¹ <http://www.fairobs.org>

² <http://www.astro.physik.uni-goettingen.de/~hessman/MONET/links.html>

desenvolupament prèvia al projecte de robotització, a la nova que es presenta en aquesta Tesi, mantenint les capacitats de la instal·lació en termes de l'assoliment del cas científic.

Aquests requeriments van servir de base per definir el disseny global de l'operació del telescopi i la seva arquitectura. En aquest punt es va establir el concepte de funcionament robòtic aplicable al TJO-OAdM.

El disseny operacional detallat i l'arquitectura del sistema constitueixen els productes resultants d'aquest primer bloc de la Tesi fruit de l'anàlisi del projecte original. L'aproximació escollida en aquest sentit es pot aplicar en la construcció d'altres telescopis robòtics, ja que es base en tres capes que cobreixen les principals funcionalitats de control: seguretat del sistema (housekeeping), flux de dades (data flow) i elements del maquinari (hardware).

El disseny operacional defineix els processos desatesos que s'executen en condicions rutinàries i no rutinàries, el tipus de tasques que pot dur a terme, i els usuaris i els diferents nivells de permisos per interactuar amb el sistema. La correcta definició d'aquest model operacional és de vital importància per poder desenvolupar un sistema que compleixi amb els requeriments, i és per això que es va dedicar un esforç important en la seva elaboració. Vam dur a terme aquest exercici en el moment en què el projecte de robotització s'havia de definir. També ha permès que el procés de validació i verificació fos més fàcil i que el projecte assolís la qualitat desitjada.

El disseny operacional va ser la base per definir l'arquitectura del sistema. El disseny general es va estructurar en dues parts principals:

- *Maquinari operacional*: Aquesta capa està composta de la instrumentació seleccionada durant la fase inicial del projecte, escollida en funció del cas científic de l'observatori. El telescopi, la cúpula i l'instrument en són els components més importants. Altres dispositius comercials van ser seleccionats o desenvolupats durant el projecte de robotització per assolir el funcionament desitjat. La selecció o disseny de la major part d'aquests dispositius es va fer en el marc d'aquesta Tesi seguint criteris de robustesa i fiabilitat.
- *Sistema de control robòtic (Robotic Observatory Control System, ROCS)*: Aquest conjunt d'elements de programari està compost per dues capes que executen les tasques involucrades en el control del flux de dades i de la seguretat del sistema. Aquest bloc té una estructura modular construïda amb diverses aplicacions (ISROCS, ICAT), eines (DDROCS, gestió de propostes) i una infraestructura que gestiona els processos utilitzats per aconseguir l'operació desitjada (OpenROCS).

Desenvolupament del sistema

El segon bloc està dedicat al disseny detallat i al desenvolupament del sistema, estructurat d'acord amb les dues parts ja mencionades: el maquinari operacional i el sistema de control robòtic (ROCS). El treball realitzat inclou el disseny general i detallat del sistema de control i de tots els seus mòduls, i el desenvolupament d'alguns dels paquets de programari.

Alguns dels mòduls de maquinari i programari inclosos en aquest bloc de la Tesi han estat desenvolupats en col·laboració o directament per altres persones (físics i enginyers) que han treballat en el projecte en algun moment. Aquest grup de científics, referit en el text com l'*engineering team*, ha participat en diverses fases del projecte i ha contribuït amb un treball molt valuós. Ells han liderat les fases de disseny detallat i/o de desenvolupament de diferents paquets, alguns dels quals s'han inclòs en el text d'aquesta Tesi per donar una descripció completa del sistema. S'ha indicat de forma adequada en aquest cas.

➤ *Maquinari operacional*

L'arquitectura del sistema que compona l'observatori va evolucionar des de la configuració inicial fins a la versió implementada descrita en aquesta Tesi. Estem convençuts que disposem ara d'un sistema estable i robust que treballa amb un nivell de risc baix i que proporciona dades d'una qualitat propera a la màxima que pot proporcionar l'equipament del TJO-OAdM.

A la Tesi es descriuen els canvis realitzats en el maquinari científic original: telescopi, instrument i cúpula. Aquests canvis es van fer per millorar la seva funcionalitat i la qualitat de les dades, però també per incrementar la fiabilitat del sistema. Es van dur a terme canvis importants en el sistema de protecció contra la llum directa (baffle) en el tub del telescopi que van permetre augmentar la qualitat òptica. Els canvis en l'electrònica i del programari de baix nivell van millorar el sistema de seguiment del telescopi i van afegir funcionalitats, com són el control de les tapes del mirall primari i de la cúpula. La càmera CCD de l'instrument i l'electrònica de la cúpula també van ser modificades per augmentar les seves prestacions i fiabilitat. Tots aquests canvis han significat un increment substancial de

l'estabilitat del sistema. Finalment, es va dissenyar i instal·lar un nou sistema per calibrar el detector de l'instrument.

També presentem en aquesta Tesi els nous dispositius afegits per completar l'arquitectura del sistema. S'han instal·lat més proteccions contra l'impacte directe de llamps, es disposa ara d'un alt nivell d'aïllament elèctric, s'ha augmentat la robustesa del sistema d'alimentació elèctric i s'ha completat el sistema amb línies de control redundants pels elements més crítics (obertura de la cúpula i sensors meteorològics). Aquests canvis han sigut importants per reduir la probabilitat que es produeixin certs escenaris d'alt risc, o per minimitzar l'impacte que aquests poden produir. Altres accions s'han enfocat a incrementar el nivell de control i el retorn en temps real del sistema que és necessari per detectar una font d'error en el menor temps possible i proporcionar un control dinàmic. La majoria de nous dispositius són equips comercials d'una alta robustesa.

➤ *Sistema de control robòtic (ROCS)*

ROCS va ser dissenyat com a un paquet de programari que gestiona el flux de control d'un observatori robòtic amb la mínima interacció humana i el màxim retorn científic. En el disseny a alt nivell, es va especificar les aplicacions necessàries per dur a terme les diferents funcionalitats, la llista detallada dels possibles estats de l'observatori i les seves característiques, la llista de les tasques que ROCS havia d'utilitzar per controlar el flux de dades i la seguretat del sistema, i la llista de les interfícies definides entre els diferents mòduls.

El desenvolupament d'aquest paquet va donar lloc a les aplicacions i eines resumides a continuació.

OpenROCS

OpenROCS és el programari responsable de resoldre qualsevol situació que pugui posar en risc la integritat del sistema i activa accions correctives o accions que minimitzin el dany produït. És un administrador del flux de dades i de treball dels processos de l'observatori i un planificador dels processos que el sistema ha d'executar. OpenROCS monitoritza l'estat global de l'observatori i assegura que el sistema executa les operacions rutinàries i no rutinàries alhora que es maximitza el retorn científic.

Aquest programari connecta de manera homogènia el conjunt d'aplicacions desenvolupades per executar tasques particulars amb components de maquinari i programari de l'observatori. Proporciona una arquitectura modular i escalable que incrementa la fiabilitat del sistema i s'allunya del concepte monolític per implementar el sistema de control d'un observatori.

S'ha desenvolupat dues versions d'aquest programari. La primera versió estava basada en un *middleware* (Ice) i proporcionava una estructura complexa i monolítica difícil d'adaptar a diferents necessitats i configuracions. Una segona versió va ser programada amb llenguatge PHP de *scripting* i amb una estructura més flexible que transfereix la configuració de l'esquema de control a fitxers XML. La nova implementació, actualment en ús, va venir motivada per la necessitat de disposar d'un programa fàcil de modificar i configurar, obrint la possibilitat d'utilitzar-lo en altres observatoris.

Base de dades (DDROCS) i eines de gestió de propostes

Els usuaris externs interactuen amb l'observatori principalment a través de l'eina desenvolupada per la gestió de propostes. Aquesta eina és simplement una interfície web amb la base de dades que permet a l'usuari enviar la seva proposta d'observació en dues fases. Tanmateix, darrera d'aquesta eina i de la base de dades hi ha un disseny complex estretament relacionat amb el concepte aplicat a l'observatori per la gestió dels processos relacionats amb el flux de dades.

El flux de dades aplicat al TJO-OAdM és un procés complet que comprèn des de l'entrada de la informació associada a una proposta, en la qual s'especifica els detalls per la seva execució, fins a l'entrega de les dades a l'usuari. Tots els passos intermitjos d'aquestes dues accions manuals es duen a terme de forma automàtica pel ROCS. Per tant, és molt important disposar d'una base de dades a on s'emmagatzemin les tasques associades a cada proposta d'una manera que puguin ser llegides pel sistema de control per la planificació automàtica de la nit. A més, les imatges recollides i les dades extretes s'han de poder crear fàcilment amb aquestes tasques per permetre que el sistema comprovi l'estat de completitud de cada proposta i puguin ser recuperades per l'usuari. El disseny d'aquest flux de dades té dues conseqüències principals: cal que hi hagi una clara especificació de les restriccions que l'usuari ha de definir per la correcta planificació de les seves tasques i que ha de facilitar a l'observatori, i ha d'haver-hi una base de dades que gestioni totes les dades de forma consistent. La base de dades (DDROCS) i la preparació de propostes es van dissenyar per optimitzar el procés.

La base de dades també emmagatzema altres dades (entorn, registres dels diferents processos, etc.) que poden ser necessàries per la supervisió del funcionament de l'observatori.

ISROCS, planificador de tasques (scheduler)

El planificador de tasques de l'observatori (ISROCS, per Intelligent Scheduler for the ROCS) és responsable de la selecció de la tasca més adequada per ser executada en cada moment de la nit, d'acord amb les restriccions d'execució de la tasca i de les condicions a l'observatori en el moment de fer la selecció (entorn i estat del maquinari). ISROCS s'encarrega de planificar les tasques de ciència i de calibració del sistema i és responsable de la resposta intel·ligent del sistema pel seu funcionament rutinari.

El planificador es va dissenyar per tal de proporcionar a l'observatori una eina que fés una planificació estàtica i dinàmica de les tasques científiques. La planificació estàtica calcula la llista ordenada de tasques a executar durant la nit següent (mig termini) o en un període de temps de diverses nits (llarg termini). Aquesta llista la calcula tenint en compte només les restriccions d'execució que depenen de condicions estàtiques que poden ser predites a priori. La planificació a curt termini, per contra, ve donada per la component dinàmica que afegeix les restriccions d'execució que depenen de les condicions en què es troba el sistema i l'entorn i que varien en temps real.

Ambdós nivells de planificació es calculen utilitzant un algoritme dispensador de tasques (dispatcher), d'acord amb la descripció presentada per Granzer (2004). Aquest utilitza el càlcul de la Figura de Mèrit per cada tasca i va ser escollit ja que proporciona un resultat ràpid i efectiu per la prioritització dinàmica de les tasques. En aquesta Tesi proposem diverses modificacions a aquest algoritme per tal de millorar la planificació a curt termini, tenint en compte la contribució de les restriccions a mig i a llarg termini (p.e., observacions periòdiques o tasques interrelacionades) i altres restriccions específiques del TJO-OAdM.

L'arquitectura del planificador es va dissenyar per integrar la seva execució en la gestió dels processos que du a terme OpenROCS en la gestió global de l'observatori.

La nova rutina de planificació es troba en fase de desenvolupament. Per tal de poder iniciar l'operació del telescopi amb un control completament desatès, vam dissenyar i desenvolupar un planificador simple que calcula una cua de tasques abans de l'inici de la nit seguint una sèrie de criteris preestablerts. Aquest planificador en cua serà substituït pel nou un cop finalitzi la seva implementació.

ICAT, processat de les dades

L'objectiu principal de l'aplicació ICAT (IEEC Calibration and Analysis Tool) és contribuir a la capa de control del flux de dades del TJO-OAdM amb l'extracció de dades astronòmiques d'alta qualitat i precisió. Amb aquest propòsit, les seqüències per la calibració i anàlisi de les imatges van ser definides, juntament amb un procés de control de qualitat i altres paquets complementaris destinats a gestionar l'emmagatzematge i arxivat.

Aquesta aplicació executa un processat i un control de qualitat de les imatges en temps real just després de la seva adquisició, proporcionant un retorn important al sistema per poder decidir automàticament si la imatge ha de ser repetida. Un processat final de les imatges es du a terme al finalitzar la nit per tal d'incrementar la qualitat de les dades extretes i per posar-les a disposició de l'usuari.

Per realitzar la reducció i l'anàlisi de les imatges, ICAT utilitza paquets de programari externs. IRAF, DAOPHOT, SExtractor i les llibreries CFITSIO s'han utilitzat per diferents motius. Una comparació entre els diferents paquets d'anàlisi d'imatges es va fer per seleccionar el paquet que permetés extreure els valors fotomètrics i astromètrics dels objectes amb la millor qualitat possible. Aquest procés es troba descrit a la Tesi.

L'ICAT juga un paper important en la detecció del deteriorament de la instrumentació científica (p.e., òptica del telescopi) i en el retorn proporcionat al sistema de control amb l'avaluació de la qualitat de les imatges. És un element clau, per tant, en maximitzar l'eficiència del sistema.

Eines auxiliars per la supervisió remota

S'han desenvolupat diverses interfícies gràfiques i s'han instal·lat dues càmeres de videovigilància per permetre una monitorització remota de l'observatori. Aquestes eines auxiliars es van planificar per una supervisió externa per part del personal de l'observatori i per facilitar les tasques de manteniment o de testeig dutes a terme per l'equip d'enginyers.

Control de qualitat

La robustesa i la qualitat de les operacions que realitza el sistema de control són especialment crítiques en l'operació robòtica d'un observatori astronòmic. Aquestes condicions s'han considerades en tot moment com a requeriments d'alt nivell en el desenvolupament del TJO-OAdM. En aquest procés s'han aplicat pràctiques comunes de l'enginyeria de sistemes, incloent processos de control de

qualitat (QA), per assegurar que el projecte implementat estava d'acord amb els requeriments inicials. Malgrat no s'han aplicat de forma exhaustiva i formal, aquestes pràctiques han tingut una funció important en el desenvolupament del projecte i d'aquesta Tesi.

Hem definit i aplicat diverses pràctiques de control de qualitat durant les diferents fases del projecte: gestió de riscos per assegurar que els riscos del projecte estaven correctament identificats i protegits amb accions orientades a eliminar-los o a minimitzar-ne els efectes; control de verificació per confirmar que els requeriments especificats s'havien complert i que les solucions implementades es comportaven de forma adequada; activitats de testeig dissenyades i executades per complementar el control de verificació; i gestió i control de configuració centrat en les activitats realitzades per gestionar els diferents tipus de documents, dispositius, paquets de programari i procediments de test.

Un procediment de test s'ha dissenyat també com a un element important del control de qualitat per demostrar que cada un dels subsistemes de l'observatori es comporta tal i com defineixen els seus requeriments i que el sistema complet opera de manera robusta per un període llarg de temps. El procediment de test descriu la campanya de tests necessària per validar l'operació robòtica del TJO-OAdM mitjançant el test individual de cada subsistema i simulant diferents escenaris.

Les operacions realitzades a l'observatori des de la finalització del projecte de robotització (finals de 2010) s'han centrat en testejar el sistema de control. Els tests han demostrat el bon comportament de l'operació robòtica. Tot i així, els tests van permetre detectar alguns defectes en l'òptica, la mecànica, en el programari de baix nivell i en l'electrònica del telescopi TJO; i una part important del temps dels darrers mesos s'ha dedicat a corregir-los. El sistema proporciona actualment dades amb el màxim nivell de qualitat, però el nombre de tests que finalment s'ha pogut fer ha sigut baix.

En el moment d'escriure aquesta Tesi s'està executant el procediment de test de forma sistemàtica per obtenir prou informació per avaluar el funcionament del sistema de control. L'eficiència del sistema és encara baixa quan es compara el temps invertit en realitzar observacions científiques amb el temps nocturn disponible. Això és conseqüència de treballar en mode robòtic supervisat, utilitzat actualment i que requereix la presència de personal a l'observatori mentre s'executen les tasques, fet que redueix el temps disponible per les operacions. La falta d'eficiència també ve causada per l'execució del procediment de test que implica aturar l'adquisició de dades científiques. Un cop aquest procés de test verifiqui la correcta resposta del sistema a situacions crítiques i les operacions desateses es duguin a terme de forma rutinària, s'espera que l'eficiència augmenti fins el nivell esperat.

Treball en curs i objectius futurs

El disseny general dels diferents mòduls que componen el telescopi TJO-OAdM, la implementació de l'arquitectura física i la majoria de paquets de programari es troben ja desenvolupats i funcionant a l'observatori.

El treball en curs es centra principalment en el test del sistema de control robòtic, en la repetició de les tasques de comissionat del telescopi i l'instrument, en millorar alguns dels mòduls de programari ja implementats i en incorporar noves funcionalitats ja dissenyades, però no implementades. El treball futur consistirà bàsicament en la instal·lació de nova instrumentació.

Tot aquest treball es du a terme en col·laboració amb l'equip d'enginyers de l'IEEC.

Tests

El procediment de test és executat de forma aleatòria cada vegada que el personal de l'observatori es troba al Montsec per fer observacions robòtiques en mode supervisat. Aquest tipus d'operació està previst que duri fins el mes de maig de 2012. Les operacions robòtiques començaran a partir de llavors.

Les dades astronòmiques que es recullen estan relacionades amb propostes específiques dissenyades per repetir l'anàlisi fet en el comissionat de la instal·lació, per tal d'avaluar la millora del telescopi i de l'instrument MEIA després dels canvis realitzats.

Millora dels mòduls de programari

Diversos mòduls s'han refet per corregir alguns errors i per millorar la seva implementació. L'objectiu principal és simplificar el manteniment augmentant la modularitat i la homogeneïtat de llenguatges i, alhora, millorar el seu funcionament. La gestió de les propostes i la base de dades, per exemple, s'estan modificant per tenir una implementació més flexible i completa. Un esforç important s'està duent a terme per tal que tots els mòduls desenvolupats siguin fàcilment traslladables a altres

observatoris. Aquesta política s'aplica a OpenROCS i a ICAT. TALON és un altre dels mòduls de programari que està sent depurat per proporcionar una millor comunicació amb el sistema de control.

Noves funcionalitats

Un dels mòduls principals en desenvolupament és la nova aplicació ISROCS per la planificació de tasques per tal que incorpori la resposta dinàmica i millori el procés de selecció.

Una altra tasca important és la integració de les dades d'entorn recollides per l'estació meteorològica del Telescopi Fabra-ROA Montsec situada al mateix observatori. Aquesta nova col·lecció de dades proporcionarà al TJO-OAdM una redundància en la detecció de pluja i, per tant, es podrà eliminar un dels punts de fallada crítica (SPF) de l'observatori.

Nova instrumentació

En un futur proper s'instal·larà nova instrumentació al Montsec:

- *ARES*: Aquest espectrògraf ja es troba instal·lat a l'observatori, però ha de ser integrat en el sistema de control. S'ha construït una habitació específica a la planta baixa del TJO-OAdM perquè l'ocupi aquest instrument. L'eina de gestió de propostes, el planificador de tasques i el sistema de control hauran de ser modificats, i una nova rutina de processat de dades serà necessària per avaluar els espectres adquirits.
- *Autoguiat i front-end*: La instal·lació d'ARES s'haurà de fer en paral·lel amb la construcció d'una nova interfície opto-mecànica (front-end) que permeti la selecció automàtica de l'instrument (MEIA o ARES) i el correcte posicionat de la fibra de l'espectrògraf. Un sistema d'autoguiat també serà instal·lat per tal de proporcionar un bucle tancat de control que millori el sistema de seguiment. La capacitat d'autoguiat ja ha estat incorporada a TALON i al programari de baix nivell de les targetes electròniques que controlen els motors del telescopi (CSIMC).
- *RoboDIMM*: L'observatori disposa d'un monitor automàtic de l'estabilitat atmosfèrica, però no s'ha construït un recinte on es pugui instal·lar per ser operat de forma autònoma. Actualment només s'utilitza si s'instal·la manualment fora de l'edifici principal de l'observatori durant les hores nocturnes.

Experiència i nous projectes

El TJO-OAdM es va construir com a un telescopi que pogués ser utilitzat per la comunitat científica per dur a terme treballs de recerca observacionals. Aquest tipus de telescopi és molt útil per programes que requereixin una gran cobertura temporal. L'operació robòtica proporciona una alta eficiència i la possibilitat d'incloure aquest telescopi en xarxes d'abast mundial.

Una segona, però no menys important, motivació va ser la d'utilitzar-lo com a banc de proves pel desenvolupament d'instrumentació astronòmica. Alguns dels objectius eren l'aprenentatge de la tecnologia relacionada amb els telescopis robòtics i dels processos involucrats en l'enginyeria de sistemes, fer recerca en qüestions obertes i en el desenvolupament de noves solucions. Aquests objectius han portat a desenvolupar alguns dels mòduls de control en base a especificacions molt més estrictes del que seria necessari per aquest telescopi. Això pot haver significat un esforç excessiu en algunes ocasions en termes dels recursos destinats i el cost en el retorn científic. El desenvolupament d'un planificador dinàmic de tasques n'és un exemple: el component dinàmic del planificador representa una petita optimització del temps d'observació quan es compara el cost de l'operació del telescopi amb el cost per desenvolupar una eina d'aquestes característiques. Tot i així, l'experiència adquirida durant el procés de recerca obra la possibilitat d'aplicar aquest coneixement en altres projectes on pugui ser necessari. A continuació descrivim alguns projectes que ho exemplifiquen.

SuperWASP Qatar Follow-up Telescope (SQFT)

SQFT és un projecte internacional per la construcció d'un telescopi robòtic d'1 m equipat amb un instrument de dos braços i dedicat a la caracterització d'exoplanetes. Es va instal·lar a l'observatori del Roque de los Muchachos (ORM, Illes Canàries, Espanya) a principis de 2011 i té una arquitectura molt similar al TJO-OAdM.

L'IEEC participa en el programari de baix nivell i en l'electrònica del telescopi, construït pel mateix fabricant que el TJO; contribueix en el desenvolupament de l'arquitectura del sistema i en l'instrument; i proporciona el sistema de control que gestiona les operacions del sistema, basat en el paquet

OpenROCS. Les especificacions d'aquest telescopi són similars a les del telescopi del Montsec, amb petits canvis en la tecnologia utilitzada.

El finançament per aquest projecte es va aconseguir gràcies a una crida competitiva de projectes que es va fer internament a l'IEEC.

Calar Alto high-Resolution search for M dwarfs with Exoearths with Near-infrared and optical Échelle Spectrographs (CARMENES)

Aquest espectrògraf (Quirrenbach et al. 2010) és un instrument de nova generació que s'ha de construir pel telescopi de 3.5 m de l'observatori de Calar Alto (Almeria, Espanya). La construcció la realitza un consorci d'institucions alemanyes i espanyoles. L'objectiu científic es centre en la recerca d'exoplanetes al voltant d'estrelles nanes de tipus M i utilitzarà dos canals per l'adquisició simultània d'espectres en la zona de l'infraroig proper i de l'òptic. Recentment ha superat l'avaluació de PDR i la primera llum de l'instrument es preveu per principis de l'any 2014.

L'IEEC lidera el desenvolupament del sistema de control de l'instrument, del planificador i del procés de control de qualitat gràcies a l'experiència adquirida al TJO-OAdM. Aquests tres paquets són anàlegs a les aplicacions OpenROCS i ISROCS i a la rutina pel control de qualitat inclosa a l'ICAT, però amb unes especificacions més rigoroses. Una major eficiència de CARMENES representarà un fort increment del retorn científic i, per tant, una optimització del cost. Això pot ser una conseqüència directa del correcte funcionament d'aquests tres paquets dels quals som responsable. La nostra participació en el projecte està finançada pel Ministeri d'Innovació i Ciència espanyol.

Cherenkov Telescope Array (CTA)

El projecte CTA (CTA Consortium, 2010) és una iniciativa internacional per construir un observatori terrestre de nova generació dedicat a la detecció de rajos gamma d'alta energia. CTA estarà compost per telescopis Cherenkov de dimensions petita, mitjana i gran situats en dos emplaçaments diferents (hemisferi Nord i hemisferi Sud). S'instal·laran més de 60 telescopis en cada un d'ells. El projecte es troba en la seva fase preparatòria que acabarà a finals de 2013, quan s'espera que comencin les primeres operacions.

Amb aquest alt nombre de telescopis, CTA estarà preparat per ser operats en un ampli espectre de configuracions i permetrà des de l'estudi profund d'objectes individuals amb una sensibilitat mai assolida fins al monitoreig de desenes d'objectes potencialment interessants. Aquesta flexibilitat serà assolida amb l'ús de subgrups predefinits de telescopis.

L'IEEC participa en el paquet de treball d'aquest projecte que ha de proporcionar el programari de control del conjunt de telescopis. Som els responsables de desenvolupar el planificador, el qual juga un paper important alhora de proporcionar l'alta flexibilitat requerida per l'operació global de l'observatori. L'eficiència d'una instal·lació d'aquestes dimensions està fortament vinculada a la planificació òptima de les tasques, tenint en compte l'estat del conjunt de telescopis, les condicions ambientals i la configuració de cada un dels subgrups. La nostra participació està finançada pel Centro de Desarrollo Tecnológico Industrial (CDTI), entitat que depèn del Ministeri d'Innovació i Ciència espanyol. Aquest treball es porta a terme en col·laboració amb l'empresa GTD S.l.

International Concordia Explorer (ICE-T)

ICE-T (Strassmeier 2007) és un telescopi robòtic binocular dissenyat per realitzar mesures fotomètriques contínues de gran camp i alta precisió des de l'Antàrtida (Base Concòrdia, Dome C). L'objectiu principal és investigar els efectes combinats dels exoplanetes, l'activitat estel·lar magnètica i les pulsacions no radials en l'estructura i evolució de les estrelles, així com també la seva dinàmica interna. L'observació contínua amb una alta precisió fotomètrica és necessària per la detecció de planetes i per caracteritzar l'evolució de les estructures estel·lars superficials. ICE-T es troba actualment en la fase de disseny del telescopi i la infraestructura.

Ens encarreguem del paquet de treball de processat de dades (WP 7000), l'objectiu principal del qual és obtenir dades fotomètriques d'alta qualitat i precisió. S'utilitzarà una aproximació similar a la utilitzada amb l'aplicació ICAT, encara que els requeriments de precisió fotomètrica i la configuració dels dos detectors (basats en xips CCD de 10kx10k) imposen diferents opcions de processat.

Chapter 1

Introduction

This document is written to describe the work done by the author for the complete robotization of the Joan Oró Telescope at the Montsec Astronomical Observatory (TJO-OAdM¹). It is submitted to obtain the degree of Doctor of Philosophy.

1.1 Motivation of the Thesis

The project for the construction of the TJO-OAdM telescope begun in the mid-90's, when the *Montsec Sostenible* initiative was approved by the Catalan government to promote the Montsec area in Catalonia. This envisaged the construction of a new facility for science outreach (the *Centre d'Observació de l'Univers*) and an observatory for astronomical research (the *Observatori Astronòmic del Montsec*, OAdM), equipped with a small-class telescope (the biggest in Catalonia) to be operated in a completely unattended fashion.

The initial definition of the TJO-OAdM project had very loose design requirements. The construction of the observatory was mainly based on the acquisition of a telescope assumed to be fully robotic. But multiple shortcomings hindered a complete and robust unattended operation of the whole observatory just after the installation of the telescope in March 2004. The project had to be subsequently modified, elaborating a new one (Colomé et al. 2006) for the complete robotization of the TJO-OAdM observatory. This project under the leadership of Dr. Ignasi Ribas, referred in this text as *the robotization project*, was approved in 2007 by the Generalitat de Catalunya and extended until 2010. The work presented in this Thesis has been carried out in the framework of this project and has contributed to the achievement of its goals.

1.1.1 Robotic operation

There are numerous advantages in operating an astronomical observatory using robotic control instead of human attendance. These have been widely described in publications over the last decade (Strassmeier et al. 2004). This mode of operation implies complex technology that is commonly used in space observatories or certain industrial applications rather than in classical ground-based observatories. Its use has been extended to ground-based observatories thanks to the evolution of hardware and software capabilities and motivated by the clear advantages it offers in terms of time optimization to maximize the scientific return. However, achieving reliable operation is still an issue for most of these facilities, which must be adapted to work on remote site locations and, sometimes, in extreme environment conditions.

The robotic control definition applied to the telescope operation has had different formulations along the last four decades. This is mainly related with the evolution of the technology used to implement it that has enabled an increase in autonomy. Four primary routes can be identified and associated to the required technology level (Baruch et al. 2004, Baruch 1993):

- *Automatic telescopes*: These instruments are led through a series of movements as a form of training and then left to repeat the movements indefinitely. In the case of a telescope, the star,

¹ <http://www.oadm.cat>

its comparators, standards and sky are acquired by the human observer and then the telescope is left to monitor the group of objects as it transits the sky. Such systems usually include some crude environmental sensors that close it down at dawn or if the weather deteriorates into rain.

- *Automated telescopes* are telescopes able to find stars and centre their photometers on the target stars. They can follow a preloaded observing sequence and move around the full sky. These telescopes imitate the developing flexibility of industrial robots that are able to mix a number of tasks and respond through their software programming to complex requirements.
- *Remote telescopes*: These are tele-operated telescopes that run operations following the control commands defined by an observer.
- *Robotic telescopes*. Robotic telescopes are the only ones operated autonomously. Human interaction is not required to operate this kind of facility, which provides an increase of efficiency despite the development of complex engineering systems.

This robotic telescope definition is the one considered in this Thesis any time this operation mode is mentioned. However, the robotic telescope concept used worldwide has historically covered a wide spectrum of control levels thus adding confusion: from those operated remotely to completely unattended facilities. The misconception has persisted to today, and it is still quite usual to find papers in the literature describing robotic telescopes that have some level of attendance. See, for instance, the description of the dome control system at the NMSU 1 m *robotic* telescope, given by Holtzman et al. (2010), whose opening is synchronized with the dome status of the ARC 3.5 m attended telescope at the same site (Apache Point Observatory). Or the Physics Innovations Robotic Astronomical Telescope Explorer (PIRATE) project, a remotely-operable telescope (see Holmes et al. 2011), whose name clearly defines this as a *Robotic Astronomical Telescope*. These are just two examples that illustrate the common albeit incorrect use of the robotic term to describe attended facilities.

This generic use of the term *robotic* for astronomical observatories has led to some confusion when describing the global characteristics of this kind of facilities and an underestimation of the engineering complexity behind them. The latter has been quite common for those projects that have been based on the acquisition of commercial off-the-shelf (COTS) telescopes described by the manufacturers as *robotic* instruments. The industrial solutions are normally designed to provide automatic acquisition of the objects and, therefore, lots of time and resources are spent achieving their autonomous mode. Furthermore, the results are very poor for fundamental reasons associated with the design of the telescopes. For instance, they hardly comprise a full system, including the enclosure and environment control. In addition, they are commonly based on proprietary technology that does not use standard solutions, which means that they cannot be easily modified to integrate them into the control system of the hosting observatory. These reasons have led, in many cases, to a reduction of the project goals by falling back to a remote- or human-operated system.

1.1.1.1 Constraints for a robotic operation

A ground-based observatory has several constraints when it has to work without human interaction. These constraints can be divided into intrinsic and site-dependent.

Unattended observatories have to emulate human response to all those events that perturb the normal workflow. Any decision has to be planned at the design phase. Malfunctioning of any hardware or software element must be detectable and a suitable response must be available to ensure reliability and safety. Special attention must be paid to those points that could cause damage to critical parts of the system. These issues are quite standard for all the observatories and could be defined as intrinsic constraints, because they are system dependent.

The specific environment at each site also introduces particular requirements. An observatory must have a planned response to any weather variable, for example, that could have an effect on the correct performance of the system or the quality of the registered data. Instrument safety and working efficiency depend on a good response to environment conditions.

The human attendance level of an astronomical observatory is also established by the confidence on the system to respond to environment changes and to possible alarms. This point determines the risk of damage at any time for a particular facility. The robotic operation is reached when human attendance is not required to keep the risk under an acceptable level. This level may also be lower or higher depending on the isolation of the site.

1.1.1.2 Robotic control levels

We can distinguish different levels of unattended control among the projects developed so far. We can group them according to the operations they perform without human interaction and using two different

criteria: the completeness of the unattended operation for the data flow execution and housekeeping control and the level of dynamic response to changing conditions.

Regarding the number of processes executed with no human interaction, robotic telescopes have historically been focused mainly on automatizing the execution of tasks and the processes carried out for the control of housekeeping (i.e., monitor the environment and ensure the system safety). The majority of them execute a list of predefined tasks following a routine operation based on starting up the system, taking data of the science targets and stopping the full system. Other processes involved in the data flow are done manually and/or off-line. The efficiency of an astronomical facility can be much increased with its complete automatization. It should include all the steps from the proposal submission up to the data retrieval by users, and mainly focusing on task scheduling, task execution, and data processing routines. This is especially important for multipurpose and multiuser observatories that handle different types of programs and require a more intelligent management.

The level of the dynamic response of a system to the changing conditions is defined by the type of feedback the system has for any of the running processes and how this feedback is used to automatically react with corrective or reactive actions.

Static (or open-loop) observatories are those that do not react to changing conditions except for those actions executed to minimize the risk under bad weather (i.e., dome closure when it rains). Dynamic observatories (or closed-loop), on the other hand, use the data from the system health monitoring or the quality of the acquired data to run contingency actions to restore any malfunctioning component or to modify the schedule. Dynamic facilities are prepared to react efficiently according to the events (routine and non-routine) providing a high level of feedback.

The majority of the robotic telescopes are designed to perform a simple data flow control, mainly devoted to the execution of tasks, and are based on the static control approach. Their autonomy is quite low, but they are a good solution as they represent a simpler development. They are suitable for telescopes located on non-isolated sites, where human interaction is possible for immediate responses to critical alarms or to reschedule the night planning.

A small number of robotic telescopes use smart (or dynamic) schedulers and/or on-the-fly processing routines to complete the unmanned data flow control. These are still quite unexplored issues for such kind of ground-based observatories, although they have proved to substantially increase the system efficiency in terms of data acquisition and scientific return. And there are also some telescopes that have been built to operate dynamically in order to achieve a high level of autonomy. Some examples of telescopes based on these technologies are: the RAPid Telescopes for Optical Response (RAPTOR, Vestrand et al. 2002), the Bradford Robotic Telescope (BRT, Baruch 2007), the STELLar Activity project (STELLA, Granzer et al. 2010) or the Liverpool Telescope (LT, Steele et al. 2004).

The next step within this technological field is the construction of networks of high dynamic response robotic telescopes. Some examples of this are currently under development mainly based on two different approaches. Within the first group, each network is under the leadership of a unique consortium or institution, avoiding the need to find common technologies or use standard interfaces and protocols to build the network. We can mention three projects included in this group: MONET (Bischoff et al. 2006), Las Cumbres Observatory Global Telescope Network (LCOGT, Pickles et al. 2010), and BOOTES (Castro-Tirado et al. 2012). As for the second, several initiatives are ongoing to define a protocol standardization for the communication of events, the scheduling of tasks, the massive storage of data, etc., among different observatories. They are carried out to enable the collaboration among different telescopes spread all around the world. Among them, we emphasize the eSTAR (Allan et al. 2004), the Heterogeneous Telescope Network (HTN, White 2007), or the Virtual Observatory Event (White et al. 2006). All these projects will *open new windows of scientific discovery* thanks to a coordinated operation of multiple robotic telescopes.

1.1.2 Thesis goals

The TJO-OAdM robotic telescope proposed in the mid-90's represented a technological challenge for the astronomical instrumentation field. It was conceived as a multipurpose telescope with a completely unattended operation. At that time, there were a few facilities around the world already running unmanned telescopes (i.e., BRT – Baruch 2007, Fairborn Observatory¹ - Henry 1995), although most of them were only capable of task execution control (i.e., telescope pointing, data acquisition). These leading projects were also normally not located on isolated sites, thus making human supervision common. And, finally, they were devoted to specific science cases and for a particular community of

¹ <http://www.fairobs.org>

users that did not impose a hard requirement on supporting different operation modes or execution programs (fixed time, monitoring, etc.).

The initial development of the TJO-OAdM system did not fulfill the expected performance. It had several deficiencies mainly caused by a deployment based on COTS instrumentation not suitable for an unattended and reliable control, and an incomplete definition of the system architecture to guarantee the operation of the full facility and the complete data flow processes.

Therefore, and taking into account these previous obstacles, the main goals of this Thesis have been the development of the envisioned robotic telescope based on the original TJO-OAdM project, and providing a system with a high level of autonomy.

The work is focused on applying the changes needed to provide a system capable of giving intelligent responses at any situation (based on the dynamic model) as well as a completely unmanned data flow control. The specifications cover from program preparation and submission to image processing and data retrieval by users. This operational concept imposes a high level of intelligent control to maximize the system efficiency and the use of completely unattended and reliable end-to-end data flow and housekeeping control. Site isolation and harsh environment conditions at the Montsec mountain add a requirement of robustness to the system's development. Its reliability and robustness and the quality of the extracted data are also considered fundamental for a maximization of the efficiency and the scientific return.

Therefore, the Thesis is structured in three main blocks. The detailed operational design and the system architecture constitute the product of the first block and they result from a deep analysis of the original project and a specification of all the requirements for the complete telescope robotization. The approach chosen can be applied to build other robotic telescopes, since it is based on three general layers that cover the main control functionalities (housekeeping, end-to-end data flow and hardware).

The second block is devoted to the design and development of the system. The Robotic Observatory Control System (ROCS) and the implementation hardware are the main components. The work involves the general and detailed design of the control system and all of its modules, and the development of some of the software packages. It is important to mention that the software infrastructure developed in the framework of this Thesis, dubbed OpenROCS (Open Robotic Observatory Control System), is mainly responsible of solving any situation that could threaten the system's integrity (housekeeping control) as well as to administrate the workflow of the observatory processes (end-to-end data flow control). The software is designed on the basis of event-action management scenarios. This software layer connects, in a homogeneous way, the suite of applications developed to execute different particular tasks with the hardware layer, avoiding a monolithic conception of the observatory control system and providing it with a modular and scalable architecture while increasing the system reliability.

This second block is also characterized by the set of hardware components that conforms the hardware layer and several software applications. Two software packages are critical for the system efficiency while working under unattended control: the scheduler and the data processing routines. The scheduler (ISROCS, standing for Intelligent Scheduler for the ROCS) is responsible for the selection of the best task to execute at any time, according to the task execution constraints and the conditions (environment and hardware) at the observatory when computing the selection. It is based on a dispatch-scheduling algorithm and is also used to obtain a static queue of tasks for the entire night or for several nights in order to optimize the point in time when an observation is made. This application is responsible for the system's intelligent response and for maximizing the scientific return of the facility. The data processing routine (called ICAT, for IEEC Calibration and Analysis Tool) performs an on-the-fly processing and quality control of the image right after its acquisition, providing an important feedback to the system to let it decide automatically if the image should be repeated. Final processing of the image is done off-line to increase the quality of the extracted data and to make them available to the end-user with an easy and reliable retrieval procedure. ICAT output can be used to detect deterioration of the scientific instrumentation (i.e., telescope optics). A few additional applications or modules (database, execution control software, proposal management and user interface) are also important for the end-to-end data flow control, although they are mainly used as support tools and they are not exclusive of unattended observatories.

Some hardware and software modules included in the second block of the Thesis have been developed in collaboration with or by other physicists or engineers. This group of scientists, referred as the *engineering team* in the text, has participated in the project during specific moments in its life cycle and has contributed with valuable work. They have led the detailed design and/or the development phases of several packages, which are included in this Thesis for a complete description of the system. This involvement is highlighted accordingly.

The third and last block is devoted to ensure the quality of the implemented system and determine whether it meets the project needs. It involves system-engineering practices, mainly focused on quality assurance processes to monitor and evaluate the work during all its phases. Although these practices have not been exhaustively implemented and applied, they have played a relevant role in this Thesis.

This Thesis has evolved in parallel with several other robotic telescopes around the world based on a similar approach (i.e., LT – Steele et al. 2004, STELLA – Granzer et al. 2010, Faulkes – Smith et al. 2004), which have different levels of completion at the time of writing this document. Although the TJO-OAdM telescope presents several similarities with the previously mentioned projects, in which comparable technologies and designs have been implemented, it is a pioneering project in several other aspects. This encompasses the different tools developed for the management of the proposal data, the intelligent scheduling of tasks, the data processing, which returns a quality index of the acquired data, or the software infrastructure that implements the general control of the system. These solutions are complemented with others devoted to strengthen the robustness (i.e., redundant systems) and to implement contingency actions that increase the system's human independence, which are fundamental for such an isolated site and harsh environment conditions. The emergence of new technical implementations during the work development has also helped to find different solutions.

The features of the TJO-OAdM telescope and its size make this telescope quite singular and unique in comparison with the other robotic telescopes currently in operation (see, for reference, the fairly extensive compilation produced by F. V. Hessman¹).

1.2 TJO – OAdM General Description

1.2.1 Generalities

The TJO-OAdM is a small-class observatory working on completely unattended control.

The Montsec Astronomical Observatory (OAdM) project is supported by the Catalan government and owned by its local official institution: the Consorci del Montsec (CdM). Several research facilities are installed on the OAdM site: the TJO telescope, owned by the CdM and installed and managed by the Institute for Space Studies of Catalonia (IEEC); the Fabra-ROA Telescope at the Montsec (TFRM) of the Real Academia de Ciencias y Artes de Barcelona and the Real Observatorio de la Armada; an IEEC all-sky camera for meteor and bolide detection; a station for the monitoring of airborne particles (contaminants) of the Catalan Government (Dept. Medi Ambient); and a weather station of the Servei Meteorològic de Catalunya (SMC, Catalan Government), one of the existing nodes in Catalonia for weather reporting.

Several research institutes and universities have actively contributed with scientific and technical supervision and manpower to the development of the TJO-OAdM project: the Institute for Space Studies of Catalonia (IEEC²), the University of Barcelona (UB³), the Technical University of Catalonia (UPC⁴), the Spanish Research Council (CSIC⁵) and the Joan Oró Foundation (FJO⁶).

1.2.1.1 Site description

The OAdM is located at an altitude of 1570 m on the Montsec mountain range, 50 km South of the central Pyrenees and 50 km North of the city of Lleida (Catalonia, Spain). Site testing campaigns were conducted during the years 1997-1999, with measurements of air stability, transparency and weather conditions, and seeing. These tests showed that the night sky of the Montsec mountain is suitable for high-quality astronomical observations. Light pollution at the OAdM was found to be very low. The sky background brightness at the zenith in moonless nights averages 22.0 mag-arcsec⁻². This value indicates that this is a particularly dark site.

Weather variables and air stability are being monitored continuously since 2004 (weather) and 2009 (stability) and their evaluation confirms that conditions for astronomical observations remain very good.

¹ <http://www.astro.physik.uni-goettingen.de/~hessman/MONET/links.html>

² <http://www.ieec.cat>

³ <http://www.am.ub.es>

⁴ <http://www.upc.edu>

⁵ <http://www.ice.cat>

⁶ <http://www.fjo.es>

1.2.1.2 Instrumentation overview

The TJO-OAdM is equipped with the 0.8 m Joan Oró Telescope (TJO). It was supplied by Optical Mechanics Inc. (OMI) and installed in March 2004. The 6.15-m dome was manufactured by Baader Planetarium GmbH and is fully automatic. The first light instrumentation is a 2048x2048 ProLine Finger Lakes Instrumentation CCD camera, with a CCD42-40 Marconi back illuminated chip with a pixel size of 13.5 μm , and a FOV at the Cassegrain focus of 12.4x12.4 arcmin (0.36 arcsec-pixel⁻¹). The telescope is also equipped with an automatically controlled 12-position filter wheel, and the 5 filters of the Johnson-Cousins photometric system (U, V, B, R, I). Several instruments for environment monitoring are acquiring data continuously: two weather stations, a GPS antenna, a storm detector, etc. A radio link antenna with 10 Mbps bandwidth provides external communication necessary for remote control. This connection will be substituted by a fiber cable link in the coming months, which will significantly increase the available bandwidth.

A complex software architecture manages all observatory operations. Basic telescope and dome control is conducted through the TALON software.

As complementary instrumentation, an all-sky CCD camera is operating to detect bolides and meteors. As a mid-term objective, its images will be used to generate cloud maps that will allow the selection of clear areas in the sky.

1.2.1.3 Operation mode

Robotic operation is a hard requirement for the observatory routine use, due to the science case needs and the isolation of the site. Human interaction is only foreseen to provide observing targets and to retrieve the data and acquired images. The primary guideline of the project on the design and development phases has been to guarantee such operation mode for the collection of high-quality astronomical observations.

1.2.2 Science Case

The TJO-OAdM is a general purpose facility and, as such, will carry out a variety of observations related to various science cases (Isern et al. 2000). Given its size, the main niche for TJO-OAdM is the time-domain astronomy, where high-cadence, continuous observations are the primordial requisite. It is obvious that TJO-OAdM cannot compete with the current largest telescopes, which have a collecting area 170 times greater. However, its advantage is a flexible operation mode allowing for the monitoring of sources for extended time periods and also the possibility of a rapid reaction time, potentially as short as a minute or less. Given such features, the possible science cases for TJO-OAdM include:

- Exoplanet research (possibly follow-up of known transiting planets or targeted searches of individual objects).
- Eclipsing binaries (to understand stellar properties and structure).
- Pulsating variables (probing the stellar interior).
- Evolved variable stars (supergiants, post-He flash).
- Stellar activity (to understand the magnetic dynamo and to calibrate the time-decay of such activity).
- Variability of active galaxy nuclei (related to the stochastic accretion process).
- Solar System objects (follow-up of asteroids, near-Earth objects, comets).
- Supernovae (with the added value of obtaining early photometry).
- X-ray binaries (rotational variability and accretion phenomena).
- Novae (also with possible early data).
- Optical counterparts of Gamma Ray Bursts (GRBs).
- Any transient phenomena in general.

The science cases above require important flexibility in the night scheduling, which allow the system to react rapidly to observational alerts related to GRBs, new supernovae, and similar time-critical events. The participation to the networks of robotic observatories is considered to carry out, for example, observations requiring continuous time coverage. Similarly, the TJO-OAdM can be used as a support facility for space missions to collect photometric and astrometric data. An obvious example is the Gaia mission (with prominent participation of the IEEC), which is scheduled for launch in 2013 and for which ground-based support (pre- and post-launch) will be an important contribution. The TJO-OAdM will offer time to the astronomical community via competitive proposals peer-reviewed by a Time Allocation Committee (TAC).

1.3 Thesis structure

This Thesis is divided into three main parts. The first part (Chapter 2) is focused on the general analysis of the initial TJO-OAdM concept to describe the project motivations and to illustrate the conditions under which the Thesis started and has evolved. We first provide a description of the TJO-OAdM from a historical point of view, and we continue with a formal presentation of the robotic operational concept. A list of requirements was specified to correct the system and achieve the one envisioned. The **qualified robotic control** concept is introduced to summarize the high level requirements for this telescope. The detailed operational design and the system architecture presented in Chapter 3 are based on these requirements and represent the final step of this initial part.

The second part describes the work achieved to deploy the designed architecture, including an overview of the physical implementations (Chapter 4), the control system for the robotic control (Chapter 5 and Chapter 6) and, finally, the different applications that complete the software suite in charge of most of the control operations (Chapter 7 to Chapter 12).

The third and last part (Chapter 13) includes those items related with quality assurance: project life cycle, documentation, verification and validation, and testing procedures to check the system's performance are described.

Finally, the conclusions of the Thesis are presented in Chapter 14.

Chapter 2

Qualified Robotic Control at the TJO-OAdM

In this chapter a general analysis of the observatory operation is presented. It illustrates how the TJO-OAdM telescope was conceived from the very initial phases to provide the science return envisioned and described in the science case (Sect. 1.2.2). The general requirements derived from this operational analysis are introduced in the last sections and define the work carried out in this Thesis.

The evolution of the project from its inception to its final implementation is presented first. It is important to describe the different stages the project has undergone and the impact these stages have had on its design and development.

2.1 Historical perspective

Multiple factors, institutions and a considerable amount of people have contributed to the TJO-OAdM project development for more than fifteen years. The construction process can be divided into two main phases that may be named the “pre-robotization project” and the “robotization project”; the latter being the phase during which most of the work of this Thesis has been done. These phases were not defined a priori, and they are only introduced here to justify the initial conditions of this doctoral work. The selection process of several of the observatory components has not been included in the Thesis work since these were done during the first phase. The astronomical instrumentation (telescope, dome and instrument), selected to address the science case and to enable the automatic operation, is an example of such initial constraints.

The following subsections illustrate the project life cycle.

2.1.1 Pre-robotization project

After conducting site-testing measurements on the OAdM site, the project to build an astronomical research facility was initiated in the late 90's.

The initial design of the system architecture (Isern et al. 2000) was mainly determined by the science case goals. The performance of the telescope optics and mechanics and the instrument features were clearly defined, and there was a general description of a basic set of components for the control system.

The development carried out during the pre-robotization phase was focused on the selection and acquisition of COTS equipment that fulfilled this general design. At the end of this phase, the full set-up was mainly based on the 0.8-m telescope manufactured by OMI and the TALON software for the management of the observatory operations, provided by the same company. A classical dome equipped with an electronic interface and a CCD camera completed the science instrumentation. Other elements that constituted part of this initial system were the computers programmed to run the control system and store the data as well as the support devices to monitor the environment conditions, to generate electrical power, and to provide internet connection. The civil work was also

carried out during this phase, which included the definition of the communication and electrical wiring to interconnect all the equipment on-site (Colomé et al. 2003).

Several deficiencies of this initial stage were detected during the system commissioning. They were a consequence of an incomplete definition of the control architecture and several design issues of the COTS devices. For instance, the OMI telescope and the software TALON promised reliable unattended control at the time of purchase, but it was noticed during early stages of the project that a number of aspects had to be improved and new features had to be added to achieve a reliable, safe, and efficient robotic control. This occurred right after the inadequate and faulty installation by the manufacturer, which implied subsequent extensive efforts to reach proper working conditions.

The system provided by the telescope manufacturer, for example, did not carry an application to control the dome nor any redundancy in case the main dome control channel failed; just a predefined small number of environment sensors could be connected, without the possibility of adding a redundant weather station, a storm detector, etc.

TALON also proved to be very unreliable and incomplete from the very beginning. It was developed to control astronomical observatories and planned to be in charge of the entire TJO-OAdM robotic operation control. But it was not ready to be used as the control software for an entire observatory, nor as software for unattended control. Robust (including redundancies) housekeeping monitoring or dome control were not implemented, and efficiency in the end-to-end data flow was really poor. Several examples support this statement:

- There is no data processing routine to check for comprehensive data quality in real time.
- TALON uses a simple queue scheduler to sort the observations before the beginning of the night, without real-time response capability to any kind of alarm (i.e., environment changing conditions) or feedback from the acquired data quality.
- There is no management of the proposal information for an efficient traceability of its status or data archiving.
- There is no interface for complex proposal submission.
- There is no interface for data retrieval for users that do not have access to the observatory archive.
- TALON does not store all relevant information into the image header (proposal identifiers, image type, etc.).

The evidence of the unsuitability of TALON was, actually, the first motivation for the development of the new observatory control system presented in this Thesis: a new control system had to be developed to act as a superstructure to cover all missing features of TALON. The software weaknesses derived also in new requirements on hardware devices needed for robotic control.

Therefore, the mentioned shortcomings and other required modifications (see Chapter 4 for a detailed description) made the system unsuitable for the unattended and reliable control and motivated the definition of the robotization project.

2.1.2 Robotization project

Identifying the additional elements needed to achieve a reliable and completely unattended system was possible thanks to a revision of the telescope operational concept (see Sect. 2.2) and an analysis of the initially developed system. These elements were specified in a requirements document for the complete robotization of the TJO-OAdM. These requirements can be found in Sect. 2.3 and constitute the basis for the robotization project.

The IEEC submitted the robotization project to the Generalitat de Catalunya (Catalan government), through its local representative CdM in late 2006. It was approved in 2007 and IEEC received a special funding to execute this task. Dr. Ignasi Ribas was the Principal Investigator of the project up to its completion, in April 2010, after a System Acceptance Review carried out by an independent panel.

Routine operations of the TJO-OAdM telescope started at the end of 2010, although a significant fraction of the time was still invested in a second commissioning of the equipment to check the system performance in terms of control and acquired data quality. This process has permitted the adjustment of the system configuration and the identification of improvements done on hardware and software implementations that are under development in parallel with night operations.

2.1.3 Chronological description

Some events are used here to illustrate the chronological description of the project phases. Milestones are highlighted in bold text.

2.1.3.1 Timeline of the pre-robotization project

- Analysis and design phase
 - 2000: The project definition was established, which included high level requirements and a preliminary and high level design (Isern et al. 2000). It was submitted to the Catalan government, who approved its funding.
- Development phase
 - 2000-2004: This phase included the selection and purchase of the science instrumentation and support equipment, the definition of the communication and electrical wiring, the detailed civil work project, and the web page for the project outreach.
 - 2002: The civil work to build the TJO building started.
- AIV phase
 - 2004: The civil work ended, although some important shortcomings were detected, mainly related to power generation and distribution, and building insulation.
 - March 2004: The **telescope was installed** at the observatory.
 - August 2004: A direct lightning impact produced severe damage to the observatory that required the replacement of most of the electronic boards of the telescope, some electronic components of the dome, the main control computer, and the router. This major incident motivated a deep analysis of the system and the installation of multiple protections to filter any overcurrent produced by electrical storms.
- Commissioning phase
 - 2004-2005: Important shortcomings on the telescope and TALON were detected.
 - 2005-2007: Actions to improve the system were carried out during this period, although the system stability was still very poor. Commissioning work for the astronomical characterization of the observatory was conducted, although very few results were obtained due to the telescope operation difficulties.

2.1.3.2 Timeline of the robotization project

- Analysis and Design phase
 - 2006-2007: The analysis of the initial system shortcomings, the definition of requirements for upgrade, and the definition of the robotization project (Colomé et al. 2006) were conducted in this period.
 - November 2006: IEEC submitted the robotization project to the Generalitat de Catalunya.
 - Early 2007: The **robotization project was approved**.
 - 2007-2009: The high-level design of the new system architecture was carried out during 2007. The detailed design for each of the layers and modules of the new architecture started in 2007 and ended in 2009.
- Development phase
 - 2008-2010: The development of the system modules (hardware and software) was conducted in this phase.
- AIV phase
 - AIV is done gradually as each control module development ends, increasing the number of unattended operations every time a new one is added. Unit testing is performed before the integration.
 - April 13th 2010: A **System Acceptance Review** (SAR) was held to evaluate the system readiness once the development was completed. The external panel for the system evaluation was composed by: M.I. Andersen (Niels Bohr Institute), D. Galadí-Enríquez (Centro Astronómico Hispano-Alemán, CAHA) and F. Herrada (Spanish Science and Innovation Ministry). A small portion of the development and verification phases was still ongoing at that time and subject to the new commissioning phase.
- Commissioning phase
 - September 2010 – October 2011: During this period, the commissioning of the unattended operation and system was performed. Some defects on the optical and mechanical parts of the telescope and the dome were detected after the SAR had taken place, and, therefore, actions to correct them had to be performed. The control system software was also upgraded to increase its reliability, mainly when handling events and alarms of the housekeeping control layer.
 - September 2010 – October 2011: Science observation programs and calibration tasks were done using both supervised robotic operation (global control without human

intervention, but with people on-site supervising the process) and interactive operation (some processes being executed by people on-site) modes. This work was done in parallel to the commissioning tasks, investing a significant fraction of the total time. The first scientific results were published (Herrero et al. 2011).

At the time of writing this Thesis, a final phase is foreseen and it is under design:

- Routine operation phase
 - November 2011: Routine fully unattended robotic operations start in this phase.
 - May 2012: A **System Operation Review** will be held after 6 months of fully unattended operation in order to evaluate the success level of the final implementation.

2.2 TJO-OAdM operational concept

A description of the operational concept of the TJO-OAdM telescope is introduced in this section derived from the initial general design of the project given in Isern et al. (2000).

The operational concept is used to formally describe how the observatory is planned to perform in its operational environment and to provide procedural information regarding the sequencing of activities during routine (before, during and after an observation) and non-routine operations.

2.2.1 Observatory operations

Several control, operation and scheduling modes are considered to carry out the tasks. They are defined and described in this section.

2.2.1.1 Control mode

TJO-OAdM has multiple science cases to cover, including transient phenomena. This kind of multipurpose observatory requires the capability to change the observatory scheduling in real time and maximizes its efficiency by locating the decision taking capability in a control system developed to work without human interaction. Most processes included in the different task routines are defined to be executed autonomously, which is defined as **robotic control operation**. This mode of operation is mandatory for routine use of the observatory due to the isolation of the site.

2.2.1.2 Operation tasks

The tasks supported by the observatory and to be managed by the control system are:

- **Science observations:** The proposals submitted to the observatory define a set of science observations to obtain images and data in order to fulfill a specific science case.
- **Calibration tasks:** Calibration processes are necessary to reduce the data obtained with the science observations.
- **Engineering tasks (maintenance and tests):** Periodic maintenance tasks are performed to minimize the risk of unexpected malfunctioning of any subsystem component. Testing is also eventually foreseen to prepare system upgrades or to evaluate new instrumentation.

2.2.1.3 Operation modes

The observatory was conceived in the original project to operate with a small-class telescope, a classical dome and two main instruments: a photometric imaging camera and a spectrograph. The initial development was done to have the imaging camera operative with the start of the routine operations. Once this step is accomplished, the implementation of the second instrument will follow. Only the operation mode for photometric data acquisition is available presently. The integration of other operation modes is out of the scope of this Thesis, although adding new instrumentation could be done in a straight forward manner thanks to the modularity of the control system design, one of its main assets.

Common science devices to all the operation modes are the TJO telescope and the dome, which are described in Chapter 4.

MEIA: Photometric imaging with the MEIA CCD camera

Only this instrument is currently used to perform the operational tasks following any of the control modes. MEIA, which stands for Medium-format Efficient Imager for Astronomy, is composed of a CCD camera, a set of photometric filters and a specific calibration set-up and it is used to obtain photometric and astrometric data.

Future modes

A second mode of operation will be defined in the future when the new instrument ARES (Astronomical mid-REsolution Spectrograph) is installed at the observatory.

And, finally, a third control mode could be defined in case an autoguiding camera is integrated in the control system, providing the possibility of selecting the mode of operation with or without the autoguiding functionality.

2.2.1.4 Observation preparation

Science observations are entered into the system by submitting a proposal to the observatory. Each proposal submission process is divided into three different phases: submission of the general description and the detailed observation pattern, revision and verification of the submitted data and, finally, data retrieval.

A basic set of calibration images are also defined to be used by the system for automatic reduction and analysis of the science images. They are also delivered to the user to enable for an alternative image reduction process.

2.2.1.5 Task scheduling

An automatic scheduling routine selects the most suitable task/s to be performed by the observatory, according to predefined criteria and with the goal of maximizing the time invested in science operations. The probability of having a proposal executed should depend on its scientific priority, but also on the conditions defined by the observational constraints.

2.2.1.6 Task execution

The execution of a science observation or a calibration task includes the monitoring and control of all the hardware involved (telescope, instrumentation and dome) and monitoring of the environment data. Regarding hardware control, it includes pointing and tracking, filter and focus positioning, image acquisition, and housekeeping tasks that ensures the system safety in case of bad weather conditions.

2.2.1.7 Data processing and storage

Every time a new science or calibration image is acquired, a process for its reduction and analysis is executed. A quality check is then performed to determine if the image is obtained in compliance with predefined criteria, and to provide feedback to the observatory control system. Images and data extracted are finally stored.

2.2.2 End-to-end data flow

The observatory end-to-end data flow is illustrated in Fig. 2.1. The diagram represents the sequence followed when a new proposal enters the system and, also, the sequence followed when the control system performs the nominal run workflow for the observations associated to the approved proposals. Routine operations that are not linked to a specific proposal (calibration and maintenance tasks) follow the same workflow scheme, but their associated tasks are entered into the system by the observatory operators only once and they are permanently active except for updating purposes.

The diagram in Fig. 2.1 is also useful to illustrate the different applications involved in the frame of unattended actions. Human-controlled and unattended processes have to be identified, and also the role they play in the end-to-end data flow control has to be defined. This helps defining the framework in which the control system must operate and the boundaries of the robotic system responsibilities.

Regarding human interaction, the observatory nominal working process implies the following processes with human interaction:

- Submission of proposals (Phase 1).
- Proposal revision and verification (Phase 2).
- Data retrieval (Phase 3).

And the following unattended processes:

- Storage of proposals in the observatory database.
- Mirroring of the database at the observatory site facilities.
- Operations control (science and calibration image acquisition), including:
 - Observatory start-up and power down routines.
 - Selection of targets in two steps: a first pre-selection process and a real time selection, based on a dispatch scheduler.

- Hardware control for telescope, instrumentation and dome to perform the task execution.
- Data processing and storage.
- Housekeeping, hardware and software application monitoring and control.
- Data backup.

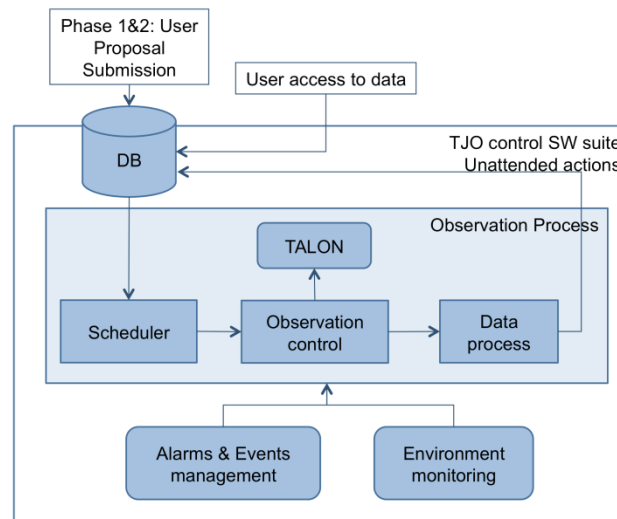


Fig. 2.1: Data flow at the TJO-OAdM

2.3 Robotic control requirements

The described operational concept implies that the telescope needs to operate under robotic control and with a high level of feedback from the system to apply corrective actions or change the schedule of tasks when necessary.

Taking into consideration the operational concept and the initial set-up established by the end of the first phase, we defined a list of requirements in order to develop the new TJO-OAdM observatory control system and fix the errors and malfunctioning of the available components. These are specified in this section, grouped in high- and low-level requirements.

“Shall” and “should” are used in each requirement description in the following manner: “shall” is used for requirements whereas “should” is reserved for guidelines; requirements are mandatory and guidelines are not mandatory, although their fulfillment should be strongly pursued.

2.3.1 High level requirements

Four characteristics summarize the robotization of the TJO-OAdM: reliability, robustness, efficiency and quality. They have been considered as high-level requirements from the design to the development phases. These important concepts are described here and mentioned throughout the Thesis any time the selection criterion between possible solutions has to be justified.

2.3.1.1 Reliability

Reliable systems always respond to a specific event with the same predefined action. High reliability systems must ensure the predefined performance with a high level of confidence and for a specific lifetime.

The TJO-OAdM control system design shall ensure a highly reliable unattended operation.

2.3.1.2 Robustness

The functionalities of a robust system do not diminish when being affected by disturbances. Risk level is minimized in robust operation systems.

Solutions implemented at the TJO-OAdM for hardware and software shall be selected and/or developed mainly considering this important feature.

Environment conditions at the Montsec mountain are very unfavourable for electronic devices due to thunderstorms, more frequent in Summer months, high humidity or the accumulation of snow during

Winter. Special care shall be taken to ensure that the installed equipment is prepared to operate in such conditions.

2.3.1.3 Efficiency

The output of an efficient system is maximized whatever the operation conditions are.

Scientific return is the output parameter considered at the TJO-OAdM.

The system efficiency shall be maximized or, at least, optimized at the TJO-OAdM. Some key aspects can have an impact on this and shall be included in the system design and development: the system response to every specific event, an efficient control system that enables an optimized selection of the task to execute at any time (goal of the scheduler), quasi real-time processing of the acquired data, an Exposure Time Calculator to be used by multiple applications (proposal handling, scheduler, data processing), etc.

The efficiency can be measured in terms of the time invested to perform science observations compared with the available night time for that purpose. The value resulting from this comparison shall be also weighted by taking into consideration the quality of the acquired data. The latter is important mainly to ensure that the system not only executes the tasks in an efficient way, but it also retrieves data with the expected quality.

2.3.1.4 Quality

We consider a good quality operation when the overall system performs in a reliable, robust and efficient way. This operation quality is strongly linked at having good quality software (high and low level) and also good quality electronics for instrument control (telescope, dome, detectors, housekeeping, proposal and data management and processing, etc.).

However, we also measure the quality of the system in terms of the quality of the obtained scientific data. The latter involves not only the mechanical and optical performance of the telescope, whose stability and accuracy is critical, but also the way all the proposals are inserted into the system and the way data are extracted from the acquired images and processed for user retrieval and to be employed as a feedback by the control system (end-to-end data flow).

The robotic control envisioned at the TJO-OAdM could then be summarized with the following global requirement:

TJO-OAdM shall achieve a qualified unattended operation that implies having a good quality operation together with extracted data of good quality.

The chapters of this Thesis contain the description of the aspects related with the operation and data extraction procedures to show how the unattended operation has been achieved in an optimized way to maximize the quality at the TJO-OAdM.

2.3.2 Low-level requirements

The detailed list of requirements provided in this section was elaborated for the definition of the new control system.

Each requirement is labelled with a prioritization index (from 1 to 5 increasing priority) describing its level of impact to achieve a robotic control system with the high level needs described above. In the following chapters expanded lists of requirements are provided, if necessary, to describe the development of specific modules.

Some modifications on the hardware included in the initial observatory setup (pre-robotization project) are mentioned as Development requirements (Sect. 2.3.2.8). We do not include in this section those hardware devices that did not require any modification to be adapted to the new observatory control system. All the hardware components (initial ones and new developments) are described in the architecture overview given in the following chapters.

2.3.2.1 Functional requirements

FUN-1. Observatory operation: The robotic control system shall be designed and developed to operate the TJO optical telescope, initially composed by a Ritchey-Chrétien 80cm telescope, a Johnson-Cousins filter set (U, B, V, R, I), a mid-size CCD camera and a dome of 6.15-m in diameter. All these instrumentation was originally designed to be operated automatically. *Priority: 5*

FUN-2. Scientific return: The TJO telescope shall be capable of executing the scientific cases described in Section 1.2.2. This requirement is strongly dependent on the telescope and the instrument features that shall be kept during the entire observatory life time. <i>Priority: 5</i>
--

2.3.2.2 Performance requirements

PER-1. System efficiency: The system shall maximize the scientific return. <i>Priority: 5</i>
PER-2. Operation overheads: The operation overheads shall be minimized. Operation overheads besides telescope and dome pointing, target acquisition and detector readout should be zero or very small (< 10 seconds). <i>Priority: 5</i>

2.3.2.3 Quality requirements

QUA-1. Housekeeping control: The system shall be reliable and robust for the observatory housekeeping control. It shall have redundant equipment at the identified single-points-of-failure. <i>Priority: 5</i>
QUA-2. End-to-end data flow: The quality of the images acquired shall be checked as a feedback control for the system operation. <i>Priority: 5</i>
QUA-3. Science data quality: The robotic operation control system shall not produce any decrease of the quality of the data. Data quality shall be only affected by mechanical and optical telescope performance. <i>Priority: 5</i>
QUA-4. Qualified operation: The control system shall have the ability to check the quality of the system operation and the quality of the images. <i>Priority: 5</i>
QUA-5. Efficiency, reliability and robustness: The system shall operate efficiently, reliably, and with a high level of robustness. Hardware and software development shall be done accordingly. <i>Priority: 5</i>
QUA-6. Maintainability: Maintenance cost shall be minimized. The system shall be serviced once per month or less with operations that imply human presence on the observatory. <i>Priority: 5</i>

2.3.2.4 Operation and environmental requirements

OP-1. Observatory operation: The observatory shall be operated without human intervention. Any reaction to unexpected or to routine events shall be done automatically. <i>Priority: 5</i>
OP-2. Time to repair: In case of exceptional system control failure, palliative solutions shall be executed to ensure the system's safety for more than 120 minutes. The system shall be capable of sending external alerts with status reports to the observatory staff. <i>Priority: 5</i>
OP-3. Observatory modes: Science, calibration, commissioning and engineering operations shall be accommodated without any controlling conflict. <i>Priority: 5</i>
OP-4. Observatory users: The robotic (unattended) and the superuser (human attended) controllers shall be the two possible operators of the system. The robotic controller shall be the routine operator for science and calibration tasks, whereas the superuser controller shall be activated to perform commissioning and engineering tasks. <i>Priority: 5</i>
OP-5. Human interaction with the observatory: The observatory nominal working process shall have human interaction for the submission of proposals, for the proposal revision and verification, and for data retrieval tasks. And it shall have the following unattended processes: proposal storage in the observatory database, database mirroring at the observatory site facilities, observation control, and data backup. <i>Priority: 5</i>
OP-6. Observation control processes: The following shall be included: observatory start-up and power down routines, selection of targets (pre-selection process and real time selection), hardware control for telescope, instrumentation and dome, data process and storage, housekeeping, hardware and software application monitoring and control, and data backup. <i>Priority: 5</i>
OP-7. Environment: The system operation shall be prepared to work on extreme environment conditions. It shall be designed to cope with high humidity (>99%, with condensation), low temperatures (< -10 C deg), dust, and electric storms. <i>Priority: 5</i>

2.3.2.5 Control system requirements

CTR-1. Global management program: The TALON software supplied with the telescope is extremely limited and rigid when it comes to the full control of an observatory. A new scalable and fully configurable system is needed for reliable robotic operation. A new software suite shall be designed and developed. It shall act as an envelope to the TALON software and take over some of its functionality to improve reliability and flexibility. <i>Priority: 5</i>
CTR-2. TALON usage: TALON shall be the application to use for telescope and instrument control. Some functionalities shall be added (mirror cover control, dome control), improved (pointing and tracking) or removed (environment monitoring, management of the housekeeping associated alarms, proposals management, data processing and scheduling). <i>Priority: 5</i>
CTR-3. End-to-end data flow control applications: The control system shall contain several applications to execute a complete end-to-end data flow process and for complete hardware control. These applications should be adapted from existing tools to minimize development time, although completely new developments could be done if necessary. <i>Priority: 5</i>
CTR-4. End-to-end data flow processes: The end-to-end data flow shall include proposal preparation and review, scheduling of observing proposals, execution process control, calibration and imaging, data processing, data delivery and archiving, and archival research. <i>Priority: 5</i>
CTR-5. Event and alarm generation and response system: The system needs to identify and react to a number of circumstances. An event and alarm generation software shall be developed to monitor all relevant environment variables, telescope/dome operation, communications, computers, etc, to detect faulty behaviour, to issue system-wide events and alarms according to a defined protocol and to perform the necessary actions to preserve the integrity of the system. The actions shall be performed in an efficient way to avoid conflicts and ensure the system operation quality. <i>Priority: 5</i>
CTR-6. Housekeeping: The system shall have complete knowledge of the housekeeping status and it shall have the capability to execute actions to reduce risk in unattended operations and to optimize the system performance. <i>Priority: 5</i>
CTR-7. Scheduling: The TALON software comes with a basic static scheduling system (telsched). The control system shall provide scheduling services that select the optimum target at each point in time owing to the consideration of different factors (e.g., phase coverage, environment information, and brightness). A new static scheduler with more functionality (calibration images, additional information for the image headers, launch reductions, etc) shall be developed. An optimized dispatch scheduler for optimal use of telescope time and maximum science output should also be developed. <i>Priority: 5</i>
CTR-8. Data processing: The TALON software does not perform data processing of sufficient quality (just a very basic flat fielding, dark current and bias corrections and image analysis). The control system shall provide data processing routines to perform image calibration, analysis, quality control, data archiving and storage. A full reduction and analysis software suite shall be developed. The software shall calibrate the images (co-adding necessary bias, dark and flat field frames), perform preliminary calibration on the science images, extract astrometry and photometry and calibrate it for all point-like sources in the field. The data products should be analysis-ready time-series astrometry (WCS) and photometry. The pipeline shall perform data quality analysis that shall feedback the scheduler to decide on the successful acquisition of science data according to internal quality parameters and quality criteria provided by the user. <i>Priority: 5</i>
CTR-9. VO compliance: Archiving shall be compliant with VO standards. <i>Priority: 4</i>

2.3.2.6 Boundary conditions

BOU-1. Observatory control: The system shall have access to all the housekeeping data in real time to trigger the corresponding actions to solve any critical situation. Data extracted from the acquired images shall also be used to characterize the system performance and react accordingly. <i>Priority: 5</i>
BOU-2. Observation constraints: Each user shall provide all the observation constraints in order to let the automatic scheduler select the best operation to execute at any time. <i>Priority: 5</i>
BOU-3. Observatory output: The control system shall provide all the necessary information to characterize the status of the observatory in real time, to check the system performance for

a specific period of time and compute statistics, and to retrieve the user with all the data and images associated to a proposal. *Priority: 5*

2.3.2.7 User interface requirements

USR-1. **Proposal management tool:** The only tool for the users to interact with the observatory shall be a high-level software interface providing ways to access the database and manage the target list. The users shall be able to propose their science programs (phase I), prepare the observations (target data, constraints, observing modes, integration time) if the program is approved (phase II) and retrieve their data products upon successful observation (phase III). *Priority: 5*

USR-2. **Observatory status overview:** The superuser shall have a graphical interface to have a status overview of the observatory and easily detect any malfunctioning system. *Priority: 3*

USR-3. **Observatory configuration:** The superuser shall have a graphical interface to modify the system configuration in a straightforward manner. Configuration shall include the observatory control system and all the associated applications. This interface shall be unique. *Priority: 3*

USR-4. **Pipeline interface:** There shall be a graphical interface for image processing based on the pipeline used in the data processing routine. The user shall be capable of reproducing the image reduction and analysis with a manual configuration of the process. The output of the process shall be stored temporarily during the user session life time. *Priority: 2*

2.3.2.8 Development requirements

➤ Original instrumentation

DEV-1. **Initial configuration:** The instruments in the initial setup (telescope, instrument, dome, support devices) shall be included in the new global control system developed. A particular device should only be replaced or modified when its reliability is significantly increased with no decrease of the system performance. *Priority: 5*

➤ Telescope

DEV-2. **Stray light:** Original observations showed a high level of stray light. Because of design problems, the original baffling system was inadequate. A new baffling system in accordance with the mirror cover design shall be developed to eliminate stray light (down to a level not higher than the background). *Priority: 5*

DEV-3. **Exposure of critical elements to open environment:** The entire power system was installed by OMI at the pier of the telescope with no protection and operated by manual custom-made switches with no insulation. The RA and DEC axis encoders were left unprotected. A redesign and installation of the telescope power system in a safe location with no open connections shall be done. Electronic and optical elements shall be protected from the environment. *Priority: 5*

DEV-4. **Counterweighing:** The telescope was left by OMI with inadequate counterweighing. A new scalable counterweighing system shall be designed and implemented with minimum distance from the optical axis of the telescope and preserving cylindrical symmetry around such axis so that non-symmetric flexures are avoided. *Priority: 5*

DEV-5. **Mirror covers:** The covers were provided by OMI but their control was not integrated in the system. The mechanical design had serious flaws that posed a serious threat to robotic operations. Design, manufacturing and implementation of a robust and reliable mirror cover system shall be done and it shall include the necessary electronics and software for their operation. *Priority: 4*

DEV-6. **Telescope pointing:** The pointing model had low accuracy and a deficient mathematical modelling led to large deviations close to the North Equatorial Pole. A new pointing model, potentially with learning capability (i.e., the addition of pointing corrections automatically within the system), shall be developed. It shall provide a pointing accuracy better than 18 arcsec throughout the sky (1/40 of the entire nominal FOV). *Priority: 3*

DEV-7. **Focus temperature dependence:** The value of the focus of the telescope depends on various factors, most notably the ambient temperature. The temperature dependence of the focus of the telescope should be characterized. *Priority: 3*

DEV-8. **Telescope finder:** A finder telescope system with a high-speed camera should be installed and integrated in the system. *Priority: 1*

➤ **Dome**

DEV-9. **Reliability of control electronics:** The dome control electronics supplied by Baader were subject to malfunctioning and failures of some specific components (relays and integrated circuits). Being the dome a critical component, this is not acceptable. A new electronic control system shall be designed and developed by dropping the external IR remote control and using instead push buttons. Optical coupling is mandatory to avoid overcurrents and failures. *Priority: 5*

DEV-10. **Encoder:** The encoder provided by Baader suffered frequent total failures probably caused by the lack of shielding. The system (mechanics and electronics) shall be adapted to install a shielded encoder for reliable operation. *Priority: 5*

DEV-11. **Integration of the dome control system:** The telescope software control TALON was supplied without the possibility of controlling a dome (in spite of the requirement at the time of purchase). The software (low and high level) necessary to control the dome (opening, closing, rotation) from the TALON software shall be developed. *Priority: 5*

DEV-12. **Redundant control:** Being the dome the most critical element of the observatory (essential to preserve the integrity of the telescope and electronics), it is crucial to ensure a certain level of redundancy in its control. Electronics, software and computer solutions to provide extreme reliability dome control (only shutter closing) shall be designed and developed. The system needs to ensure that the dome will be closed in case of weather alarm or rain detection. The latter is deemed so important that a hardwired direct connection is required to the rain sensor. *Priority: 5*

➤ **MEIA CCD camera and filter set**

DEV-13. **Vignetting of the CCD FLI4240:** The CCD was supplied by FLI with significant field vignetting at the detector edges. This was deemed unacceptable. A new casing shall be provided so that the field vignetting is totally suppressed on the CCD chip. *Priority: 5*

DEV-14. **Flat fielding system:** The CCD requires of flat field images for calibration. The optimal use of sky flat or dome flats shall be tested. If the latter are selected, a flat-fielding system in the dome (screen, illumination, remote control) shall be designed and developed. *Priority: 5*

DEV-15. **Characterization of CCD:** Flat fielding measurements, dark current values, optimum working temperature, shutter map were not available for the CCD. A fully characterization of the CCD camera for proper analysis of the resulting images shall be performed. *Priority: 4*

DEV-16. **Characterization of UBVR filter set:** The filters were not characterized (i.e., their transformation coefficients to the standard system and necessary focus offsets). The filter set shall be fully characterized with transformation coefficients to the standard system and focus offsets with respect to the nominal position (e.g., for the V filter). *Priority: 3*

DEV-17. **Use of the Apogee Alta U16:** Use of alternative cameras should be made available, starting with the Apogee Alta U16. No software driver compatible with TALON was available for this camera and a new one should be developed. *Priority: 2*

➤ **Support devices**

DEV-18. **Environment control system I:** The original telescope supplied by OMI included a weather station and GPS as the sensor set. Many additional sensors are needed to ensure that the telescope operates safely. Several new environment sensors shall be installed and the software for their monitoring and control shall be developed. Redundant weather data shall be cross checked to detect any malfunctioning operation of a specific sensor. New instruments to install and configure: a redundant weather station, a cloud sensor, a rain sensor. *Priority: 5*

DEV-19. **Power supply and communications:** The TJO-OAdM is prone to electrical thunderstorms during Summer months. Experience acquired over several years demonstrates that direct lightning hits take place at least every other year. Protection against power surges and an uninterruptible power supply (UPS) system are a must for a robotic observatory such as the TJO-OAdM. A set of UPS systems supplying the necessary power to the telescope, dome, computers and electronics shall be installed and configured. Surge filters shall be installed at all cable connections of all electronic devices and power lines (serial port

connections, USBs, Ethernet, etc). To further strengthen the reliability, optical connections should be used within the subsystems that are physically separated by more than 2 m. Fiber optics couplings are suggested. <i>Priority: 5</i>
DEV-20. Remote monitoring: Two webcam devices for visual inspection of the dome and building shall be installed for remote monitoring. <i>Priority: 4</i>
DEV-21. Security: An intrusion security system shall be installed and integrated. <i>Priority: 3</i>
DEV-22. Environment control system II: An All-sky camera and a robotic seeing monitor for cloud and air stability monitoring, respectively, shall be installed and integrated. <i>Priority: 2</i>

➤ Control system

DEV-23. Power control: The turning on and off of electrical devices is critical to the operation of an unattended facility. Power resets are often needed to solve issues caused by the electronics (e.g., USB port communications often need resets). Hardware and software necessary for computer-controlled operation of power switches shall be used. A distributed power switch system that enables selective shutdowns and restarts shall be designed. <i>Priority: 5</i>
DEV-24. System modularity: The control system shall be clearly separated from the hardware implementation to easily adapt it to new hardware devices or to be used at other astronomical observatories. <i>Priority: 5</i>
DEV-25. Software infrastructure: The control system shall be based on a software infrastructure easy to configure and easy to tune to improve its performance after the system calibration. <i>Priority: 5</i>
DEV-26. User interface development: The proposal management tool should be a web-based interface. The entire user graphical interface should be developed as a web interface. <i>Priority: 3</i>

2.3.2.9 Non-functional requirements for software development

NONFUN-1. Open source and platform compatibility: Software development shall be based on open source software and shall be multiple-platform compatible (or compatible with Linux distributions at least). <i>Priority: 5</i>
NONFUN-2. Software programming: Software development shall be programmed in a modular way and it shall be compatible with different programming languages (C, Java, C++, Python, etc.). <i>Priority: 4</i>
NONFUN-3. Development tools: A Source Control Management shall be used. Any Software change shall be traceable. A bugtracking system shall be used. <i>Priority: 4</i>
NONFUN-4. Documentation: Software shall be self-documented. <i>Priority: 4</i>
NONFUN-5. Interoperability / Compatibility: The International System of units shall be used. <i>Priority: 4</i>
NONFUN-6. Modifiability / Maintainability / Extensibility / Scalability: More functionalities shall be easily added and shall be modular. The software shall have low coupling: low dependency between classes, low impact in a class of changes in other classes, high reuse potential. The software shall have high cohesion. It attempts to keep objects appropriately focused, manageable and understandable. <i>Priority: 4</i>
NONFUN-7. Testability: Each system and/or subsystem shall be tested alone. <i>Priority: 4</i>
NONFUN-8. Operability: The system architecture shall be defined in a hierarchical manner. It shall be clearly defined which system could control other (sub)systems. A subsystem shall be controlled by only one parent system (at the same time). <i>Priority: 3</i>
NONFUN-9. Quality: Coding Standards document should be used. Metrics should be used. <i>Priority: 3</i>
NONFUN-10. Usability: The system shall be completely controllable for the end user (step by step). <i>Priority: 3</i>

Chapter 3

System Architecture Design

In this chapter we present the operational design and the system architecture defined for the TJO-OAdM telescope.

The operational design section includes a detailed description of the different solutions we have selected to manage the overall operation of the observatory. The system architecture to develop the control system is based on these solutions and it is presented subsequently at conceptual, logical and implementation levels. These describe the design of the control system for the robotic operation of the observatory, based on the requirements specified in Chapter 2.

The different components of the hardware, together with their functionality and their interconnection, are presented in Chapter 4. The software elements, or the components of the control system, are presented in the subsequent chapters. They are all introduced here.

3.1 Operational design

Some important operation items are described here. They are designed to fulfill the requirements of the telescope operational framework and are based on the Operational Concept specified before (Sect. 2.2).

3.1.1 Control modes

The control system of the observatory works in a fully unattended manner, or specifically under **robotic control**. Most of the processes included in the science observation and calibration task routines (see Sect. 2.2) are executed without human interaction.

On the other hand, the sporadic engineering tasks are controlled under manual mode with human supervision.

Therefore, two different control modes are necessary:

- **Nominal:** It corresponds to the unattended process control used for science observations and calibration tasks. Tasks can also be scheduled using the interactive execution mode (see section below for a detailed definition).
- **Engineering:** This mode corresponds to high-level control for maintenance and testing tasks. Tasks are scheduled under interactive and manual execution modes and the operator has the possibility to interact with all subsystems via the control system. Some engineering time is reserved to perform these tasks.

3.1.2 Observatory operators

A robotic or **ROCS** (unattended) and a **Superuser** (human interactive) controllers are the two possible operators of the system. The robotic controller is the routine operator for science and calibration tasks, whereas the superuser controller is only activated to perform engineering tasks.

3.1.3 Operation mode

Only one operation mode is enabled at the moment of writing this document: photometric imaging with the MEIA CCD camera. This is the instrument currently used to perform the operational tasks in both control modes. It is composed by a medium-format CCD camera and a Johnson-Cousins filter set devoted to imaging observations. A specific set-up to acquire flat field calibration images is also available as support equipment to this mode of operation.

The operational design described here only takes this mode into consideration, although the general design is conceived to be modular enough to easily integrate supplementary modes in the future. New modes of operation are foreseen once a new autoguiding system and a new instrument (ARES spectrograph) are installed at the observatory.

3.1.4 Task preparation

In this section, we describe the steps to enter new science observations and calibration or engineering tasks.

Science observations are generally structured as task proposals submitted to the TJO-OAdM following a predefined process. Nominal calibrations or engineering tasks are defined and introduced into the system following the same process in order to use the same scheduler routine, to plan their execution and to enhance optimization.

Some of the concepts introduced here are inspired by other projects with advanced tools for observation preparation purposes (LT¹ – Steele et al. 2004, Spike for the Hubble Telescope – Johnston 1990).

3.1.4.1 Proposal submission phases

Science observations are entered into the system by submitting a proposal to the observatory. Each proposal submission process is divided into three different phases: submission of the general description and TAC evaluation (Phase 1), submission of the detailed observation pattern and its revision and verification (Phase 2), and, finally, data retrieval (Phase 3).

The execution pattern of the nominal calibrations and the engineering tasks are entered in a similar way.

3.1.4.2 Science observations

The proposals submitted to the observatory define a set of science observations (targets and observation patterns) to obtain images and data in order to fulfill specific science cases. Such observations are structured through "Observation Block" (OB) units, as defined below. The control system takes observation blocks into consideration to schedule the observations.

➤ *Observation types considered*

Different observation types are considered and accommodated by the scheduler:

- **Flexible time:** Observations can be carried out at any time when the conditions are appropriate.
- **Time-Critical or Phase-Critical Observations:** Proposals may request that TJO-OAdM observations be made at a specific date and time or phase, or within a range of specific dates or phases, when scientifically justified. Some examples of such cases are: astrometric observations, observing specific phases of periodic variable stars, monitoring programs, observations coordinated with another observatory.
- **Target of Opportunity (ToO):** The observation of unexpected phenomena can be entered into the system, eventually overriding ongoing processes. A ToO is initially checked by the system in terms of feasibility and scientific priority.

➤ *Type of targets*

Science observations can be defined for any of the following target types:

- **Fixed target:** Target that can be specified by its name and coordinates, according to the catalogued values.
- **Generic target:** Target that can only be described in terms of astronomical characteristics or general location on the sky (i.e., a specific region in an extended object). Separate targets

¹ <http://telescope.livjm.ac.uk/Info/PropInst/>

should be defined and listed if the user plans to acquire observations at several locations within an extended object. A generic target must be specified with its Equatorial coordinates.

- **Solar system target:** Targets with high proper motion that cannot be specified with its Equatorial coordinates, but with its orbital elements.

➤ **Science observing time distribution: Observation Block**

Science observations included in a proposal are specified through OB units, an atomic element of observation that contains a series of one or more exposures on a target and following a defined sequence pattern. OBs are grouped into larger units called Target Groups (TGs) that include all the OBs for a specific target according to the number of iterations and the scheduling requirements. A specific target can have several TGs. Finally, a complete proposal is composed of the sum of TGs for all targets. Special calibration images and standard stars to execute absolute photometry programs are also specified at the time of proposal submission.

Instrument overheads (e.g., the time required to read out the data) are also included in the total amount of required time to execute an OB.

The observation constraints for any OB contain the following components:

- Target definition (i.e., Equatorial coordinates)
- Control mode (nominal, engineering)
- Operation mode (MEIA)
- Observation pattern: binning, exposure time, defocusing, number of repetitions, filter, etc.
- Environment conditions (weather)
- Sky darkness (distance to Sun and Moon, Moon phase and altitude above the horizon)
- Time scheduling requirements
- Special calibration observations
- Expected data quality
- OB priorities

Data obtained from the execution of an OB may not have the expected quality, in terms of one or more observation constraints that are not satisfied. In this case, OBs can be executed more than once to guarantee that the observation meets the user requirements.

OBs are atomic (i.e., indivisible) and they are not considered finished until they have been completely executed with no interruptions.

OBs and TGs may have complex interdependencies. Two main types can be identified. Firstly, there can be ordering dependencies. Secondly, there can be time constraints to such ordering. For example, certain types of measurements may require calibrations to be scheduled at more or less fixed intervals in time both before and after the primary measurement. Thus, the OBs to execute such pattern are not only ordered in time, but are constrained to be carried out within certain intervals of time.

Every particular OB has five possible states once the proposal is submitted:

- **Revise:** Validation of the observation constraints and target description is pending.
- **Rejected:** OB has been rejected after validation.
- **Ready:** OB parameters have been validated and ready to be considered by the scheduler.
- **InProcess:** OB has been selected by the scheduler and is under execution.
- **Done:** OB has been observed and the data quality control is passed.

➤ **Priority**

In order to compute the ranking of science observation tasks, a priority index is computed for each proposal with different contributions:

- **Scientific priority:** This value is based on the scientific merit of the submitted proposal. Three levels of scientific priority are defined (High, Medium, and Low or filling proposals) and assigned by the TAC when doing the proposal evaluation.
- **Executive priority:** This component is associated to the institution, and/or country the person who submits the proposal belongs to. It shall be defined according to the observatory access policies that are not yet defined at the moment of writing this document.
- **TG and OB priorities:** The user can rank internally the observations included in the proposal by giving them a different priority.

3.1.4.3 Calibration tasks

Data from TJO-OAdM observations are normally provided to the Principal Investigator (PI) after full calibration (raw images and master calibration frames are also supplied in case the PI specifies such requirement). Baseline calibrations include bias, long integration time darks and flat-field images, and are mostly taken at the beginning and at the end of the night.

Projects may need to include special calibration images. A number of bias or dark images can be specified for each observation and they will be taken just before or after the science image. Extra flat calibration images cannot be requested as they cause an important impact on the overall execution process and time overheads. And a set of standard stars can also be included in the proposal definition in case a target requires such calibrations. The latter is done by defining interdependencies among the observations of targets associated to the same proposal. Special calibration tasks must be included in the total request for observing time, and justified explicitly.

➤ *Calibration effects*

The effects that have a strong influence on the image quality when using a CCD detector to obtain high precision photometry are: thermal noise, cosmic rays, readout noise, saturated areas and non uniformity of the detector. Some of them can be corrected. One exception is the readout noise, which is caused by the intrinsic uncertainties coming from the readout amplifiers and the analog-to-digital converter. However, in modern detectors, the readout noise is kept at very low levels (a few electrons per pixel) and this poses no problems to reach the required photometric accuracy, except for faint objects.

These are the relevant effects that need to be corrected at the TJO-OAdM during image processing:

- **Bias current:** The bias current is a positive value that the CCD electronics adds to every image in order to obtain positive values when reading all the pixel data. It is commonly evaluated at every image using a virtual area of unexposed pixels known as the overscan area. There is also a smaller contribution to the bias current that has a smooth 2D variation across the CCD frame, the pattern of which can be obtained combining a set of images with zero exposure time (bias current images or just bias).
The MEIA detector is not equipped with the overscan area and, thus, the two contributions are corrected together using the bias images.
- **Dark current:** The dark (or thermal) current is a contribution to the pixel extracted data due to thermally-induced electrons. It has an exponential dependence with the temperature of the detector and it also depends on exposure time. Its value can be estimated from the combination of a set of dark current images (or just darks) of long exposure time and removed from science images. This effect can be negligible for those images with short exposure time.
- **Flat fielding:** A homogeneous illumination produces different response for different pixels of a CCD chip. That is caused by the non-uniformity of the CCD detector (different quantum efficiency per pixel, ghosts, etc.) and must be corrected by using images of uniformly illuminated fields. The flat fielding process also corrects the image from other effects with different causes, like the presence of dust or ice on the telescope optics. The technique used to obtain flat field images is discussed below.
- **Shutter effect:** Exposures taken by CCD cameras with mechanical shutters can yield different exposure times as a function of the position on the CCD frame. For accurate photometry, the images need to be corrected for this difference using a shutter map that must be applied during image correction process. A description of the shutter map calculation for the main CCD camera at the TJO-OAdM is given in Colomé et al. (2007).
- **Cosmic rays:** The impact of high energy particles can be registered in an image and they appear as bright dots or streaks. The presence of cosmic rays is less important on back illuminated CCDs and on images of short integration time. The MEIA operation mode complies with these two statements and the images taken are unlikely to be affected by this effect. This will not be the case of the future spectrograph (ARES) that will take images of longer integration time.
- **Bad pixels:** A bad pixel mask must be designed to identify those pixels that have non-linear behaviour (either cold or hot pixels). Once established, the bad pixel mask does not need to be recalculated often because it has only a weak dependence on environment or instrumental factors. This correction can be applied at any stage of the data processing routine and certainly prior to the extraction of photometric measurements.

All these contributions to the image spoilage are treated separately by executing a sequence to obtain the master calibration images that are then used for the reduction of the science images.

➤ **Acquisition procedures**

According to the considerations above, three main types of calibration images are considered for TJO-OAdM: bias, darks and flats. Dark current contribution is likely to be negligible for most of the science images, although could be necessary in some cases. The acquisition procedure is specified in a way to minimize the time invested in the execution.

The periodicity to produce a new master bias, dark and flat image depends on their stability as a function of time. It is evaluated on site and the calibration image collection is scheduled accordingly, mainly during twilight hours.

Master bias frame

Bias images can be taken during the twilight or when there is any circumstance that makes it impossible to take flat images or science images. Every run of bias should consist of a set of 10 or more images that are combined to produce a master bias frame (also minimizing the random error level). Once a combined image is obtained, a two dimensional function is fitted to the bias surface to smooth the profile that will be applied to the dark current frames, flat-field and science images.

Master bias frames usually have some dependence on temperature and other environment variables. Time variability and correlation with housekeeping data must be studied to define the acquisition policy.

Master dark frame

Dark current images can be taken during the twilight or when there is any circumstance that makes it impossible to take flat images or science images. Every run of dark should consist of a set of 10 or more images that are combined to produce a master dark frame (also minimizing the random error level). Once a combined image is obtained, a dark current per second and per pixel is computed. Thanks to the CCD chip linearity, this image can be then applied to the flat-field and science images multiplying it by the image exposure time.

Flat field acquisition procedure: twilight flats and dome flats

Flats field images can be acquired during twilight while sky is bright enough. The procurement of sky flats is triggered either by the solar altitude, or by the residual background brightness measured on the science field. This flat field acquisition policy is difficult to carry out in a completely unattended observatory, due to the possible presence of clouds. Screen flats are then considered.

A set of flat-fielding images must be taken frequently, and must have an exposure time long enough to obtain well-sampled images, with high count levels but comfortably below the non-linearity limit. A detailed description of the dome flats set-up and acquisition policy of flat field images is provided below. Several tests were performed to find the best quality image frame, which is a mandatory process to achieve the envisioned photometric quality.

Master flat frame

Once flat-field images are taken, they are corrected from bias using the master bias frame. Finally, contemporaneous flat-field images are co-added and the result is normalized to create the master flat.

Time variability and correlation with housekeeping data must be studied to define the acquisition policy: periodicity, images to use in case a recent flat master frame is unavailable, etc.

➤ **Time distribution**

Calibration tasks are specified using the OB unit as a basis, as defined for science observation time distribution, to provide a homogeneous control of all the routine tasks. A permanent set of OBs describing the execution constraints specifies the basic calibration tasks to be considered by the control system. They contain a series of one or more exposures for every different kind of calibration image and following a defined sequence pattern. Only the observatory superuser operator can modify these constraints.

The task constraints for any calibration OB contain similar parameters as for the science observations. The target is defined in this case with the type of image (bias, dark, flat) instead of the target coordinates.

➤ **Priority**

Calibration tasks have priorities defined to guarantee that they will be scheduled with the correct execution policies and thus ensure the quality of the data.

Calibration tasks have lower priority than time-critical science observations.

3.1.4.4 Engineering tasks (maintenance and tests)

Periodic maintenance tasks are defined in the observatory. They play an important role to minimize the risk of unexpected malfunctioning of any subsystem component.

They are scheduled following the execution policies defined in the Maintenance and Security Plan (under development at the time of writing this document) to keep the quality of the data and preserve the safety of the equipment. Time-variable priority may be defined for the maintenance tasks. Such priority should be greater as the conditions for their execution degrade or as time between maintenance actions exceeds the specified constraints.

They can be carried out during daylight or during the night when the environment conditions do not permit the acquisition of science data. The latter is only possible for those tasks that could be executed without human attendance, although most of them are defined to be carried out under human supervision.

Testing is also eventually foreseen to prepare system upgrades or to evaluate new instrumentation. Such operations are done only under human supervision and, therefore, do not have execution policies to be integrated into the global automatic scheduling.

3.1.5 Task scheduling

The scheduler selects the most suitable task/s to be performed by the observatory according to predefined criteria and with the goal of maximizing the time invested in science operations. It also provides a dynamic response to those conditions that change in real-time. The selection is done by executing a global rating of all the active tasks (science operations and calibration and maintenance tasks) and creating a ranked list. We provide here a brief description of the different aspects considered. See the scheduler description (Chapter 9) for more details.

3.1.5.1 Scheduling cycle

The scheduling module performs the following planning cycles:

- **Long-term (static):** The long-term routine performs a long-term planning of all the active tasks for an entire cycle (i.e., six months).
- **Mid-term (static):** The mid-term scheduler is useful to execute specific plans to be carried out over a night.
- **Short-term (dynamic):** It performs a dynamic scheduling according to instantaneous conditions (housekeeping), cross-correlated with the required execution constraints.

3.1.5.2 Task rating

The scheduler algorithm used to rank the pending tasks (science observations, calibration and maintenance tasks) is based on the Figure of Merit (FOM) calculation for each OB based on predefined criteria. The FOM values obtained are used by the scheduler to select the optimum OB among all the available ones.

3.1.5.3 Proposal priorities

The probability of a proposal to be carried out mainly depends on its scientific priority, but also on the conditions for execution based on the observation constraints.

3.1.6 Execution of tasks

The execution of each specific task is carried out under fully unattended control as this is the default mode. This condition is relaxed to perform some engineering tasks.

3.1.6.1 Execution modes

We foresee four execution modes for the observation scheduling:

- **Robotic:** In robotic mode a short-term scheduler produces a ranked list according to FOM of all the tasks tagged as ready. The task with higher FOM is executed by the system. This is the **TJO-OAdM default mode** of operation.

- **Interactive:** The automatic scheduler furnishes a ranked list of OBs, but the operator can select any OB from the list (not necessarily the one with higher FOM). The operator can also modify the observation constraints for the targets to observe: exposure time, filter, binning, etc. Once these constraints are defined, the observation sequence follows the same pattern as that with robotic control.
- **Manual:** A manual mode is available to instrument operators to test new observing procedures. This mode is used only in exceptional circumstances. In manual mode there is no scheduler providing a ranked list of OBs but rather the telescope and instrument is under the control of an outside agent (i.e., operator), who submits commands directly to the control system via control language or a Graphical User Interface (GUI). The user performs the selection of the OB to observe. The user can also decide on running the Data Processing routine after the acquisition of the observation.
- **Simulation:** The scheduler is capable of running in simulation mode. The simulation mode provides the users a tool to check the observation pattern of their target and evaluate its feasibility before storing it to the instrument database to be considered by the scheduler. The simulation mode can also be used by the observatory TAC to evaluate and adjust the priority of the observations.

3.1.6.2 Nominal operation

Nominal operations perform science observations and calibration tasks running under robotic mode. Several states are defined for the observatory operation and are changed by the control system according to the running routine or non-routine conditions.

We describe here the nightly nominal operation and we give next a more detailed description specific for each type of task, including the boundary and environment conditions and workflow.

➤ System states

We define different states in the nominal operation of the observatory, each of them involving a set of processes:

- **Stop:** It is the default state for the system when not performing science or calibration tasks. Science devices or software applications involved in image acquisition processes (telescope, dome and instrument) are stopped in this state. Subsystem computers, the control system and the hardware and software for the environment monitoring are running.
- **Calib:** The system initiates the instrument in this state (power and control software) and, if required, starts the acquisition of calibration images (Dark and Bias) and the process quality check. Telescope motors are also powered up as the system must imitate the real operation conditions.
- **Init:** System is performing actions to initiate telescope (axis homing) and dome (power up and axis homing) in this state. The dynamic scheduling routine is started.
- **Flat:** It involves switching on the flats lamp and telescope and dome pointing for dome flat acquisition. The system can acquire flat field calibration images and run the quality check.
- **Thermal:** It includes the actions devoted to thermalize the dome. At the TJO-OAdM, this involves the opening of the dome enclosure and comparison of the outdoor and indoor measured temperatures.
- **Science:** The acquisition of science images and quality check are carried out in this state. Dome shutter is opened.
- **Park:** The system is stopping all the processes and moves all the hardware to the stow position following a predefined sequence.
- **Shutdown:** The system executes a complete shutdown of all the science and support devices and the control layer.

The control layer and the routines for the housekeeping control and the execution of the data processing (described below) are operative in all the states, except for *Shutdown*.

The time to change from one state to another in a routine operation with no weather alarms is defined by the selection of tasks performed by the scheduler and some time events and time offsets for different system operations, as listed here:

- Time events:
 - T_{FirstCal} = time when the first (or any) calibration task is scheduled. The static schedule computed a priori determines this value (see Chapter 9 for more details). The system response is different depending on the calibration task selected or the time of selection. In Table 3.1, for instance, T_{FirstCal} , corresponds to the selection of the first

flat calibration task, whereas T''_{FirstCal} is the time scheduled for morning calibration tasks.

- T_{FirstSc} = time when the first (or any) science observation is scheduled. The static schedule computed a priori determines this value. It is adjusted in real time by the dynamic scheduler.
- T_{Last} = time when the last scheduled task is finished.
- Time offsets:
 - T_{Calib} = mean time for the instrument warm-up (operating temperature reached).
 - T_{Init} = mean time to initialize the telescope and dome.
 - T_{Flats} = mean time to warm-up the flats lamp.
 - T_{Thermal} = mean time required to thermalize the dome.
 - T_{Park} = mean time to perform all the parking actions.

This time evolution is translated into an event-driven process when implementing the nominal operation work flow as described in the control system design (Chapter 5). This translation is due to the dependency of some of the mentioned time events with other variables (i.e., T_{Thermal} depends on the difference between the inside and the outside dome temperature that has seasonal variations).

➤ *Nightly nominal operation*

Table 3.1 describes how the status evolves during a night without weather or unexpected events:

	Time	State	Process
Evening	$t \ll T_{\text{FirstCal}} - T_{\text{Calib}}$	Stop	None
	$t < T_{\text{FirstCal}} - T_{\text{Calib}}$	Stop to Calib	<ol style="list-style-type: none"> 1. Initiate control software. 2. Power up the instrument. 3. Start instrument cooling system.
	$t \approx T_{\text{FirstCal}}$	Calib	<ol style="list-style-type: none"> 1. Bias and Dark images: <ol style="list-style-type: none"> a. Acquisition and on-the-fly processing.
	$t \approx T'_{\text{FirstCal}} - T_{\text{Calib}} - T_{\text{Init}}$	Init	<ol style="list-style-type: none"> 1. Initiate control software. 2. Power up the system: switch on power supplies for telescope, dome, and support equipment. 3. Telescope and dome homing, opening telescope mirror covers.
	$t \approx T'_{\text{FirstCal}}$	Flat	<ol style="list-style-type: none"> 1. Flat calibration images: <ol style="list-style-type: none"> a. Dome lamp warming-up. b. Telescope and dome pointing. c. Filter and focus positioning. d. Acquisition and on-the-fly processing. 2. After last evening calibration image is acquired: computation of master calibration frames.
	$t \approx T_{\text{FirstSc}} - T_{\text{Thermal}}$	Thermal	<ol style="list-style-type: none"> 1. Close telescope mirror covers. 2. Open dome enclosure. Time required for dome thermalization is T_{Thermal}. 3. Open telescope mirror covers.
Night	$t \approx T_{\text{FirstSc}}$	Science	<ol style="list-style-type: none"> 1. Science observations: <ol style="list-style-type: none"> a. Telescope and dome pointing and tracking b. Filter and focus positioning c. Image acquisition and on-the-fly processing.

Morning	$t \approx T''_{\text{FirstCal}}$	Flat and Calib	<ol style="list-style-type: none"> 1. Close telescope mirror covers. 2. Close dome enclosure. 3. For Flat calibration images: <ol style="list-style-type: none"> a. Open telescope mirror covers. b. Telescope and dome pointing and switch on flats lamp. c. Flats lamp warming-up. d. Filter and focus positioning. e. Acquisition and on-the-fly processing. 4. For Bias and Dark calibration images: <ol style="list-style-type: none"> a. Acquisition and on-the-fly processing. 5. After last morning calibration image is acquired: computation of night master calibration frames.
	$t \approx T_{\text{Last}}$	Park	<ol style="list-style-type: none"> 1. Move telescope and dome to park position, close telescope mirror covers. 2. Power down the task execution system: switch off power supplies for telescope, dome, instrument, and support equipment. 3. Stop task execution control software.
	$t > T_{\text{Last}} + T_{\text{Park}}$	Stop	<ol style="list-style-type: none"> 1. Final processing of all the science images.

Table 3.1: Nightly nominal operation

3.1.6.3 Task execution policies

We give here the specific conditions to execute science observations and calibration tasks. Differences between them are indicated in the text.

The execution policies must guarantee the stability of the calibration frames for their subsequent application. A quality control process is executed for each frame and notified to the control system.

A data set characterizing the contributions to the image that require calibration is obtained with the following periodicity:

- **Bias current:** Daily measurement.
- **Dark current:** Daily measurement.
- **Flat fielding:** Daily measurement.
- **Shutter effect:** Static pattern correction, to be computed at higher periodicity (> 1 month).
- **Bad pixels:** Static pattern correction, to be computed at higher periodicity (> 1 week).

A basic set of calibration images (bias, dark and flat) is acquired every night and delivered to users together with their science observations. The data processing policies and their implementation to optimize the image quality are analyzed in Chapter 10.

➤ **Boundary conditions**

The control system handles the operation routines based on a set of input and output data:

- **Inputs:** These include the database with parameters describing execution constraints, housekeeping data (environment, hardware health, etc.), subsystem status and feedback from the data processing routine after each image acquisition.
- **Outputs:** These are the images and the data produced by their analysis, the quality control results, the updated task status and the archival of results in the database.

➤ **Environment and system conditions**

The task execution run is carried out under some specific conditions. They are continuously checked by the control system.

- **Sun altitude:** Science data are acquired when the Sun is below the horizon. Scientific observations can be scheduled during twilight if execution constraints so require it, but they are usually carried out when the Sun is at a lower altitude than the one determined by the astronomical twilight. Calibration tasks are carried out during daylight, twilight or during the night when the environment conditions do not permit the acquisition of science data.
- **Weather variables:** Weather variables must be in a range of accepted values for the routine operation of the hardware (telescope, dome, instrument, and support equipment). Accepted

ranges have to be determined during the commissioning phase of the observatory. The weather variables considered are: outdoor and indoor dome humidity, outdoor and indoor dome temperature, wind speed, and detection of clouds, rain or electric storm. Calibration images can be obtained in those weather conditions that are unsuitable to enable opening the dome shutter, but good enough to work with the scientific devices without any risk of damage. The latter would not be true for sky flat fielding.

- **Hardware status** at the time task execution can start after finishing the system initialization:
 - **Telescope:** Electronics are ready and communicating with the control system. Motors are switched on. Science observations and flat field images require homing all the axis (HA, DEC, Focus, filter wheel), and opening the mirror covers.
 - **Dome:** Science observations and flat field images are carried out when the electronics are ready and communicating with the control system, and homing is complete. The enclosure (flap and shutter) must be opened, and the dome must be completely thermalized in case of acquisition of science data and sky flat fields.
 - **Calibration:** The acquisition of dome flat fielding images is executed when the flats lamp has reached the working temperature.
 - **Instrument:** The instrument must be initialized and the nominal working temperature of the detector must be reached.
 - **Support equipment:** The power supply system and environment sensors are ready and working properly. Redundant dome shutter control is operative in case of science observations or acquisition of sky flat fields.

➤ **System initialization before starting the nominal operations**

The initialization process is carried out just in case it has not been initialized before. This process involves all the actions required to change the state from *Stop* to *Calib*, in case Bias or Dark images are scheduled. In case of science observations or acquisition of sky flat fields, the system state is changed from *Stop* to *Science*, that is: *Init*, *Calib* and *Thermal*. For dome flat fields, *Flat* state must be reached by changing the system from *Stop* to *Init* and, then, to *Calib* and, finally, to *Flat*.

➤ **Nominal run workflow**

Science operation runs are executed continuously by the control system during all the time in which environment conditions are ready for their execution in order to maximize the time invested in these kinds of tasks.

At least, a full set of calibration runs is executed twice per night, according to the environment conditions. That is, just before (after) the astronomical twilight ends (start) in the evening (morning). Start time in the evening is adjusted by the scheduler to complete the full set of calibration images.

Calibration images could also be acquired in two special situations:

- Calibrations performed under science environment conditions, but with weather variables not allowing science observations.
- Extra calibration images required by the users. Images are taken following the user requirements. Flat fielding images are not allowed as it would mean losing the dome thermalization and a large overhead for a specific proposal.

Table 3.2 provides a description of the workflow after system initialization:

1. Scheduler	1.1. Task selection with the short-term scheduler (OB). 1.2. IF Interactive mode: The operator can change the task selection and some of the observation constraints. 1.3. The control system changes the status of the selected task in the database as task under execution or InProcess (not to be considered by the scheduler until the status is changed to Ready or Done). 1.4. Send the selected task to the control system, with the parameters describing the task to be performed: type of image, telescope, dome and instrument configuration, exposure time, etc.
2. Telescope and dome	2.1. IF science observation or acquisition of a flat field is selected: The control system checks for telescope and dome status, and executes the pointing routine. In case of a science observation, it starts the autoguiding routine; whereas it checks for the flats lamp status for a dome flat field acquisition.
3. Observation	3.1. The control system checks for instrument status and starts image

(MEIA mode)	acquisition. 3.2. Acquisition finishes at the time of integration defined. 3.3. Start reading image.
4. Post acquisition processes	4.1. New task selection and data processing. 4.1.1. The scheduler executes the selection of task (the current task is not considered). 4.1.2. The control system computes time to change set-up configuration (instrument + telescope + dome) for new target/task (t_{NT}). 4.1.3. The control system computes/knows the time to execute the entire data processing routine (t_{DP}). The latter includes the data quality control. 4.2. IF $t_{DP} < \alpha \cdot t_{NT}$ (α : coefficient to calibrate the system response): 4.2.1. The control system does not start new selected task execution until Data Processing finishes. 4.2.2. When image is downloaded: 4.2.2.1. Data processing routines are executed. 4.2.2.2. The control system archives the data, the image files (IF they exist), and updates the database. 4.2.3. IF the quality control result is positive (data obtained has good quality): the control system continues with the new selected task → Workflow loop back 4.2.4. IF the quality control result is negative (data obtained has poor quality): change last task status to Ready to enable its execution and the control system executes again the scheduler selection process. 4.3. IF $t_{DP} > \alpha \cdot t_{NT}$: 4.3.1. The control system starts the new task without taking into consideration the last one (task with the state InProcess) in the selection process performed by the scheduler → Workflow loop back (new thread process) 4.3.2. These processes run in parallel when last image is downloaded: 4.3.2.1. Data processing routines are executed. 4.3.2.2. The control system archives the data, the image files (IF they exist), and updates the database.

Table 3.2: Nominal task run workflow

3.1.6.4 Operation process control

Each process included in the night operation is executed by the observatory control system. Some of those processes involve a direct control of the hardware layer described in the Physical Architecture (Chapter 4). All the rest involve the execution of distributed software processes among different subsystems. The software infrastructure and the software applications described next are responsible for this operation control.

3.1.7 Data processing and storage

A data processing routine is executed each time a new image is acquired. The main goal of this module is the extraction of high-quality data from the observatory running on a particular operation mode. Photometric and astrometric data are expected for the MEIA imaging camera.

The following steps are designed to perform the data extraction with the expected quality and to provide feedback to the control system in an efficient way:

- **Image quality control:** These routines evaluate the quality of each frame acquired in terms of file parameters, photometric and astrometric values and scheduling conditions.
- **Image calibration and analysis pipeline:** The pipeline performs the science and calibration image reduction and the science image analysis.
- **Data archiving:** This step involves archiving of data and generation of new registries into the observatory database and into the logging system for all the images.
- **Storage and backup:** This step is responsible for the file storage design and on- and off-site storage policies for images and data backup.

An on-the-fly execution of the data processing routine after their acquisition is foreseen to enable the dynamic response of the system according to the evaluation of science and calibration images.

Finally, an off-line process is also executed at the end of the night operations and using all the images to improve the quality of the reduction and analysis results.

3.1.8 Housekeeping

Housekeeping control centralizes all the system health and environment data and provides them to all the subsystems that so require.

It is also responsible for managing the events and alarms arising during the observatory operation. It executes contingency actions that restore the nominal system operation or minimize the damage produced to the instrumentation. It provides a high level of feedback to the observatory control system.

3.2 Architecture conceptual design

We defined three different layers for the system design and implementation to achieve the complete unattended operation and following the operational design given in last section:

- **Hardware layer:** The lowermost layer contains the hardware elements of the system. It is composed by the science devices to acquire astronomical data (telescope, dome, and instrument), the sensors and other support devices for housekeeping purposes.
- **Housekeeping control layer:** This software layer is in charge of monitoring the system health, all the environment sensors and the support devices, and handling events and alarms generated by this layer and their corresponding actions.
- **End-to-end data flow control layer:** This software layer is in charge of executing all the data acquisition routine from user proposal submission to user data retrieval and including data processing feedback for the optimization of the operation scheduling. Handling events and alarms generated by this layer is also under its own responsibility.

Fig. 3.1 describes the architecture conceptual design.

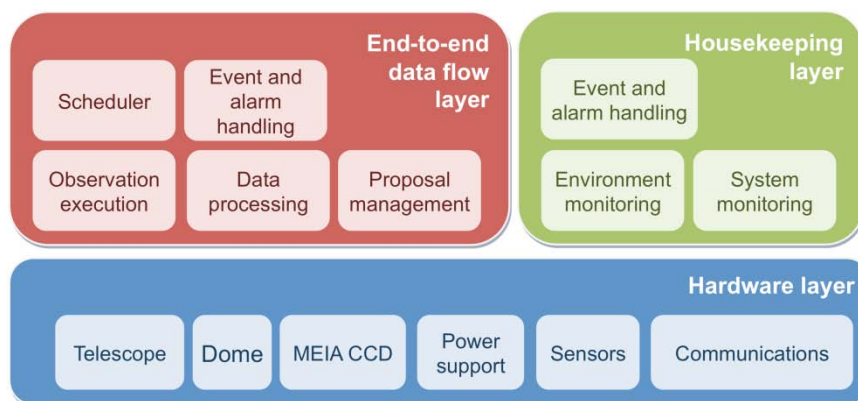


Fig. 3.1: Architecture conceptual design

The housekeeping layer is mainly responsible for the robustness and reliability of the unattended operation, whereas the data flow layer has a direct impact on the system efficiency. The overall quality of the system performance depends on the quality of the particular implementation of each one of these layers and the subsystems to control them and, also, the implementation of the interfaces.

3.2.1 Single Points of Failure and Redundancies

A critical step on the system analysis to design a robust robotic observatory is to identify those points of special weakness that would threaten the integrity of the observatory equipment in case of failure or malfunctioning. Special care must be taken on these points when designing the system architecture and its implementation, mainly at the level of the Housekeeping and Hardware layers.

We identified two single points of failure at the TJO-OAdM:

- **Dome aperture control:** The dome shutter is the most critical point for the safety of the telescope, which is the core element of the observatory. Any error on its response would be fatal if it happened during precipitation.

- **Environment monitoring:** Another critical point is the reliability of weather data. The system must have the capability to measure all the critical weather variables and verify if the sensors are performing as expected.

In order to minimize the frequency of faults, redundant control components or sensors were included in the architecture implementation design and robust equipment was chosen at the development phase. Finally, a complementary set of actions, together with support equipment, were defined in the Housekeeping control and the Hardware layers to detect any error related with these points and execute palliative processes. They minimize the risk of severe damage and trigger an external alarm for the intervention of the observatory operators in case the system cannot restore the normal operation by itself. The modules involved are highlighted in Fig. 3.2. Red boxes correspond to the single points of failure mentioned before, whereas related hardware and software are represented in yellow. An extended description of these components is given in the sections where we present the architecture implementation (Sect. 3.3) and the detailed Housekeeping layer (Sect. 5.2) design.

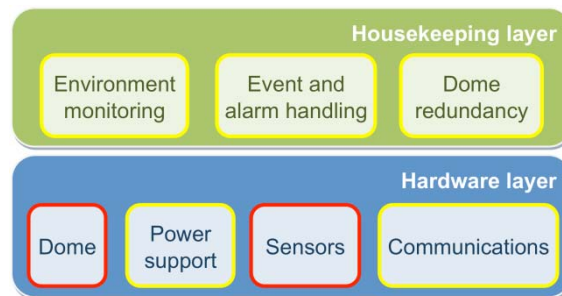


Fig. 3.2: Modules involved in the single points of failure analysis (highlighted in red)

3.3 Logical architecture

3.3.1 Software layers

A set of applications running on different subsystems implements the logical architecture for the software layers described above. As mentioned before, the component layers are:

- **Housekeeping control:** This layer includes the environment and system health monitoring, and the management of alarms and events.
- **End-to-end data flow control:** It includes proposal management, scheduler, task execution, and data processing.

A third component, the software infrastructure, is added to these layers in order to increase the software homogeneity and to guarantee its efficiency with a clear definition of interfaces and integrating common functionalities (i.e., handling of events and alarms):

- **Software infrastructure:** A general control system based on this infrastructure acts as an agglutinating element for all the applications developed. It is also responsible for handling the events and alarms, as well as logging and backup, considered as common services to the other layers of control and included into the software infrastructure development.

The applications and the software infrastructure conform the control system (or ROCS, standing for Robotic Observatory Control System) for the TJO-OAdM.

3.3.2 Subsystems

The software layers and their associated services (applications) must be implemented on a unique or multiple subsystems running in computers, embedded systems (i.e., Programming Logic Controllers – PLCs), etc.

We defined five subsystems to control the entire observatory operation on a distributed task scheme (Fig. 3.3): four onsite (TJO-OAdM) and one offsite (currently IEEC-CSIC in Bellaterra, Spain). The definition of each particular subsystem is done, at this logical level, by just considering the actions and services it performs. It does not imply any particular hardware implementation, i.e., more than one subsystem could be installed on the same machine or a single subsystem could be installed on several machines.

The motivation of using up to five different subsystems is to reach a compromise between the risk of having too many services in a single computer (all of them are lost if the computers fails) or having a different computer per service (every new computer is a potential source of errors). The number of subsystems has been minimized joining as many services as possible in the same machine, but using different computers for redundant services or for services that have to be accessed by other instruments located at the observatory site.

The distribution of applications among these subsystems implements a top-down control structure that manages the workflow and housekeeping functionalities of the observatory.

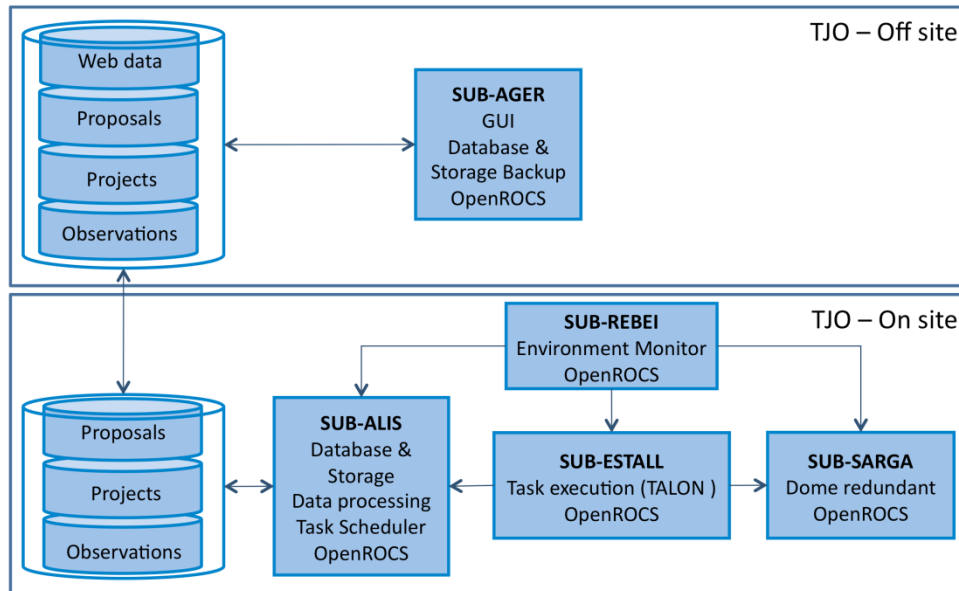


Fig. 3.3: Logical architecture

Hereby we describe the purpose of each subsystem and the applications under supervision grouped by the software layer they are intended to implement:

- **End-to-end data flow and Housekeeping**
 - **SUB-ESTALL** (on site): This subsystem includes the control of all the task execution process and the monitoring of the hardware involved: telescope, MEIA CCD camera and dome. The TALON software is used for telescope, instrument and dome control.
 - **SUB-ALIS** (on site): This subsystem is in charge of data storage and backup, data processing and task scheduling routines. It is also used as the main server and observatory gateway to connect to the outside world. A secondary subsystem, Router-LAN, manages the communications among internal computers and also with the Internet.
 - **SUB-AGER** (off site): Off-site supervision of the observatory is carried out by this subsystem. It is also dedicated to massive data storage (4 TB), it hosts the observatory database, the web-based user interface and the project development system (tools such as GIT, SVN, TRAC, etc.).
- **Housekeeping**
 - **SUB-REBEI** (on site): The environment monitoring is running on this computer. It collects and compares all the environment data. It also interfaces physically with the weather sensors that use a serial connection. A secondary subsystem, **SUB-RUBIES**, located on site (not shown in the Fig. 3.4), is implemented on an MSWindows-based computer whose only purpose is to download data from the redundant weather station (SMC) and provide it to the SUB-REBEI subsystem. LoggerNet is the software used to communicate with this specific weather station.
 - **SUB-SARGA** (on site): This subsystem is in charge of the dome redundant control.

All subsystems are managed by the Open Robotic Observatory Control System (OpenROCS) software suite, which acts as a software infrastructure and also controls the event and alarm generation and the corresponding actions to ensure the reliability of the entire system.

3.3.3 Interfaces

Software interfaces among all the applications and subsystems are needed to integrate new packages with the commercial ones, and to have a software suite that manages the complete observatory in a consistent manner.

Arrows connecting the subsystems in Fig. 3.3 illustrate dependencies among the subsystems that require a well-defined interface:

- SUB-REBEL interacts with all the subsystems that require environment data access: SUB-ALIS for data logging, SUB-SARGA for the dome redundant control and SUB-ESTALL for the execution of the observation operation.
- SUB-ESTALL interacts with SUB-ALIS for the data processing associated services and the scheduling routine that provides the next operation to execute, and it is also connected to SUB-SARGA to detect any malfunction of the dome redundant control.
- SUB-SARGA interacts with SUB-ESTALL to detect any malfunction of the dome nominal control (based on the TALON software).
- SUB-ALIS interacts with SUB-AGER for database mirroring, data storage backup services and remote supervision of the observatory.

3.3.3.1 User interface

A web-based proposal submission tool is necessary to store all the information in the database and it is thought to be the main user interface with the observatory. Information is also accessed there by the TAC to evaluate the proposals and by the automatic scheduler. The images and the resulting data products are stored in the observatory archive and can be easily accessed using the web interface. The user can download the data identified with a proposal code after logging in a private area of the observatory web page.

Remote monitoring of the observatory is necessary to supervise the system operation when a critical alarm is received. A web-based interface will be installed on SUB-AGER and used for that purpose.

3.3.4 Modularity

The TJO-OAdM control system design is based on a distributed task scheme with several subsystems, each of them containing specific routines subject to the elements it controls. This modular approach enables the separation of critical tasks into different control elements and provides the possibility of adding redundant equipment to single points of failure or adding other facilities to work under the same control system.

For a distributed control system, the trigger of an alarm by one of its elements may derive on a global cascade depending on the origin of the problem. But, global cascade actions should not produce conflicts for the efficient control of the observatory.

3.4 Physical architecture

From the physical architecture point of view, the three layers mentioned in the architecture conceptual design are grouped into two different parts (see Fig. 3.4): the operational hardware and the control subsystems. The operational hardware includes all those instruments, and their software control drivers and firmware, whereas intelligence lies in the subsystem. The subsystems, already introduced in the logical architecture description, are composed by the computers and the software (middleware infrastructure, applications and interfaces) and they are responsible for collecting all the information provided by the hardware and for taking the best decision at any time.

Fig. 3.4 shows the connection between each subsystem and the hardware device/s it controls. The database and the archive are also included to illustrate that this software element is accessed by several subsystems to perform their actions.

The distribution of subsystems is designed in a modular way to be adaptable to a different hardware layout with just a few modifications on the configuration. This feature is more extendedly described in the following chapters. An example of how other projects can take profit of such versatility is the development done for the SuperWASP Qatar Follow-up Telescope (SQFT), a project for exoplanet characterization and installed in El Roque de los Muchachos (Canary Islands) in May 2011, whose control system is based on the work presented here.

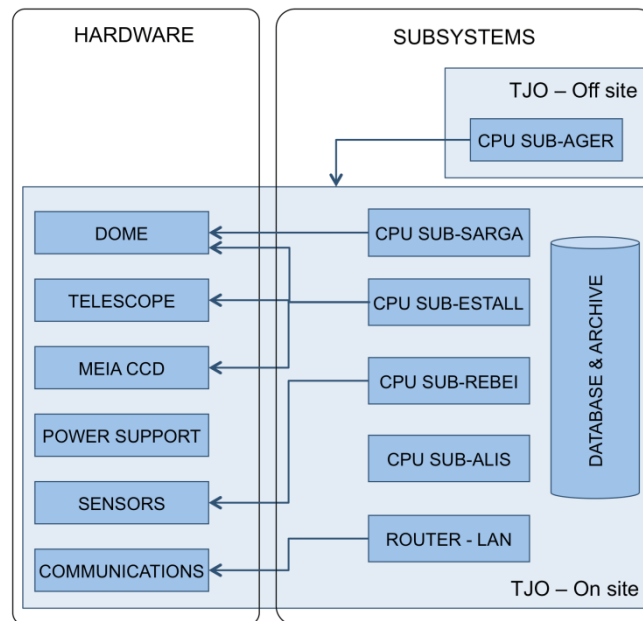


Fig. 3.4: Physical architecture

3.4.1 Operational hardware

The operational hardware components cover all the functional and development requirements stated in Section 2.3. Some of them (dome, telescope and instrument) are part of the original setup and their selection was mainly based on the science case the observatory has to achieve. They are grouped in six categories:

- **Telescope:** It includes the TJO telescope optics, mechanics and electronics, the mirror covers, the filter wheel and the set of filters, the CCD camera mechanical interface and the control software TALON that can control all these components. This device and software is completely dedicated to data flow operations.
- **Dome:** It includes the dome mechanics, the electronics and the low level control software installed in the electronics interfacing the dome motors and the control system. Regarding the shutter motion control, electronics and software provide a combination of two control modes, nominal and redundant, in order to increase the system robustness. The nominal dome control is devoted to data flow operations, whereas the redundant mode is included in the housekeeping control.
- **MEIA CCD:** It includes the CCD camera and the software drivers that are designed to contribute to the data flow processes of the observatory. The Johnson-Cousins filter set and the set-up to acquire flat field calibration images are also included in this category.
- **Power support:** It includes the power distribution units, with remote control capability based on computer network access protocols, the uninterruptible power supply units, also accessed through the network, and a set of passive filters for protection against power surges. These devices ensure a reliable housekeeping control.
- **Sensors:** The suite of sensors, with their corresponding control drivers, monitor the weather environment variables for housekeeping purposes.
- **Communications:** This system provides internal, among all the devices, and external communications. It is devoted to both data flow and housekeeping tasks.

3.4.2 Subsystem computers

The observatory is composed of five subsystems. Each subsystem is based on one or more computers with the necessary software according to the operations it perform. The implementation architecture of each subsystem has the structure shown in Fig. 3.5.

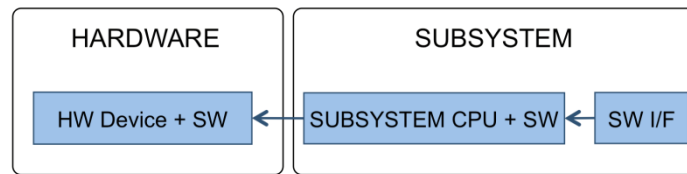


Fig. 3.5: Subsystem implementation architecture

The software included in every subsystem is designed to communicate with the low level software that controls the hardware device (embedded software in the hardware or drivers installed in the subsystem computer) and it also interfaces with the other software modules.

The distribution of applications for each subsystem was given in the logical architecture description (Sect. 3.3).

Chapter 4

Physical Architecture Implementation

A description of the hardware devices included in the system physical architecture of the TJO-OAdM (see Fig. 3.4) is presented in this chapter. The low level software is also described when it is considered necessary for a more complete picture of the system.

Some new hardware components developed in-house or based on COTS instrumentation are presented in more detail for those cases in which design and implementation the author has been deeply involved. This description illustrates how accurate the implementation has been to ensure a positive contribution to the quality operation of the TJO-OAdM.



Fig. 4.1: TJO-OAdM areas (Author: J.Bas, from www.parcasrastronomic.cat)

4.1 Site areas

The main hardware components are located in four different areas (Fig. 4.1):

- **Dome area:** Housing of all the science devices (telescope, instrument, calibration and dome).
- **Control area:** The subsystem computers are located here.

- **Communications area:** The communication system (internal and external), the main server computer (SUB-ALIS) and the UPSs for a stable power supply are installed in this area.
- **Communication tower** (not shown in the figure): It holds the RF antennas for the Internet connection and multiple sensors of the SMC and Davis weather stations.

The environment sensors are mostly located outside the building, whereas the power grid connection, the support generator and solar panels for power generation are located in a small building in the Northern side.

4.2 Telescope - Joan Oró Telescope (TJO)

The TJO telescope, manufactured by OMI, has a 0.8-m primary mirror with an overall F/9.6 optical system in Ritchey-Chrétien configuration. The telescope has an equatorial fork mount. The instrumentation is placed at the Ritchey-Chrétien focus. A segmented mirror cover protects the primary mirror from dust when the telescope is stowed. All these elements are automatically controlled.

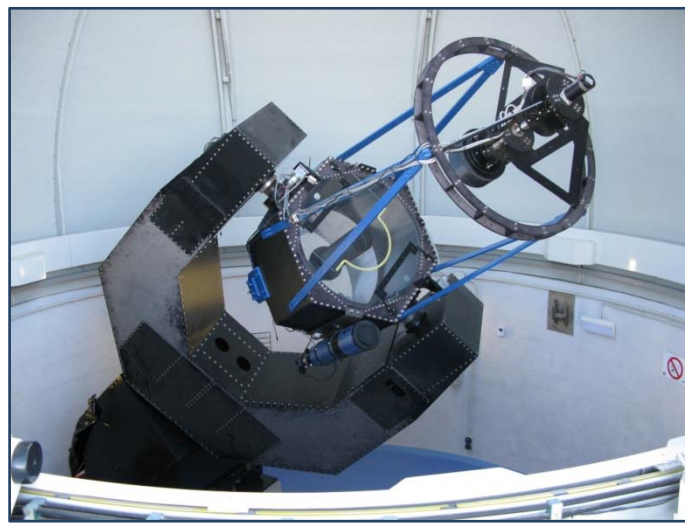


Fig. 4.2: TJO telescope

4.2.1 TALON

The basic equipment supplied by the telescope manufacturer included TALON, built on C and shell scripting programming and to be run on Linux, enabling the control of different elements to automatically manage astronomical observations, and which is now a GNU license software. The hardware elements controlled by TALON are the following: telescope, roof/dome, CCD camera, filter wheel, weather station and UPS unit (see the following sections for a description of these devices). Two of its applications (MKSCH and TELSCHED) can be used to create a list of objects to observe and to generate an observation schedule for the following night, based on simple queue (static) scheduling. Finally, TALON also has some basic applications for image reduction and photometric and astrometric analysis. In principle, the original TALON system could perform the robotic operation of the observatory facility but its low reliability, lack of redundancies and limited amount of housekeeping data would imply a high risk level and possibly intense human supervision.

4.2.2 Electronics and control

The electronics used to handle all the movements of each telescope component (Hour Angle and Declination axis, mirror covers and focus), the filter wheel and the dome are the Clear Sky Institute Motion Controller – CSIMC – boards, distributed by OMI. A network of five CSIMC boards (four of them placed on the telescope mount and pier and the other one on the dome electronics box) is used to communicate the devices for the movement control with the computer through an RS232 connection with one of the boards that serves as the gateway node (see Fig. 4.3, where red arrows indicate fiber optics communication, and the black ones are RS485). The opening of mirror covers and dome commands (shutter opening and closing, azimuth motion) are also controlled through the CSIMC boards. These standalone boards have the following features: communication with motor

controllers, reception of a closed-loop stepper axis encoder data and an open-loop stepper axis, control of two limit switches and one home and several “user-definable” opto-isolated I/O. HA, Declination, dome rotation axis and focus linear axis work in closed-loop, while the filter wheel is an open-loop stepper axis.

The distribution of axis movement control for each CSIMC board is listed here:

- CSIMC 00: Gateway node, dome control.
- CSIMC 01: DEC axis control.
- CSIMC 02: Focus axis control.
- CSIMC 03: Mirror covers and filter wheel control.
- CSIMC 04: HA axis control.

The subsystem in charge of controlling the telescope is SUB-ESTALL.

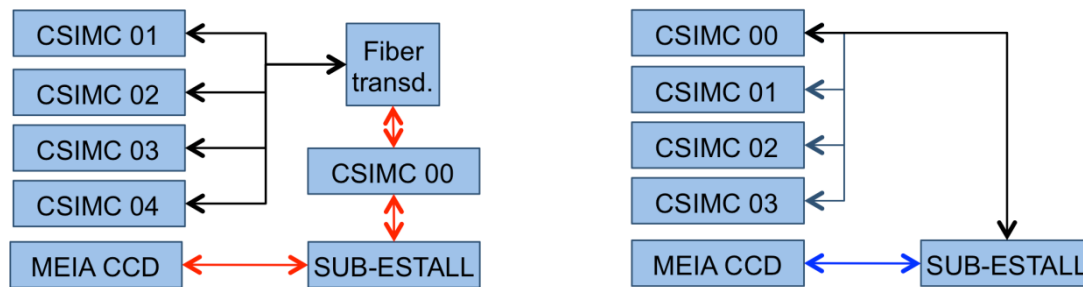


Fig. 4.3: Network of CSIMC boards: current (left) and old (right) configurations

4.2.3 Telescope modifications

Several aspects of the telescope mechanical parts, electronics and control software have been modified in order to increase the image quality, the life cycle of the components and the telescope robotic control performance:

- Baffle: An extended baffle in the primary mirror was installed to reduce stray and diffuse light effects.
- Encoder shielding: We have protected the encoders used for the Hour Angle and Declination axis control with a new shielding.
- Counterweight: The counterweight was increased after measuring a lack of weight balancing.
- Mirror covers: A new mechanical design was implemented in order to reduce their weight and to enable the baffle extension. The motors were also changed to provide a higher torque. A new electronic board was developed to interface with the CSIMC board for power control and polarity inversion required to change the motor rotation direction for opening/closing purposes.
- Wiring and fiber optics: Wiring between electronic components was revised and modified in some cases to increase the robustness. Optical fiber and fiber transducers were used to communicate the SUB-ESTALL control computer with the CSIMC board that acts as the gateway node, and also to communicate this node, the only one located outside the telescope mount, with the rest.
- CSIMC configuration: A new node was added and installed outside the telescope mount, close to the dome electronics. This new configuration improved the performance of the nominal dome control based on this new CSIMC node that communicates with SUB-ESTALL. The old configuration is shown in Fig. 4.3 (blue arrow indicates the USB connection).
- TALON: Important modifications were done on this software. Mirror covers and dome control were added; pointing and tracking functionalities were improved; and environment monitoring, management of the housekeeping associated alarms (environment and power supply), data processing and scheduling were removed or discarded.
- Integration into the control system: We have developed new solutions to integrate TALON in a more general software structure providing a reliable and safe global control. A new layer of software was added to communicate the control system routines with the telescope at the hardware layer. The latter is composed by the low level part of TALON installed on SUB-ESTALL computer and the embedded part at the CSIMC boards. This work is extensively described below.

- Power stabilization: A dedicated UPS was installed to power the telescope motors and the electronics providing the capability to move the telescope to the stow position, in case of power outage, and also to ensure a high quality electrical signal.

4.3 Dome

The dome, manufactured by Baader Planetarium GmbH., has 6.15-m in diameter and it is equipped with a shutter and a flap that provide an aperture of 97×29 square degrees for the telescope sky vision. The azimuth rotation and the aperture opening and closing movements are nominally controlled from SUB-ESTALL through a CSIMC board and using the software TALON. Communication between the CSIMC board and the dome electronics is based on TTL protocol.

The robustness of the aperture control is a must to ensure the robotic operation and represents the most critical single point of failure at the observatory. A correct control of the azimuth rotation is also required to ensure the system efficiency and the operation quality, minimizing the observation overheads and optimizing the dome pointing routine. Both aspects and the poor performance of the initial set-up motivated a deep redesign of this operational hardware component. We carried out an analysis of the control requirements and modified the design of the aperture control to ensure an implementation of high reliability. The work done in this Thesis is described in the following subsections. Final implementation was done in collaboration with the engineering team and the Insercad Electronica S.L., a company with expertise in developing safety critical electronic systems.

4.3.1 Dome control analysis

The initial set-up (see Fig. 4.4) of the dome control system presented the following weaknesses:

- The electronic components used by the dome manufacturer (micro chips, electrical wiring, encoder) proved to be very unreliable thus affecting the system robustness. Signal processing performed by the original dome electronics, for instance, was based on a very unstable microchip (ref.BC547C). In addition, dome electronics (including an IR receiver for manual operation) were installed inside a plastic box located close to the motor power regulators, what caused frequent damage and a high level of noise on these components.
- Communication between the CSIMC board and the dome electronics was based on a long (> 15 m) electrical wire connection, which caused frequent induced currents due to the proximity with the dome motors and to environment effects, like electric storms.
- The cabling connection at the CSIMC board was highly unreliable and not correct to carry out the control of the dome movements.
- The software routine to control the dome was not correctly implemented in TALON.
- No redundant control was available to respond to a faulty behaviour of the SUB-ESTALL control computer or the interface electronics.
- Shutter closing due to rain detection was software controlled, without a hardwired connection with the dome electronics and adding a time delay for the system to respond after detection.

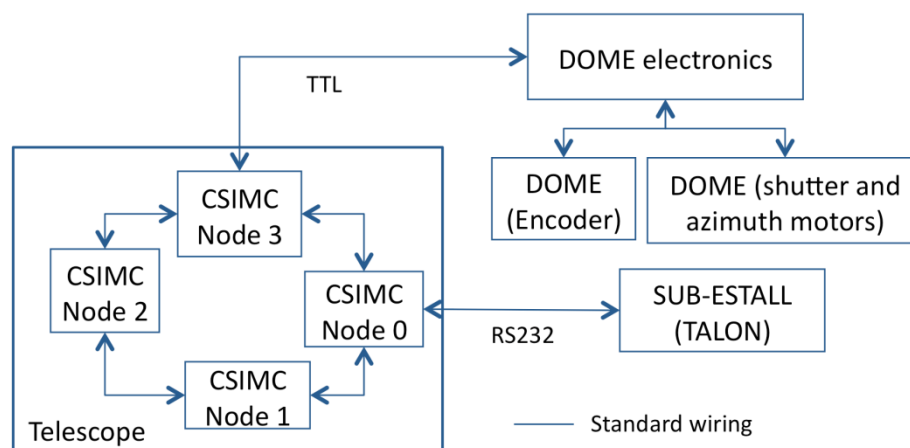


Fig. 4.4: Diagram connection of the initial set-up for the dome control

We ended the analysis for the dome control system redesign with the specification of the general requirements that are summarized in the following list.

The dome control system had to provide:

- Communication among the following devices: telescope control system (CSIMC electronic boards and TALON software), dome control electronics (manual buttons), redundant control system and rain detector.
- Communication of all signals for the dome control: azimuth rotation, open/close shutter operation, shutter status, home position, and encoder signal reading.
- Prioritization of all signals coming from the telescope (CSIMC boards), the manual control buttons, the redundant control system and the rain detector.
- Combination of the rain detection signal coming from two independent sensors, thus introducing redundancy in the detection.
- Software control (TALON and redundant control) integrated with the observatory control system.
- Dome control using a redundant control system, based on a robust computer and software implementation. Redundant control programmed to take over control of the dome in case of weather alarms or system failure.
- Maximized system robustness.
- The system had to provide watchdog functionality to execute aperture closing in case of no communication with any of the controlling devices.
- The physical location of the system, the electronic components, the employed communication system, the installed protections, etc., had to permit the minimization of electrical noise and the risk of induced power surges. For this reason, insulation against conducted and induced electrical noise had to be guaranteed to have a high level of operation reliability of the devices together with extended durability.
- Power supply to dome aperture motors had to be guaranteed at any time and, at least, to perform the aperture closing in case of a power outage.
- The conditions in which the devices had to operate were those typical of extreme environments (comparable to industrial applications: extreme temperatures, high and low humidity, etc) and in such extreme situations is when the system had to guarantee nominal behaviour. They are summarized in the following table (Table 4.1):

	Nominal conditions	Operation limit	Survival limit
Temperature	-10°C to +35°C	N/A	-20°C to +40°C
Thermal variation (nocturnal, in 4 h)	0°C to 5°C	N/A	N/A
Relative humidity	0% to 99%	N/A	0% to 100% with condensation

Table 4.1: Dome control operating conditions

4.3.2 Design and development

The design and development of the new electronic devices and the software applications used for dome control were driven by the requirements and specifications described above.

The major tasks derived from the abovementioned requirements were:

- The nominal control system based on TALON, one of the CSIMC boards and the dome electronics was improved.
 - A new version of the software routine in charge of the nominal dome control included in TALON was developed.
 - The original dome electronics were modified and simplified by removing the more fragile components (i.e., the BC547C microchip or the IR receiver for manual control).
 - The encoder used to read the dome rotation was also changed by a shielded one, designed for industrial applications. The new encoder is a HENGSTLER Incremental Shaft Encoder Type RI 30. The installation of the new encoder required the modification of the mechanical assembly to the dome and some metal work on the encoder shaft.
- A new subsystem (SUB-SARGA) was installed to implement the redundant control of the dome aperture. It was based on a robust computer.
- The standard wiring connection was changed by a new one based on optical technology (fiber optics).

- Two new electronic interfaces were installed to integrate and to prioritize the different input signals coming from those devices responsible for the dome operation. They are based on two electronic boards, named Fiber Signal Converter 1 (FSC1) and Redundant Control and Rain Detector (CRRD). Their functionalities are described below.
- The rain detector was integrated into the housekeeping control. Its communication wiring was optocoupled and connected to the SUB-REBEI subsystem using a serial port (RS232). A software driver was developed to read those data and store them into the housekeeping monitoring.
- The rain detector was also hardwired to the FSC1 board to close the dome with no time delay in case of rain.
- A dedicated and stable power line was installed for the dome motors and electronics. It was based on a new UPS (see the UPS features in a section below).

The mentioned work improved the system performance and robustness, enabling the control of the dome rotation and the aperture movements and reducing the noise induced at the communication signals. Fig. 4.5 illustrates the new system configuration and how the telescope electronics and the subsystems interact with it. Fig. 4.6 shows the block diagram of the implemented electronics and interfaces for the dome control system.

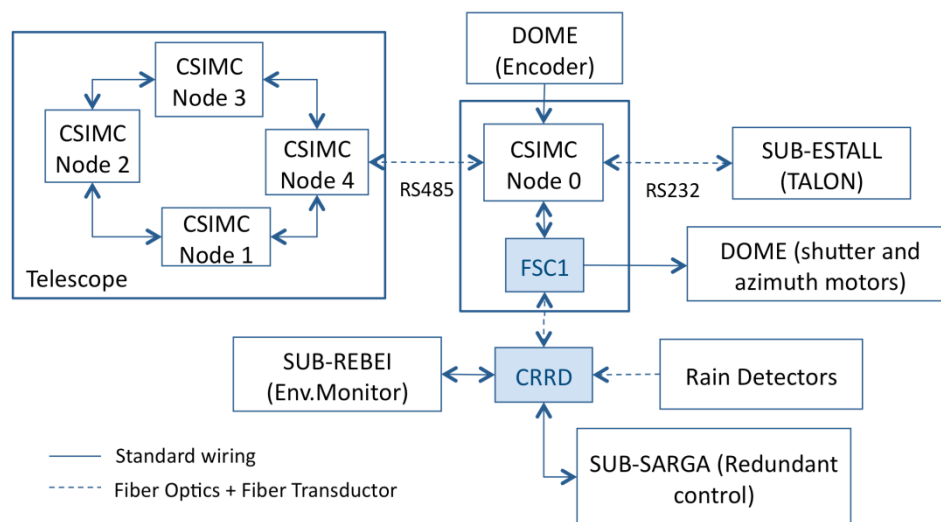


Fig. 4.5: Final version of the dome control block diagram

The main elements developed and the main system features are described in the following subsections.

4.3.2.1 Fiber signal converter - 1

The FSC1 board is located close to the dome electronics and is responsible for:

- Managing the communication between the CSIMC board that acts as node 0 to provide the link with the telescope control system installed on the SUB-ESTALL subsystem, with the redundant control system installed on the SUB-SARGA subsystem, with the signal from the rain sensor, with the manual control switches and with the azimuth rotation, shutter and flat open/close motors.
- Establishing the prioritization of each of the signals according to a predefined scheme.
- Executing the dome closing when communication is lost with any of the subsystems to which it is linked (watchdog functionality).

It uses a microcontroller (STM32F103RBT6) that contains the control firmware necessary for the operation of the device.

4.3.2.2 Redundant Control and Rain Detector

The CRRD device is located close to SUB-REBEI and it provides a connection interface, based on fiber optics, with part of the control equipment (SUB-SARGA and rain detector). Implementation constraints (computer and dome power electronics are located in different rooms) were the main motivation for it, as a direct connection among FSC1 and these devices would have been a better approach.

CRRD is responsible for:

- Communication of the dome redundant control system with FSC1.
- Relay the signal from the rain detector/s to FSC1.
- Relay the signal from one of the rain detectors to the environment control system installed on the SUB-REBEL subsystem.

The device implements an OR logic gate to enable the use of two rain detectors operating with redundancy, although it is not used in the final configuration since a new weather station equipped with a similar detector and accessible through an Ethernet connection was installed. The latter covers the need of redundancy for rain detection, but it has two important drawbacks: it is not directly hardwired to the CRRD board and, so, it produces a software activated action; its location is far from the dome. A new detector could be acquired in the near future to improve further on the robustness of the system.

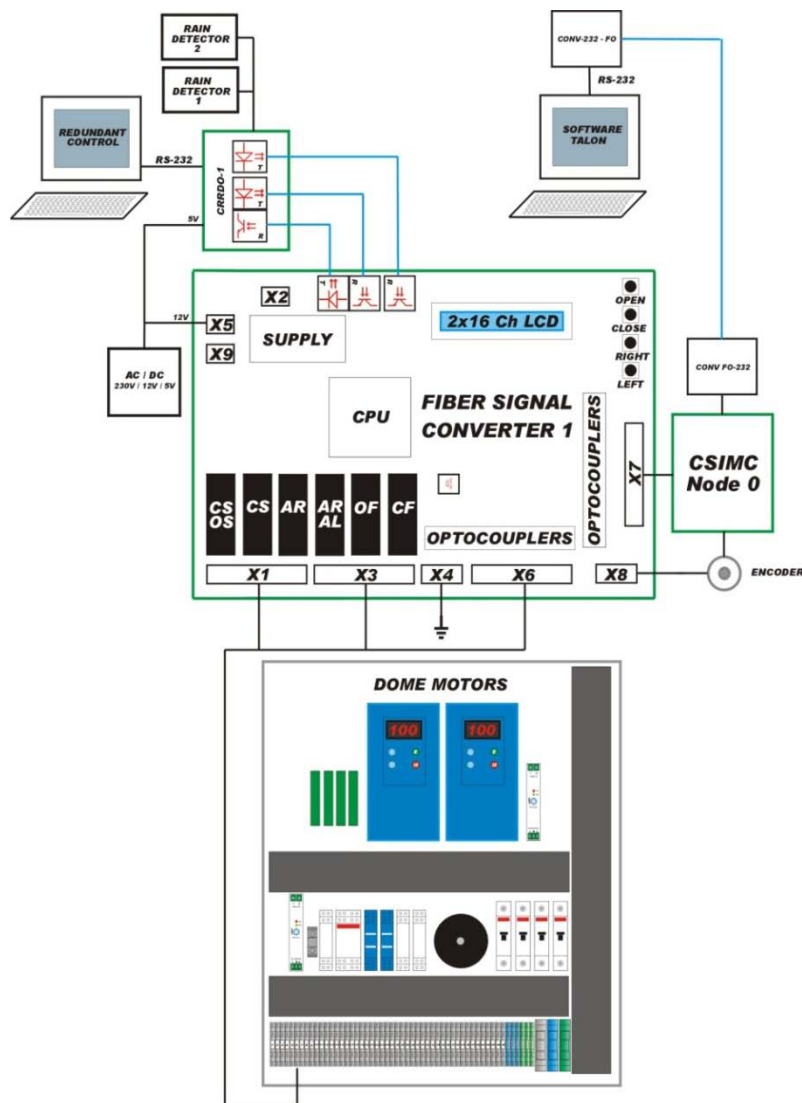


Fig. 4.6: General connection diagram

4.3.2.3 System features

➤ *Prioritization*

One of the requirements was the prioritization of the signals sent to the motors depending on where they were generated. The commands coming from the rain detector are the top priority ones, followed by the manual buttons on the FSC1 board, the OpenROCS redundant control and then the CSIMC board controlled by TALON.

➤ **Noise reduction and durability**

To minimize the electrical noise and the risk of induced power surges, fiber optics were used for any cable length exceeding a few meters and optical coupling was used in all input and output connections. This solved the existing problem with the identification of logic levels caused by the large difference in voltage (> 3.5V) between the electrical grounding of the dome and the fiber optics electronics. The new mentioned encoder (industrial version) was also installed to provide reliable operation and accurate encoder measurements.

➤ **Reliability**

Robust electronics were considered for the implementation: opto-coupled devices, communications based on fiber optics and electronic components selected to maximize long-term robustness and reliability. All the components were chosen with time stability and adapted to the harsh environment conditions specified before.

➤ **Modifications of the original control system**

A detailed analysis of the electrical connections in the electronics box provided by the manufacturer (Baader Planetarium) was carried out to understand the system and thus to devise possible ways to increase the robustness and reliability of the system. The study of the system concluded that the manual remote control provided by the manufacturer had to be eliminated. This was based on the use of electronic components of high fragility, as was repeatedly experienced after the installation of the dome. The manual control was substituted by a set of four buttons installed on FSC1.

➤ **Extra-features**

We also considered the following points in the implementation:

- The signal is multiplexed to use only one fiber optics cable to communicate the dome electronics with the rest of the TJO-OAdM subsystem.
- The dome is controlled using the telescope electronics and the TALON software. The TTL protocol and inverse logic is used for the communications between the telescope electronics and the dome, as required by the CSIMC control boards.
- An ACK/NACK system is implemented so that the dome control system executes the close command and stays in locked mode if any of the control elements it communicates with is not responding.
- A manual control feature (through a set of four buttons: open, close, right, left) for direct access to dome operations is included. Manual operations override automatic operations and lock the system for a configurable period of time.

4.3.2.4 Software

Three different software packages are necessary to implement the dome control system: FSC1 microcontroller firmware, TALON dome control routine, redundant control system software.

The firmware was designed following the specifications given in Colomé (2010) and implemented on the microcontroller installed in FSC1. The priority diagram follows the policy defined above. A development tool provided by the company that built the system (Insercad Electrónica S.L.) needs to be used for the maintenance and update of this firmware.

The TALON nominal control and the software for redundant control were developed by the software engineers at IEEC, and their description is not given here.

➤ **Flux of the commands and implemented protocol**

What follows is a description of the operation of each device and the response from the FSC1 control system:

- **Rain Detector (RD):** The closing signal is kept active while the rain detector indicates the existence of rain. The sensor closes the dome when it first detects rain (5 rain drops trigger TRUE signal of the RD) and it does not deactivate until no rain is detected for 5 minutes (subject to the configuration of the device). Once the sensor returns FALSE the system lifts the lock on the dome opening operation as requested from the manual control or the telescope system (CSIMC). The rain event does not prevent azimuth rotation of the dome.
- **Manual control buttons:**
 - One click on Close to close everything.
 - When close is completed, the system blocks the opening.
 - One click on Open to unlock opening.

- One click on Open to open everything if the system is unlocked.
- One sustained click on Close for manual close. The closing operation will stop upon lifting the button.
- One sustained click on Open for manual open. The opening operation will stop upon lifting the button.
- **Redundant Control:** The closing order is sent by the redundant control software via the command line (exec close), which executes complete dome closing and activates the opening lock. An unlock order (exec unlock) must be sent from the redundant control prior to the successful execution of an opening command coming from the telescope (CSIMC). The redundant control can also open the dome shutter (exec open).
- **CSIMC:** It operates as specified within TALON (the closing order stops as soon as the “dome closed” message is returned). It does not implement lock and unlock features but this is not necessary in our case since this is the only device that will automatically activate the opening.

4.3.2.5 Pictures and connection diagrams

We show here some pictures and connection diagrams of the implemented system (Fig. 4.7 to Fig. 4.10).

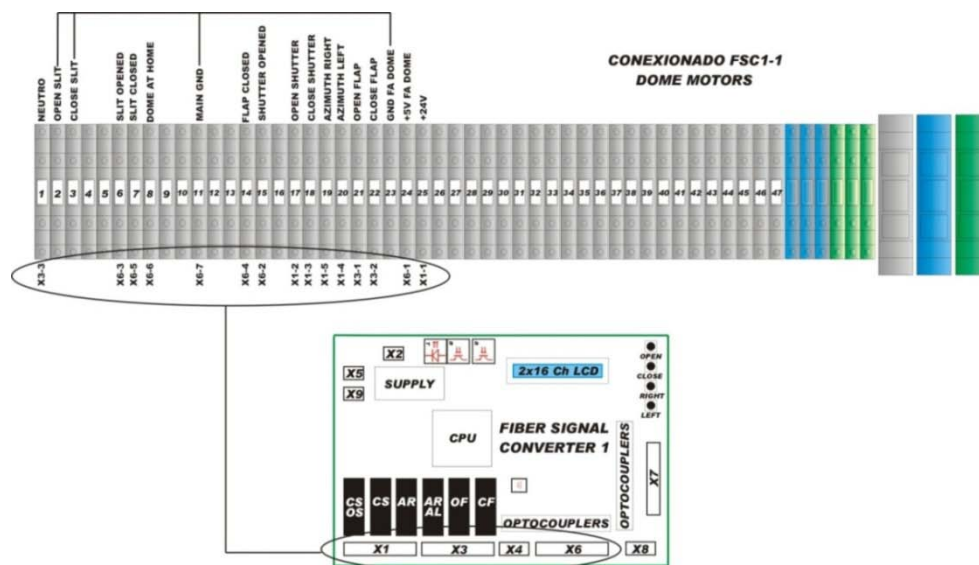


Fig. 4.7: FSC1 and dome electronics connection

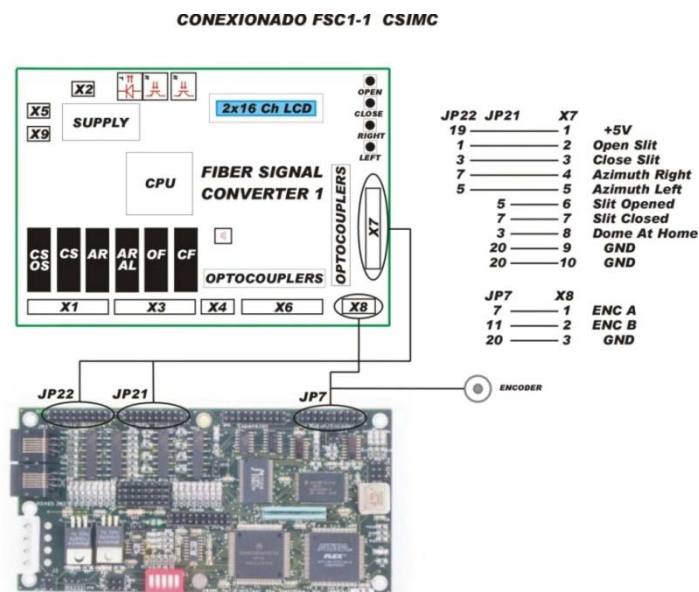


Fig. 4.8: FSC1 and CSIMC board connection

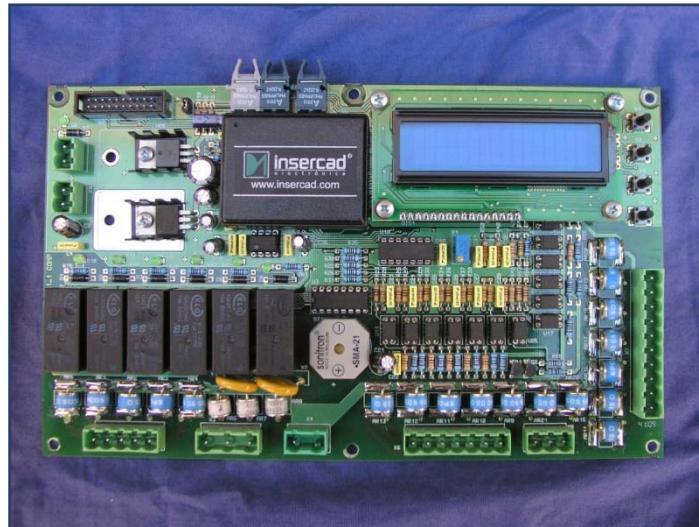


Fig. 4.9: FSC1 board



Fig. 4.10: CRRD board

4.4 MEIA Instrument

MEIA (Medium-format Efficient Imager for Astronomy) is the main instrument of the TJO telescope. MEIA is composed by a high-efficiency CCD camera, a set of Johnson-Cousins filters and a set-up for the acquisition of flat field calibration images described below.

4.4.1 CCD camera

The camera is a ProLine 4240 (model PL4240-1-B), with a back-illuminated 2kx2k chip (see Fig. 4.11 for the detector quantum efficiency curve – Basic Midband Coated), manufactured by Finger Lakes Instrumentation (FLI). The camera is physically attached to the filter wheel casing. It uses a USB connection to the computer (SUB-ESTALL). The low-level software for cooling and image acquisition control is based on the Camera routine of TALON that uses the FLI drivers for Linux.

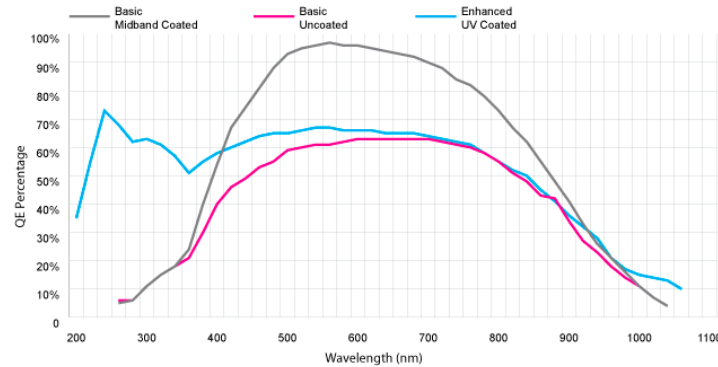


Fig. 4.11: CCD camera Quantum Efficiency (FLI, www.flicamera.com)

4.4.2 Filters

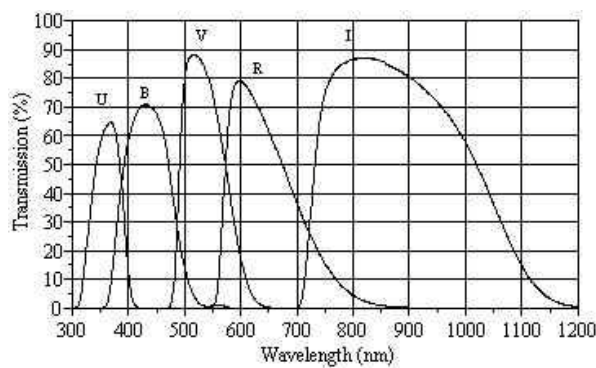


Fig. 4.12: Transmission curve of the Johnson-Cousins filters

There are two main components related to the photometric filters:

- The filter wheel: It is physically coupled to the telescope at the back of the primary mirror assembly. It can hold up to 9 3-inch filters.
- The Johnson-Cousins photometric filters (Fig. 4.12): The 5 filters that are currently installed are U, B, V, R, and I, and can be placed at the optical axis of the telescope by rotating the filter wheel.

4.4.3 Instrument calibration and dome flats

The initial approach regarding the set-up to acquire the necessary calibration data did not foresee any supplementary hardware, apart from the scientific devices already mentioned. We defined in the system operational design that the tasks to obtain the master Bias, Dark and Flat calibration frames had to be executed with a daily periodicity (see Sect. 3.1.4). They could be obtained with the set of components mentioned so far (telescope, dome and instrument), although it implied flat fielding images had to be taken by pointing the telescope at sky areas with homogeneous illumination during the twilight. This procedure has some drawbacks that turn to be critical for the unattended operation: the selection of the sky areas is quite difficult in case of clouds, exposure time to obtain the required signal level is highly dependent on the acquisition time and the filter used, and tuning exposure time is time consuming and can drastically reduce the number of images obtained for each filter. The mentioned points motivated the implementation of an alternative set-up to perform dome flats, although sky flats were not discarded. In any case, this set-up would be also necessary when bad weather conditions made impossible to obtain sky flats.

We finally prioritized the Flat frames acquisition policy based on dome flats images. A comparison between both possible configurations is ongoing to check for significant differences in terms of image quality. Dome flats are more adequate for an unattended observatory to ensure that a whole set of flat field calibration images are obtained every day. Meanwhile the system stability is not tested (routine operations will determine this point), at least two sets of dome flats images are scheduled every day during the twilight (dusk and dawn).

It follows a description of the new set-up we implemented.

4.4.3.1 Dome flats design and development

We designed the implementation of this new set-up for dome flat field acquisition based the following general requirements:

- Flat field images have to be obtained with homogeneity better than 1% RMS.
- The operation has to be done automatically and integrated into the observatory control system.
- Task execution definition has to enable its selection by the automatic scheduler routine without any interference with the science observation tasks.

The system designed to obtain dome flats at the TJO-OAdM is composed by three main elements, a white screen mounted on the inside of the dome, a full-spectrum light source that illuminates the screen and a control system to turn on the light when required by the observatory control system.

➤ *The Flat Screen*

The screen used for the dome flats is a 4:3 ratio white MovieBoard from Projecta. The screen is 129×169 cm and the viewing area 120×160 cm, giving a diagonal of 200 cm, and has a weight of 14 kg. The screen is matte white and has a reflection value of 1.0 and a viewing angle of 50° left and right. The screen is mounted on a black aluminum frame. The actual screen is a sandwich PVC board filled with polystyrene foam.



Fig. 4.13: Flat screen mounted inside the dome

The screen is mounted inside the dome attached to two supports of the dome shutter opening mechanism (Fig. 4.13 and Fig. 4.14). Cable tensors were used to tighten the cables holding the screen firmly in place. These tensors also allow the angle of the screen to be adjusted, so as to position it with respect to the telescope and flat light.



Fig. 4.14: Coupling system to hold the flat screen

➤ *Flat light*

Flat images must be taken using the filters U, B, V, R, and I. This range is from 320 nm to 400 nm for U, and 700 nm to 900 nm for I, thus it is important that the light used covers this spectrum.

Unfortunately, virtually all lights on the market have special coatings which filter out UV light, as it is dangerous for humans. Thus we decided to make a compromise and use a lamp that covered well the other parts of the spectrum (BVRI) and to simply do longer exposure times for the U filter.



Fig. 4.15: DLH200S Daylight Soft Light (Dedolight)

The selected lamp, DLH200S manufactured by Dedolight (Fig. 4.15,), has daylight spectrum with its energy distribution peaking at 5500 K and with rather uniform output in all wavelengths within 400 nm and 900 nm. In the Fig. 4.16 transmission of the filter on the DLH200S lamp is shown (with the UV coating in place). As can be seen it drops rapidly below 400 nm. UV protection coating could be chemically removed to allow UV light to pass through. This option was finally discarded because the MEIA instrument was not optimized in that wavelength indeed (see Fig. 4.11), and just a small fraction of science proposals would have required the use of the U J-C filter. Safety reasons also justified the decision.

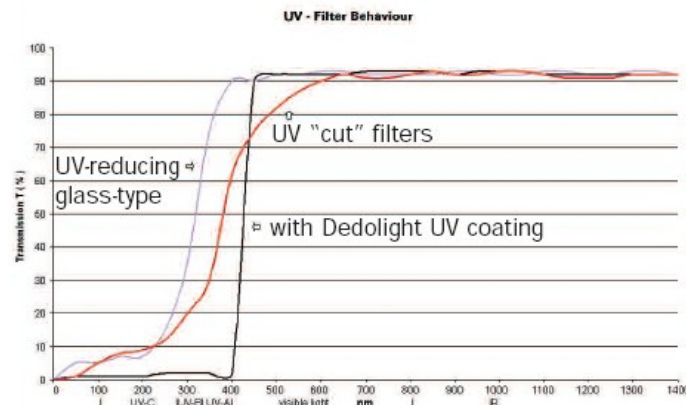


Fig. 4.16: Transmission of the DLH200S Daylight Soft Light (Dedolight)

The flats lamp is located inside the dome and it is installed adequately to obtain dome flat images with minimum illumination gradient.

The lamp is attached to a 0.5-m articulated steel arm that is in turn attached to the wall with a steel plate. This setup gives quite a bit of freedom of movement, allowing the flat light to be positioned to illuminate the screen, while not sending stray light towards the telescope. To help assure that the light is uniformly emitted towards the screen, a diffuser surrounds the light (Fig. 4.17).

The 200-W power source for the light is mounted on the wall beside the light setup, and is housed within a protective case.

➤ **Light control**

The powering on and off of the lamps is controlled by SUB-ESTALL. The power source is plugged into an outlet of a Power Distribution Unit (PDU, see below), and can thus be turned off and on via the control system software and using SNMP protocol. Thus, it is possible to perform flatfield images automatically.



Fig. 4.17: Flat light mounted on the wall, with the diffuser surrounding the light

Regarding the control configuration, the control system must wait the warming up time for the flat lamp to reach the nominal temperature where the light stability is at its maximum. At that time, the task execution control can start acquiring flat field images for the different filters. Finally, the lamp should not be turned off until the nominal temperature is reached to avoid a reduction of the lifetime of the lamp.

➤ *Final configuration*

The manufacturer recommendation to apply a 5 minutes delay before taking flat images to let the light warming up first and reach a stable light level was found to be an overestimate. Tests performed showed that the light reaches its maximum in less than 2 minutes.

To test the uniformity of the flat screen and illumination system flat images of the screen were taken with the telescope in equivalent positions with 180 degrees rotation. These two images can be subtracted from one another and measure the gradient on it (the difference should be zero if there was no gradient). Various light orientations and screen inclinations were tested and it was found that by pointing the light towards a black fabric sheet on the ground (to diffuse the light more) and by adjusting the screen inclination a maximum gradient across the flat screen of approximately 1% was obtained.

The optimal exposure times measured for each filter are shown in Table 4.2. These exposure times give an average pixel count of around 40000 (out of a maximum of 65500).

Filter	U	B	V	R	I
Exposure (s)	200	20	12	7	7

Table 4.2: Mean exposure time per filter (flat fields)

4.4.4 Instrument modifications

- Fiber optics: Optical fiber and fiber transducers were installed to communicate the SUB-ESTALL control computer with MEIA. It constituted a great increase in reliability and robustness of the instrument.
- Some parameters describing the camera configuration (gain, readout noise, etc.) were not included by TALON in the acquired image headers. In order to include them automatically, the newly developed data processing routine was designed to also add these missing keywords in the image header.
- The CCD was supplied with a significant field vignetting at the detector edges. The manufacturer changed the camera casing to solve this problem, based on the casing and electronics of the ProLine camera model.
- A new set-up for the acquisition of dome flat field images was designed and developed.

4.5 Housekeeping

Housekeeping is the most critical aspect when considering safety in robotic operations. It involves real-time environment knowledge and hardware reliability, aspects that have been dealt at the TJO-OAdM with the appropriate equipment.

4.5.1 Support devices for environment monitoring

The use of just a single weather station would be suitable for most observatories, but it is insufficient for a robotic facility. There shall be, at least, as many sensors as the number of environment variables to monitor. For those variables most critical for the observatory safety, redundancies are mandatory. The environment control at the TJO-OAdM is carried through the set of devices described here. Some particularities about the sensor connections are mentioned.

- **Previstorm:** This sensor, manufactured by INGESCO, records the variations of the electrostatic field (kV/m) to predict thunderstorms. The sensor is located close (10 m) to the main building entrance. A coaxial cable connected to the Data Acquisition Module, placed inside the building, is used for powering and receiving data from the sensor. An RS232 connection is then used to read the data from the SUB-REBEI computer for data logging. A specific Linux driver was developed to read the data. The Previstorm is configured to disconnect several power distribution lines, using specific relays, when the measured electrostatic field exceeds a configured value.
- **Weather stations:** The TJO-OAdM has two different outdoor weather stations to ensure reliable data:
 - Davis Vantage Pro II (EM-Davis): Davis weather station with three different nodes. One node is located in the communications tower outside the building and it is equipped with sensors to measure wind, humidity, temperature and pressure. A second node is in the dome and it has temperature and humidity sensors. Last node is in the control room and it provides also temperature and humidity data. All three nodes are radio-connected to a central hub located in the control area where SUB-REBEI is connected for data transmission executed via an RS232 cable link and using a Linux based driver provided by the manufacturer.
 - Campbell Sci. weather station (EM-SMC): Campbell station belonging to the Catalan Meteorological Service (SMC) owned by the Catalan government and installed in the communications tower. The signal is transmitted from the weather station to the SUB-RUBIES computer using RF antennas and RF to RS232 transceivers. Communication is done executing the LoggerNet 3.3 (Campbell Sci.) software, an MSWindows application that enables a data acquisition run with a preconfigured frequency (currently set to 1s). Data is served through the network by a server-client routine. This station is included in a network of weather stations distributed over Catalonia that are continuously monitored and maintained, providing high quality and high reliability data.

Comparison of the measurements from the two stations guarantees safe decisions on humidity and wind speed to preserve the integrity of the system. Data links to SUB-REBEI (via SUB-RUBIES in the case of the SMC) provide the data to the control system.

- **Rain detector:** The rain detector is located on the terrace above the control area. It is an IRSS88 model manufactured by Eigenbrodt. A new custom-made device had to be manufactured to communicate with a standard serial port (RS232). The detector signal (aperture and closure of a switch) had to be converted to a digital signal while maximizing the robustness of the channel. Special care was put on the design and development of this device (finally manufactured by Insercad Electronica S.L.) as it is the most critical element on the system security. A Linux driver was developed to read the status through the serial port.
- **Cloud sensor:** The Boltwood Cloud Sensor II is located on the terrace above the control area. It provides differential information about the sky brightness temperature (and thus cloudiness) to SUB-REBEI. The cloud sensor is not integrated in the environment monitoring system, yet. An RS232 connection is available and it is physically connected to the SUB-REBEI computer, but no effort has been done to develop a Linux driver to read the data as its reliability and utility to derive the level of the sky coverage by clouds or the sky transparency is not clear (Marchant 2008). Cloud sky coverage is planned to be measured by an allsky camera recently installed on the observatory site. The cloud sensor, then, will be probably used just as a low priority indicator.

The information obtained with these sensors increases safety, but a tool to manage the data and the generated alarms is required. The Environment Monitoring module was developed with that purpose and included in the observatory control system (Sect. 5.2.1.2).

4.5.2 GPS

A GPS (Garmin) antenna for the time-base determination is located in the main building above the control area. The signal is transmitted via a shielded communications cable directly to SUB-ESTALL using RS232 protocol and a Linux-based driver installed with the TALON software is used to read the data.

4.5.3 Power and Electronics

The reliability of the power supply and the protection of signals against induced current and perturbations was also a matter of concern. We increased the power supply security, its stability and reliability by using several Uninterruptible Power Suppliers (UPSs), software controlled switches (PDUs) and electric insulation (opto-couplers).

The observatory is equipped with these instruments to ensure a continuous supply of energy and to avoid any unforeseen voltage distortion or power outage. These are mandatory due to the isolation of the site and the characteristics of the local weather, with frequent occurrence of lightning during storms, especially in Summer. The initial design of the building considered the installation of two lightning rods, a deep connection to ground and the storm detector mentioned above for lightning protection. But some events showed that more protection and filtering was needed: voltage peaks produced by induced current in the communication wires, perturbation of the communication signal due to proximity of power lines, random power outages, etc.

4.5.3.1 Power

The OAdM is powered by main frame power from the Endesa Company. The facility also has a backup 35-kW generator and solar panels that provide sufficient energy to the system to continue operations even in the case of a power outage. The warm-up time of the generator (about 35 seconds) before it switches on power production is covered by the UPSs.

4.5.3.2 UPS

Three UPS units (PowerWare Eaton 9120), using double-conversion online technology (level 9, the maximum level of protection, with batteries always connected to an inverter, so no power transfer switches are necessary) were installed inside the main building to supply power to the equipment that is very sensitive to power fluctuations and require electrical isolation: astronomical instrumentation, sensors, computers and dome. These UPSs provide 3 kVA of power each and allow a typical backup time of 15-30 min. They have the capability to execute a computer-controlled switch off.

The UPSs were configured to trigger a system shutdown that is monitored with the NetWatch software running on the different subsystems. The connections and devices powered by each of the UPSs are detailed below:

- **UPS-TJO:** It powers exclusively the TJO electronics and drives (20% load in peak telescope operation)
- **UPS-Dome:** It powers exclusively the dome (30% load in peak dome operation).
- **UPS-CPUs:** It powers the entire computer and communication system (35% load during operation).

The notification of the power outage is done some time (60 s) after the AC power is lost and the UPS shuts down after a wait period (600 s for UPS-Telescope and UPS-Dome, and 900 s for UPS-CPUs). Power is supplied 300 s after the AC power is back (30 s in the case of UPS-Dome). These timeout values are subject to configuration and enable a soft shutdown of the system at any time.

A remote monitoring of the UPS status based on SNMP protocol commands is also possible (and under implementation) providing the control system with the capacity to generate proactive events and actions when, i.e., a power outage is detected.

4.5.3.3 Electric insulation for lightning and power surge protections and noise reduction

The facility is appropriately protected against lightning strikes and induced power surges using different kinds of filters (for Ethernet, RS232, radio link, etc). Protection elements were installed in all electronic connections when communication between components required a cable length above a few meters. Commercial solutions, manufactured by Cirprotec, were used for standard serial ports and network connectors, and opto-coupler elements were implemented for the rest. Three filter class, also from Cirprotec, are also used in the power distribution lines depending on the potential danger of the device to large overcurrents.

Different grounding for physically linked components also caused a distortion of electric signals. This could only be solved by electric insulation.

➤ **Fiber Optics Transceivers**

Reduction of noise in the communication wiring was achieved by using opto-couplers or fiber optic cables. The fiber optic transceivers transform electric signals into light signals. Using optical fibers, those devices that are most vulnerable to induced power surges were isolated. Such devices are used to intercommunicate the TJO-OAdM telescope, the dome and subsystem computers (SUB-ESTALL and SUB-SARGA).

➤ **Electrical noise**

Electrical noise was a big concern for dome encoder signals, because the cable transmitting this signal was very close to dome azimuth and shutter motors. These three-phase AC motors produce current voltage peaks when movement starts, which caused false signals and frequent encoder damaging. The use of a new encoder with high noise interference immunity in combination with fiber optic wiring drastically reduced the noise and corrected the position detection problem.

4.5.3.4 Power Distribution Units (PDUs)

The PDUs permit remote or automatic control of the power supplied to the different devices. Three APC PDUs (AP7920) with eight outlets each are installed at the TJO-OAdM:

- **IP-Switch 1:** It is located in the control area. It powers all the computers, except SUB-ALIS, the RS232 to fiber optics transceiver for the connection with the CSIMC node 0, the flat lamp, the FSC1 box (containing the RS232 to fiber optics transceiver for the connection with the CSIMC node 0, the FSC1 board and the RS485 to fiber optic transceiver), the indoor webcam light focus (see Sect. 4.5.4), the environment sensors, the GPS and the CRRD board. It is powered by the UPS-CPUs output line.
- **IP-Switch 2:** It is located on the ground floor at the pier of the TJO. It powers the electric panel situated inside the dome behind the telescope and, through this, the entire telescope electronics, drives and the MEIA CCD camera. It is powered by the UPS-Telescope output line.
- **IP-Switch 3:** It is located in the communications area on the ground floor. The two power supplies of SUB-ALIS, the communication system (Router-LAN, Ethernet switches, RF device antennas, radiolink fiber optics transceivers), the outdoor webcam light focus (see Sect. 4.5.4) and other devices installed in the communications tower are powered by this PDU. It is powered by the UPS-CPUs output line.

PDU outlets states are operated by the control system based on SNMP protocol commands.

4.5.3.5 Electrical Wiring

The electrical wiring of the TJO-OAdM is shown in the figure below. The source of energy is the main frame power (FECSA on the figure) or the on-site generator (EPG; automatic switching). The UPSs and PDUs distribute the energy to all devices and ensure electrical power for enough time in case of electric power cut. All devices are protected against power surges using filters.

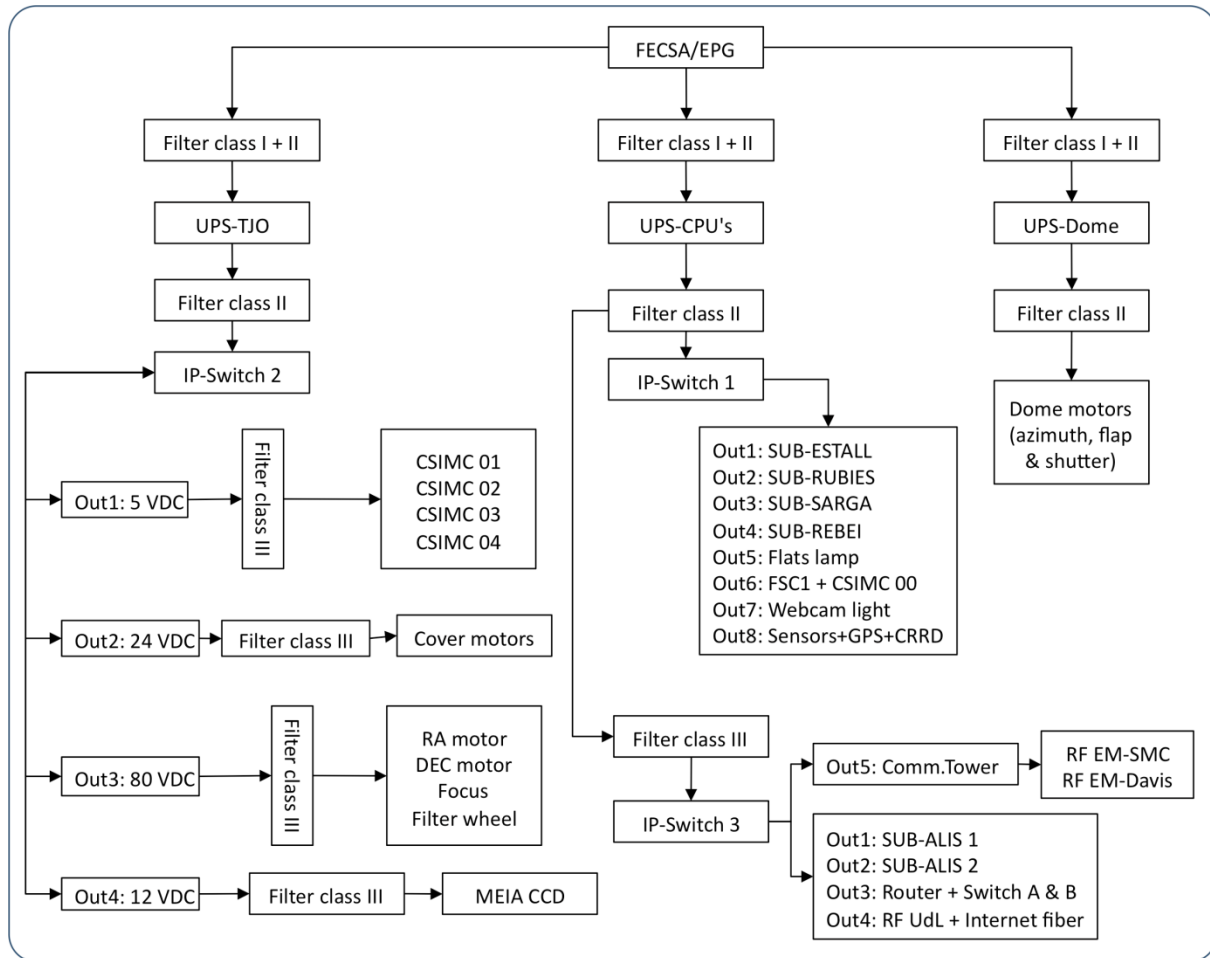


Fig. 4.18: Scheme of the electrical wiring at the TJO-OAdM

4.5.4 Visual remote monitoring

A remote monitoring system to visualize the state of some components was deemed important to increase the security, although it is not strictly necessary for a robotic telescope.

Two webcam devices (AXIS 211) were installed at the observatory for remote monitoring of the main critical areas and devices of the observatory:

- **Indoor camera:** This camera gives a general view of the dome area, including the telescope and the dome shutter.
- **Outdoor camera:** The TJO-OAdM building, the main entrance and the dome are externally seen in real-time thanks to the images taken by this webcam.

These two cameras are both complemented with a light focus that is switched on to illuminate the areas when a visual inspection is required during night hours, for the outdoor vision, or when the dome shutter is closed, for the indoor vision. These focuses, located close to the cameras, are controlled through the outlets of two PDUs (IP-Switch 1 and 3), as described before.

These devices (cameras and light focuses) are provided with web-based user interfaces that are linked directly from the observatory web page.

4.5.5 Modifications on the housekeeping

The description given above for the housekeeping physical architecture is a compilation of all the components that integrate the final system implemented. The following list shows the devices not included in the original configuration that we added or integrated:

- Environment monitoring: Rain detector and EM-SMC integration in the control system, upgrade of the EM-Davis and installation of new nodes inside the building.
- Power and electronics: UPSs, PDUs, electric insulation (i.e., fiber optics), protection filters.
- Visual remote monitoring: Webcams and focuses.

The modifications increased the system robustness and reliability and provided redundancy on one of single-points-of-failure previously identified: the environment monitoring reliability.

Fig. 4.19 and Fig. 4.20 illustrate the differences between the initial and final physical architecture designs.

4.6 Communications

Communication among all the devices is based on standard ports and protocols. A high level of homogeneity was envisioned, although not completely achieved due to the different communication ports used by the commercial solutions found. Ethernet and serial port connection (RS232 or USB) are the most common ones at the TJO-OAdM.

4.6.1 Internet and LAN

Internet communications is one more element to take into account when facing the safety of the observatory. A radio link antenna of 10 Mbps bandwidth is used for external communication. It connects the observatory with the Anella Científica (Scientific Ring), a high-speed communications network (2 Gbps bandwidth) managed by CIESCA (the Supercomputing Center of Catalonia), that connects universities and research centers in Catalonia. The network is connected to RedIRIS, the Spanish research network. Through RedIRIS, it reaches the most advanced international research networks. Continuous monitoring of the internet connection is done by CIESCA and also by the control system of the observatory that checks the link with SUB-AGER, the off-site subsystem, at a high cadence. Both CIESCA and the control system activate an external and an internal alarm protocol, respectively, when the connection breaks. The internal protocol is designed to stop any operation at the observatory to minimize the risk of the damage that could be produced in case a severe alarm happened in simultaneously (i.e., dome shutter blocked while closing).

LAN monitoring is also done locally at the observatory by checking the connection with the router, and an alarm manager is in charge of the execution of an alarm protocol if necessary.

Internet and LAN errors are analyzed in other chapters (see below).

The systems related to communications of the TJO-OAdM are:

- **Router:** It is located in the communications area in the main building inside a computer rack. The router provides NAT and firewall services and prioritizes the communications depending on the IP address.
- **Switch A and B:** They are located in the router rack and connect all the computers to form a LAN.
- **Radiofrequency antennas:** They are installed outside the building in a tower (called communication tower) to attain direct vision with the receiving antenna in the campus of the University of Lleida (UdL). This provides a link to the Internet with a bandwidth of 10 Mbps.
- **Radiofrequency amplifier:** It is located inside the building on the ground floor inside the router rack.

4.6.2 Communications Wiring

Fig. 4.19 and Fig. 4.20 show the final and the initial wiring design for the system internal communications. Optocoupled links (RF antennas, fiber optics and filters) are used everywhere to avoid induced power surges and problems with grounding. Black arrows correspond to Ethernet connections, red arrows to optical fibers, green arrows to serial ports (RS485 for the CSIMC network, USB for the CCD and RS232 for the rest) and blue arrows to standard wiring connections.

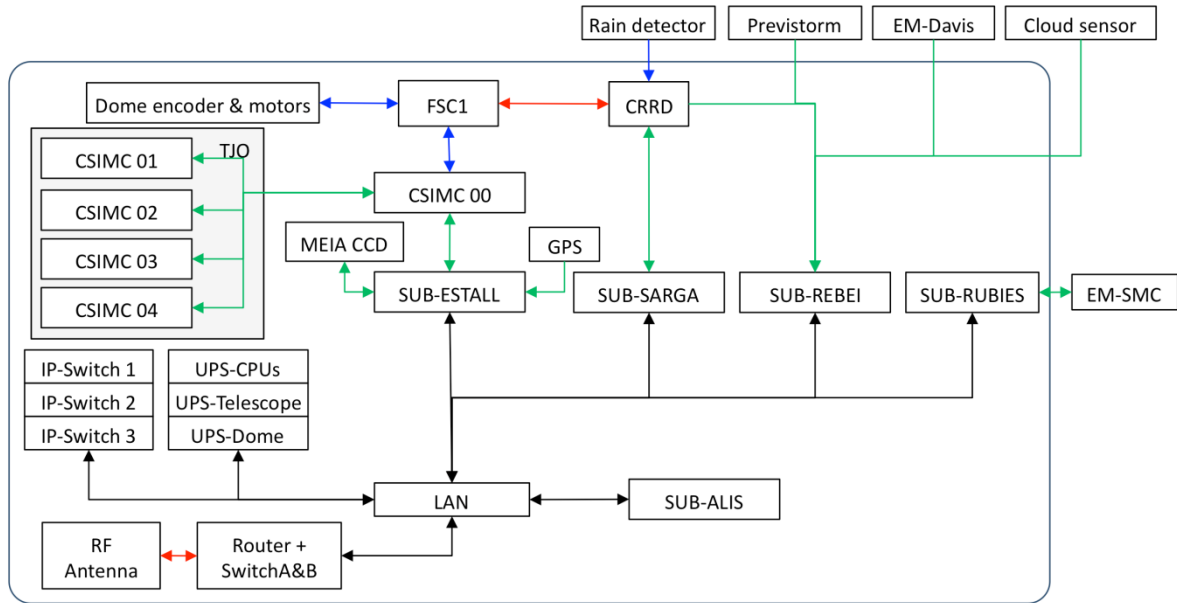


Fig. 4.19: Scheme of the communications wiring at the TJO-OAdM

4.6.3 Modifications on the communication architecture

The communication architecture design at the pre-robotization project phase (Fig. 4.20) had several drawbacks that decreased the system robustness, modularity and efficiency. In terms of robustness, optocoupling, for instance, had not been implemented for all the connections, although some of them had already been protected with filters (Ethernet and RS232) after the first lightning strike that occurred at the observatory in 2004.

During the robotization project, the subsystems were better defined and separated to reduce their interaction and the possibility of the concurrence of errors. An independent Environment Monitor, for instance, was implemented in a single subsystem (SUB-REBEI) to keep the collection of data running in case the main telescope control computer (SUB-ESTALL) suffered a shut down. This subsystem was based on a robust computer and some electronics of high reliability.

And, finally, some devices were integrated by developing new communication channels or modifications were made on the existing ones increasing the number of support elements. This work was done during the system redesign and contributed to achieve a better performance and higher efficiency.

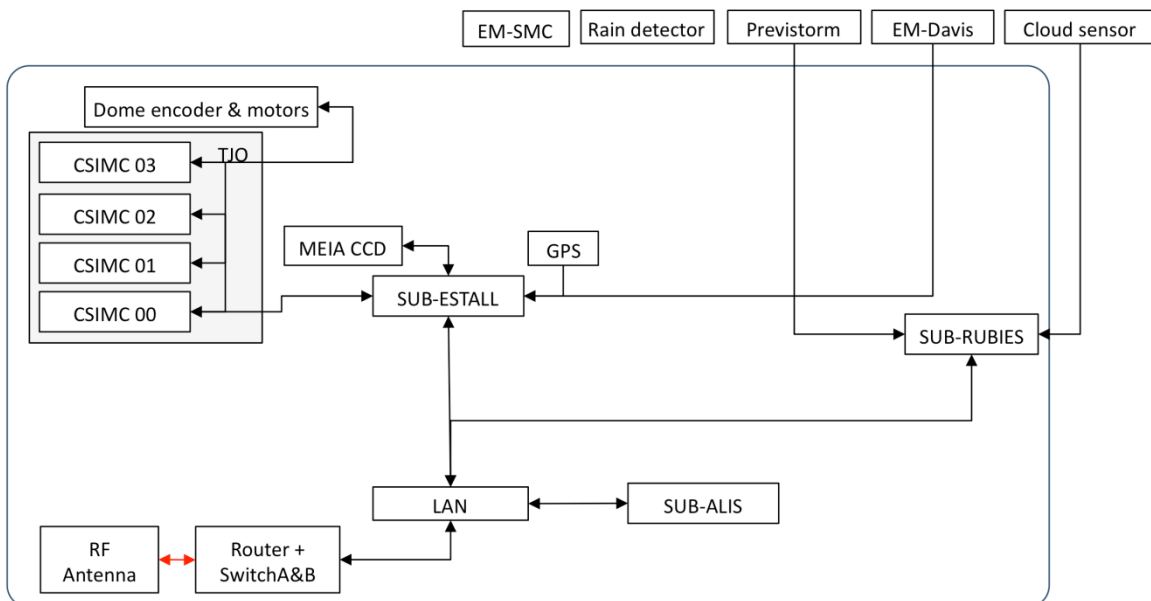


Fig. 4.20: Scheme of the communications wiring at the TJO-OAdM (initial design)

4.7 Subsystem computers

The TJO-OAdM has six computers implementing five subsystems (see Fig. 3.3), which supervise the correct behaviour of the TJO telescope, the control system and the housekeeping. Four of the computers (SUB-ESTALL, SUB-SARGA, SUB-REBEI, SUB-RUBIES) are located in the control area at the TJO-OAdM main building (first floor) next to the telescope, the instrument CCD camera, the dome and the flat lamp. A fifth one (SUB-ALIS) is in the communications area, next to the communication system (router, switch, radiolink electronics). The computers are interconnected configuring a LAN net. Finally, another computer (SUB-AGER) is located off-site to remotely supervise the observatory, to act as a final repository of data and to provide the required user interface for proposal submission and data retrieval.

Most of the computers run on Linux, chosen to be a high reliability OS, and have the OpenROCS software installed, with the exception of SUB-RUBIES that runs on MSWindows.

SUB-SARGA and SUB-REBEI were added when designing the control system during the robotization project phase. They were introduced to increase the system robustness on the two identified single-points-of-failure described below. Industrial computers were chosen with that purpose.

A functional description of each subsystem was given in the logical architecture introduction. The main features of the computers in charge of their control are summarized here:

➤ SUB-ESTALL

- **Computer:** Efficient and reliable DELL computer (OptiPlex 780 MT) with high processing capacity (Intel® Core 2 Quad Q9650, 3GHz,1333MHz,12MB) and an internal RAID1 with three hard disks of 500 GB each.
- **Communications:** SUB-ESTALL communicates directly with the TJO and the dome through an RS232 link (optical fiber channel) to one of the CSIMC cards (node 0). The system is also directly connected with the MEIA CCD camera via USB (through a fiber optics link, too). The GPS data is received directly by this system via an RS232 communication link. The system has software control over flat lamp via a PDU. The system monitors UPS-TJO via the NetWatch software.
- **Software:** SUB-ESTALL runs on Scientific Linux and operates the entire TJO control software (TALON) and OpenROCS.
- **Power supply:** SUB-ESTALL is powered by IP-Switch 1 that receives electric energy from the UPS-CPU.

➤ SUB-REBEI

- **Computer:** iEi industrial computer with ULV Intel Celeron M 1 GHz fanless system processor and 1 GB 266 MHz DDR SDRAM memory. It is an extremely robust and compact computer, equipped with 5 RS232 and 4 USB ports.
- **Communications:** SUB-REBEI receives data from the Previstorm monitor, the Davis weather station and the cloud sensor via RS232. It also collects data from SUB-RUBIES on the redundant SMC weather station. The system provides an environment data stream to any client requesting it.
- **Software:** OS installed on this computer is Linux Custom ArchLinux with squashFS and UnionFS. It is built on an SD memory card in read-only fashion. The system has driver software to process the data streams from the Previstorm, the Davis weather station and the cloud sensor. OpenROCS also runs on the system.
- **Power supply:** SUB-REBEI is powered through IP-Switch 1 that receives electrical energy from the UPS-CPU.

➤ SUB-RUBIES

This subsystem is included at SUB-REBEI as its only purpose is to provide data to the Environment Monitor installed in it.

- **Computer:** Desktop computer with standard processing and disk capacity.
- **Communications:** SUB-RUBIES is connected to the EM-SMC weather station via an RS232 link (radiofrequency).
- **Software:** SUB-RUBIES runs on MSWindows Server 2003. Logger-Net 3.3 software is installed to administrate the connection to the EM-SMC weather station and to format the weather data so that it can be delivered to the environment control system.
- **Power supply:** SUB-ESTALL is powered by IP-Switch 1 that receives electric energy from the UPS-CPU.

➤ SUB-SARGA

- **Computer:** iEi industrial computer with ULV Intel Celeron M 1 GHz fanless system processor and 1 GB 266 MHz DDR SDRAM memory. It is an extremely robust and compact computer, equipped with 5 RS232 and 4 USB ports.
- **Communications:** SUB-SARGA is connected with the dome and the rain detector through the CRRD system. The system monitors the UPS-Dome via the NetWatch software.
- **Software:** OS installed on this computer is Linux Custom ArchLinux with squashFS and UnionFS. It is built on an SD memory card in read-only fashion. The system runs OpenROCS and takes care of the CRRD of the dome.
- **Power supply:** SUB-SARGA is powered through IP-Switch 1 that receives electrical energy from the UPS-CPU.

➤ SUB-ALIS

- **Computer:** HP server (Compaq ProLiant ML350) with redundant power supply, high processing (two Intel Xeon 2.2 512 Kb and 1280 Mb PC2100 DDR SDRAM memory) and storage (276 GB RAID5, implemented on 4 hard disks) capacity and two Ethernet cards.
- **Communications:** SUB-ALIS receives data through the LAN from SUB-ESTALL, including the images and logging information. The system is also monitoring the UPS-CPU via the NetWatch software.
- **Software:** SUB-ALIS runs on Linux (Debian, to be upgraded to CentOS) and has the database and the scheduler and data processing software. The back-up system is installed in this computer (back up of data, images and logging). OpenROCS also runs on the system.
- **Power supply:** SUB-ALIS has two power supply units and both are connected to IP-Switch 3, in turn receiving energy from the UPS-CPU.

➤ SUB-AGER

- **Computer:** Efficient and reliable server computer with high processing (Genuine Intel® Core™ i7 CPU 950 3.07GHz) and storage (RAID6, implemented on 4 500 GB hard disks) capacity with redundant power supply.
- **Communications:** SUB-AGER only uses the Ethernet connection.
- **Software:** SUB-AGER runs on Linux (CentOS) and has the database (mirror of the SQL database at SUB-ALIS), the back-up system and the web-based user interface, requiring web server software (Apache). OpenROCS also runs on the system.
- **Power supply:** SUB-AGER is installed at the IEEC facilities in the UAB university campus, powered by a stable and reliable electric connection. A dedicated UPS (PowerWare, 2.2 KVA) is used as a backup power supply in case a hypothetical power outage happens.

Chapter 5

Robotic Observatory Control System

The control system is the fundamental software layer to achieve the qualified robotic operation, as introduced in the Logical Architecture design description given in Sect. 3.3. It is responsible for the decision making capability of the observatory and it has a direct impact on the system performance. We describe in the following sections the design and development of this control layer for the TJO-OAdM.

The acronym ROCS, standing for Robotic Observatory Control System, is used to designate the control system of the TJO-OAdM robotic telescope.

5.1 Main control layers

The conceptual design of the control system is based on the two layers of software introduced in Chapter 3: the Housekeeping control layer and the End-to-end data flow control layer. Both layers are integrated into a software infrastructure that implements the distributed control of any subsystem and provides a homogeneous way to implement and execute all the software applications. Fig. 5.1 illustrates the general design of the TJO-OAdM control system.

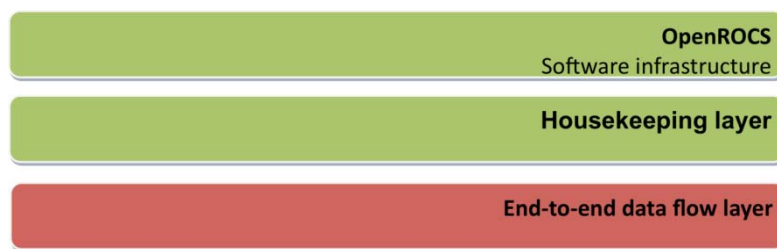


Fig. 5.1: Control system conceptual design

We initially included the capability to handle events and alarms in each particular layer, but we finally moved it to the software infrastructure during the implementation to provide a more homogeneous response and to ensure the integrity of the control system.

The need to provide the observatory with a suitable software infrastructure and a complete suite of applications to manage all the required processes is derived from the system architecture design. Its major components are listed here and widely described in the coming sections and chapters of this Thesis (see the architecture design in Fig. 5.2):

- **Housekeeping control layer:**
 - Integral environment and system (hardware and software) monitoring.
 - Redundancy control.
- **End-to-end data flow control layer:**
 - Proposal management tool.

- Scheduler for task selection (ISROCS).
- Observation process control software to interface with the TALON software.
- Data processing package (ICAT).
- **OpenROCS:** This tool takes the role of the software infrastructure that controls all the necessary routines that implement responses to identified events and alarms. It also interfaces with all the applications running at the TJO-OAdM observatory for data flow control purposes and the database (DDROCS). And it implements those applications devoted to housekeeping tasks that require interfacing with the hardware layer components.

OpenROCS is the core element of the control system. We designed it as an independent layer to be exported to other robotic observatories and will be released under public license in the future.

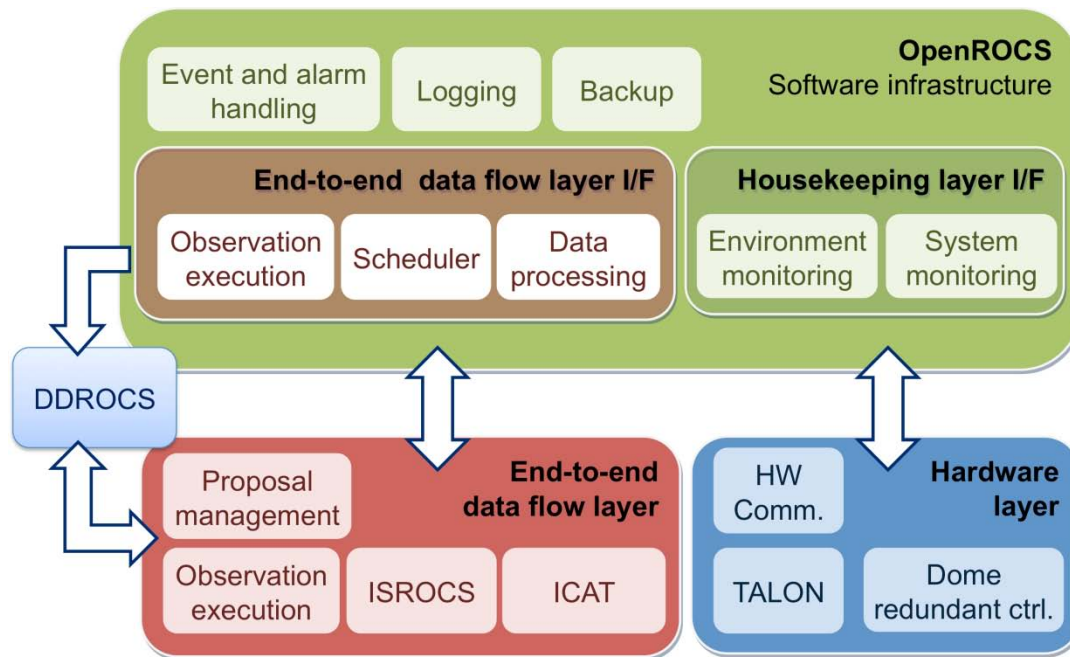


Fig. 5.2: Control system architecture design

5.2 Housekeeping

The Housekeeping control layer was defined in the System Architectural design (Chapter 3) as the software layer in charge of monitoring all the environment sensors and system components health, and handling events and alarms produced by this layer and their corresponding actions. This layer of control is the most critical one regarding the system reliability and robustness. It is important to stress that the two identified single points of failure mentioned in Section 3.2 fall in this layer.

We describe in this section the main modules required for the Housekeeping layer.

5.2.1 Housekeeping analysis

The figure below (Fig. 5.3) illustrates the structure of the Housekeeping control layer as it was conceived for its implementation. Elements with white background are related to the unattended housekeeping control. Red boxes correspond to the components of the hardware layer and the applications developed to run some data-flow services, whereas the blue one and the red arrows are designed for manual control or remote monitoring.

5.2.1.1 Housekeeping actors

The Housekeeping control layer is mainly devoted to perform actions to ensure the system health at any time during the unattended operation. That is, under the robotic operator (or ROCS). Human interaction with this layer is considered for a small fraction of the time when maintenance and testing tasks are performed or for remote supervision of the system. The latter is just provided to the TJO-OAdM staff (superuser operator).

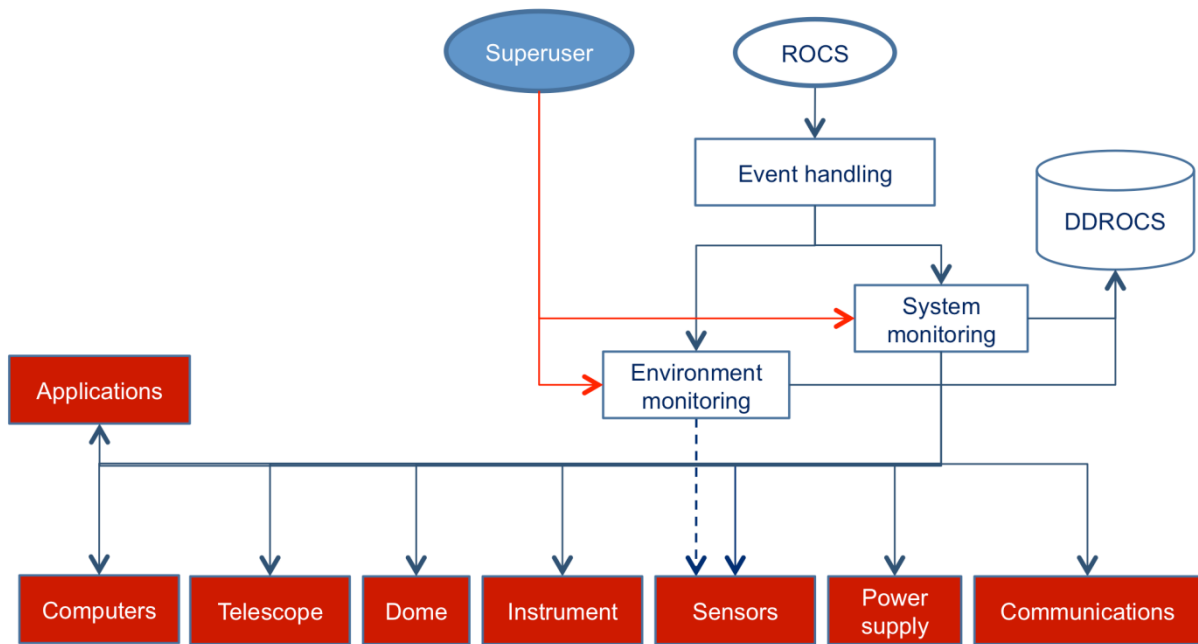


Fig. 5.3: Housekeeping control layer design

5.2.1.2 Housekeeping modules

This control layer is composed by two different modules:

➤ **Environment monitoring**

Environmental data have to be collected and distributed from a variety of heterogeneous sensors in a transparent manner and the data have to be provided in a homogeneous way, thus allowing the system, and other potential clients, to have a simple and standard way of accessing these data. This functionality is one of the core modules of the Housekeeping software layer.

Structure

The environment monitoring has to be implemented by reading the data collected from the sensors and publishing them to be accessible to the clients connected over the network. Multiple sensors are required to provide at least one value for each environment variable. Redundancy is mandatory for the most critical ones (i.e., humidity). Fig. 5.4 (Francisco et al. 2009) illustrates this structure design, where multiple driver interfaces are used to integrate the sensor data into a common data array.

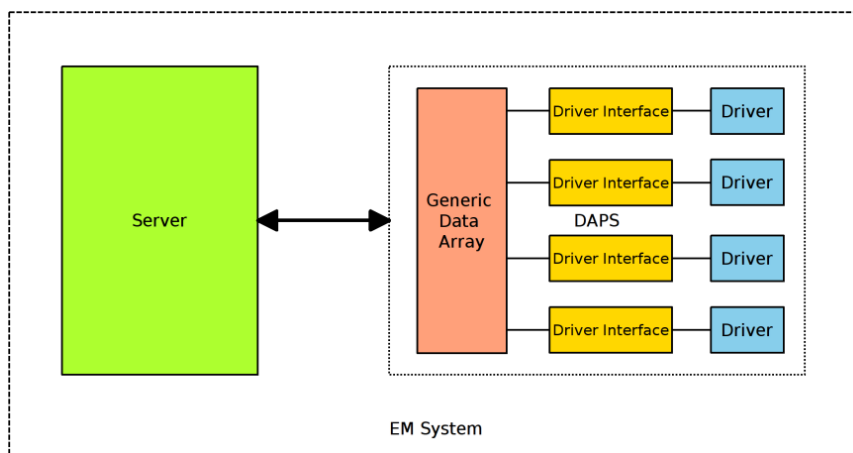


Fig. 5.4: Environment monitoring structure (Francisco et al. 2009)

Clients

There are two main clients for the environment monitoring module: OpenROCS and the user interface for remote supervision. Environment data must be provided to both clients with the required cadence.

Environment monitoring was also designed to store the collected data into the database (see Chapter 7 for more details).

Sensor redundancy

Some weather variables are critical for the system security: rain and storm detection, humidity and wind speed. All these variables are measured with redundant sensors, except for the rain detection, as already described in the Physical Architecture Implementation (Chapter 4). A new weather station equipped with a rain detector was installed at the observatory site in early 2011 by the TFRM project and the data was made accessible to the TJO-OAdM, although its integration has not been programmed yet. Once it is integrated, the TJO-OAdM will have three sensors for humidity and wind speed measurement and two rain detectors, providing the required redundancy.

Redundancy of sensors reduces the risk of the unattended operation and enables the control system to perform two important actions:

- **Check the quality** of the measured data and detect any trend due to a degradation sensor calibration. This feature requires the execution of a comparison of the weather data in real time: the comparison of humidity and wind speed is based on statistical considerations. The difference between the values of the various data points is compared with the standard deviation, which value has to be into an accepted range. Rain detection has to prioritize any positive detection and to activate a shutting down action in case a conflicting measurement persists.
- **Continue operating** the system when a sensor does not work. This would represent running the system with no increased risk, in case two sensors were still alive, or with high risk, if just one sensor was available. The high risk operation could be assumed to perform high priority tasks during a short time interval, although it is not considered in the current implementation.

➤ **System monitoring**

The Housekeeping control layer also includes routines to interface with the Hardware layer components and with the software applications that implement the system services. These routines are required to monitor the system health. The collection of data and the execution of commands acting on the different layer components must be doable in a homogeneous way. This control layer must be programmed in a modular way to easily change the Hardware layer configuration or the software applications at any time.

Structure

The components of the Hardware layer are (see Chapter 4 for more details):

- **Telescope:** telescope axis (HA, DEC) and focus, mirror covers, filter wheel and electronics.
- **Dome:** nominal and redundant control.
- **MEIA Instrument:** CCD camera.
- **Power supply:** public grid, support generator, UPSs and PDUs.
- **Sensors:** weather stations, rain detector, storm detector, cloud sensor.
- **Communications:** internet connection and internal LAN.
- **Subsystem computers**

All this equipment has the capability to be controlled automatically.

The applications monitored by this module are those used to run the following services:

- **Task preparation:** ToO, for unattended task injection; database for the storage of proposal data (definition and obtained).
- **Task scheduling:** ISROCS, for task selection and night scheduling.
- **Task execution:** TALON, for telescope, dome and instrument (Camera routine) control; and routine for dome redundant control.
- **Task processing:** ICAT, for image calibration, analysis, storage and quality control.

These applications are described in the following chapters.

The Housekeeping layer should be programmed to communicate with the Hardware layer and the software applications with a common communication protocol.

Clients

The system monitor has the same clients as the environment monitor. And it was also designed to store the collected data into the database.

Redundancy for critical hardware

Highly reliable hardware is chosen to perform all the tasks at the observatory, but paliative operations are designed to restore or protect the system in case of failure of any component. Special attention was put on the dome shutter control, which was identified as a single point of failure of the system.

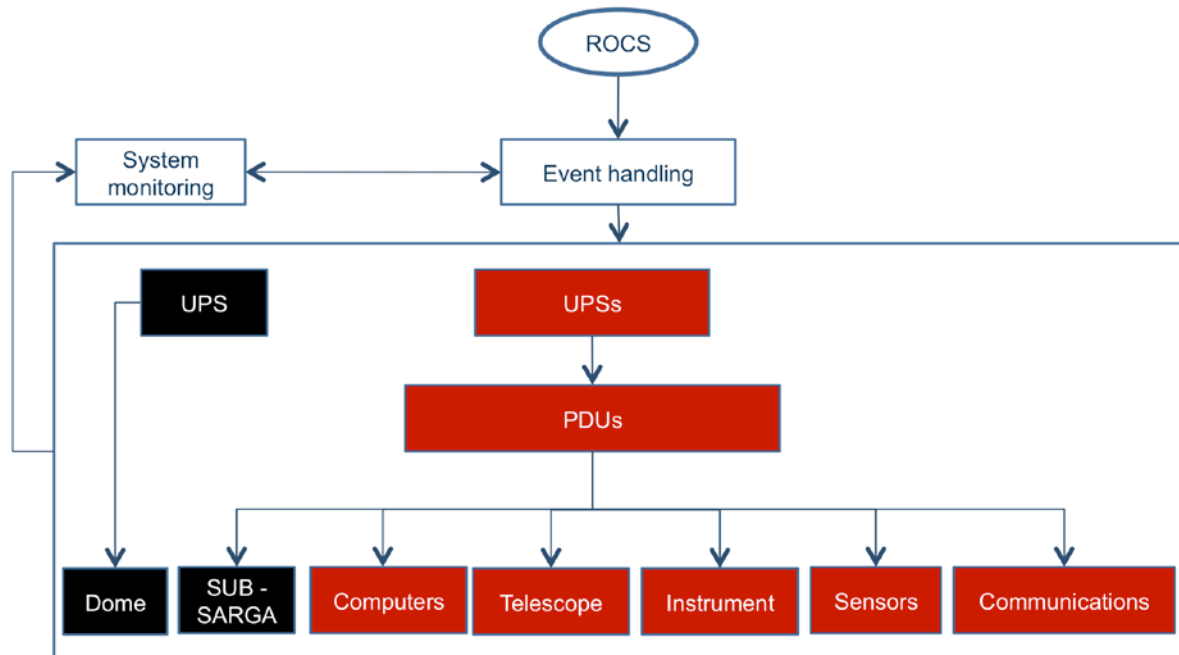


Fig. 5.5: Soft (red) and hard (black) redundancy levels

Two levels of redundancy (Fig. 5.5) were defined: a soft level, designed to react to most of the hardware events foreseen at the TJO-OAdM; and a hard level, for the robustness of the dome shutter control. Both levels require the monitoring and control capability, implemented using the devices detailed before and grouped as follows:

- **Soft redundancy level:** Hardware included here is mainly devoted to restore the components with faulty behaviour or to shut them down if there is no possibility of recovery.
 - **PDU**s are used to enable the reboot of almost all the critical hardware for some specific events and alarms. Computers, telescope power and electronics, instrument, sensors and communication system are powered through the outlets of the PDU. The Housekeeping layer is responsible to trigger the associated actions to restore every particular device executing a hard reset by rebooting the power line if necessary. In that sense, hardware components have to be monitored and PDU must be controllable remotely by the Housekeeping layer.
 - **UPS**s (UPS-CPU and UPS-Telescope) are used to enable a soft shutdown of the system in case of a long power outage or to keep the system running when a short outage is produced. A short power outage is not covered by the observatory support power engine, due to its warming up time; and, therefore, this support system is required. A remote monitoring of the UPS status is also required.
- **Hard redundancy level:** Hardware included here is devoted to close the dome shutter when required and under any circumstance. All the control path for this redundancy level is focused on this critical component.
 - **Dedicated UPS:** A dedicated, overestimated and remotely controlled UPS (UPS-Dome) is used to power the dome shutter motors under any circumstance.
 - **Dedicated redundant computer:** A robust computer is used to perform a redundant control of the shutter motors. Both nominal (SUB-ESTALL) and redundant (SUB-SARGA) control computers have to be continuously monitored by the Housekeeping layer that executes a dome shutter closing action when faulty behaviour is detected.
 - **Dedicated electronics:** A robust electronic board (FSC1) was developed to enable the redundant control based on signal prioritization. Control signals come from: the nominal and the redundant control computers; the rain detector, a sensor signal hardwired to the electronic board; and, finally, from the firmware at the electronic

board itself that executes a dome shutter close action when no signal is received from the input channels (whatchdog functionality).

These two redundancy levels ensure the system reliability and robustness.

5.2.2 Housekeeping development

According to the Housekeeping design presented, this control layer is composed of hardware and software components whose main functionality is to monitor the behaviour of all the elements at the observatory and recover the system in case of any malfunctioning.

The Housekeeping layer components are summarized here:

- Environment monitoring
 - **Environment monitor**: This application is used to collect data and serve them to any subsystem, providing a transparent communication interface. It is a scalable module in case the number of sensor changes.
 - **Data evaluation monitor** is used to determine environment events due to bad weather conditions or any malfunctioning sensor.
- System monitoring
 - **Monitor of the Hardware layer**: This application is used to detect any faulty behaviour of the hardware components.
 - **Monitor of all the software applications** required for data flow processes: task preparation, task scheduling, task execution and task processing.
 - Support hardware used to keep the system running or to execute a soft shutdown in case of power outage: UPSs. And hardware used to execute a global system shutdown or a restart routine at some specific subsystem or device: PDUs.
 - Dome redundant control software and hardware: FSC1, SUB-SARGA, control software and power supply (UPS-Dome).

All the monitoring functionalities listed here (highlighted in bold and underlined text) were finally integrated into the OpenROCS software infrastructure. This approach improved the control system development by joining the system monitoring and the generation of events to solve, palliate or avoid any undesired situation. The latter is managed by a set of well defined event-action pairs included in the OpenROCS software and thanks to a clear definition of interfaces between these two frames (monitoring and event-action pairs).

5.3 End-to-end data flow

The End-to-end data flow control layer was defined at the System Architectural design (Chapter 3) as the software layer in charge of executing all the data acquisition routine from user proposal submission to user data retrieval and including data processing feedback for the optimization of the operation scheduling. We designed and implemented the general data flow scheme, the different processes involved, and the way they interact or transfer data among them to maximize the system efficiency.

A fundamental tool to build this layer is the database. All the information associated to each task entered into the system for its execution is stored in the observatory database. And it is, then, accessed by the software services in charge of managing the task selection, execution and processing, which also store the processing results back into the database to provide feedback to the system. Finally, it is used to easily retrieve the data by the end user or by the observatory staff supervising the operations. The database is described in Chapter 7.

We give in this section a description of the end-to-end data flow concept. Each one of the processes and applications involved in the data flow scheme is widely described in next chapters.

5.3.1 End-to-end data flow design

The end-to-end data flow for the TJO-OAdM is illustrated in the Fig. 5.6. White boxes correspond to unattended tasks and blue ones are those which require human interaction. Those boxes with a gradient background correspond to tasks that can be executed under both control modes.

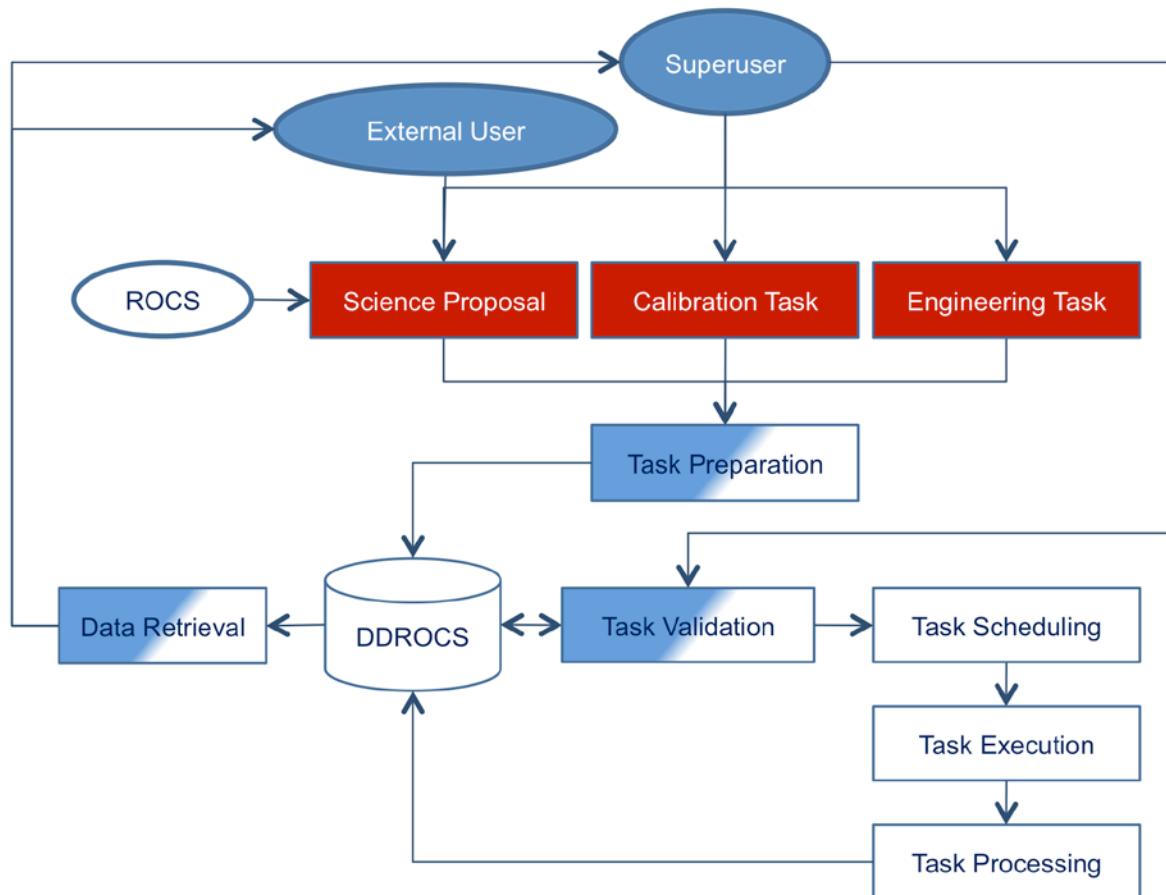


Fig. 5.6: End-to-end data flow design

Three different components integrate this scheme: the actors interacting with the system, the tasks to be performed and the system database.

5.3.1.1 Data flow actors

We define three main actors to interact with the system at different points of the data flow:

- **External user:** This actor submits proposals to the observatory to get data for a specific science case. Interaction with the system is produced for proposal submission and data retrieval.
- **TJO-OAdM staff (or superuser):** The observatory staff members are responsible for introducing the basic set of calibration tasks executed by default in the routine operations. These calibration images are a product to deliver to the external user. They are also in charge of validating the tasks included in each science proposal to confirm that their execution constraints fulfill the approved conditions. Finally, they introduce the engineering tasks for maintenance and testing operations.
- **ROCS:** User managing the unattended data flow processes. Some tasks may be automatically introduced into the system by this robotic operator: observation of transient phenomena enter into this category.

The definition of the actors is important as it clarifies the design of interfaces and the information required by the system to specify the execution constraints for each task. Final purpose is to minimize human interaction with the system and to maximize the homogeneity of task preparation and submission processes. The latter provides a set-up to use the scheduler task in most of the scenarios, not only for the nominal science and calibration operations but also for the less frequent maintenance and testing tasks. And this is an important issue as system efficiency is strongly linked with the use of the scheduler.

Human interaction capability is foreseen at any stage of control, although the data flow scheme is focused on a fully unattended operation from task submission to data retrieval under the control of a robotic operator. A system supervision or an attended operation mode under the control of a superuser operator are compatible with the same definition of tasks used for robotic operation, as

illustrated in Fig. 5.7 (red arrows indicating possible alternative control or data flow paths). Interfaces and communication protocols are defined accordingly (see Chapter 12 for a description of the observatory user interfaces).

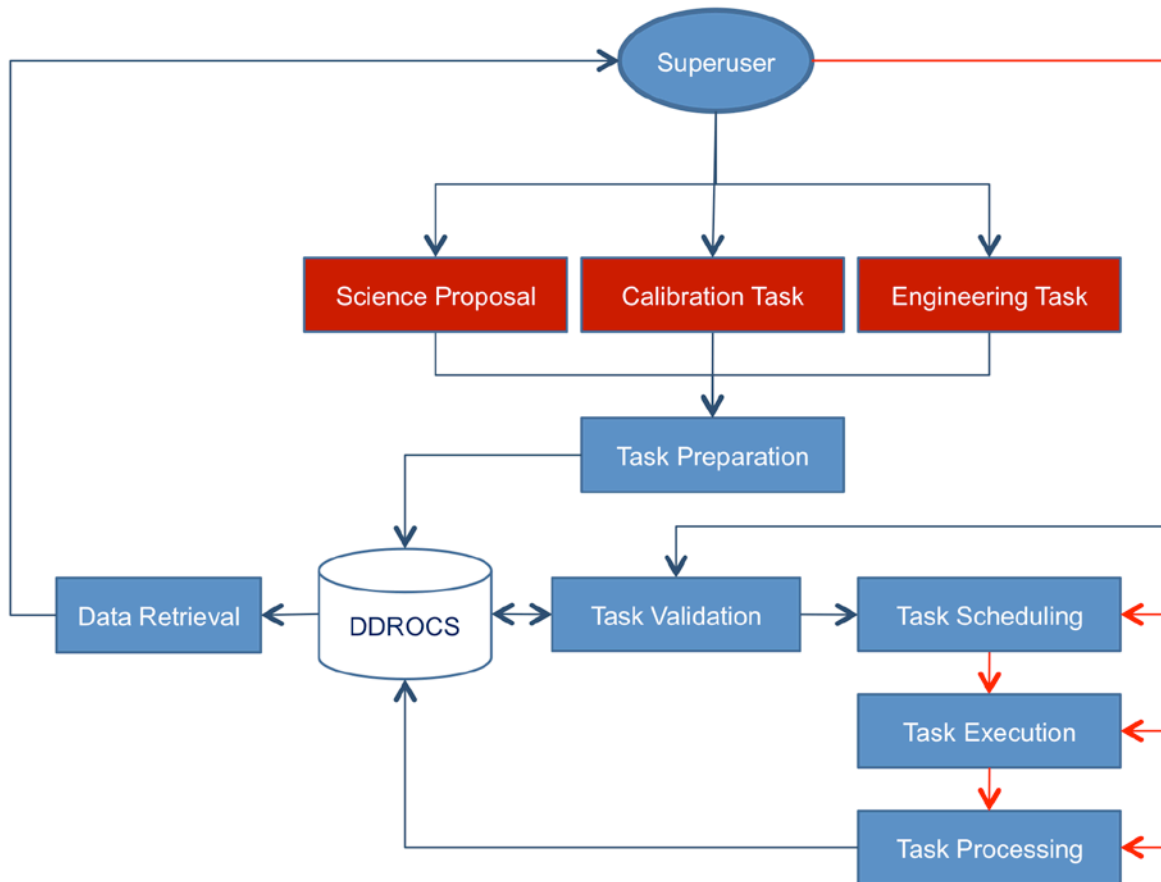


Fig. 5.7: Data flow for human supervision or attended control mode

5.3.1.2 Data flow tasks

Five tasks (or modules) are involved in the end-to-end data flow. We summarize them here, indicating the level of human attendance when the system is operated under robotic mode.

➤ **Task preparation and validation**

Task submission

The definition of the execution constraints for any specific task is done by the user submitting the task. External users submit their science proposals in two phases: first one for general information to be evaluated by the observatory TAC and second one for specific execution constraints that have to be validated by the TJO-OAdM staff. If necessary, the superuser operator can introduce new tasks directly by specifying the task execution constraints included at the first and second phases of submission without the TAC evaluation.

A fully automatic injection of tasks related with transient phenomena is foreseen, but the majority of processes for task submission are done with a relatively high level of human interaction.

Database storage

Tasks are stored in the database at the time of submission. This process is done automatically because the mentioned actors use a graphical tool that directly interfaces with the database. This interface also performs a validation of the submitted data before storing them. It is not recommended for the superuser, and not allowed to external users, to introduce the task parameters directly into the database to avoid any possible error.

Validation

An automatic validation is initially performed by the graphical interface by doing a rough check of the parameters (i.e., variable format). But, the majority of the task parameters must be validated by staff

members through a manual operation. Tools to execute a quick check and validation of the data are useful to optimize the process. The scheduler, for instance, is designed to have a simulation mode that can be used to validate the time scheduling constraints defined by the user and compare with those included in the approved proposal.

Database mirroring

Database is stored at the off-site server (SUB-AGER) and it is automatically mirrored at the observatory server (SUB-ALIS). External users submit the execution data to the off-site server, whereas the superuser is able to introduce new tasks at SUB-ALIS directly during attended operations, for instance, during the commissioning phase. This possibility is managed with the databases mirroring configuration described below.

➤ **Task scheduling**

Selection of targets is done automatically and in two steps. A mid-term scheduler computes an ordered list of tasks for the whole night. During the night, a short-term scheduler, based on a dispatcher, does a real time selection that can slightly modify the static planning. The scheduler (called ISROCS, Chapter 9) uses only those tasks stored in the database that have already been validated.

➤ **Task execution**

The execution of tasks is done automatically when nominal operations are performed. Some maintenance or test operations may also be done automatically. Processes involved in the execution of tasks are: observatory start up and power down routines; hardware layer control for telescope, instrumentation and dome to perform the task execution.

➤ **Task processing**

Data processing routines are in charge of image reduction and analysis, executing a quality control validation, archiving the data and the images, updating the database and the status of the pending tasks and, finally, ensuring a secure backup policy. All these processes are done automatically at the TJO-OAdM with the ICAT package.

➤ **Data retrieval**

Data extracted from the images obtained are stored in the database and in the archive in a way they can be cross-identified with the original proposal. Therefore, following the status of the tasks and data retrieval is done in a straightforward manner using the database interface and the proposal identifiers. This might be defined, then, as a semi-manual process.

5.3.1.3 Data flow and the observatory database

Those processes involved in the system control (Scheduling, Execution and Processing) interact with the observatory database (Chapter 7) to extract the relevant data for their execution.

Other processes (Preparation, Validation and Retrieval) interface also with the observatory database that contains the information required for the applications involved. The web-based general user interface is designed with this purpose (see Chapter 12).

5.3.2 End-to-end data flow development

According to the design presented, this control layer is integrated by hardware and software components whose main functionality is to manage and execute all the tasks involved in the work flow of proposals.

We summarize here the End-to-end data flow layer components:

- Task preparation: It involves the **user interface** or the automatic routine (for **ToO** tasks only) to enter the task execution definition into the system.
- Task scheduling: It carries out the task selection procedure to optimize the execution of science programs at the TJO-OAdM. **ISROCS** is the application in charge of this functionality.
- Task execution: The hardware devices (telescope, dome, instrument) and the software for their configuration and control to execute the tasks are included in this component. **TALON** and its routine **Camera** to control the MEIA CCD are used with that purpose.
- Task processing: **ICAT** is the component that processes every image acquired and extracts a quality flag. Storage and archiving is also included in the list of features of this application.

- **Database:** This is a fundamental component to track the observatory history and to improve the system efficiency. It is also used to retrieve the data to the end user.

All the applications (highlighted in bold and underlined text) are modules of the data flow layer. They all were developed in the frame of the TJO-OAdM robotization project and this Thesis, except TALON (and Camera), and they are extendedly described in other chapters of this work.

Data flow execution is managed by a set of well defined event-action pairs included in the OpenROCS software and thanks to a clear definition of interfaces among this software and the mentioned applications.

5.4 OpenROCS

A major step for the robotic control system design was to establish the software that could handle the events and alarms of the system. Most of the actions that execute the data flow or housekeeping control applications are triggered by well defined events and alarms. Therefore, a direct consequence of this assumption is that this software, named OpenROCS, acts as the primary actor for the overall observatory control. The software infrastructure takes this role on a distributed control system, and it is also a software framework to develop the set of applications in a homogeneous way. Three major steps were identified then as fundamental for the OpenROCS design:

- **Interfaces and applications:** It includes the specification of interfaces between OpenROCS and the software control layers. Development of any software application is done accordingly.
- **Events/alarms and actions:** It includes the clear definition of the full set of events, alarms and actions for any possible control operation at the observatory.
- **Software infrastructure:** It involves the development of a suitable software infrastructure.

These three design levels are presented in this section. A general list of the requirements for the software infrastructure and a discussion about the system states are given before.

5.4.1 General requirements

The general requirements for the software infrastructure of the TJO-OAdM are:

- **Automatic alarm generation and management:** The system has to be prepared to trigger a set of actions when an alarm or event occurs and to check if these actions are executed correctly and reacted on accordingly. Non-routine events associated to housekeeping monitoring modules and routine events to manage the telescope operation sequences and the end-to-end data flow processes have to be considered.
- **Reliability, duplication and distribution:** The system has to be based on reliable technology and be able to work even in dire conditions. For the observatory being fully autonomous, the housekeeping system has to be the first one to start up and the last one to shutdown to check all the other subsystems in the observatory. In that sense, the system must have the capability to be replicated and distributed among different computers. It has to be able to transparently restart a service if it fails, or at least notify the failure to take action.
- **Software infrastructure:** The system has to act as an infrastructure to implement the control applications and services in a homogeneous way.
- **Log generation and collection:** A logging system has to be deployed to trace any action or data related to system control. It has to be replicated to allow failover logging.

5.4.2 System states

The control system must have the capacity to determine the current state and change it according to the actions triggered by the events produced by the housekeeping or the end-to-end data flow control.

5.4.2.1 General system states

A higher level state definition is considered for the development of OpenROCS to let the Superuser define the operation mode at the observatory (Fig. 5.8):

- **Stop / Shutdown:** Subsystem computers, the control system and the hardware and software for the environment monitoring are running or they are completely stopped.
- **Robotic:** The observatory is operated in a completely unattended fashion. It is the nominal control mode.

- Engineering:** The user interacts with the system to perform the required operations. OpenROCS and all the subsystem computers are running. He can use OpenROCS to let the system configure and control the work flow execution, but having the capacity to add or remove events at any time.

We took these high level states into consideration during the OpenROCS design and when configuring the user interface. See next chapter for a detailed description.

A sub-set of states are enabled when running the system under unattended control, as described below.

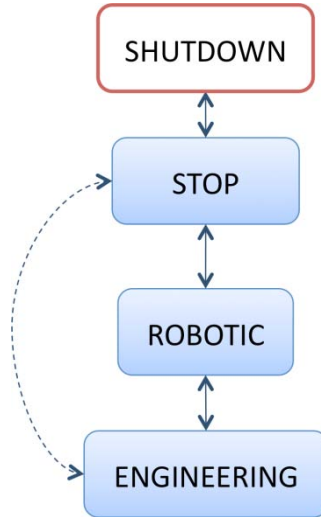


Fig. 5.8: General states of the control system

5.4.2.2 States for the nominal control

We defined different states at the observatory for the robotic mode of operation: Stop, Calib, Init, Flat, Thermal, Science, Park, and Shutdown. We already described them when we introduced the nominal operation of the system (Sect. 3.1). These states are mainly defined to handle the task execution process based on event-action pairs managed by the software infrastructure.

The hardware and software status at each state are given in Table 5.1.

State	Stop	Calib	Init	Flat	Thermal	Science	Park	Shutdown
Telescope								
Power (motors, electronics)	Off	On	On	On	On	On	On	Off
Electronics (CSIMC)	Off	Ready	Ready	Ready	Ready	Ready	Ready	Off
HA, DEC and Focus	Off	On	Home	Home & Pointing	Home	Home & Tracking	Home & Stow	Off
Filter wheel	Off	On	Home	At position	Open	At position	Home & Stow	Off
Mirror covers	Off	On	Open	Open	Open	Open	Closed	Off
Control software	Off	On	On	On	On	On	On	Off

State	Stop	Calib	Init	Flat	Thermal	Science	Park	Shutdown	
Dome									
Power (electronics)	Off	Off	On	On	On	On	On	Off	
Dome status	Off	Off	Home & Closed	Home & Closed & Pointing	Home & Open	Home & Open & Tracking	Home & Closed & Stow	Off	
Control software	Off	Off	On	On	On	On	On	Off	
Instrument									
Power (instrument)	Off	On	On	On	On	On	On	Off	
Instrument status	Off	Ready	Ready	Ready	Ready	Ready	Ready	Off	
Control software	Off	On	On	On	On	On	On	Off	
Flats lamp	Off	Off	Off	On	Off	Off	Off	Off	
Control system									
Data flow SW	ToO	On/Off	On/Off	On/Off	On/Off	On/Off	On/Off	On/Off	Off
	DB	On	On	On	On	On	On	On	Off
	ISROCS	On	On	On	On	On	On	On	Off
	ICAT	On/Off	On/Off	On/Off	On/Off	On/Off	On/Off	On/Off	Off
	TALON	Off	Off	On	On	On	On	On	Off
	Camera	Off	On	On	On	On	On	On	Off
HK SW & HW	UPSs & PDUs								
	Sensors & Communications								
			Dome redund.	Dome redund.	Dome redund.	Dome redund.	Dome redund.		
	Environment & System monitors								Env.Monitor
OpenROCS	On	On	On	On	On	On	On	On	
SUB-ESTALL	On	On	On	On	On	On	On	Off	
SUB-REBEI (SUB-RUBIES)	On	On	On	On	On	On	On	On	
SUB-SARGA	On	On	On	On	On	On	On	Off	
SUB-ALIS	On	On	On	On	On	On	On	Off	
SUB-AGER	On	On	On	On	On	On	On	On	
Environment									
Indoor conditions		Ok	Ok	Ok	Ok	Ok	Ok		
Outdoor conditions					Ok	Ok			

Table 5.1: HW and SW status for each state

We defined additional states to specify the unattended operations that the system performs out of the night nominal workflow. These are for task preparation, task scheduling and task processing. All the involved processes run on SUB-ALIS. These states are:

- **ToO**: System performs the actions to inject a new and unexpected science observation task into the system.
- **ISROCSStatic**: The mid-term schedule is computed during this state.
- **ICATEnd**: Quality control is executed off-line at the end of the night for all the images acquired.

5.4.2.3 Housekeeping state diagrams

State transitions produced by actions designed to ensure the system security are shown in Fig. 5.9. The control system must check the current state machine any time a transition between two states is triggered and, then, execute the sequence to move to the new state.

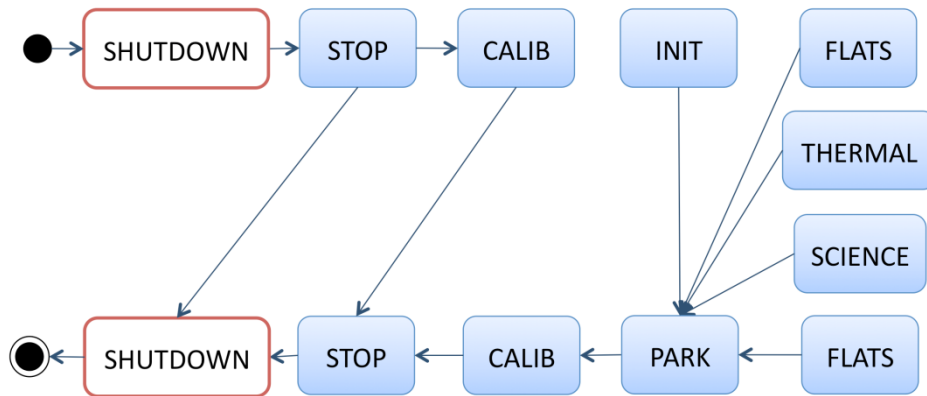


Fig. 5.9: Housekeeping state transitions

5.4.2.4 End-to-end data flow state diagrams

State transitions produced by actions designed to control the data flow are shown in Fig. 5.10. Straight lines indicate the standard nominal work flow; whereas dashed arrows indicate alternative state transitions that can happen depending on the scheduler task selection.

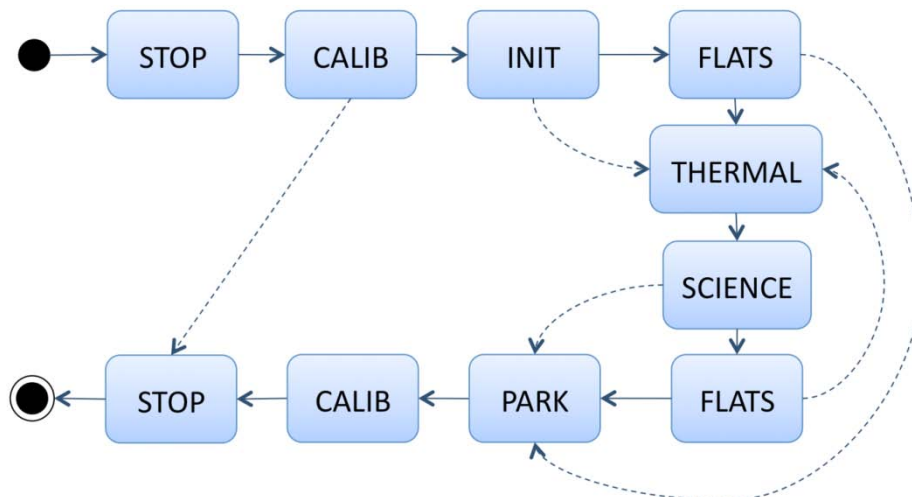


Fig. 5.10: Data-flow state transitions

Processes that must be executed to move the system set-up to specific state conditions can be derived from the transitions shown in this figure. The following transitions must be executed depending on the type of task selected by the scheduler (origin state can be different if the system is not on Stop when starting the transition, as it happens for the standard work flow):

- Bias or Dark calibration task: Stop → Calib
- Flat calibration task: Stop → Calib → Init → Flats
- Science observation: Stop → Calib → Init → Thermal → Science

Once the system is initialized, it must be on Park state before it can be stopped.

5.4.2.5 State transition prioritization

System robustness and safety always has a higher priority to ensure its integrity and, therefore, actions triggered for housekeeping control purposes and devoted to change the system state are done with higher priority.

5.4.3 Interfaces with housekeeping and end-to-end data flow control

The exchange of information to OpenROCS from every subsystem executing a process is the base for this software infrastructure to create events or alarms that will immediately (or at a specific time delay) trigger actions to manage the housekeeping and data flow control. Actions can be executed at the same or at different subsystems. It is, then, very important to have a clear definition of the interfaces.

5.4.3.1 Housekeeping interfaces with OpenROCS

We define the interfaces between OpenROCS and the environment and system monitoring modules in such a way that a highly reliable and robust housekeeping control is achieved by the control system. A flux of information and a list of commands to execute the required actions are, then, specified for every application in order to transfer the housekeeping control to OpenROCS. The latter is summarized in the following sections.

Interfacing with the observatory database was also foreseen for environment data storage and hardware status logging. All these data are used to remotely supervise the system behaviour and to detect any performance diminishing with time.

➤ *Environment monitoring module interface*

The data for the environment monitor interfaces are all specified in Table 5.2. Event and alarm handling involved in this module is also specified.

Data	
Item	Values
OpenROCS – Environment data	Environment monitoring data are transferred to OpenROCS
Sensors	<p>Data taken from weather stations (multiple sensors for redundancy on some variables) and the GPS antenna is read by the environment monitoring:</p> <ul style="list-style-type: none"> • Air temperature (K): EM-Davis (indoor – Dome – and outdoor) and EM-SMC (outdoor) • Pressure (hPa): EM-Davis, EM-SMC • Relative humidity (%): EM-Davis (indoor – Dome – and outdoor) and EM-SMC (outdoor) • Wind direction (deg) and speed (m/s): EM-Davis, EM-SMC • Accumulated rain (mm): EM-Davis, EM-SMC • Solar radiation (w/m²): EM-SMC • Rain detection (yes/no): IRSS88 • Storm detection (V/m): Previstorm • Sky cloud coverage (%): Cloud Sensor • GPS signal for time base determination <p>Each value is accompanied by the sensor identifier and the timestamp.</p>
Commands	
addEvent	Add an event to trigger the associated actions.
removeEvent	Remove an event.
getCurrent	Get data from the environment monitoring module.
setData	Publish environment data to be used by the execution task modules.
evalData	Evaluate if data values are in the accepted range.
validateData	Validate data from redundant sensors by comparing the read values.
getState	Determines the current system state.

setState	Set the system state, depending on the data values evaluation and validation. One or multiple state transitions can be required to reach the goal state. The latter is done with a generic command: chgStart2End.
chgStart2End	Execute a state transition between current and goal states. System configuration is changed at every transition by using the setTask commands. Transition enabled sequences were defined before.

Table 5.2: Interface definition for environment module

➤ **System monitoring module interface**

The data for the environment monitor interfaces are all specified in Table 5.3. Event and alarm handling involved in this module is also specified.

Data	
Item	Values
OpenROCS – System data	System monitoring data is transferred to OpenROCS
System data	<p>Data of the hardware components to be read by the hardware monitoring:</p> <ul style="list-style-type: none"> • Telescope: encoder/position values and status for telescope axis (HA, DEC), focus, and mirror covers), communication with the device (CSIMC nodes), and low level software (<i>telescope</i> daemon). • Dome: encoder values, shutter status (opened, opening, closing, closed), communication with the device (CSIMC node 0 and FSC1 electronics) and the redundant control subsystem (low level software) • MEIA instrument: CCD working parameters (temperature, shutter status, etc.), communication with the device (USB connection), and low level software (<i>camera</i> daemon), filter wheel status and position, flat lamp status. • Sensors: low level software (drivers), and status of the communication channels. • Power supply: status of the UPSs and their batteries, and the PDUs, communication with the devices. • Communication: status of the internet (connection with SUB-AGER), the intranet lines and the router-LAN (connection with the router). • Subsystem computers: status of the control computers (SUB-ESTALL, SUB-SARGA, SUB-REBEI, SUB-RUBIES, SUB-ALIS, SUB-AGER). <p>Status of the software applications:</p> <ul style="list-style-type: none"> • TALON and dome redundant control • Database • Scheduler (ISROCS) • Image processing (ICAT)
Commands	
addEvent	Add an event to trigger the associated actions.
removeEvent	Remove an event.
getCurrent	Get data from the system monitoring module.
setData	Publish data to be used by other modules.
evalData	Evaluate if data values are in the accepted range.
getState	Determines the current system state.
setState	Set the system state, depending on the data values evaluation and validation. One or multiple state transitions can be required to reach the goal state. The latter is done with a generic command: chgStart2End.
chgStart2End	Execute a state transition between current and goal states. System configuration is changed at every transition by using the setTask commands. Transition enabled sequences were defined before.

Table 5.3: Interface definition for hardware module

5.4.3.2 End-to-end data flow interfaces with OpenROCS

Interfaces between OpenROCS and the data flow modules have to be defined in such a way that a highly reliable and efficient execution is achieved without extra time overheads. We specified, then, a flux of information and a list of commands to execute the required actions for every application in order to transfer the data flow control to OpenROCS. The latter, also done before for the housekeeping control, is summarized in the following sections.

Those processes involved in the system unattended control (Scheduling, Execution and Processing) interact with the OpenROCS system following its interface definition. All the software packages executing these processes have to be managed by this software infrastructure to optimize the system efficiency.

The initial implementation of the data flow control was not done using OpenROCS interface definition, but using a call of every particular application by the preceding one in the flow of control process. I.e., the task execution routine (TALON) calls the data processing applications just after the acquisition of the image. Work to move the control of every task to the OpenROCS software is underway and it is based on the interfaces defined here.

➤ **Task preparation and validation interface**

Task preparation and validation is a manual process that does not interface with the robotic control system. Storage of task data in the database and the database mirroring are automatic processes, but they are performed without any interaction with OpenROCS.

Transient phenomena

The observation of Targets of Opportunity (ToO) is the only process foreseen to introduce a new proposal with no human interaction. This kind of science observation, included in the science case of the observatory, requires an interface definition to communicate with the control system. The automatic sequence to be performed in that case would be: receive the alert, validate its feasibility with the TJO-OAdM instrumentation, configure the execution constraints and insert the task in the database and in the static night schedule computed by the mid-term scheduler (see Chapter 9 for more details about the scheduler). This automatic process should be done by a specific service, not implemented at the time of writing this document, whose execution should be triggered by event-action pairs of OpenROCS. This service should, then, interface with OpenROCS to exchange information and commands with a sequence as follows:

- OpenROCS generates an event when the alert is received, triggering the action to execute the service sequence.
- Once the first action is triggered, the service reports information periodically to OpenROCS about its status. Service states are: service is running, ToO received, ToO under evaluation, ToO configuration, ToO injection.
- The service reports to OpenROCS with the task execution constraints derived from the ToO evaluation. It includes the task feasibility and a science priority.
- OpenROCS generates an event when the service is at ToO injection status and the action to inject the execution constraints to the database is triggered. The same event triggers the scheduling task to calculate the maximum value of the Figure of Merit for that task as a function of time. It then inserts the task in the corresponding time-slot of the static night schedule previously computed by the mid-term scheduler.
- In case the best time for the task execution is before the end of the ongoing task, OpenROCS triggers the execution of the short-term scheduler that selects the one with higher priority. If the ToO is ranked first, OpenROCS creates an action to make the task execution service abort the current task and to start executing the ToO task with the constraints determined before.
- In case the ToO execution is planned for a later time-slot, no more actions are triggered as ToO will be considered by the nominal night execution.

Table 5.4 summarizes the data the ToO service should send to / receive from OpenROCS and the commands OpenROCS defines.

Data	
Item	Values
Alert data received	Data describing the ToO received from the outside world.

ToO service status	Running, ToO received, ToO under evaluation, ToO configuration, ToO injection, ToO rejection, Stopped.
ToO execution constraints	Science observation constraints.
Commands	
addEvent	Add an event to trigger the associated actions.
removeEvent	Remove an event.
getAlert	Get data from the alert system (outside world).
getStatus	Evaluate the service status.
setStatus	Publish service status to notify it to other modules.
evalData	Evaluate the received alert data to check its feasibility at the TJO-OAdM.
rejectAlert	Reject the alert task.
getMeritMax	Send query to the mid-term scheduler to compute maximum value for the FOM at the current night.
setAlert	Insert the execution constraints into the observatory database and into the mid-term planning list.
startISROCS	Send query to the short-term scheduler to select the task with the highest priority in order to check if the alert task has to override the current one.
abortTask	Send query to the task execution module to abort current task (use abortImage command) and execute the alert task.

Table 5.4: Interface definition for ToO service

➤ **Task scheduling interface**

The scheduler (see Chapter 9 for more details) is integrated in the observatory control system as a service module. Mid and short-term scheduling execution is managed by the event monitoring and action execution functionalities of OpenROCS. The scheduler routine is in charge of providing OpenROCS with the most suitable task to be executed by the TJO-OAdM at any time.

Scheduler and OpenROCS interfacing is done by defining the information and commands that must be sent from one to the other. OpenROCS events trigger the execution of the mid- and short-term planning for a full night or a dynamic task selection and sorting, respectively. Some information must be sent from the scheduler to the OpenROCS, and viceversa, and some commands must be executed by the scheduler as actions triggered by system events. An interface is also required between the observatory database and the scheduler to provide the scheduler with all the pending task data.

The data and commands for the scheduler interfaces are all specified in Table 5.5.

Data	
Item	Values
Database	Task execution constraints taken from the database for all the pending and validated tasks.
Mid-term list	Ordered list of tasks for a complete night. It is used as input data for short-term scheduling.
OpenROCS – Housekeeping data	Environment data (weather and atmospheric variables) and system health (readiness of all software and hardware).
Short-term selected task	Task selected and its execution constraints.
Commands	
addEvent	Add an event to trigger the associated actions.
removeEvent	Remove an event.
getDB	Get data from the pending tasks stored at the database.
getMeritMax	Execute the mid-term scheduler to compute the maximum FOM value for the pending list of tasks.

sortMeritMax	Select the tasks feasible for the following night and sort them by their maximum FOM value (mid-term planning list).
getHK	Get data from the housekeeping control modules.
getTask	Execute short-term scheduler over the sorted list of selected tasks (mid-term planning list) for the current night and get the selected task.
setTask	Publish the selected task and its execution constraints.

Table 5.5: Interface definition for scheduler service

➤ *Task execution interface*

The execution task controls the hardware layer components to perform the observatory operations involved in data acquisition. OpenROCS manages the process by triggering actions to specific events and according to the system state, the task selected by the scheduler and the housekeeping data.

The data and commands for the execution task interfaces are all specified in Table 5.6.

Data	
Item	Values
OpenROCS – Mid-term list of tasks	List of tasks for a full night. Starting time for the routine operations is taken from this list: time to change the system state from Stop to Initialization.
OpenROCS – Task execution parameters	Task identifiers and task execution constraints to specify the execution details (telescope, dome and instrument configuration).
OpenROCS – Environment data	Weather data that are necessary to tune the execution system (i.e., environment temperature for the focusing system).
Execution results	Acquisition data to be stored in the image header and execution data to be stored in the logging system (also transferred to the control system for housekeeping purposes, as described in Section 5.2).
Commands	
addEvent	Add an event to trigger the associated actions.
removeEvent	Remove an event.
getMT	Get data from the mid-term list of tasks.
setInit	Determines the time to start night operations with the state transition from Stop to Init. Time is determined from the mid-term list of tasks.
getTask	Send a query to the short-term scheduler and get the task selected and all the parameters defining its execution constraints.
getHK	Get data from the housekeeping control modules.
setTask	Send the task execution parameters to the execution task control software. Different commands are used at every particular control level: setTel, setDome, setInstr, setCtrl, startDevice, stopDevice, shutdownDevice, startApplication, stopApplication.
setTel	Set the status of the telescope control software, power supply, position of the telescope axis (HA, DEC), focus, and mirror covers, and state (home, slew, track). Housekeeping data (air temperature and pressure) are also transferred.
setDome	Set the status of the dome nominal and redundant control software, power supply, azimuth position, state (home, slew, track) and dome shutter position (open, closed).
setInstr	Set the status of the instrument control software, power supply and instrument configuration (shutter position, cooling temperature, filter, flat light).
setCtrl	Set the status of the subsystem computers (on, off), data flow and housekeeping software.
startDev	Start a hardware device by switching power on (PDU outlet).
stopDev	Stop a hardware device by switching power off (PDU outlet).
shutdownDev	Shutdown a computer.

moveTo	Change a particular component status to the required position or state (i.e., move telescope HA axis, dome shutter or instrument cooling temperature).
startApp	Start application on a specific subsystem.
stopApp	Stop application on a specific subsystem.
getImage	Execute the camera control software to take an image with specific acquisition constraints (exposure time, binning, flats lamp, etc.).
abortImage	Abort the camera process for the current image acquisition.
setImage	Publish the image parameters (successful, aborted, name, etc.) just after its acquisition.
getState	Determines the current system state.
setState	Set the system state, depending on the kind of task (calibration or science). One or multiple state transitions can be required to reach the goal state. The latter is done with a generic command: chgStart2End.
chgStart2End	Execute a state transition between current and goal states. System configuration is changed at every transition by using the setTask commands. Transition-enabled sequences were defined before.

Table 5.6: Interface definition for execution service

➤ *Task processing interface*

The processing task performs the routines required to obtain the final output data, according to the image analysis and file quality and storage defined processes. OpenROCS manages the process, as well, by triggering actions to specific events and according to the system state. An interface is also required between the observatory database and the data processing task to register the image information and to update the proposal status.

The data and commands for the processing task interfaces are all specified in Table 5.7.

Data	
Item	Values
OpenROCS – Task processing parameters and system state	Information required to configure processing routines: file name and system state. The latter is given to let the task know the required processing option. Instead of the system state, the current implementation uses a string with value “LAST” specified by the image acquisition software to let the processing task know that master calibration frames must be computed.
Processing execution results	Processing data to be stored at the logging system. Processing outputs are stored in the image header, in the image data files and into the database. Image quality must be transferred to the control system to update the proposal status in the mid-term list for the current night and in the database.
Commands	
addEvent	Add an event to trigger the associated actions.
removeEvent	Remove an event.
getProcess	Get the processing information (image, process system state).
moveFits	Move image to process from the acquisition subsystem to the final storage computer.
setICAT	Configure ICAT and execute the processing routine.
getICAT	Get the ICAT processing results.
setOutput	Store processing output at the archive and the database.
updateTask	Update the task status in the mid-term scheduler list and at the database.
setOffsite	Execute the off-site backup copy.

Table 5.7: Interface definition for processing service

➤ *User interface for data retrieval*

Data storage into the local database and automatic mirroring to the off-site server provides a direct access for the external users to retrieve the acquired data using the observatory web-based interface. The latter is possible thanks to the proposal and tasks identifiers that are used through all the data flow process.

Users are automatically notified about the progress and level of completion of their proposals. This can be done by a new service of OpenROCS at the end of the processing task for each acquired image, but its implementation is still under consideration.

5.4.4 Specification of events/alarms and actions

A correct design and implementation of the alarm and event management is a critical element for the robustness of a robotic observatory. We specify here the set of events and alarms foreseen for the TJO-OAdM and the actions they should trigger, as a baseline for the development of the software infrastructure that handles them.

5.4.4.1 Publisher/subscriber and actions

In order to efficiently propagate the events and alarms triggered by the system, the control system has to be capable of distributing them without blocking the communication network or collapsing the system with a waterfall of actions. The control system has to have two subsystem behaviours (publisher and subscriber) used to send (publish) and retrieve (subscribe) events/alarms from other subsystems. Each subscriber in this configuration responds only to those events/alarms that it is concerned with (subscribed). Once an event/alarm is received, it is processed according to the action configuration for each subsystem.

5.4.4.2 OpenROCS nodes

Concerning the conceptual design, an OpenROCS node manager in charge of the aforementioned actions is required for each subsystem. Each manager monitors the housekeeping and data flow variables and it generates events and alarms. It finally triggers the corresponding actions, when necessary.

The set of node managers considered (one per subsystem) are called ROCS@SUBSYSTEM (or just @SUBSYSTEM, for simplicity). This is the list of nodes: ROCS@ESTALL, ROCS@ALIS, ROCS@REBEI, ROCS@RUBIES, ROCS@SARGA, and ROCS@AGER.

Each node must have the interfaces with the Housekeeping and the Data flow layers as defined in this section.

5.4.4.3 General list of events and alarms

In the event and alarm definition, it has been assumed that most of the actions that execute the routine data flow or housekeeping control applications can be triggered by well defined events and alarms.

The events and alarms considered at the TJO-OAdM are grouped in:

- **Data flow Events:** expected (or unexpected, only for a ToO occurrence) events involved in the nightly nominal operation work flow. Defined according to the Operational Design (Sect. 3.1).
- **Housekeeping Events and Alarms:** unexpected events and alarms caused by the environment or by a hardware or software component malfunction.

We summarize in Table 5.8 the events foreseen at each group and the action we have designed for any case.

Category	Description	Actions	Severity
End-to-end data flow events			
Start observation sequence	Time to start configuring the system to be prepared for the first task execution (calibration or science) as selected by the mid-term scheduler. The time value can be: <ol style="list-style-type: none"> 1. $T_0 = T_{\text{FirstCal}} - T_{\text{Calib}}$ 2. $T_0 = T_{\text{FirstCal}} - T_{\text{Calib}} - T_{\text{Init}} - T_{\text{Flats}}$ 	Execute different sequences: <ol style="list-style-type: none"> 1. Change the system state and take Bias or Dark images. 2. Change the system state and take Flat images. 3. Change the system state and take Science images. 	Non critical

	3. $T_0 = T_{\text{FirstSc}} - T_{\text{Calib}} - T_{\text{Init}} - T_{\text{Thermal}}$ Where T_{Calib} , T_{Init} and T_{Thermal} is the amount of time required to change the system state.		
Task sequence iteration	Last image acquisition finishes.	New target selection, performed by the short-term scheduler.	
	If new task requires changing the system state.	Change system state.	
	If state does not need to be changed.	Configure system according to task execution constraints and execute task.	
	Image acquisition and readout ends.	Start image process and quality control.	
End observation sequence	End of the last scheduled task execution.	Sequence finishes by moving the system to the Stop state.	
		Final process and quality control for all the images acquired.	
Housekeeping events and alarms			
Weather events	Bad weather conditions: rain, storm, high humidity or high wind speed.	Abort science observations. Move system to Calib, Stop or Shutdown state.	Critical
	<i>Humidity variation is subject to a hysteresis cycle. This effect implies two humidity thresholds (H_1 and H_2). The system shutdown or initialization is executed depending on the humidity value and the variation slope sign (see Fig. 5.11). The bold line indicates the humidity region where the weather alarm is triggered and maintained.</i>		
Science device alarm	Telescope error (axis motion, encoder positions, mirror cover status, etc.).	The subsystem is restarted or stopped in case the problem remains.	Critical
	Dome error (shutter status, azimuth position).		
	MEIA error (CCD status, filter position, flats lamp status).		
Computer alarm	Computer not responding.	Some services running on a failed computer can be executed on a different computer. If not, the system executes a shutdown or is kept running on a mode with less functionality.	Critical
Application alarm	Application not responding.	The application is restarted. If the problem remains, the system is stopped or is kept running with less functionality.	Critical / Non critical
Support device alarm	Weather sensor error (data are not updated, no active link, reported data are out of accepted range, significant discrepancies between individual sensor data, etc.).	The device is restarted. The system is stopped when the device delivers critical data and the alarm remains.	Critical
	Power supply error (power outage, UPS error, PDU error, etc.).	The system is stopped in case of a long duration power outage or a UPS or PDU error.	

Table 5.8: List of events and alarms considered at the TJO-OAdM

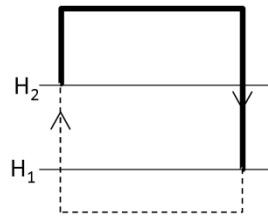


Fig. 5.11: Humidity variation, where H1 and H2 are high and critical values, respectively.

The level of severity (critical or non-critical) depends on the risk for damaging the system equipment, but it has no direct relation with the system efficiency.

The event and alarm generation and management system is designed according to the needs of the TJO-OAdM observatory, although most of events and alarms are generic for unattended telescopes.

5.4.4.4 Detailed list of events/alarms and actions

The list of the events/alarms is summarized here grouped in control layers (Housekeeping and End-to-end data flow). Subsystem/s generating the events and the subsystems subscribed to them are specified.

At a second level of classification, events are distinguished depending on the severity of their occurrence. Two levels of severity are defined (critical, C, or non-critical, NC), depending on the risk for damaging the system equipment, but with no considerations about the system efficiency.

The wide and exhaustive analysis of events and actions presented here is done in order to ensure the system efficiency, reliability and robustness.

➤ **Housekeeping event-action pairs**

Environment monitoring

Code	Event	Action	Child events	Publisher	Subscribers	Severity
HK_EnvGet	New data required at specific cadence	Query the environment monitoring module that returns the latest data.		@ESTALL @SARGA @ALIS	@REBEI @RUBIES	C
HK_EnvSet	New data received	Check if data values are in the accepted range. If not, create an environment event: <ul style="list-style-type: none"> Environment outdoor: outdoor sensor data is out of range (humidity, wind speed, rain, clouds). Environment indoor: indoor sensor data is out of range (humidity). Storm detection produces also an indoor event, as system must be moved to Stop state. 	HK_EnvOut HK_EnvIn	@ESTALL @SARGA	@ESTALL @SARGA	C
		Critical parameters (rain detection, outdoor humidity and windspeed): execute a comparison between redundant data to detect any possible error on the environment sensor. In such case, a sensor error event is created.	HK_ErrSens1 HK_ErrSens2	@ESTALL @SARGA	@ESTALL @SARGA	C

Code	Event	Action	Child events	Publisher	Subscribers	Severity
HK_EnvOut	Environment outdoor	Change system state to Calibration.	DF_EXsetstate	@ESTALL @SARGA	@ESTALL @SARGA	C
HK_EnvIn	Environment indoor	Change system state to Stop.	DF_EXsetstate	@ESTALL @SARGA	@ESTALL @SARGA	C

Table 5.9: Environment monitoring events

System monitoring

Code	Event	Action	Child events	Publisher	Subscribers	Severity
HK_ErrTel	Telescope error: Goal status is not achieved.	Change system state to Calib. The system is restarted again setting the state to Init. This process restarts the telescope and dome control software and hardware. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@ESTALL	@ESTALL	C
HK_ErrDome	Dome error: Goal status is not achieved.	Change system state to Calib. The system starts again setting the state to Init. This process restarts the telescope and dome control software and hardware (nominal and redundant). Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@ESTALL @SARGA	@ESTALL @SARGA	C
HK_ErrMEIA	Instrument MEIA error: Goal status is not achieved.	Change system state to Thermal and restart the instrument control software and hardware. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@ESTALL	@ESTALL	C

Code	Event	Action	Child events	Publisher	Subscribers	Severity
HK_ErrSens1	Critical sensor error: Significant difference between sensors measuring the same variable, communication error, and sensor data not updated.	Change system state to Park and restart the environment monitoring module (software and hardware), including the SUB-REBEI and SUB-RUBIES computers. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@all	@ESTALL @SARGA @REBEI @RUBIES	C
<i>Critical sensors are those giving redundancy (minimum two sensors) and/or measuring: rain and storm detection, indoor and outdoor humidity, wind speed.</i>						
HK_ErrSens2	Nominal sensor error	Conitue operating without this functionality in case the problem remains and generate a Notification event.	DF_EXsetstate HK_ErrNotify	@all		NC
<i>Nominal sensors are those giving extra redundancy (more than two sensors) and/or measuring: cloud coverage.</i>						
HK_ErrPW	Power supply error: Power outage, data not updated, no link to UPS or PDU	Change system state to Stop. Generate a Shutdown state if power supply error involves the UPS-CPU's or the IP-Switch 1 or 3. An Alarm event is generated.	DF_EXsetstate HK_ErrAlarm	@all	@all	C
HK_PWstart	Power supply recovery	If system at Shutdown state, change system state to Stop.	DF_EXsetstate	@REBEI	@REBEI	C
HK_ErrLAN	Internal network error: No connection with router.	Change system state to Stop.	DF_EXsetstate	@all	@ESTALL @SARGA	C

Code	Event	Action	Child events	Publisher	Subscribers	Severity
HK_ErrInternet	Internet error: No connection with SUB-AGER	Change system state to Stop.	DF_EXsetstate	@all	@ESTALL @SARGA	C
HK_ErrCPU	Computer error: Computer does not respond.	Change system state to Stop and restart the computer. Stop state is kept and an Alarm event is generated in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@all	@all	C
HK_ErrToO	ToO service error	The application is restarted using the commands defined at the task execution module. Continue operating without this functionality in case the problem remains and generate a Notification event.	HK_ErrNotify	@ALIS	@ALIS	NC
HK_ErrDB	DB service error	The application is restarted using the commands defined at the task execution module. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@ALIS	@ALIS @ESTALL	C
HK_ErrISROCS	ISROCS service error	The application is restarted using the commands defined at the task execution module. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@ALIS	@ALIS @ESTALL	C
HK_ErrTALON	TALON error: error on telescope, dome or MEIA control level	The application is restarted using the commands defined at the task execution module. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@ESTALL	@ESTALL	C
HK_ErrDRC	Error on dome redundant control SW	The application is restarted using the commands defined at the task execution module. Generate a permanent Stop state and an Alarm event in case the problem remains.	DF_EXsetstate HK_ErrAlarm	@SARGA	@SARGA @ESTALL	C

Code	Event	Action	Child events	Publisher	Subscribers	Severity
HK_ErrICAT	ICAT error	The application is restarted using the commands defined at the task execution module. Continue operating without this functionality in case the problem remains and generate a Notification event.	HK_ErrNotify	@ALIS	@ALIS @ESTALL	NC
HK_ErrOnBP	On-site backup error: Low disk storage capacity	Continue operating by storing the data in a different computer (SUB-ESTALL). Generate a Notification event.	HK_ErrNotify	@ALIS	@ALIS @ESTALL	NC
HK_ErrOffBP	Off-site backup error	Continue operating without this functionality. Generate a Notification event.	HK_Notify	@ALIS	@all	NC
HK_ErrNotify	Notification	Send a notification to the observatory staff. Response is required in 24h.		@all		NC
HK_ErrAlarm	Alarm	Send a critical notification to the observatory staff. Response is required as soon as possible (access time estimation is 20-30 min).		@all		C

Table 5.10: System monitoring events

➤ **End-to-end data flow event-action pairs**

Task preparation: ToO

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_TOOget	New ToO alert received	Get the ToO description data and evaluate its feasibility.		@ALIS	@ALIS	NC
DF_TOOreject	ToO service state: Rejection	Reject the task. Log the process, but the task is not injected.		@ALIS	@ALIS	NC
DF_TOOinject	ToO service state: Injection	Insert task execution constraints into the database.		@ALIS	@ALIS	NC

Code	Event	Action	Child events	Publisher	Subscribers	Severity
		Compute maximum value for the FOM for the current night with the task scheduler.		@ALIS	@ALIS	
		Add target into the mid-term planning list.		@ALIS	@ALIS	
DF_TOOshort	Maximum FOM for the ToO is before the end of current observation	Execute short-term scheduler: get the task with the highest FOM.	DF_SCHgettask	@ALIS	@ALIS	NC
		Abort observation if ToO has the highest FOM and execute ToO task.	DF_EXrunscience	@ALIS	@ALIS @ESTALL	

Table 5.11: Task ToO events

Task scheduling events

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_SCHgettime	Mid-term computing time	Execute mid-term scheduler to obtain the ordered list of tasks for the full night.		@ALIS	@ALIS	NC
DF_SCHgettask	System requiring a new task to execute	Get current environment data and run the short-term scheduler for the selection of the task to execute.	HK_EnvGet	@ALIS	@ESTALL @ALIS	NC

Table 5.12: Task scheduling events

Task execution events

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_EXstart	Night operations starting time is reached	If first task is Calibration: Change system state from Stop to the goal state Calib.	DF_EXsetstate	@ESTALL	@ESTALL	NC
		If first task is Science: Change system state from Stop to the goal state Science.	DF_EXsetstate		@ESTALL @SARGA	
DF_EXgettask	State changing processes are finished or last task execution finishes. No task is under execution.	Query the scheduler for a new task to execute.	DF_SCHgettask	@ESTALL	@ESTALL @ALIS	NC
DF_EXsettask	New task (calibration or science) selected and delivered by the scheduler	Execute the sequence iteration: 1. Check current state. 2. Select option: a. If not at the required state: change the system state. b. If at the required state: execute the task following its specific constraints.	DF_EXsetstate DF_EXruncalib DF_EXrunscience	@ESTALL	@ESTALL	NC

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_EXsetstate	State transition	Execute the processes to reach the required state: <ul style="list-style-type: none"> • Calib: no previous state is required • Init: execute Calib • Flats: execute Calib and Init • Thermal: execute Calib and Init • Science: execute Calib, Init and Thermal • Park: transition to this state from Init, Flats, Thermal or Science. No previous state transition is required. • Stop: execute Park • Shutdown: execute Park and Stop 	DF_EX[Shtdwn2Stop, Stop2Calib, Calib2Init, Init2Flats, Flats2Thermal, 2Park, 2Stop, Stop2Shtdwn]	@all	@all	NC
DF_EXShtdwn2Stop	State transition: from Shutdown to Stop	Execute the sequence: <ol style="list-style-type: none"> 1. Power up the computers. 2. Start OpenROCS on every computer. 		@REBEI	@ESTALL @SARGA @ALIS	C
DF_EXStop2Calib	State transition: from Stop to Calib	Execute initialization of the instrument: <ol style="list-style-type: none"> 1. Power up instrument. 2. Start the instrument control software (Camera). 3. Start instrument cooling system. 		@ESTALL	@ESTALL	NC
DF_EXCalib2Init	State transition: from Calib to Init	Execute initialization of the hardware layer components involved in image acquisition: <ol style="list-style-type: none"> 1. Power up telescope electronics (CSIMC). 2. Start the control software (TALON). 3. Power up telescope motors. 4. Execute homing of telescope axis and dome. 		@ESTALL	@ESTALL	NC
DF_EXInit2Flats	State transition: from Init to Flats	Execute the sequence: <ol style="list-style-type: none"> 1. Switch-on flats lamp. 2. Telescope and dome pointing to predefined position. 3. Wait until flats lamp warming-up process is finished ($>T_{lamp}$ timeout). 		@ESTALL	@ESTALL	NC

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_EXFlats2Thermal	State transition: from Flats (or Init) to Thermal	Execute the sequence: <ol style="list-style-type: none"> 1. If previous state is Flats: switch-off flats lamp (lamp time of operation must be $> T_{lamp}$ timeout). 2. Move telescope to stow position. 3. Close mirror covers. 4. Open dome shutter. 5. Open mirror covers. 		@ESTALL	@ESTALL	NC
		<i>The thermalization process finishes when the indoor and outdoor temperature difference is lower than the specific value of a configuration parameter. A thermalization timeout is used instead as a first approach.</i>				
DF_EX2Park	State transition to Park	If the telescope and dome is not initialized, no process is done. In other case, execute the sequence: <ol style="list-style-type: none"> 1. Move telescope and filter wheel to stow position. 2. Close mirror covers. 3. Close dome shutter. 4. Move dome to stow position. 		@ESTALL	@ESTALL @SARGA	C
DF_EX2Stop	State transition to Stop	Execute the sequence: <ol style="list-style-type: none"> 1. Power down instrument. 2. Stop instrument control software (Camera). 3. Power down telescope motors. 4. Power down telescope electronics (CSIMC). 5. Stop TALON. 		@ESTALL	@ESTALL	C
DF_EXStop2Shtdwn	State transition from Stop to Shutdown	Execute a shutdown of all the computers, except SUB-REBEI, the Router-LAN and SUB-AGER.		@ESTALL @SARGA @ALIS	@ESTALL @SARGA @ALIS	C
		Power down computers once shutdown finishes.		@REBEI	@REBEI	

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_EXruncalib	Execute calibration task	If dome Flat image, execute the sequence: <ol style="list-style-type: none"> 1. Get the required environment data for telescope and focus positioning. 2. Move filter and focus to the specified position. 3. Acquire image. If Bias or Dark images, acquire them with no previous process.	HK_EnvGet	@ESTALL	@ESTALL	NC
DF_EXrunscience	Execute science task.	Execute the sequence: <ol style="list-style-type: none"> 1. Get the required environment data for telescope and focus positioning. 2. Telescope and dome pointing to predefined position. 3. Move filter and focus to the specified position. 4. Acquire image. 	HK_EnvGet	@ESTALL	@ESTALL	NC

Table 5.13: Task execution events

Task processing events

Code	Event	Action	Child events	Publisher	Subscribers	Severity
DF_PRset	New image acquired	Execute data processing routine for the new image.		@ESTALL	@ESTALL @ALIS	NC
DF_PREnd	Image processed: ICAT end signal	Execute update of the task status in the mid-term scheduler list and at the database. Store results of the task execution in the observatory archive and database.		@ALIS	@ALIS	NC

Table 5.14: Task processing events

5.4.5 Software infrastructure development

Different approaches were considered when defining a way to implement the communication among the applications at the observatory and manage the events and alarms. Finding an existing well tested and under public license software application was initially explored, but a new one was finally developed. The application is programmed on the PHP scripting language and it is configured with XML files, providing a simple and reliable software easy to maintain.

A short summary of the programming solutions explored during the development of OpenROCS is given here and an extended description of its implementation is given in the following chapter.

5.4.5.1 Development framework

We considered the following points when evaluating the possible solutions to implement the software infrastructure once the detailed list of event-action pairs and the interfaces were defined:

- **Number and typology of subsystems:** The TJO-OAdM control system is based on multiple subsystems that have to work together in a transparent way. One of the subsystems (SUB-AGER) is located in a different network, and there are different operating systems running in some subsystems.
- **Applications running at each subsystem:** Applications are distributed in different subsystems and programmed in different languages, introducing a level of non homogeneity.
- **Events/alarms to be considered:** A significant amount of events and alarms, with internal dependencies, and their corresponding actions characterize the system complexity.
- **Applications required to control the data flow process and the hardware operation:** Data-flow process has to be executed in a transparent way, whatever the number of computers is required to run the applications and operate the hardware.

5.4.5.2 Analysis of the programming tools

The first attempt to program the software infrastructure was to base this tool on simple shell scripting commands, executing a continuous supervision of the running processes on each computer. This approach was identified as too complex early on and not appropriate in terms of reliability, scalability and maintenance.

The second approach was trying to use existing software, such as NAGIOS (Barth 2006) or Ice (Henning et al.). Actually, a first prototype of OpenROCS (v1.0) was implemented using Ice, although this technology proved to be unsuitable for the observatory control architecture and it was discarded when less than 30% of the complete software code was written. Ice (and NAGIOS) is mainly designed to perform a distributed control of processes in farm computing. The distribution of services at the TJO-OAdM is quite limited instead, mainly due to the dependency on hardware that can only be controlled by one specific computer.

The final implementation of OpenROCS (v2.0) was, then, developed using: PHP, for the software services; XML files, for the configuration of the system responses (events and actions); and commands based on standard protocols and small programs to carry out some actions. This was done from scratch, although some libraries and routines of the first version were reused.

The two programming tools that were analysed in detail (Ice and PHP-XML) are presented here, together with a comparison with the two implementations of OpenROCS. It is important to mention that the tool or language selection is somewhat linked to the expertise of the development engineers, although the features of the software tools are the first level criteria for the selection.

➤ *Internet Communications Engine (Ice)*

After a deep analysis of the possible existing software tools to handle actions produced by predefined events and alarms (application and machine status supervision through the network), we considered a CORBA-based middleware as the first solution. As stated in the literature: "CORBA enables separate pieces of software written in different languages and running on different computers to work with each other like a single application or set of services and it is a mechanism in software for normalizing the method-call semantics between application objects residing either in the same address space (application) or remote address space (same host, or remote host on a network)". The development of a distributed control system based on this kind of middleware seemed to offer the features required for the TJO-OAdM in terms of the capability to supervise applications running in different machines, compatibility with multiple programming languages, event/alarm handling, etc. And it also provides a general infrastructure to implement all the software applications in a transparent and homogeneous way. A general control system based on this infrastructure would also act as a glue element for all the applications developed.

The participation at the SPIE¹ course "An Introduction to Scalable Frameworks for Observatory Software Infrastructure" during the SPIE Astronomical Instrumentation meeting held in Marseille in 2008 gave us the opportunity to learn about this kind of programming for the development of observatory control software. Two years later, we had a second interaction with this kind of programming thanks to the collaboration with the Cherenkov Telescope Array (CTA) project (CTA Consortium, 2010). The CTA control system is based on the ALMA Common Software (ACS, Chiozzi

¹ <http://www.spie.org>

et al. 2002), a middleware developed by the European Southern Observatory (ESO¹) for the Atacama Large Millimeter/submillimeter Array (ALMA²) project and implemented in CORBA. The advantages to use this approach were again made evident and seemed to be the best solution for the TJO-OAdM system. At that time, the expertise of the engineers working at the TJO-OAdM made this approach affordable to develop such an infrastructure.

We decided to develop this infrastructure using Ice³ software developed by ZeroC (Henning et al.), after a comparison between different existing implementations of CORBA-based middleware. Ice is developed on the same principles as CORBA but trying to solve some of its drawbacks. One of its major strengths is the extra services it offers. The main features of Ice are listed here (extracted from the Ice webpage):

- Operating systems and programming languages: Ice is a general-purpose object-oriented middleware platform. It runs on a large number of operating systems and provides language mappings for C++, Java, C#, Objective-C, Python, Ruby, and PHP.
- Application Programming Interfaces (APIs): Ice provides a set of APIs that emphasize simplicity and ease of use.
- Advanced technology: Ice supports synchronous as well as asynchronous calls, co-exists with firewalls due to its support for bidirectional connections, allows messages to be batched for efficiency, and permits sophisticated control of threads and resource allocation.
- Fault tolerance and load balancing: Ice allows creating systems that are fault tolerant. Multiple instances of a server can be deployed on different machines, with transparent fail-over if a machine crashes or is disconnected from the network.
- Performance: At the network level, Ice uses an efficient binary protocol that minimizes bandwidth consumption. Ice uses little CPU and memory, and its highly efficient internal data structures do not impose arbitrary size limitations. Data can be transmitted at whatever speed is supported by the network, so Ice does not create any performance bottleneck.

➤ **PHP and XML**

PHP is a general-purpose scripting language and is especially suited to server-side web development where PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on web sites or elsewhere. It can also be used for command-line scripting, programming of client-side GUIs and applications. PHP can be deployed on many operating systems and platforms, and can be used with many relational database management systems.

PHP is widely used and supported by the community providing a lot of years of testing. It is similar to Python or Perl, but uses syntax similar to the C programming language. Some of the features used to implement OpenROCS are:

- `pcntl_fork` and `pcntl_signal`: They allow starting scripts in background mode with a C-like communication.
- `register_tick` function: It allows programming a periodical task. It is used as watchdog control system.
- `socket_create`, `socket_bind`, `socket_listen` and `socket_accept` with specific timeouts: They are used to publish a network service.

PHP also provides a lot of features to manage arrays, matrixes and trees, parse strings and XML files, use of variables by reference, the variable variable feature and more.

XML (standing for Extensible Markup Language) files are used to specify the configuration of the software in a human friendly and machine-readable form. The tasks (events, actions, monitoring, etc.) performed by the system and the variables can be set in these files.

➤ **Comparison of the two implementations**

The first version of OpenROCS based on Ice was partially implemented. It envisioned a transparent distribution of services, with a high flexibility to adapt it to the configuration of the subsystems (and their computers). The initial integration tests showed a good performance, although only a basic set of functionalities for the unattended operation were deployed. Some important drawbacks appeared when the deployment continued and new services were added.

¹ <http://www.eso.org>

² <http://www.almaobservatory.org/>

³ <http://www.zeroc.com/ice.html>

The resulting software permitted a homogeneous interfacing with the different modules running at each subsystem. It had the capability to execute a fraction of the actions specified to manage the system work flow and the housekeeping according to the events described in Section 5.4.4.4. But, on the other hand, the software was very difficult to modify and maintain because it had a big, complex and monolithic structure, and several programming languages had been used in the deployment. The configuration of the actions was also an arduous task. These were a consequence of the codification used in order to gain in transparency by separating this control infrastructure from the Hardware layer components.

The first implementation approach was frozen at a stage that it enabled the execution of some unattended operations with the TJO-OAdM telescope. Robotic observations could, then, start. At the same time, an analysis of the programming alternatives was done and a new implementation started using PHP programs and XML files with a more modular and flexible architecture. The CORBA middleware and, thus, the transparent distribution of services were discarded. The latter is not critical for the TJO-OAdM telescope, where most of the services are hardware-dependent and they can only be executed at one specific computer, although they had given us the expertise to use this kind of deployment in other observatories with a more complex configuration. The development of a new implementation was resolved to be more efficient for the project in terms of efficiency, reliability and maintenance work load, resulting in a new version of OpenROCS.

The second version of OpenROCS implements the software with a few and short PHP programs that execute the main functionalities, and transfers the logic of the system to the XML files. These XML files define the events and actions the software execute in such a modular way that any modification on the system configuration can be done easily.

Chapter 6

OpenROCS

OpenROCS is a software for the control of robotic astronomical observatories, whose development is based on the design considerations presented in Sect. 5.4.

It acts as a software infrastructure and executes all the necessary routines to implement responses to the identified events and alarms that can appear in the routine and non-routine operation workflow. A set of tools are included in this software to manage data flow processes and to monitor the environmental conditions and the hardware components health, and to manage and generate events according to these conditions. All the applications running at the TJO-OAdM observatory and the hardware components interact with it.

The implementation was carried out in two phases, as justified in Sect. 5.4.5. We summarize in the first section of this chapter the description given in Francisco (2010a) and Francisco (2010b) that describes the features of the first prototype (OpenROCS1.0) based on the Ice software. It is included here to give a complete description of the work done, although this version was only used during an early stage of the unattended operation. The final version of this software (OpenROCS2.0) is described next, as described by Sanz (2011).

6.1 OpenROCS1.0, the first prototype

We describe in this section the first prototype of OpenROCS (OpenROCS1.0). The level of completion of this software when its development was aborted enabled a fraction of the functionalities required for the specified unattended control. The nominal operation to acquire science and calibration images was possible, but it did not handle the complete data flow and it had a very low level of feedback with the system. So, it could be used to operate the telescope in a first stage and with a supervised mode.

6.1.1 Architecture overview

The main goal of OpenROCS1.0 was having a software infrastructure that handled events and alarms and communicated with the running applications in a transparent manner. A suitable architecture (Fig. 6.1) was defined with this purpose using the Ice middleware.

6.1.1.1 Middleware layer

The lowermost layer in this architecture is the middleware layer used to communicate the various elements of the system. The Ice services used to compile this layer are *IceGrid*, *IceBox* and *IceStorm*, described below:

- *IceGrid* is the location and activation service for Ice applications and it simplifies the deployment of these applications in grid networks and heterogeneous environments. It is one of the central parts of the OpenROCS1.0 environment as it provides the reliability, distribution, replication and failover capabilities.
- *IceBox* is an Ice Applications container and service manager that allow several of these applications to run in the same server space, providing better memory and connection management.

- *IceStorm* is a Publisher/Subscriber service for Ice Applications. *IceStorm* can be seen as a system to propagate messages in a one-to-many and many-to-one way. It is also a central part in OpenROCS as it powers the Environmental Data Collector and the Alarm/Event Generator and Manager subsystem.

6.1.1.2 Core

The middle layer is the core part of the OpenROCS1.0 suite, where all the central and common parts of the system are implemented and the framework, API and libraries needed to implement the modules are defined.

One of the elements contained in the OpenROCS-Core is the *Locator Service*. A locator service is a feature of the IceGrid system that allows the developer to call an object through its proxy without knowing its endpoints, just knowing its name. It is, then, up to the deployer to choose where this object will be located.

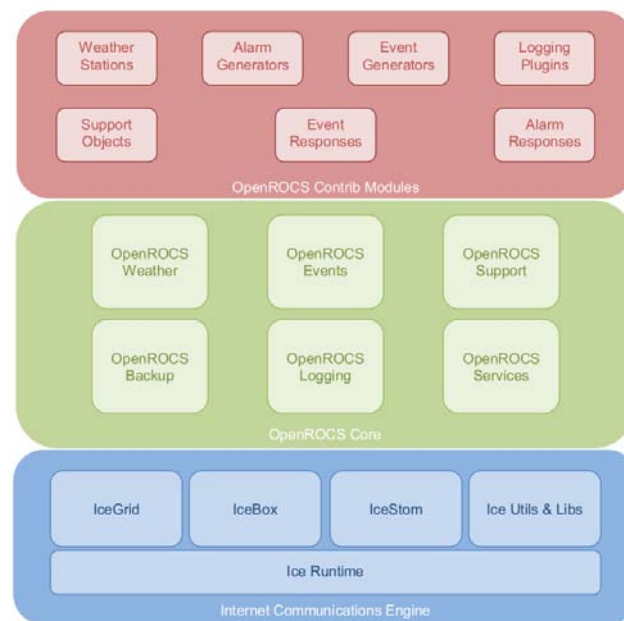


Fig. 6.1: OpenROCS1.0 global architecture

OpenROCS-Events is the module that implements the alarm/event generation and management. It includes the definitions of the alarm and event generators and the Event Monitor, described below.

OpenROCS-Weather is the subsystem that manages the collection and distribution of environmental data and also monitors how many weather stations are up and running. It is designed using the replication capabilities provided by IceGrid and the publisher/subscriber architecture provided by IceStorm.

OpenROCS-Support is a framework that permits the interaction with other elements in the observatory such as the TALON software and the dome redundant control.

OpenROCS-Logging is an infrastructure built on top of Ice logging system, because its maturity level was deemed sufficient to fit our project needs.

OpenROCS-Services contains a framework to automatically start and stop other OpenROCS1.0 modules such as weather stations and alarm/event generators. This may be used, for example, to start a service when an event occurs.

OpenROCS-Backup is a module that contains a framework to implement a common backup infrastructure independent from the underlying backup system and software.

6.1.1.3 Contrib Modules

The upper layer is where all the different modules that use the OpenROCS services are implemented. All the weather stations, alarm generators and actions and logging plug-ins (among other components) go into this layer.

6.1.2 OpenROCS1.0 modules

Several modules were designed and developed for the OpenROCS1.0. These are those in charge of the basic functionalities of the unattended operation: management of events (Event module), monitoring of the environment status (Weather module), and communication with hardware components (Support monitor).

6.1.2.1 Event module

The event module is the set of subsystems that allows the generation, collection and management of the events and alarms that occur on the observatory. It is also the one that contains the systems and libraries needed to execute a predefined set of actions when an alarm or an event appears.

The Event Monitor (EM) is the core of the Event module. It allows the system to execute an action when an alarm or an event appears, it keeps tracking of the active alarms and events and their state and it provides the status of an alarm or an event when required by the alarm and event generators.

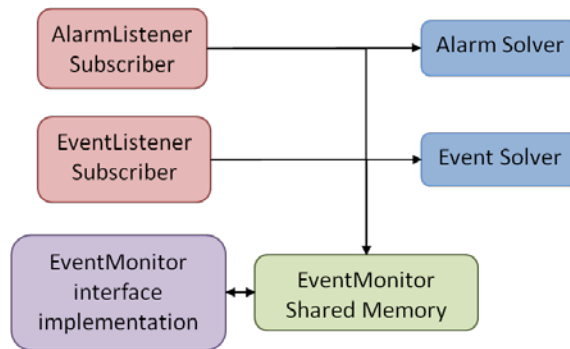


Fig. 6.2: Event Monitor block diagram

Fig. 6.2 contains a block diagram of the EM. It implements a subscriber for both the “alarms” and “events” channel in *IceStorm* so the system knows when an alarm or an event appears into the system. An XML file is loaded at startup, containing the actions that must be executed when an action trigger or an event appears or disappears. When an action trigger or event appears, the AlarmListener/EventListener Subscriber notifies the EventSolver or the AlarmSolver, which starts the actions. Fig. 6.3 contains the flow diagram for an event appearing/disappearing and the solving process for an Alarm.

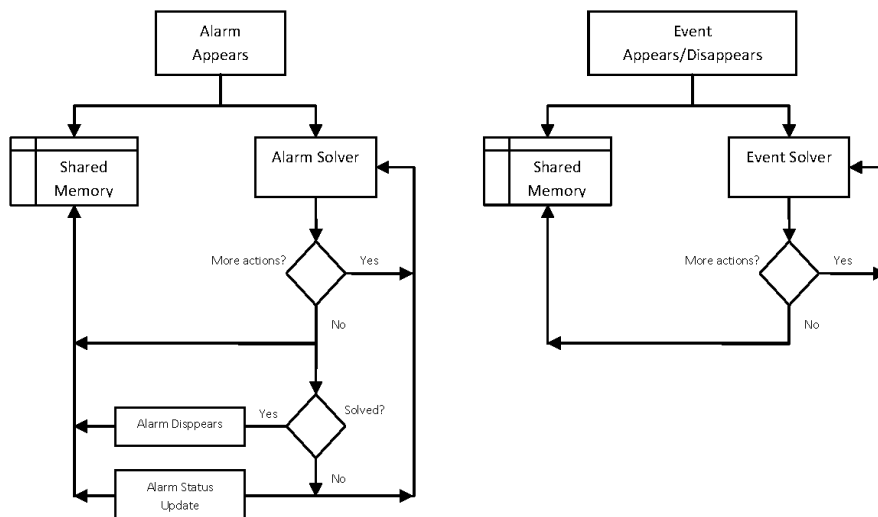


Fig. 6.3: Alarm and event solving flow diagram

```

<root>
  <action name location /> #one or more
  <alarm name> / <event name> #zero or more
  <appear> / <disappear>
    <actionpack index> #zero or more
    <requirements> #may be null
    <alarm name status count /> #zero or more
    <event name status /> #zero or more
    <requirements>
    <action index name param delay /> #zero or more
    <delay>#seconds</delay>
    <actionpack>
  </appear> / </disappear>
</alarm> / </event>
</root>

```

Fig. 6.4: Event monitor configuration XML file

Fig. 6.4 illustrates the specification of the XML configuration file for the event monitor. It contains the definition for different subsets of actions (action packs) with a set of requirements (such as active events in the system or alarms present) that will be executed when an alarm or an event appears or disappears. Most of the system configuration lies in this XML file, which establishes the general behaviour of the observatory following the specifications given in Sect. 5.4.4.

➤ Alarm and event generator

The alarm and event generators are services that constantly monitor the status of one or more variables (weather, disk space, Sun position...) and send events or alarms to the *IceStorm* channels whenever needed.

Alarm generators constantly analyze the status of the system and whenever an anomaly appears they generate the corresponding alarm and introduce it into the system.

One of the most important generators at the TJO-OAdM is the weather alarm generator. It defines the thresholds of the weather values and it generates alarms according to these thresholds. It is also the one that continuously monitors the status of the weather stations. It generates alarms related to humidity, temperature, wind speed, errors on sensors, insufficient number of sensors for specific variables (some of them require redundancy), outdated data, rain detection, and measurement error. It also performs a comparison between data coming from different sensors. The latter is a fundamental feature with a direct consequence on the system reliability providing redundancy in the housekeeping control. A configuration file defines the accepted data ranges for each variable, the sigma and threshold to compare values for the same variable measured with different sensors.

The sunset event generator triggers the events that are used to control the start and stop of all the nominal operation for the astronomical observations. This generator is included in the OpenROCS-Contrib module and is written in Java.

Other alarm generators include disk space, hardware, process monitors, etc., although their implementation is not finished.

Event generators, on the other hand, are services that generate events according to some conditions: Sun position, time of the day, service is running. They notify an event to the system.

➤ Action

An action is a service that is executed as a consequence of the appearance/disappearance of an alarm/event in the system. They are small programs that do just one or two operations (such as SNMP notifying, file copying, event propagation, computer reset and shutdown) when required.

6.1.2.2 Weather module

The OpenROCS-Weather module defines how a weather station interacts with the system and how the environmental data are sent. It is a simpler module as most of the work is done by the weather station programs and drivers. It is composed by two elements: the environmental data publisher and the weather station monitor. Each weather station monitor has to connect to the environmental data publisher to send the data collected to the system.

- **Environmental data publisher/subscriber:** Weather data are sent to the system using the *IceStorm* publisher/subscriber architecture. Any data are transferred to a structure that contains all the environmental data understandable by the system and also an error type, used when the collector detects a problem in the station.
- **Weather station monitor:** It is in charge of monitoring whether a station is active and sends its data to the system.

The weather data collectors are the active part of the weather module. Each data collector is tied to a weather station and acts as the bridge between the driver and OpenROCS1.0.

➤ **Weather libraries**

The weather libraries are included at the OpenROCS-Core module and contain a set of tools or methods to ease the implementation of weather station monitors. Mainly written in C++, they also provide a utility class to compute the Julian Date for a given UTC time used to add a time stamp to every single value measured at every sensor.

➤ **Weather sensors and monitors**

A description of the weather station monitors developed is shown in Table 6.1. It includes the subsystem in charge of each monitor.

Station	Module	Description	SUBSYSTEM	Language
EM-Davis station	Contrib	Collection of data from the EM-Davis weather station. It has 3 sets of sensors distributed at different locations in the observatory, as described in Chapter 4.	SUB-REBEI	C++
Previstorm	Contrib	Electrostatic field measurement to detect electric storms approaching the observatory by measuring big gradients on that variable.	SUB-REBEI	C++
Rain detector	Contrib	Measurement of a switch status change, assigning True if raining and False if not.	SUB-REBEI	C++
EM-SMC station	Contrib	MSWindows service that reads the data collected from the EM-SMC station at SUB-RUBIES. It then distributes the data (temperature, humidity, daily rain, wind speed, wind direction, solar radiation and pressure) over the network. The stream of data is read and parsed to handle these data.	SUB-REBEI	Python/Java
Sun position	Contrib	It provides the solar altitude above the horizon to be used to establish sunrise and sunset time and, then, managing the nominal operation of the observatory	SUB-REBEI	C++
EMDS bridge	Core	The EMDS (Environment Monitor Data Stream) bridge collects the weather data coming from the stations implementing a Monitor. Every time a station sends data it is collected and stored in the memory. It is then read by a plain socket connection, getting the last data from each sensor. It is used to maintain the compatibility with older modules that use the EMDS to get environmental data from the stations in a plain-text format.	OpenROCS-Support	Java

Table 6.1: Weather station monitors

6.1.2.3 Support module

This module contains several routines to interact with different components of the observatory. At the time the implementation of OpenROCS1.0 was stopped only two of them had been specified and partially deployed (TALON and FSC1 integration). Those devoted to interface with data flow applications (scheduler, data processing, etc.), for instance, were missing.

- **TALON integration:** The entire decision making was stripped off TALON, and just the telescope, dome and CCD camera control was kept. TALON employs a POSIX Shared Memory interface to inform about the status of the different elements (telescope, dome, mirror covers, etc.) and a set of named pipes to interact with the hardware. To interact with it, an `Ice::Object` was defined that bridges the POSIX Shared Memory to the OpenROCS system. It

is up to the implementation of an event or alarm generation to read the status of the hardware element and act accordingly.

- **FSC1 Integration:** FSC1 is the core of the redundant system that controls the dome and allows OpenROCS1.0 to bypass TALON in the hardware control of the dome. This module acts as a bridge between OpenROCS1.0 and the serial interface provided by the FSC1.

6.1.3 OpenROCS1.0 code structure

The OpenROCS1.0 code is based on the Ice middleware and, therefore, it is implemented following this middleware instructions and using several of the services it provides.

New code development mainly involved the Core and the Contrib layers, already described in the software architecture overview:

- **OpenROCS-Core:** It contains the base system and configuration files. They were defined from a general perspective in order to build a software suite suitable to be used at any kind of unattended observatory.
- **OpenROCS-Contrib:** It contains the implementation of modules to be used at the TJO-OAdM. Modules were deployed to respond the control requirements of each particular subsystem. Modules implemented in more than one machine are stored at SUB-ESTALL as it contains most of the OpenROCS1.0 implementation at the TJO-OAdM. Modules were generally grouped by the programming language used (C++, Java and Python).

6.2 OpenROCS2.0

The second version of OpenROCS (OpenROCS2.0) provides more flexibility to adapt this software infrastructure to different observatory configurations and event-action specifications. It is based on an abstract model that is independent of the specific hardware or software and more configurable. Interfaces to the system components are defined in a simple manner to achieve this goal.

This new implementation is operative and has proven its reliability to perform the required actions in an efficient way.

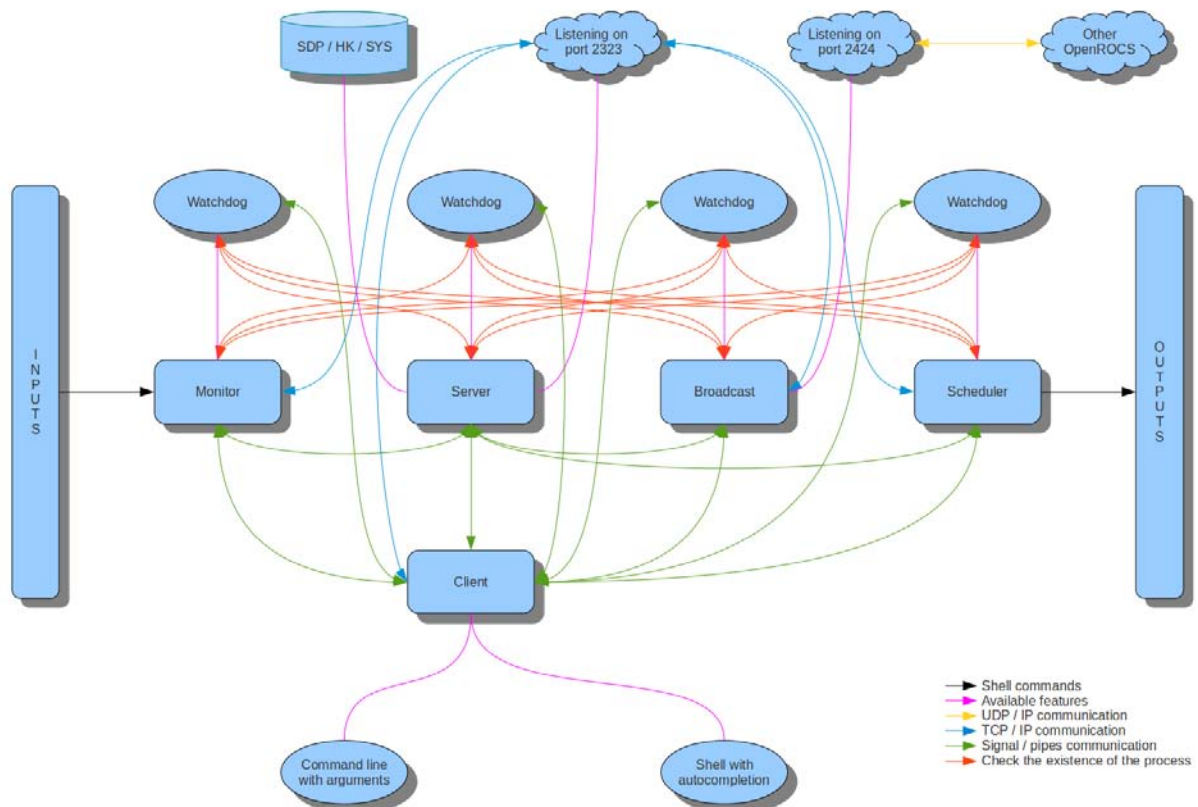


Fig. 6.5: OpenROCS2.0 global architecture

6.2.1 Architecture overview

Fig. 6.5 shows the software architecture. The different modules and the communication channels among the running processes are illustrated.

The software is composed for four main services: *Monitor*, *Server*, *Broadcast* and *Scheduler*. Input data are composed for the values of the environment and system variables, which are taken by the *Monitor* service. The actions performed on the system by the *Scheduler* service represent the software output. A client is also provided to let the system actors (Superuser and ROCS) manage the services. Several XML files configure the system tasks.

This architecture corresponds to an instance of the software running in a subsystem. Several instances of OpenROCS are running at the TJO-OAdM according to the number of subsystems that compose the full observatory. Each one has a specific configuration and communicates with the rest thanks to the *Broadcast* service functionalities.

6.2.2 OpenROCS2.0 modules

The OpenROCS2.0 services carry out different actions, although they are structured in a similar way and use the same codification to implement the watchdog functionality, the communication among processes and the process management. These common parts ensure the system integrity and robustness by performing a continuous check of the entire system state. The watchdog routine, for instance, is used by each service to monitor the status of the other services and to detect a faulty situation.

6.2.2.1 Server

Fig. 6.6 illustrates the main parts of this service. It executes a TCP/IP server on port 2323 and handles requests from the other services using this socket communication. It provides remote storage of the system variables grouped in three different categories:

- System Data Pool (SDP): The system state and all the variables associated with the packages that manage the observatory workflow (ISROCS, ICAT, etc.) are included in this category.
- Housekeeping (HK): It includes the weather and sky quality data.
- System (SYS): It includes the system health data.

The data are stored using total or incremental timestamps. The *Server* also provides a language to manage stacks, variables and their values.

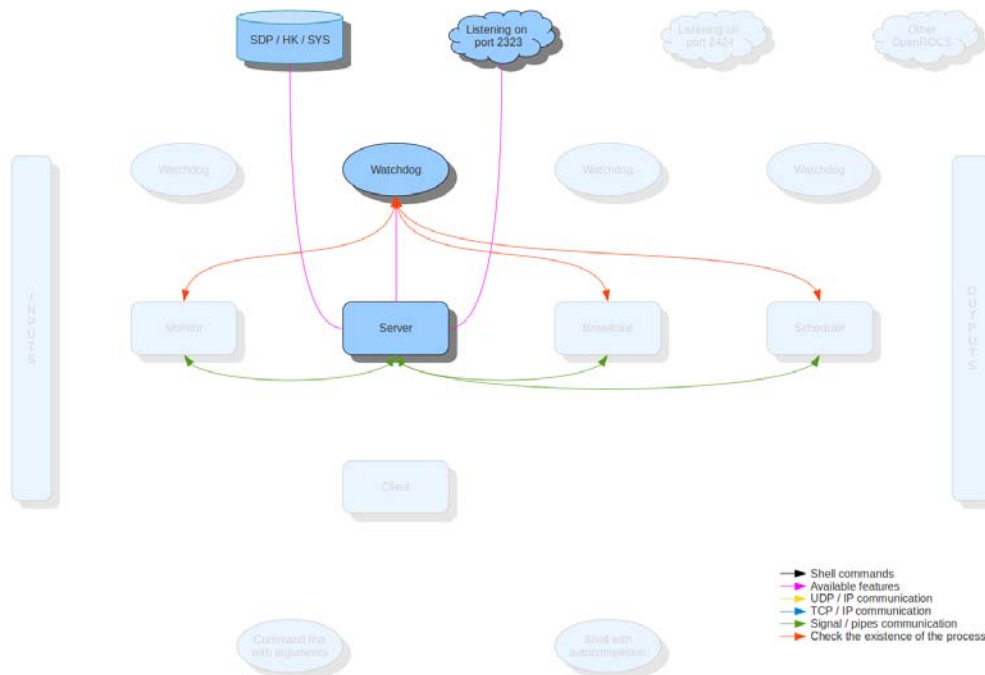


Fig. 6.6: Server service structure

6.2.2.2 Broadcast

The main goal of this service is the synchronization of multiple instances of OpenROCS2.0 using broadcasting. The other instances can be running in other computers distributed on a network. Its main functionalities are:

- Execute a UDP/IP server on port 2424 to handle requests.
- Send the discovery packets to the broadcast.
- Update the stacks located on *Server* with the new information

Fig. 6.7 illustrates the structure of this service.

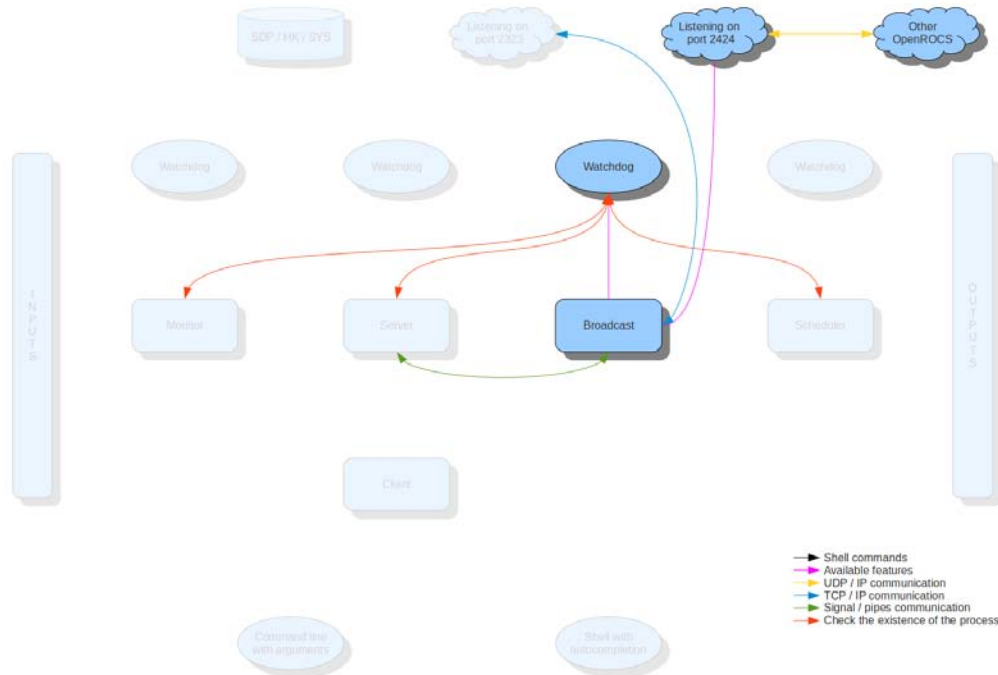


Fig. 6.7: Broadcast service structure

6.2.2.3 Monitor

The main objective of this service is the execution of periodic tasks to perform a continuous evaluation of the system and determine its state. It is mostly responsible for the management of the housekeeping events specified in Sect. 5.4.4. Fig. 6.8 illustrates the structure of this service.

It has the following features:

- It carries out a periodic execution of preconfigured tasks. It adds, removes or updates the values of different stacks (SDP/HK/SYS) depending on the output of the task.
- The tasks can be executed at time intervals or at a specific frequency. A time interval specifies the time between the end and start times, whereas a frequency determines the separation the starting times of consecutive tasks.
- It controls the task execution using timeouts and alternate actions in case the timeout is reached.
- It performs a multiple evaluation control of the output generated by each task. This evaluation is based on the use of the choose/when/otherwise language structure.
- It communicates with the *Server* service to set the obtained data.

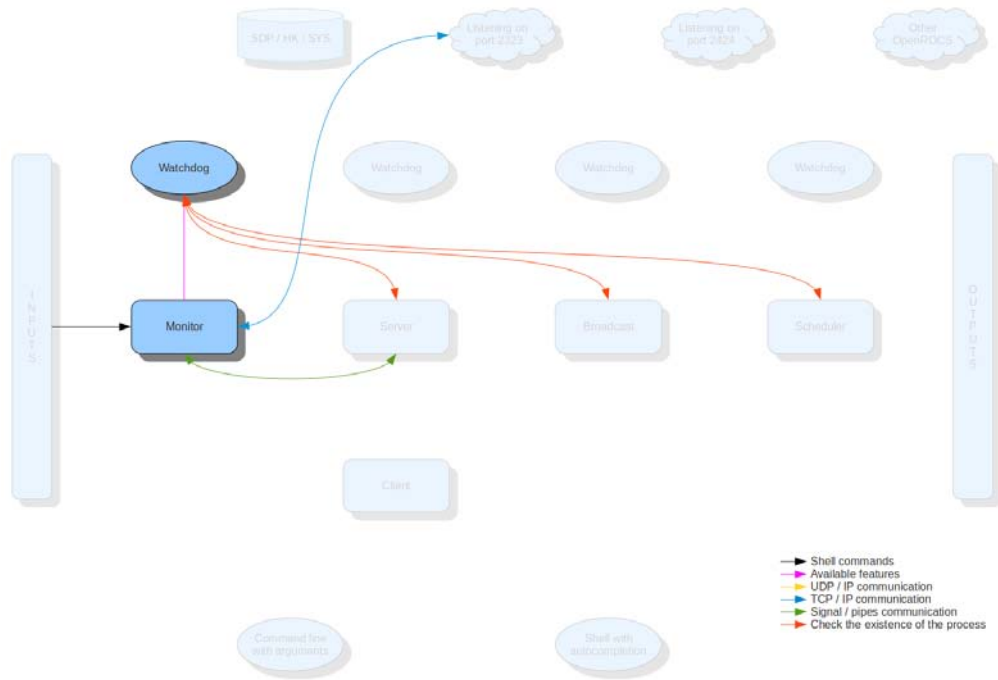


Fig. 6.8: Monitor service structure

6.2.2.4 Scheduler

The Scheduler service executes actions when preconfigured SDP or HK variables change. It is used to control the entire telescope workflow, implementing the data flow events mentioned in Sect. 5.4.4. Its structure (Fig. 6.9) implements the following features:

- It gets and sets data from and to the *Server* service. Only those stacks and variables that are added, updated or removed are reported in order to reduce the data rate and increase the service efficiency. This is carried out using a control timestamp.
- Multiple execution work flows and variables can be specified in the *Scheduler* configuration.
- The execution of a task can be triggered by the *Monitor* service when it detects a significant change on a SDP/HK/SYS variable that affects the observatory workflow.

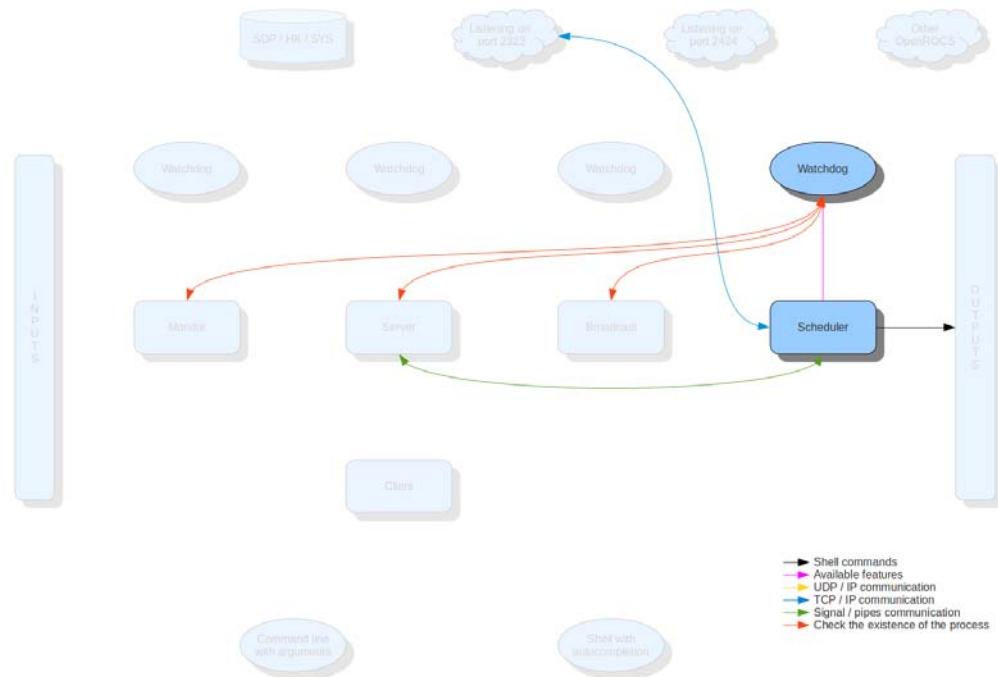


Fig. 6.9: Scheduler service structure

6.2.2.5 Client

The *Client* module enables a direct control of the service status (start/stop/restart/reload) and provides two modes of use (command line with arguments or shell with autocompletion). Its structure is shown in Fig. 6.10.

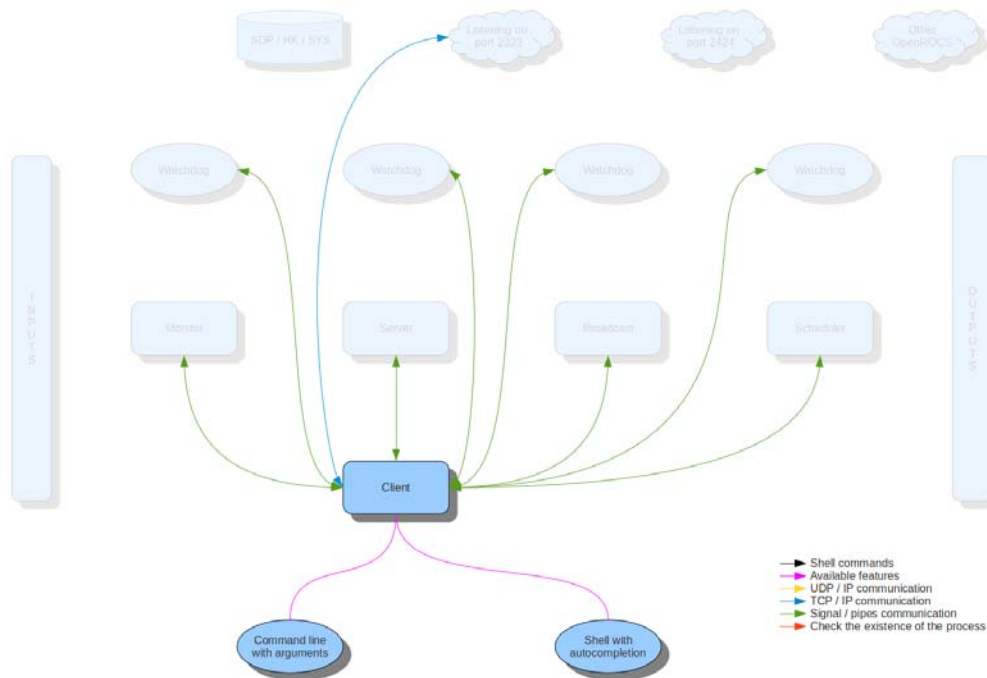


Fig. 6.10: Client module structure

It was designed to provide the actors that control the telescope (Superuser or ROCS) with a tool to modify the running *Scheduler*. This client adds, updates and/or removes data from the SDP/HK/SYS variables. This can be done in two different modes:

- Command line with arguments: It allows executing the OpenROCS2.0 commands with arguments that are executed synchronously (blocking mode).
- Interactive shell with autocompletion: It allows the human interaction using a system shell (i.e., Bash) with autocompletion.

These two modes have the same kind of commands. They were implemented to have an interface with OpenROCS2.0 that makes the integration into different software or into a GUI an easy task.

These are some examples of implemented commands:

- Commands with no arguments:
 - shell: It opens an interactive shell (default if not command is present).
 - help: It displays the complete list of commands.
 - start: It launches the main Server process.
 - restart: It restarts the main Server process.
 - reload: It does the same action as the restart command, but without losing stored data.
 - stop: It stops the main Server process.
 - status: It checks the Server state.
 - dump: It dumps all stacks, their keys and values.
 - check: It runs a quick validation test.
 - history: It displays the command history (only exists in the interactive shell).
- Commands with arguments
 - get [none]: It returns a list with all the stacks.
 - get STACK: It returns a list with all data in the STACK.
 - get STACK KEY: It returns the value of the KEY in the STACK.
 - get TIMESTAMP: It returns a list with the modified stacks from TIMESTAMP.
 - get STACK TIMESTAMP: It returns a list with the modified data from TIMESTAMP in the STACK.
 - add/create STACK: It creates the STACK if it does not exist.
 - add/create STACK KEY: It adds the KEY to the STACK.

- add/create STACK KEY=VALUE: It adds the KEY with VALUE to the STACK.
- update/set STACK KEY=VALUE: It updates KEY with the new VALUE in the STACK.
- remove/delete STACK: It removes the STACK only if it exists and is void.
- remove/delete STACK KEY: It removes the KEY in the STACK if exists.
- stop SERVICE: It pauses the service requested.
- status SERVICE: It checks the status of the service requested.
- start SERVICE: It continues the service requested.

6.2.3 System interfaces

OpenROCS2.0 uses various interface protocols for the communication among services (*Server*, *Broadcast*, *Monitor*, *Scheduler* and *Client*) and access the data storage through the *Server* by setting and retrieving SDP/HK data.

6.2.3.1 Signals and pipe

This communication protocol is used for three different purposes:

- It is used by the *Start* command and to communicate the *Client* with the child processes created. It is used to capture the output generated (console messages).
- The watchdog process transfers the complete list of children and checks the existence of other analogue processes using this protocol.
- The *Server* also uses it to start and stop the process in other subsystems.

6.2.3.2 TCP/IP sockets

The TCP/IP is the common protocol for all the implemented services with the following features:

- The *Server* service creates a server to handle all requests through the 2323 communication port and give subsequent responses.
- The *Client* sets and gets data from and to the SDP/HK/SYS and services launched by OpenROCS2.0 as child processes.
- The *Monitor* sends all task results to the storage server (SDP/HK/SYS).
- The *Scheduler* retrieves and sends data to the SDP/HK/SYS storage variables.
- The *Broadcast* uses this channel to update the stacks with the data provided by another OpenROCS2.0 instance.

6.2.3.3 UDP/IP sockets

This communication protocol is only used by the *Broadcast* service:

- The Broadcast service executes a server that handles all the requests from port 2424.
- It sends packets with a specified periodicity to the broadcast addresses to detect other OpenROCS2.0 instance running in the same network.

6.2.3.4 Shell commands

OpenROCS2.0 does not implement specific software to communicate and act on the telescope and other components of the observatory (i.e., weather station). The command line shell is the only method used to communicate with the system components to retrieve input data and send actions. This provides the aforementioned abstraction level that is required to have a system easy to modify and to adapt to a new configuration set.

Shell commands allow using standard tools. Two examples are given below:

- SNMP protocol commands are used to get or set parameters from several devices (PDUs, UPSs, weather stations). OpenROCS2.0 defines external commands based on them: `snmpget` and `snmpset`.
- The INDI¹ (Instrument-Neutral-Distributed-Interface) standard control protocol will be used in the future to communicate with the telescope. This protocol is designed to operate astronomical instrumentation.

6.2.4 OpenROCS2.0 configuration

The OpenROCS2.0 uses XML files to define the system configuration, the variables, the data structures, and the event-action pairs that this software has to manage. XML standard syntax is used.

¹ <http://www.indilib.org>

6.2.4.1 File config.xml

This file is used by all services and the Client. It specifies the *Server* configuration, the stacks, the debug flags, the shell features, and it sets the values of some variables (i.e., system environment). It includes several nodes. The main important ones are summarized below:

- Node <server>: It defines the host and port used by the *Server* service and creates the stacks at start up.
- Node <broadcast>: It configures the broadcast parameters used to execute this service.
- Node <debug>: It defines the debug flags used when adding debug data into the log files.
- Node <putenv>: It is used to specify some environment variables.

6.2.4.2 File variables.xml

This file includes a generic definition of important variables that are used in other XML files (monitor.xml and scheduler.xml). An example is shown below.

```
<variables>
  <IP>
    <SERVER>IP</SERVER>
    <HOME>IP</HOME>
    <SENSORS>IP</SENSORS>
    <TELESCOPE>IP</TELESCOPE>
    <DOME>IP</DOME>
    <CAMERA>IP</CAMERA>
    <POWERS>IP</POWERS>
    <OTHERS>IP</OTHERS>
  </IP>
  <INTERVAL>10</INTERVAL>
  <PDU>
    <ON>1</ON>
    <OFF>2</OFF>
  </PDU>
  <SAI>
    <ONLINE>2</ONLINE>
    <OFFLINE>3</OFFLINE>
  </SAI>
  <OID_POWERS>.1.3.6.1.4.1.318.1.1.12.3.3.1.1.4</OID_POWERS>
</variables>
```

6.2.4.3 File actions.xml

Some actions are specified in this file and can be called by the *Scheduler* and *Monitor* services. It enables grouping some tasks that have to be executed in a subsequent manner. The number of tasks included in a single action is unlimited. This scheme shows how it is implemented:

```
<actions>
  <action>
    <name>myaction1</name>
    *
    * Process nodes
    *
  </action>
  <action>
    <name>myaction2</name>
    *
    * Process nodes
    *
  </action>
</actions>
```

6.2.4.4 File monitor.xml

The tasks executed by the Monitor service are defined in this configuration file. The maximum number of tasks is unlimited. The following example shows the supported scheme:

```

<monitor>
  <name>mymonitor1</name>
  <task>
    <interval>10</interval>
    * Process nodes
  </task>
  <task>
    <frequency>10</frequency>
    * Process nodes
  </task>
  <task>
    <interval>10</interval>
    * Process nodes
  </task>
</monitor>
<monitor>
  <name>mymonitor2</name>
  <task>
    <frequency>10</frequency>
    * Process nodes
  </task>
  <task>
    <interval>10</interval>
    * Process nodes
  </task>
  <task>
    <frequency>10</frequency>
    * Process nodes
  </task>
</monitor>

```

<interval> or <frequency> tags are used to define a time interval or a frequency as a trigger to execute each task. Tasks can also be grouped using the <monitor> node. Each <monitor> node is executed in a separated process that can be suspended and resumed.

The <task> contents must conform the process parser specification described below (Sect. 6.2.5).

6.2.4.5 File scheduler.xml

This file includes the specification of tasks executed by the *Scheduler* service. It has the following structure:

```

<scheduler>
  <name>myscheduler1</name>
  <hash>
    <variable>myvariable1</variable>
    <variable>myvariable2</variable>
    <variable>myvariable3</variable>
  </hash>
  * Process nodes
</scheduler>
<scheduler>
  <name>myscheduler2</name>
  <hash>
    <variable>myvariable4</variable>
    <variable>myvariable5</variable>
    <variable>myvariable6</variable>
  </hash>
  * Process nodes
</scheduler>

```

Each <scheduler> node defines a specific task and requires the <hash> node to specify the variables used to calculate the hash. The task is triggered when the hash changes. Each <scheduler> node is executed in a separated process that can be suspended and resumed.

The <action> and <scheduler> contents must conform the process parser specification described below (Sect. 6.2.5).

6.2.5 Process nodes

The named process nodes allow specifying what commands, evaluations and flow control must use the *Monitor* and *Scheduler* to accomplish the objective. The process nodes are the basic command language of OpenROCS2.0. They are used to build any task that runs the execution of commands, parses the outputs and executes another action depending on the obtained results. Main nodes are described here.

- Node `<shell>`: It defines the command line that can be executed directly into a shell. It is based on the default system shell. It can use variables defined into the `xml/variables.xml` file
Example: `<shell>ping -c 3 -i 1 -W 1 $IP_SERVER | grep received</shell>`.
 - Node `<timeout>`: It defines the maximum time allowed for the command line execution. The execution is aborted when the elapsed time reaches this value. SIGTERM and SIGKILL system commands are used to stop the process.
Example: `<timeout>5</timeout>`.
 - Node `<ontimeout>`: It defines the action to perform when the timeout is reached. It requires the `<timeout>` tag. Example: `<ontimeout>add HK_INTERNET_SERVER_FAIL</ontimeout>`.
- Node `<php>`: This node is used to execute PHP code and retrieve the STDOUT and STDERR for its latter use. Example: `<php>microtime(true)</php>`.

And an example of a periodic task would be as follows:

```
<task>
  <php>microtime(true)</php>
  <frequency>1</frequency>
  <send>update SDP_PHP_MICROTIME=$STDOUT</send>
</task>
```

- Node `<choose>`: It defines a control statement. It can be used to specify complex decision tree structures. The `<choose>`-`<when>`-`<otherwise>` structure is inspired in the XSLT language. The following example shows the supported scheme.


```
<choose>
  <when>
    <eval>$STDOUT4>0</eval>
    <send>remove HK_INTERNET_SERVER_FAIL</send>
    <fromiter>3</fromiter> or <fromsec>30</fromsec>
  </when>
  <otherwise>
    <send>add HK_INTERNET_SERVER_FAIL</send>
  </otherwise>
</choose>
```

 - Node `<when>`: The actions specified in this tag are executed when the evaluation of the `<eval>` tag returns a true value.
 - Node `<eval>`: This tag sets a true or false value to the `<when>` node. Its contents must be a PHP expression that uses all available variables: \$STDOUT (output string of the last `<shell>` execution), \$STDERR (error string of the last `<shell>` execution), variables defined in the `variables.xml` file, available variables in the server (SDP/HK/SYS). Example: `$HK_INTERNET_SERVER_FAIL`.
 - Node `<otherwise>`: An alternate action to the `<when>` statement is defined with this tag.
 - Node `<fromiter>`: It defines the times a related action must be executed before triggering the task. Example: `<fromiter>3</fromiter>`
 - Node `<fromsec>`: A time delay is specified with this node. The system uses the timestamp to control the time elapsed from last execution. Example: `<fromsec>60</fromsec>`
- Node `<action>`: It is used to specify the actions the system executes. The actions can be defined in the `actions.xml` file. It can include functions, procedures and/or macros that can be called simultaneously from different nodes.
- Node `<send>`: This tag defines a direct communication channel with the storage server. It is used to add, update and remove stacks and variables. Example: `<send>add HK_INTERNET_SERVER_FAIL</send>`.
- Node `<log>`: It is used to define the log messages for each process node. These will appear in the `user.log` file. Example: `<log>System running correctly</log>`

6.2.6 Wrapper libraries

Several components (low level software and software applications) need an intermediate routine to interact with OpenROCS2.0. Wrapper libraries are developed as an interface among them, performing the required operations. The components that need these wrapper libraries are listed below:

- TALON and low level software for the telescope, dome and CCD camera control: The routine currently implemented is based on scripts that wrap commands to control the telescope, the dome and the CCD camera by writing and reading the fifos (named pipes) used by TALON as communication channels. The implemented features are similar than with the INDI protocol.
- Dome shutter redundant control: The communication with the FSC1 board for the redundant control requires the translation of shell commands defined in the OpenROCS2.0 XML files to the serial port commands defined in this device.
- Environment monitoring data: The weather and sky quality data have to be transferred to the *Monitor* service using the SNMP standard protocol. A conversion is required for most of the available sensors, except for the TFRM weather station that already uses SNMP.

The development of these wrapper libraries is getting full completion at the moment of writing this Thesis.

6.2.7 OpenROCS2.0 code structure

The directory structure for the software source code is organized as follows:

- root: This directory contains all software needed to run OpenROCS2.0 and the main program script.
 - xml: The software configuration files are stored in this directory.
 - php: This directory contains all the scripts used to implement the OpenROCS2.0 services. It also contains a directory (named *actions*) that stores specific scripts that execute predefined actions (i.e., start, restart, reload, or stop).
 - log: This directory is used by openrocs2 to store the activity register (logs) used to check the correct work and for debug purposes.

Chapter 7

DDROCS Database

The DDROCS database (standing for Dynamic Database for the ROCS) represents a key element in the global structure of the observatory. It plays a fundamental role in the data flow control layer, but it is also important for housekeeping purposes as it contains other information, like the system logging. We give in this chapter the general considerations about the database and a description of its development at the TJO-OAdM.

All the information associated to each task entered into the system for its execution is stored in the observatory database. Then, it is accessed by the software services in charge of managing the task selection, execution and processing, which also store the processing results into the database in a way to maximize the system efficiency. Finally, it provides the structure for easy retrieval of the data to the end user or to the observatory staff supervising the system operations.

7.1 Database analysis

The database responds to the need of having a central archive where all the information of the observatory is stored. It is considered as an alive component of the observatory as it grows with it, and that is the reason why it was called the *Dynamic Database* for the ROCS (DDROCS).

7.1.1 List of requirements

In the initial analysis, we identified the following general requirements:

7.1.1.1 Structure

The database has to be developed to contain different types of data: files and metadata. These are the data levels considered:

- **Task data:** All the proposals, from their general information to the lowest level specification for every particular task, have to be stored in the database. The execution details and the quality control results of every performed task and the data values extracted from the science images are also included in the database, conforming the TJO-OAdM archive. Science observations, calibration tasks and maintenance and testing tasks fall into this block that is mostly devoted to data flow management.
- **Housekeeping and logging data:** The status of the observatory in terms of system health and environment conditions, as well as the processes logging data, have to be stored to evaluate the system performance efficiency and to extract statistics on environment conditions for specific periods of time.
- **Web page utilities:** Some data related to the web-based user interface is also included in the database.
 - **User data:** Any user (including superuser level) must be identified when accessing the user area of the TJO-OAdM web page in order to provide the access permissions according to the profile.
 - **Contents:** Web page contents are also stored in the observatory database.

7.1.1.2 Boundary conditions

DDROCS is a tool used by software applications to store and retrieve data, but it does not perform any computation using these data. Input and output data are then coincident information blocks defined according to the applications that interact with them: OpenROCS, ISROCS, ICAT, system file architecture and a user interface.

7.1.1.3 Interface requirements

DDROCS needs to have an interface defined for several applications:

- **OpenROCS:** The logging files produced by the software infrastructure are included in the database.
- **ISROCS:** The scheduler has to take the list of pending tasks together with their execution constraints from the database.
- **ICAT:** The data processing routine results have to be included in the database.
- **User interface (web):** Data associated with new proposals to be done at the TJO-OAdM and web page contents have to be entered into the database for their later use by other applications.
- **File system architecture** for file storage: The interface with the on-site file architecture have to be defined and implemented to manage file storage.

A common access level is required to homogenize the interaction with the different applications.

7.1.1.4 Development

- **Database management system:** A relational database has to be developed providing high transaction rate, reliability, maintainability, and scalability.
- **Modularity and simultaneous data access:** DDROCS has to be programmed in a way to be accessed by different software applications and simultaneously respond to queries sent in parallel.
- Use of **existing applications:** Existing software applications, with well-known performance and under Open Source license (MySQL, SQLite, etc.) have to be considered.
- **Mirroring:** The database has to be mirrored on two computer servers, at least, with the possibility to define slave-master categories.

7.2 Database design

7.2.1 Data levels

The optimal structure of a database depends on the natural organization of the application data, and on requirements of such applications (which include transaction rate (speed), reliability, maintainability, scalability, and cost). We identified different levels for the TJO-OAdM observatory following these criteria. The list of levels is detailed below, according to the data flow control design and the specific applications that interact with each one. They are sorted from high to low level of human interaction:

7.2.1.1 End-to-end data flow

Data flow processes use the database levels where task specifications and results are stored. Three main levels are defined: Proposals, Projects and Observations.

➤ **Proposals**

It contains the general information of the proposals as specified at the first phase of their submission.

The information included at this level for each proposal is: proposal identifier, principal investigator (PI) data, list of targets, scientific justification, date of submission, acceptance and finalization, and status.

New entries have to be generated through a web user interface and, then, an automatic process is executed to create the proposal identifier and store it into the database. The acceptance date is updated manually after the submission to the observatory TAC and date of completion and status is updated automatically by the observatory control system.

➤ **Projects**

Information specified at the second phase of submission of the proposals.

It includes, for every particular task of a proposal, the detailed list of identifiers (proposal, target group, observation block, object and coordinates RA and DEC) and priorities, execution constraints, status (validation and execution results) and number of successful observations.

Regarding the execution constraints, the parameters specified are: instrument to use and configuration (filters, exposure times), number of iterations, period between iterations, environment conditions (seeing, atmospheric transmission), limits for the observation window, etc.

A user interface program is used to introduce new entries data and to perform their validation. The number of successful observations is automatically updated by the control system.

This data level is used as input by the mid-term scheduler.

➤ **Observations**

The outputs of the data processing application that manages the post acquisition steps are included at this level. A new entry is created just after every task execution with the following data: task identifiers (proposal, target group and observation block, object name and celestial coordinates), date and time of the observation, calibration and analysis status, storage directory and filenames (for the raw image and the analysis results), image quality report, error flags and quality index (value used to decide if an image fulfills user requirements). Images and extracted data files are not stored in the database. Filenames are stored and these are used to generate the links in the web page to let the user retrieve the data in an easy way.

This level is used to compute the completion level of a project, which is a comparison of the number of successful runs with the number required. And when all the projects included in the same proposal are executed, the proposal is considered as closed and a notification can be automatically sent to the proposal PI.

7.2.1.2 Housekeeping

This additional level is necessary to trace the system behaviour by storing significant output data of the applications and the control system:

➤ **Logging**

The entries stored in this level are useful for the evaluation of the global observatory status and to trace the origin of any problem occurred. Those processes that must generate new entries are: process logs for main subsystem applications (web interface, scheduler, data processing, execution task, and housekeeping data), process errors and process alarms. Each process should generate a new file every day and have a directory assigned into the logging level to store it.

➤ **Environment**

Environment data are stored in the database. These are easily accessible by the user interface (web page to external users or remote monitoring done by the observatory staff) and statistical studies can be produced to characterize the weather patterns and to calibrate the system response.

7.2.1.3 Web page

The following levels were designed to support the observatory web page. Their contents are created and updated by the observatory staff.

➤ **Users**

It includes the list of users with login, password and permissions. The latter are used to specify the levels each user is authorized to access and the list of proposals and projects that can be accessed and updated.

The users and accessibility levels are defined in Table 7.1.

Access types	Proposal phases	Proposal evaluation	Proposal data validation	ROCS monitoring	ROCS control
User	•				
TAC member		•			
Superuser	•		•	•	•

Table 7.1: User access levels

➤ *Documents*

It includes the list of private and public access documents: technical notes, press releases, conference proceedings, papers, etc. These entries are mainly used by the observatory web page in order to enable an easy access to the observatory documentation.

➤ *Contents*

The web page contents have to be stored in the database to separate them from the web design and structure. This implementation approach is used to reduce the maintenance costs of the web site.

7.2.2 Database architecture

The database architecture is based on: metadata storage in a relational database, including links to files not included into the database (images, data files); and directories with daily generated files. Below follows a description of both database structures.

7.2.2.1 Relational database

Several applications interact with the relational database, as described before, to store the relevant data and metadata (see Fig. 7.1):

- **Data flow processes:** Proposal and project data introduced by the web-based user interface during task preparation; task scheduling; and task processing, generating the observation data level.
- **Housekeeping:** Environment data collection managed by OpenROCS.
- **Web page programming:** Users, documents and contents are introduced manually by the observatory staff and accessed by the web page when required by the web server software.

A common interface is used in order to minimize errors while updating or retrieving stored metadata. This interface, called the Database Access Level (DAL), includes all the commands that correctly modify or query the database contents. Then, the applications that interact with the database can be programmed in any language and they just have to call the routines that execute the commands in the DAL interface.

7.2.2.2 Daily generated directories

Every day, a new directory is created to store images and data files associated to the executed science observations and calibrations tasks, which are linked with the corresponding task metadata included in the database. Log files are also stored at this directory structure. They all compose a different layer of the database that must be also considered. Directory structure and file naming policies are described in Chapter 11.

➤ *Images and data files*

Different data types and data levels are produced at the observatory and a specific storage and backup policy is defined to store them (see Chapter 11). Database metadata is linked to these files and the integrity of those links must be kept for the on- and the off-site database structure to avoid any error in the archiving and retrieval of data. Database mirroring policies are designed accordingly.

➤ *Logging*

Logging information of all the applications is stored by the control system (OpenROCS) in a daily file. It is managed by the software infrastructure. Applications interfacing with OpenROCS report the necessary information that is automatically stored in the logging directories, located in SUB-ALIS. OpenROCS can manage an optimized distribution of the logging data storage by balancing the use of storage capacity for the available subsystem computers. The latter enables a highly reliable storage for the logging of data, but it is not used in its current configuration that just considers the on-site server for this purpose.

7.2.2.3 Database mirroring

A mirror of the database is located at the observatory site server (SUB-ALIS), whereas the main server is at the IEEC facilities in Bellaterra (SUB-AGER). Some of the contents of the mentioned blocks are added or modified on one server, whereas other contents are added or modified on the other one. Table 7.2 summarizes the contents of the database as well as the location of the generators of these contents.

Automatic synchronization of the database contents is important to transfer the missing information from one server to the other.

This database mirror architecture is necessary to take profit of the system modularity while maximizing the security and efficiency. The off-site database is required because it supports the web page used as interface by the external users to submit proposals and retrieve data that is hosted by the off-site server. Otherwise, users accessing the observatory server would compromise the system safety and also use up the internet connection bandwidth that must be reserved for remote supervision and for data backup. On the other hand, the efficiency of the TJO-OAdM control system is highly improved if the decision making capability is kept at the observatory site, and this is strongly dependent on the database access when managing the telescope operations.

Database level	Data sub-level	Description	Main generation of data	Secondary generation of data
Data flow	Proposals	Proposal general information (Phase 1)	SUB-AGER	SUB-ALIS
	Projects	Proposal execution details (Phase 2)	SUB-AGER	SUB-ALIS
	Observations	Task acquisition details	SUB-ALIS	None
	Observations	Processing results	SUB-ALIS	SUB-AGER
	Observations	Images and data files	SUB-ALIS	None
Housekeeping	Environment	Environment data	SUB-ALIS	None
	Logging	Logging files	SUB-ALIS	SUB-AGER
Web page utilities	Users	Identification of users	SUB-AGER	SUB-ALIS
	Contents	Page contents	SUB-AGER	SUB-ALIS
	Docs	Documents and images	SUB-AGER	SUB-ALIS

Table 7.2: Database contents generation

Fig. 7.1 describes the database architecture and the group of applications that interact with each database level when the system works under nominal operation mode. White boxes indicate unattended processes, whereas blue boxes are human-supervised actions. Red arrows draw the direction of the data synchronization.

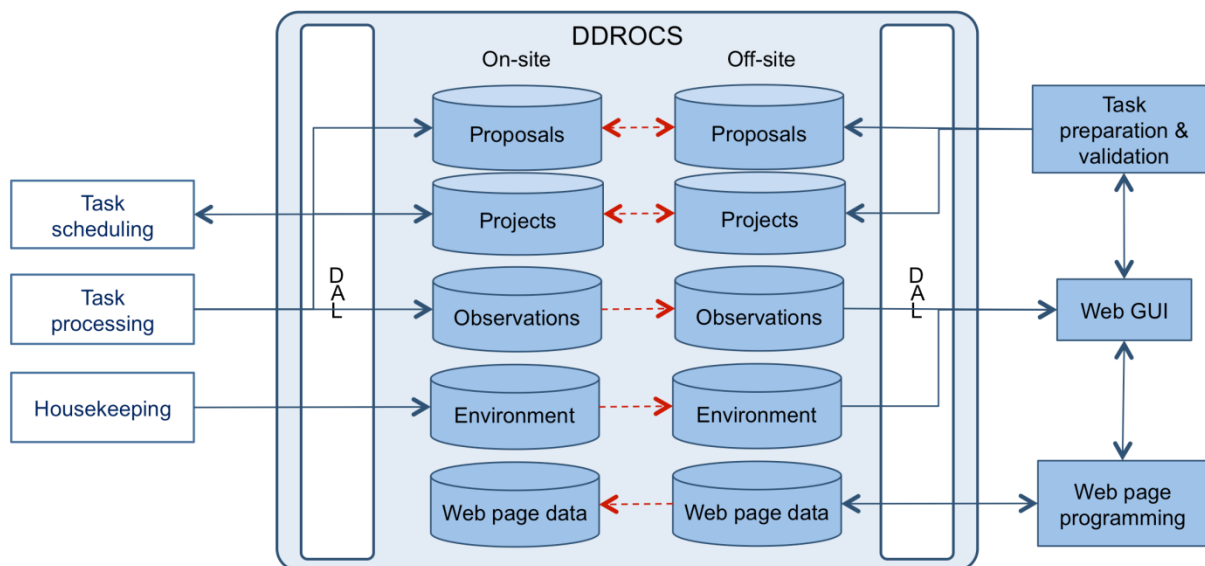


Fig. 7.1: Database architecture

7.3 Database implementation

The implementation of the TJO-OAdM database was structured using the relational model that is characterized by the presence of a common attribute used as a key identifier to link information about the same entry that is stored in different tables.

The management system used is MySQL, widely deployed and regarded as being fast. The database is then updated by different applications using SQL queries, mostly written in Python and PHP. Such programming languages widely support database interfacing.

7.3.1 Relational database

The tables below summarize the structure of the implemented database.

7.3.1.1 End-to-end data flow tables

Some of the fields included at these tables are automatically generated by the application interfacing with the database (web page or task processing). They are labelled accordingly.

Only the MEIA instrument is considered in the current implementation and no instrument option is given on the proposal submission interface. Thus, no instrument selection is stored in the database.

➤ *Proposals*

The fields included in this table of the database are used to store the general information of Phase 1 of the proposal submission. Some submitted data (scientific justification and list of targets) are stored in files created automatically on the file system of the off-site server and they are included in the list of directories to check every time both on- and off-site servers are synchronized. These file names are automatically created by using the proposal code to avoid duplicity at the storage directory and are specified at the corresponding table field.

Fields	Description
ID	Table entry identifier (auto)
code	Proposal code (auto)
pwd	Proposal password (auto)
srTitolPr	Proposal title
tipusobs	Observation science case
soProjcient	Scientific project description
filejust	File name of the scientific justification
resub, noCodi	Previously submitted proposal (yes/no) and old code
soComentarisobservacio	Other general comments about the proposal
srNomIP	PI Name
srEntitatIP	PI Affiliation
soDeptIP	PI Department
soAdrecaIP, soCiutatIP, soCPIP, soPaisIP, noTelefonIP, noFaxIP, positionIP, erCorreulIP	PI contact data
srCoinv	Co-investigator data
source	Source data (when just one target is submitted)
filesources	File name with all sources data (when more than one target is submitted)
sist	Type of observing time (flexible, monitoring, etc.)
soComentobssystem	Observing time justification
skybright, seeing, phot	Sky conditions (brightness, seeing, atmospheric transmission)
timereq	Amount of time required for the entire proposal
subdate	Submission date (auto)
status	Proposal status:

	<ul style="list-style-type: none"> • notconsidered: waiting to be considered by the TAC • rejected: rejected by the TAC • considered: accepted by the TAC • ph2submit: second phase already submitted (auto) • ready: second phase data validated and ready to be scheduled • finished: proposal finished (auto)
TACreport	TAC report
TACdate	TAC report publication date
phase2validation	Validation date of the Phase 2 data
notification	Notification date of proposal ending
results	Link to result webpage. Field no longer applicable: a section at the web page interface is dynamically created in the Status and Results section by entering the proposal code and password (see Chapter 12)
active	The proposal is active or not (yes/no)

Table 7.3: Proposal table

➤ *Projects*

Every new entry in this table includes the detailed information for the execution of a single OB associated to a particular TG of a Proposal (see Chapter 8 for a detailed description of the proposal structure in specific tasks), ready to be used by the scheduler after its validation:

Fields	Description
ID	Table entry identifier (auto)
code	Proposal code (auto)
TGID	TG identifier (auto)
target	Target name and coordinates (auto)
tgpriority	TG priority assigned by the user
OBID	OB identifier (auto)
tepx	Exposure time for the OB execution
filter	Filter for the OB execution (U, B, V, R, I)
binning	Camera binning for the OB execution
defocus	Required camera defocusing for the OB execution (yes/no)
repeats	Number of images for the OB execution
obpriority	OB priority
noninter	Non interruptible OB execution. No longer applicable: OB are defined to be atomic (non interrupted execution)
iter	Number of iterations for the TG execution
noconstr	Scheduling time constraints for the TG execution: no constraints specified
SB	Time scheduling constraints for the TG execution: time or phase interval/s between executions
afterby	Time scheduling constraints for the TG execution: time delay/s after last execution
phase	Time scheduling constraints for the TG execution: phase delay/s after last execution
lunardist	Lunar distance to the target for the TG execution
airmass	Airmass of the target for the TG execution
bias	Number of extra-calibration images (bias) for the TG execution
dark	Number of extra-calibration images (dark) for the TG execution
calibest	List of standard stars for absolute photometry proposals. No longer applicable: standard stars are added by the user as targets included in the proposal execution.
linkdata	TG interdependencies (TG identifiers and time delay between them after the TG execution)

skybright, seeing, phot	Sky conditions for the TG execution
reducauto	Automatic reduction (yes/no). Active by default.
subdate	Phase 2 data submission date (auto)
validation	Data validation status (yes/no)
validationdate	Validation date
ndone	Number of images successfully done (auto)

Table 7.4: Project table

➤ **Observations**

Every new entry in this table summarizes the metadata of an executed task. The information is introduced automatically by the task processing routine that is also responsible for computing the contents of some of the fields. Other fields are copied directly from the image header. Table 7.5 shows the structure of this database level.

Fields	Description
ID	Table entry identifier (auto)
proctime	New entry creation time (auto)
code	Proposal code (auto, from image header)
TGID	Target Group identifier (auto, from image header)
target	Target name and coordinates (auto, from image header)
OBID	Observation Block identifier (auto, from image header)
obstype	Observation type: bias, dark, flat or science (auto, from image header)
texp	Exposure time (auto, from image header)
filter	Filter used (auto, from image header)
binning	Image camera binning (auto, from image header)
focus	Telescope focus (auto, from image header)
JD	Image acquisition time (JD) (auto, from image header)
lunardist	Lunar distance (auto, from task processing)
airmass	Target airmass at the acquisition time (auto, from image header)
skybright	Sky brightness at the acquisition time (auto, from task processing)
seeing	Sky seeing (auto, from task processing)
phot	Atmospheric transmission (auto, from task processing)
reducauto	Automatic reduction (auto, from task processing)
IMGQS	Image quality summary (auto, from task processing)
IMGQF	Image quality data: File (auto, from task processing)
IMGQP	Image quality data: Photometry (auto, from task processing)
IMGQA	Image quality data: Astrometry (auto, from task processing)
IMGQC	Image quality data: Obs.Conditions (auto, from task processing)
IMGQE	Image quality data: processing Errors (auto, from task processing)
dname	Storage directory name (auto, from task processing)
fname	Raw image file name (auto, from task processing)
logfname	Log file name (auto, from task processing)
trlfname	Data file name (auto, from task processing)

Table 7.5: Observation table

7.3.1.2 Housekeeping

➤ *Environment*

Environment data are obtained automatically from the housekeeping monitor of the OpenROCS control system and stored in a table with the fields described here. Every data point is complemented also with the acquisition JD that is used by OpenROCS to detect sensor errors.

Fields	Description
datetime	New registry date (auto)
temperature_davis_indoor, datum JD	Davis weather station: temperature at the telescope control room (auto)
temperature_davis_tjo, datum JD	Davis weather station: temperature inside the dome (auto)
temperature_davis_tower, datum JD	Davis weather station: temperature outside the building (auto)
temperature_smc, datum JD	SMC weather station: temperature outside the building (auto)
humidity_davis_indoor, datum JD	Davis weather station: humidity at the telescope control room (auto)
humidity_davis_tjo, datum JD	Davis weather station: humidity inside the dome (auto)
humidity_davis_tower, datum JD	Davis weather station: humidity outside the building (auto)
humidity_smc, datum JD	SMC weather station: humidity outside the building (auto)
pressure_davis_tower, datum JD	Davis weather station: pressure (auto)
pressure_smc, datum JD	SMC weather station: pressure (auto)
windspeed_davis_tower, datum JD	Davis weather station: wind speed (auto)
windspeed_smc, datum JD	SMC weather station: wind speed (auto)
winddir_davis_tower, datum JD	Davis weather station: wind direction (auto)
solarrad_smc, datum JD	SMC weather station: solar radiation (auto)
solarpos_sunposition, datum JD	Solar altitude (auto)
rain_davis_tower, datum JD	Davis weather station: accumulated rain (auto)
raindetect_raindetector, datum JD	Rain detector status (true, false) (auto)
esfield_previstorm, datum JD	Storm detector: electrostatic field value (auto)
crc32, crc32_ok	Check code to detect transmission data errors (auto)

Table 7.6: Environment table

7.3.1.3 Web page

Users and Docs tables are summarized here. The contents table was not used in the initial web page implementation, but it is in the new upgraded one, as described below.

➤ *Users*

User login information is stored in this table. User data in a new upgraded version of the database (see below) is extended to include all the personal information (affiliation, address, phone, etc.) required to submit a proposal.

Fields	Description
index	New registry index (auto)
user	User login
email	User email
name	User full name
password	User password
profile	Access permissions

Table 7.7: Users table

➤ Documents

Any document or image included in the web page is introduced in this table, where related information is stored. Some fields only correspond to technical document files, whereas others correspond to images shown in the web page sections. Some fields may be empty.

Fields	Description
index	New registry index (auto)
date	Document date
author	Document author
title-cat	Document title (Catalan)
title-es	Document title (Spanish)
title-eng	Document title (English)
reference	Document reference
abstract-cat	Document abstract (Catalan)
abstract-es	Document abstract (Spanish)
abstract-eng	Document abstract (English)
url-cat	Document storage link (Catalan version)
url-es	Document storage link (Spanish version)
url-eng	Document storage link (English version)
doctype	Document type: <ul style="list-style-type: none"> • Technical report: technical documents or conference proceedings • Press release document • News: TJO-OAdM related news • Image: astronomical images acquired at the TJO-OAdM • Image_oadm: images of the TJO-OAdM facilities and instruments
date_cad	Date when published news related with the observatory are not shown as "latest news" any longer
url_smallimg	Image (thumbnail) storage link
url_bigimg	Image (high resolution) storage link

Table 7.8: Documents table

7.3.1.4 Database upgrade

A new database structure for some of the mentioned tables is being developed by the engineering team at the time of writing this Thesis.

Design architecture is maintained and based on the same levels described in the database design. However, the implementation structure, based on multiple tables, is modified to improve proposal data storage and management mainly for data flow sub-levels. A new user interface for proposal management and a new scheduler implementation motivated this upgrade. The new implementation, also based on MySQL, provides more flexibility and robustness and it is self-documented using Doxygen, a software documentation generator.

The web-based user interface has also been upgraded to a new implementation developed using the PHPCake¹ framework, based on a different development paradigm that uses three different layers: the web code, the web format and the web contents. The web contents are stored in the observatory database (web page level) adding new tables to the database structure. Modification of the submitted data is enabled at any time with this new web interface up to the TAC evaluation. And common information at different proposals can be recovered by the user in order to minimize the time spent to submit them (i.e., PI personal information).

7.3.2 Daily generated directories

Log files are stored in daily generated directories, as well as images and data files obtained during task execution processes. They all follow the storage and backup policies (see Chapter 11).

¹ <http://cakephp.org/>

7.3.3 Database mirroring

On- and off-site relational database synchronization is done using MySQL features and by configuring this software to work with both databases defined as masters. Information added or modified in any of these databases is always kept during the synchronization process. Directories generated every day to store log, image and data files are also synchronized on both servers following the database mirroring design.

Chapter 8

Observation Preparation

Every task to be considered by the execution system must be completely defined and inserted into the TJO-OAdM database to be subsequently used by the scheduler as already described in the Operational Design section (Sect. 3.1). We give in this chapter a description of all the details required for the specification of tasks and the process to submit them to the observatory. They are also derived from the contents of the Operational Design. Important concepts are introduced and will be the basis for a correct execution of the data flow management modules described in next chapters: scheduler, task execution, and data processing.

8.1 Tasks considered and insertion process

Two main control modes are defined for the TJO-OAdM: nominal and engineering. The nominal mode of control foresees the execution of both science observations and calibration tasks, whereas the engineering mode is used for maintenance and testing operations (see Sect. 3.1.4 for more details).

8.1.1 Task insertion

Most of the tasks are suitable to be prepared a priori by entering their execution constraints into the observatory database, although only those that are evaluated by a TAC must follow all the proposal submission phases that will be detailed next. The latter applies to all science observations except those defined for engineering purposes (i.e., instrumentation testing).

The general approach for the task preparation is to maximize the number of tasks defined using the proposal submission tools. This enables a homogenization of the task management processes and, also, the use of the scheduler (mid- or long-term) as a general planning method to determine the best time for the execution of pending tasks. Maintenance tasks, for instance, may be defined following the same execution pattern as science operations, ensuring their optimum execution according to the maintenance plan and with a good planning for the actions the observatory staff must do on-site.

Task definition is performed by different user profiles: nominal science observations are usually required by external users, whereas calibration, maintenance and test tasks are introduced into the system by staff members (superuser).

A graphical interface centralizing the injection of task constraints is the best tool to ensure system efficiency. We designed and partially implemented a web-based observation management tool through which the users can propose their science programs (phase 1), prepare the observations (target data, constraints, observing modes, integration time) if the program is approved (phase 2) and retrieve their data products upon successful observation (phase 3). This can be also used to submit execution constraints for other kinds of tasks. An extended description of this tool is given in Chapter 12.

8.2 Science task preparation

Scientists from research institutions around the world can submit a proposal to the TJO-OAdM TAC. There is a continuous open call for proposals. These are evaluated on-line by the TAC members.

Any time a new proposal to execute science observations is submitted to the TJO-OAdM it follows the scheme described in Fig. 8.1.

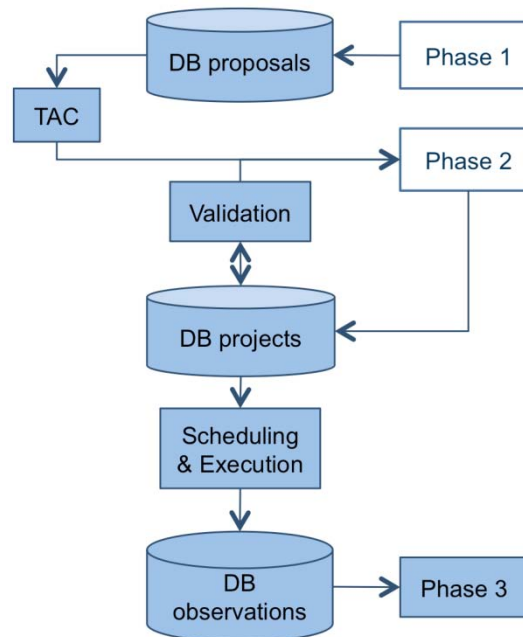


Fig. 8.1: Preparation process for science observations

8.2.1 Phases for new proposal submission

A new proposal is submitted in two phases. In Phase 1, the proposers submit a scientific justification and observation summary for peer review. The proposer must justify why the science case is critically important and why it requires TJO-OAdM and must ensure the proposal feasibility. It is the responsibility of the proposer to ensure that the proposed observations are technically feasible. Proposals that are not technically feasible are rejected.

The TAC recommends a list of programs with associated priorities to the TJO-OAdM observatory for preliminary approval and implementation. The TAC decision is notified to the proposal PI.

If the proposer is awarded time on the TJO-OAdM, the proposal can proceed to Phase 2. In Phase 2, proposals must provide complete details of the observations. This allows the TJO-OAdM to conduct a technical feasibility review, and to schedule and obtain the actual observations. Programs are not approved fully until after submission of an acceptable Phase 2 program.

Each proposal must have only one Principal Investigator (PI). Any other individuals who are actively involved in the program should be listed as Co-Investigators (Cols). The PI is responsible for the scientific and administrative conduct of the project, and is the formal contact for all communications with TJO-OAdM.

After submitting the proposal, the PI receives the automatically created proposal code and, with it, can access a specific web site location that includes the information sent in Phase 1 and the submission status. The proposer can use this page to track the evolution of the proposal (TAC considerations, Phase 2 data, etc.) and to access the data and the images obtained. All this information is stored in the observatory database.

8.2.2 Proposal storage in the observatory database

The TJO-OAdM database contains different levels according to the global control design, as described in Chapter 7. Different levels corresponding to the proposal submission phases are called DB Proposals, for Phase 1 contents, and DB Projects, for Phase 2 contents.

Proposal information submitted in each phase is stored in the database in a way the user can recover and modify it at any time until it is taken into consideration by the TAC (Phase 1) or it is validated by the observatory staff (Phase 2). Once both phases have been completed and contents are finally frozen, those tasks labelled as approved for execution will be included in the pool of pending tasks to be scheduled. The observation preparation phase is finished at this point.

8.2.3 Preparing the proposal

Some important concepts are introduced here regarding those steps must be done to prepare and submit a proposal for the TJO-OAdM.

8.2.3.1 Observation types, target specification and observing time distribution

A definition of some concepts is required for the proposal preparation: observation types (flexible, phase-critical, etc.), specification of targets (fixed, generic and solar system) and distribution of the observing time (OB, TG). We already introduced them in Section 3.1.

8.2.3.2 Exposure Time Calculator

The observatory is provided with an empirical exposure time calculator (ETC) that is used for different purposes:

- **Proposal preparation:** Users submitting a proposal can use it to estimate the required exposure time to define this execution constraint for all the targets included in their proposals.
- **Scheduler:** Scheduler night planning is based on the exposure times defined for each task, but it does not correct it as a function of current conditions at the time of execution. Air transmission and seeing have a big impact on the signal level measured at the instrument, but task execution constraints are considered fixed and unchangeable at a high priority. A different approach could be taken if many images had lower signal to noise ratio than expected and failed the data quality control. If many images had to be repeated to reach the required quality, a dynamic determination of the exposure time as a function of environment conditions may be implemented.
- **Quality control (data processing routine):** The ETC is used for the image photometric quality control when comparing catalogue magnitudes with the measured ones at the processed image. Catalogue magnitudes are converted to instrumental magnitudes by using the ETC and the environment conditions at the acquisition time (seeing, air transmission, airmass).

The ETC is programmed by adjusting a power law to exposure time vs magnitude for different filters and seeing conditions based on real data available from images taken at the observatory. No further improvements are foreseen, although an analytical function could be used in the future.

8.2.3.3 Phases

A high level step-by-step guide to a Phase 1 and Phase 2 proposal is shown in Fig. 8.2 and Fig. 8.3 and explained below:

➤ *Phase1 roadmap*

This is the sequence a user must follow to prepare the science observation for a specific proposal:

1. Determine the observation type (exoplanets, solar system, calibration, engineering, etc.)
2. Review the observatory constraints to determine if the observations are feasible (pointing constraints, bright object constraints, etc.)
3. Determine the required observing conditions to get the expected data precision. Sky brightness, seeing and photometric conditions must be specified.
 - There are two sky brightness levels, bright and dark. Bright time is defined as the time when the Moon is above the horizon and is between waxing and waning gibbous phases (more than 50% visible). Dark time is whenever the Moon is below the horizon or is between waning and waxing gibbous phases (more than 50% visible).
 - Seeing is split into three bands: Good (<0.8 arcsec), average (0.8 - 1.3 arcsec) and poor (1.3 - 3.0 arcsec). No observations are made when seeing is worse than 3 arcsec. Scheduling is always performed on the most recently available estimate of the V-band seeing at the Zenith.
 - Photometric conditions: Time allocation can be requested for Photometric and Non-Photometric conditions. If Non-Photometric time is required, the observations can be scheduled at any time. Photometric standards must be scheduled as different targets in the proposal.
4. Determine any observation constraints that could impact scheduling:
 - Flexible time: Observations can be carried out at any time during the semester when the conditions are appropriate.
 - Monitoring time: Observations of an object are required at a regular interval.

- Fixed time: Observations will only be carried out at a specified time, or not at all. This kind of constraint is suitable for occultations, or observations that must be simultaneous with a scheduled observation on some other facility. No more than 10% of time awarded by any panel may be fixed time.
5. Determine the required exposure times to complete the observations for a proposal. The following steps must be taken into account to calculate the resources required: define the execution pattern of the OBs (instrument setup, number of exposures, exposure times, etc.), define the number of iterations and the scheduling requirements that complete the TG specification and repeat these steps for each TG; finally, compute the time required for target acquisitions and instrument overheads. The following aspects must also be taken into account:
- Pointing and tracking: The current pointing performance of the TJO-OAdM is 10 arcsec RMS according to commissioning results.
 - Calculation of the time required: Use the TJO-OAdM Exposure Time Calculator (ETC) to determine the exposure time of each target, then add the following overheads: acquisition time, time to slew the telescope and dome; readout time; time to take special calibration images.

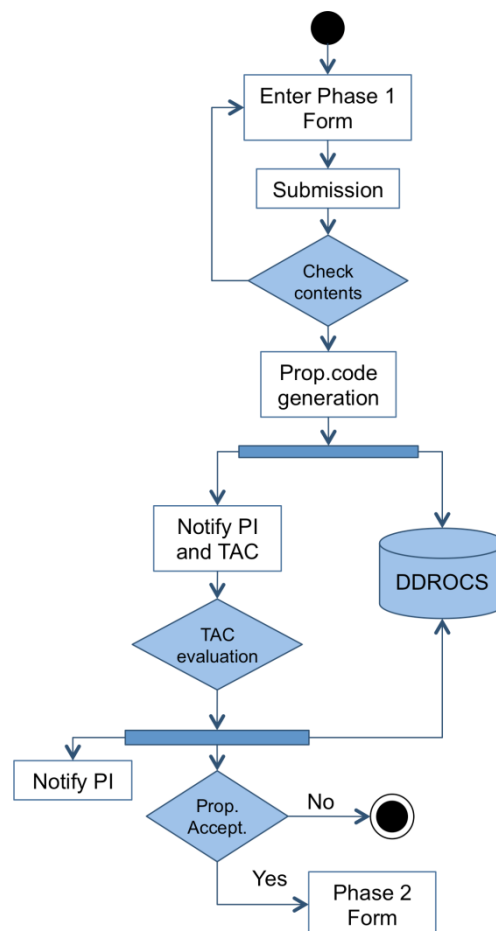


Fig. 8.2: Phase 1 proposal submission scheme

When this sequence is finished, proposal information must be filled using the web page interface: general information, investigators, source data (targets), observing conditions and observing time request. The proposal can then be submitted. Finally, the user can modify it until it is blocked to be evaluated by the TAC. The proposer can use the web page to track the evolution of the proposal (TAC considerations) and to enter Phase 2 data once execution time is allocated.

➤ Phase 2 roadmap

Phase 2 and a final revision by the TJO staff is the latest step before delivering the proposal data to the TJO-OAdM scheduler system. The Phase 2 is a transformation of an accepted Phase 1 proposal into a list of observations with the detailed information needed for implementation and scheduling of the science program.

This is a high level step-by-step guide to writing a Phase 2 proposal (Fig. 8.3):

1. Complete the remaining proposal information, if necessary
2. Complete the remaining general observation constraints, if necessary
3. Define the observation strategy (or TG) for each target. A new TG code is assigned. More than one strategy is possible for the same target, just selecting it again. A full project is finally composed of the sum of TGs for all targets in the proposal. The next steps specify a single TG:
 - a) Select a target from the list of objects submitted in the Phase 1
 - b) Select the instrument
 - c) Define the list of observations included in the OB: it creates pattern specifications for the target observation. It must include the following information for each entry:
 - o Exposure time (the TJO-OAdM ETC can be used)
 - o Filter
 - o Binning
 - o Repeats: number of exposures.
 - o Priority: it compares the observations to prioritize inside the same OB.

Such a set of parameters shall be added a number of times to complete all the required setups to fulfill the observations required. OBs are atomic, so they are planned to be executed without interruptions. Gaps between exposures may still be necessary to allow time for activities which are necessary to set up for the next exposure, such as readouts and reconfigurations.
 - d) Define the number of iterations of the OB.
 - e) Calibration images: Extra bias and dark images are permitted. The number of required images must be specified, and also the exposure time for the darks.
 - f) Priority: It compares the TGs to prioritize within a proposal.
 - g) Define the scheduling requirements:
 - o *No constraints*: Setting this requirement means that there are no scheduling constraints on the OB
 - o *Between*: Enter the range of dates or phases that all exposures defined in the OB must be observed within. If multiple Betweens are required or acceptable, then enter all values. When adding a new Between, select the option AND if it is required (observations must be made in all ranges) or select the option OR if it is acceptable (observations must be made in any of the ranges). Those requests whose exposure start in a specified phase range (<number1> to <number2>) of a periodic variation must take into account these considerations: <number1> and <number2> should be between 0.0 and 1.0; the Period <time> Zero-Phase (HJD) <date> used in the calculation must be entered at this Phase 2 step. With short periods, the phase range (difference between <number1> and <number2>) should be made as wide as possible to make scheduling easier.
 - o *After observation by*: Enter any timing or phase requirements between OBs. Timing or phase requirements between observations within an OB do not need to be specified. This is intended to capture repeated visits with spacing of multiple minutes, hours, days or greater.
4. Define links with other TGs: if one of these TGs is scheduled, then, the current one will be scheduled aswell, following one of these options:
 - o *Non-interruptible sequence*: A container can be defined including all those TGs that must be executed on a sequence. Once started it must continue until the last one is done. The TG included in a sequence must be specified using the TG code. It is also used to specify when and how to observe special standard stars for absolute photometry calibration for a particular TG.
 - o *After observation by*: Enter any timing requirements between TGs (during the same night, time interval)
 - o *No constraints*
5. Define the data delivering requirements: Data retrieval is done through the observatory web page, although some other options are considered.

When the user has finished these steps, the proposal information must be filled using the Phase 2 web page interface and the proposal can be submitted. The proposer can use the same page

mentioned in Phase 1 to track the evolution of the proposal (Phase 2 data validation, execution status, etc.) and to access the data and the images obtained.

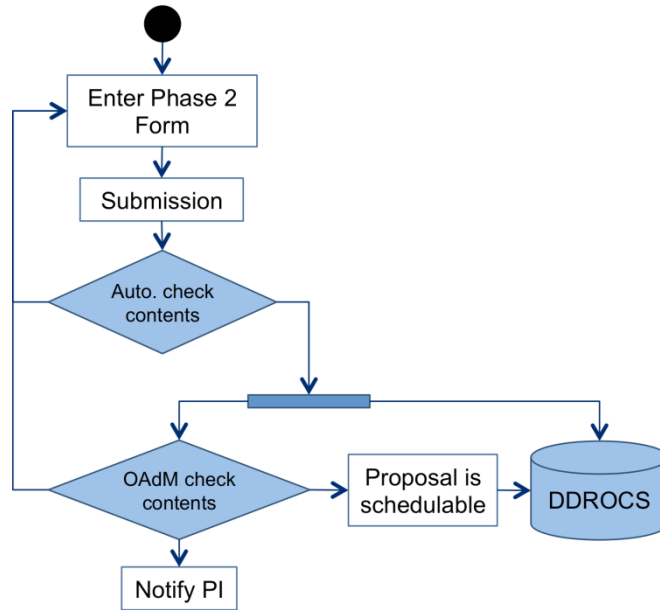


Fig. 8.3: Phase 2 proposal submission scheme

8.2.4 Proposal contents

At the end of the proposal preparation and submission phases, the constraints defined for each science observation and stored at the observatory database are those summarized in Table 8.1. TGs and OBs can have multiple entries, each one identified with a unique code.

Phase 1 – Proposal general data		
Proposal title	Title	
Principal Investigator	Person in charge of the proposal. Co-investigators are also provided.	
Targets	List of targets with coordinates and magnitudes	
Observation type	Science (and sub-types), calibration, engineering	
Observing conditions	Conditions of seeing, sky darkness, atmospheric transmission	
Observation scheduling constraints	Flexible time, monitoring, etc.	
Phase 2 – Proposal execution definition		
TG	ID (nnn)	Target name
Instrument	MEIA (ARES)	
OB	ID (nnn)	
	Exposure time (s)	
	Binning (n x m)	
	Filter (U, B, V, R, I)	
	CCD shutter (open/ closed)	
	Repetitions (frames)	
	Priority (1-10)	
Iterations	1, 2, ..., unlimited	

Time scheduling	No constrain, Between (Time or phase intervals), After by (Time distance)
Celestial object distance	Sun altitude (deg), Moon distance (deg)
Environment conditions	Sky transmission, seeing, darkness
Expected quality	Standard TJO-OAdM quality
Priority	1-10
TG interdependency	TG identifiers and dependency type

Table 8.1: Proposal Phase 1 and Phase 2 for science observations

8.2.5 Proposal validation

An automatic validation of the Phase 1 and Phase 2 submission parameters is done for those values that can be automatically revised thanks to the implementation languages used to program the user interface tool, as it is described in Chapter 12. Number formats, for instance, can be checked at any step, or the total exposure time involved in the tasks defined in Phase 2 can be cross-matched with the accepted time according to TAC approval report.

Observation data are finally validated by the observatory staff as a last step before letting the scheduler routine considers any OB included in a proposal. Validation includes the revision of those observation constraints that cannot be checked automatically and that must agree with the proposal characteristics approved by the TAC.

8.2.6 Data delivery

All science and calibration data are placed in the TJO-OAdM database, accessible through the observatory web page. Science data become immediately available to the proposal PI (Phase 3). These data may be retrieved after the PI has registered as an archive user and are normally proprietary for a period of one year. On average, the science data from TJO-OAdM are processed through the production pipeline and into the on-site database within several seconds after observation by the telescope. Then, it takes several minutes to be transferred to the off-site backup. Some data may take as long as some hours to five days, depending on the off-site backup generation frequency. The observer is notified by e-mail when the first datasets reach the archive and is provided with Web tools to track a visit's completeness and to retrieve the data generated by the pipeline. The time required for retrieving data from the Archive is typically a few hours.

8.2.7 Proposal management tool interface implementation

A web-based submission tool is implemented and running and it is used by the users to prepare their proposals, but also to insert any other kind of task. It can be accessed through the observatory web page. It is programmed using PHP, AJAX and MySQL languages. Its design and implementation is described in Chapter 12.

8.2.8 Proposal Confidentiality

Proposals submitted to TJO-OAdM are kept confidential. For accepted proposals, the scientific justification section of the proposal remains confidential, but other sections become publicly accessible, including PI and Col names, project titles, abstracts, description of observations, special scheduling requirements, and all target names and exposures. Phase 2 programs submitted for approved proposals never become publicly accessible.

8.3 Calibration tasks

Calibration tasks must be performed to enable high precision data extraction and a proper analysis of the science images.

8.3.1 Calibration tasks to reduce science images

We discussed in the Operational Design description (Sect. 3.1) the effects that have to be characterized to calibrate the science images acquired with the MEIA instrument: bias current, dark current, flat fielding, shutter and bad pixels effects. Their execution constraints (including periodicity) were also given.

The extra calibration images were also introduced. User can define them (only Bias and Darks) if the default set of calibration images obtained do not fulfill his requirements. The extra images required are scheduled just after the acquisition of the science frame.

8.3.2 Photometric standards

The observation of photometric standard stars may be necessary to perform absolute photometry programs. A generic execution pattern for night standardization was initially designed, although it was finally discarded as a consequence of:

- **Scheduling complexity:** Night standardization involves the observation of different targets (magnitudes and photometric colors) at different altitudes and at different time all over the night to establish the correction coefficients to transform instrumental to absolute magnitudes. This sequence has to be executed in those nights when a proposal that requires absolute photometric data may be scheduled. But this approach diminishes the system efficiency, especially when the targets of the proposal are not finally scheduled for any reason.
- **Ad hoc execution for each proposal:** Astronomers usually prefer to define their own standardization sequence by choosing the stars, the observation constraints, etc.

Because of the inherent complexity, specifications for the observation of standard stars for absolute photometry proposals were transferred to the proposal submitted by each investigator. The PI could include them into the proposal in a straightforward manner, as interdependency between TGs was already foreseen in the second phase of the proposal submission process. Night standardization can be, then, specified with OB execution constraints and interdependencies among TGs. The user submitting the proposal has only to be aware of including a complete list of standard stars at the list of targets to observe in order to have them available in the proposal second phase definition. Supplementary standard stars can be provided to ensure having a minimum number of them available during all the nights, although the required execution time is assigned to observe only the minimum set of them.

8.3.3 Preparing the basic set of calibration images

A basic set of calibration images (bias, dark and flat) is defined to be acquired every night. Default run execution is defined to be done twice per night during twilight hours, although bias and dark images can also be taken during night hours if any event or alarm aborts any task to be done with the dome shutter in the open position. In any case, twilight ending time determines the moment the system computes the master calibration frames (preliminary at evening twilight and final at morning twilight) by using the available calibration images at the storage directory, being acquired during twilight or night time.

The basic set of calibration images is introduced at the TJO-OAdM database by the observatory staff as a single proposal and following the two aforementioned phases. The execution constraints are defined in the same way as a science observation, although it is submitted without having to be evaluated by the TAC. This proposal for calibration tasks is also stored in the TJO-OAdM database and it is defined to have no ending time.

The proposal contents for the basic set of calibration images are defined in Table 8.2. Some comments apply:

- **Exposure times** were defined during the commissioning phase.
- A different **TG is defined for every calibration image type:** Bias, Dark and Flat.
- **Time scheduling requirements** (a time interval of 2h between iterations) ensure two iterations will be done at least for each TG. More iterations will be done during the night only if there are no science observations of higher priority or in case of bad weather conditions.
- **Celestial object distance** indicates calibration images can all be acquired during both twilight and night time. Starting (evening) and finishing (morning) when the Sun is at 0 degrees altitude ensures there is enough time to acquire the full set during the twilight. Calibration tasks are the only ones enabled during twilight, except for special science observations.
- **Flat field images for the U J-C** photometric filter have a lower priority, because the instrument sensitivity is not optimized for this wavelength range and any task to be done with this filter is very time consuming and reduces the system efficiency. Any proposal requiring this filter is, therefore, very unlikely to be accepted and it will have, in any case, a low execution likelihood.

Phase 1 – Proposal general data					
Proposal title		Calibration images			
Principal Investigator		TJO-OAdM			
Targets		Bias, Dark, Flat			
Observation type		Calibration			
Observing conditions		No restriction			
Observation scheduling constraints		Flexible time			
Phase 2 – Proposal execution definition					
TG		001			Bias
Instrument		MEIA			
OB		001			
Exposure time (s)		0			
Binning		1 x 1			
Filter		none			
CCD shutter		closed			
Repetitions (frames)		10			
Priority (1-10)		5			
Iterations		unlimited			
Time scheduling		After by			2h
Celestial object distance		Sun altitude (deg)			< 0
Environment conditions		Humidity inside the dome (%)			< 90
Expected quality		Standard TJO-OAdM quality			
Priority (1-10)		5			
TG interdependency		None			
TG		002			Dark
Instrument		MEIA			
OB		001			
Exposure time (s)		150			
Binning		1 x 1			
Filter		none			
CCD shutter		closed			
Repetitions (frames)		10			
Priority (1-10)		5			
Iterations		unlimited			
Time scheduling		After by			2h
Celestial object distance		Sun altitude (deg)			< 0
Environment conditions		Humidity inside the dome (%)			< 90
Expected quality		Nominal TJO-OAdM quality			
Priority (1-10)		5			
TG interdependency		None			
TG		003			Flat
Instrument		MEIA			
OB	001	002	003	004	005
Exposure time (s)	2	2	2	2	20
Binning	1 x 1	1 x 1	1 x 1	1 x 1	1 x 1
Filter	I	R	V	B	U

CCD shutter	open	open	open	open	open
Repetitions (frames)	5	5	5	5	5
Priority (1-10)	5	5	5	5	3
Iterations			unlimited		
Time scheduling			After by		2h
Celestial object distance			Sun altitude (deg)		< 0
Environment conditions			Humidity inside the dome (%)		< 90
Expected quality			Nominal TJO-OAdM quality		
Priority (1-10)			5		
TG interdependency			None		

Table 8.2: Proposal Phase 1 and Phase 2 for the basic set of calibration tasks

Chapter 9

ISROCS, the Observatory Time Scheduler

The large flexibility required for the TJO-OAdM operation is a consequence of its multipurpose science case (Sect. 1.2.2) and poses a challenge concerning the unattended control and the scheduling of observations.

The scheduler represents a fundamental element to fully achieve an intelligent response at any time and to optimize the use of the facility. Its main purpose is the allocation of multiple tasks to one single telescope, while minimizing the total execution time and maximizing the scientific return of the project. It receives real time housekeeping data from the control system and accesses the observatory database, where all the pending tasks (observations, maintenance operations, etc.) are stored. It, then, selects the most suitable task and delivers it to the control system for its execution.

The scheduler mainly interacts with OpenROCS, among other subsystems, and it is also used for planning the periodic maintenance operations, special testing, and system calibrations.

A description of the Intelligent Scheduler for the ROCS (ISROCS) is presented in the sections of this chapter.

9.1 Scheduling task overview

Scheduling is a task allocation issue, where the execution is conditioned by one or more constraints. Research in this field began in the mid-50s and has been developed in multiple areas given the horizontal nature of the applications (e.g. Verderame et al. 2010). The classic problem of scheduling is the job-shop problem (JSP), where N ideal tasks must be assigned to M identical resources (machines, processors, etc.), while minimizing the total execution time (Mora 2008).

Studies on dynamic scheduling have been done in the recent years. This case computes the best planning process while introducing new tasks or new constraints and / or a change in the status of the initial constraints. In most cases there is no single best approach to solve the planning system and, therefore, various mathematical algorithms are usually considered (Pinedo 1995).

9.1.1 Scheduling for astronomical observatories

Astronomical observations require specific conditions for each particular case, such as the instrument to use, the sky visibility or brightness conditions, the best time for the observation, etc. All this information, together with the scientific objectives, is included in the proposals that astronomers submit to an observatory, for an observation time request (see Chapter 8). A priority is assigned to the approved proposals and this is used to select one of them in case other proposals could overlap in time. Observations of the proposals must be scheduled for execution under the specified conditions, once accepted and prioritized. This programming is carried out by the scheduler, this being a person skilled in the sorting criteria to apply, an automatic system based on the use of computer algorithms or a combination of both (Johnston 1991, Swanson et al. 1994).

Proposal scheduling for astronomical observations is an example of multi-objective problem, since different factors must be optimized: maximize the scientific return, minimize the time between executions, tune the exposure time for each observation in terms of environmental conditions, etc. The solution to this problem is of inherent computational complexity: the number of solutions grows exponentially with the number of targets and there is no algorithm that can avoid the implicit or explicit enumeration of all solutions. Best scheduling of astronomical observations is included within the class of problems NP-hard (Garey 1979). There are many mathematical tools to solve the scheduling issue: from simple heuristics to more complex genetic algorithms or neural networks. Gómez de Castro et al. (2003) published a good description of the problem and possible solutions. A different approach to solving the problem is given by Granzer (2004), who analyzes various solutions to use in robotic observatories. The advantages of using a dispatch algorithm, based on the calculation of the Figure of Merit (FOM) for each observation, are presented in comparison with queue or optimal scheduling. Dispatch scheduling is the one chosen to be used at the TJO-OAdM observatory, as described below.

9.1.2 Scheduling cycles

Three kinds of cycles are commonly used when defining the scheduling of tasks at astronomical observatories: long, mid and short term, the latter being responsible for the dynamic response of the system. The problem to be solved by the scheduler can consist of thousands of observations per cycle, although not all objects are visible at all times. The long-term planning usually defines the engineering tasks (maintenance and calibration) and the selection of proposals for an entire cycle (i.e., six months), based on factors such as visibility above the horizon or the Moon phase, to compile the list of feasible operations to be executed. Oversubscription is permitted for the full cycle in order to ensure the complete use of available time. The mid-term scheduler selects the proposals suitable to be observed on the following night or week based on the same factors and it may be used to define the system configuration (i.e., instrument). And, finally, the short-term or dynamic scheduler selects the next best candidate to be observed.

The selection of an observation depends on external factors specified in the submitted proposal, and it is the scheduler who decides the best observation based on immediate factors. Not all observations are executed during the planned cycle as a consequence of using dynamic scheduling.

The probability of a project to be carried out depends on its scientific priority, but also the conditions for execution. Such priority may vary over time; it may be greater as the conditions for observation become unfavorable.

9.1.3 Unattended planning mode

The optimal use of mid- and short-term schedulers is usually obtained when combined with unattended operation. Thus, the sequence of tasks can toggle between observations of various proposals without the prejudice it would represent for the astronomer if he/she was at the observatory. Under unattended control, dynamic schedulers maximize the efficiency as the system response is faster and more standardized, optimizing the use of the observatory time. Many robotic observatories do not incorporate this feature in the control system and run a static sequence of observations, which is aborted, but not modified, when environmental conditions change or temporal errors arise in the system control. In this scheme, a variable priority policy is not applied and this does not optimize the scientific return of the instrument. The latter is not true on those projects that can achieve their science goal just executing a constant scanning of the night sky: they do not need such a versatile tool. Two examples of multipurpose observatories that do apply dynamic schedulers are the LT (Fraser 2004) and the STELLA (Granzer et al. 2010) telescopes. The use of automatic dynamic schedulers opens also the possibility of integrating the observatory into telescope networks, such as the Heterogeneous Telescope Network (White et al. 2007).

9.2 Scheduler analysis

The scheduler is one of the most important elements at the end-to-end data flow layer at the TJO-OAdM. We specified and considered the following list of general requirements when designing the scheduler for the TJO-OAdM observatory.

9.2.1 List of requirements

The scheduling package has to fulfill the general requirements presented in Sect. 2.3 and the design of the observatory task selection given in Sect. 3.1. The list of requirements specific for this package is summarized below.

9.2.1.1 Functional requirements

➤ **Operation mode**

The scheduler has to select tasks to be performed at the observatory working on the MEIA operation mode. New modes could be considered at the TJO-OAdM in the future (MEIA with autoguiding system and the ARES spectrograph) and they should be added in the scheduler routine without extensive effort thanks to the system modularity.

➤ **Operation tasks**

ISROCS has to select the most suitable task/s to be performed by the observatory according to predefined criteria. The operation tasks the ISROCS supports are: science observations, calibration tasks, and engineering tasks (periodic maintenance and special testing).

All these tasks are defined using the OB, as a basic unit, and the TG. The definition of OBs and TGs is in Section 3.1.

➤ **Observation types**

The scheduler has to accommodate observations with a fixed starting time, as well as monitoring or periodic observations. It also has to take into account the observation of unexpected phenomena (ToO) that can alter the short-term planning.

➤ **System efficiency**

The global operation efficiency is maximized when most of the scheduled tasks correspond to science operations. ISROCS has to maximize the time invested in science operation tasks. The latter is measured in terms of telescope time spent for the execution of science programs compared with the available time (night time with good environment conditions).

The optimization of the ratio of science and engineering tasks and a correct balance between high and low priority science tasks is also required for maximizing the system efficiency.

The scheduler minimizes time overheads caused by the scheduler execution itself. This is especially critical for the system feedback required by the ISROCS to select the next task to do. The system feedback means inserting the result of the last executed task in its description. The status of a task successfully executed is labelled as finished and it is not considered again by the ISROCS.

➤ **Task sorting and selection**

The ISROCS has to provide an ordered list of tasks for full night scheduling and sorted by execution time. Changing conditions (environment, system status) are not considered for this static planning frame. The scheduler has to dynamically select a single task among those included in the mentioned task list in order to tune the system response while considering the changing environment conditions.

A sorting function is necessary for planning on a time scale of several months. This may be used by the observatory TAC to evaluate the prioritization of science proposals and by any user to check the constraints specified in a new proposal just before its submission.

➤ **Task rating**

The scheduler has to compute a task rating by considering factors grouped as follows: execution constraints depending on system and environment conditions that can change in real-time, and execution constraints depending on static conditions that can be predicted a priori. Each particular factor contribution has to be parameterized and weighted to optimize the scheduler performance.

➤ **Execution users**

Two kinds of users have to be considered to enable automatic and manual control according to the user profiles defined in the general requirements: robotic (or ROCS), for unattended operation, and superuser, for engineering mode.

➤ **Scheduler Algorithm**

The scheduler algorithm has to be flexible and tuneable and its coefficients have to be tuned for optimum overall efficiency. This shall be done in accordance with the observatory operation policy, the distribution of program requirements and the weather statistics on the TJO-OAdM site.

Changing the scheduler algorithm used by the ISROCS has to be doable in a straightforward manner.

9.2.1.2 Boundary conditions

The scheduler has the following inputs and outputs:

- **Input for the static planning:** Execution task constraints for all the pending tasks (calibration or science) are necessary to compute the task ordering routine performed for a specific time interval (a single night or multiple nights).
- **Output for the static planning:** The scheduler has to provide an ordered list of tasks (science observations and calibration tasks) to the observatory control system. The list has to correspond to a specific interval of nights or to a single night.
- **Input for dynamic task selection:** The ISROCS dynamic task selection routine has to read the execution constraints for all the pending tasks (calibration or science) included in the ordered list of tasks for the current night when computing the task selection routine. Tasks defined as associated to transient phenomena (ToO), if not included in the night plan, are added to the list of tasks considered. Environment data (weather, air stability, etc.) and the health status of scientific devices drive the response of this scheduler routine.
- **Output for dynamic scheduler:** The scheduler provides the selected task (science observation or calibration task) to the observatory control system.

9.2.1.3 Interface requirements

ISROCS has to have the following interfaces defined:

- **OpenROCS:** The scheduler has to provide the selected tasks to the observatory control system according to the interface defined (Chapter 5) and for all the possible modes of operation.
- **Archive:** The scheduler has to interface with the archive of the observatory, where all the task execution constraints and OBs status are stored. Execution constraints contain all the necessary information for the selection of tasks by the scheduler. The scheduler has to consider only those OBs validated by the observatory staff and with a pending status.

9.2.1.4 Design and implementation

The scheduler implementation has to be based on computer resources with a high processing capacity, in order to run the selection algorithm in the minimum required time. It is especially important for short-term planning.

9.2.1.5 Non-Functional Requirements

At the analysis level, the non-functional requirements given in Section 2.3 were considered as recommendations.

9.2 Execution block diagrams

The scheduler package is integrated in the main data flow scheme (see Sect. 2.2.2), and needs to interact with several modules in order to give a correct response any time a new planning must be computed.

Once a user has entered the observation details in the database, through the web-based interface, and the observatory staff has validated the data, the scheduler takes these data into consideration when selecting the tasks suitable to be executed during the coming night or when selecting the best task for the next image acquisition. The first process is performed by a mid-term scheduling routine, whereas the second one is carried out by a short-term or dynamic scheduler.

Calibration and engineering tasks are also entered into the system to be considered by the scheduler at the time sorting and selection of tasks are performed.

The next step after the task selection is the execution of the acquisition process, which ends with the image processing routines and the quality control. The latter decides whether the task must be rescheduled or is deemed complete, and updates the database accordingly.

The following schemes (Fig. 9.1 and Fig. 9.2) summarize the foreseen scenarios (red-dashed lines correspond to human attended processes):

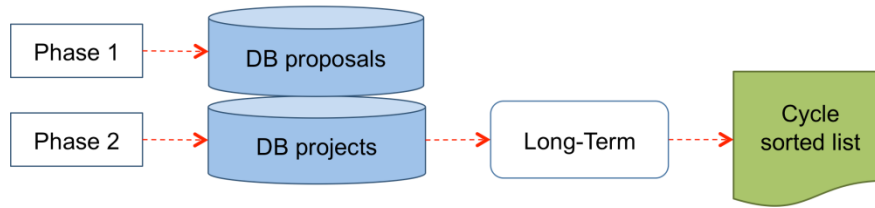


Fig. 9.1: Long-term planning block diagram

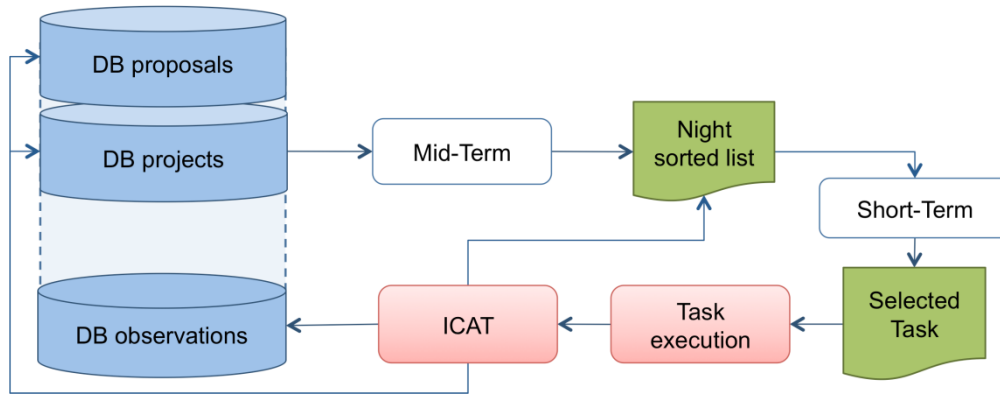


Fig. 9.2: Mid- and short-term planning block diagram

9.3 Scheduler design

The design presented in this section is based on the list of requirements and the execution work flow diagrams described in Sect. 9.2.

9.3.1 Planning modes

We define four types of planning modes:

- **Manual mode:** A manual mode is available to observatory operators for testing new observing procedures. This mode is to be used only in exceptional situations. In manual mode there is no scheduler operative that executes OBs; rather the telescope is under the control of an outside agent (i.e., operator) submitting commands directly to the control system via a control language.
- **Automatic mode:** In automatic mode, mid-term and short-term planning are executed to select the best tasks according to predefined constraints and deliver them to the system for execution. The automatic mode is the default mode of operation.
- **Interactive mode:** In interactive mode, the automatic scheduler furnishes a list of OBs in ranked order to the operator and the operator can choose to override the top selection. The only difference between automatic and interactive modes is that the observatory operator makes the decision as to which OB to execute next rather than the automatic scheduling algorithm.
- **Simulation mode:** The scheduler is capable of running in a simulation mode. It provides users with a tool to check the observation pattern of their proposal for long-term scheduling and evaluate its feasibility before submitting it to the observatory TAC. The simulation mode is also used by the TAC to evaluate and adjust the priority of the proposals. It is possible to use the simulation mode to run the scheduling off-line using recorded historical weather data, or model atmospheric data, as input.

9.3.2 System integration

The scheduler package runs whenever the TJO-OAdM software system is operational, regardless the mode of the telescope, in order to monitor the progress of events within the life of proposals.

The scheduler is integrated in the OpenROCS software suite performing the observatory control. The OpenROCS software manages the events generated by the system, including those controlling the routine operations. Some of these event cases trigger the execution of the mid- and short-term scheduler as specified in Sect. 5.4.

9.3.3 Scheduling use cases

We considered several use cases for the development of the scheduling package and they were the basis for its implementation. In some cases, two different responses are defined depending on the type of user that is executing the process (interactive or automatic):

9.3.3.1 Task rating

The scheduler executes a task selection algorithm considering all pending OBs and based on predefined criteria. The values obtained are used by the scheduler to select the optimum OB among all the available ones and to return a sorted list of tasks.

The mentioned predefined criteria are grouped into two main contributions:

- **Dynamic contribution:** execution constraints depending on system and environment conditions that can change in real-time.
- **Static contribution:** execution constraints depending on static conditions that can be predicted a priori.

9.3.3.2 Scheduling cycles

Long-, mid- and short-term scheduling cycle levels are considered.

All three levels are considered for manual, interactive and simulation planning modes, whereas the telescope executes the OB at the highest position in the ranking of tasks performed by the short-term scheduler in the automatic and interactive modes. The latter is the default mode of operation.

We group these scheduling cycles into the two levels described below: static and dynamic.

➤ **Static scheduling, for long and mid-term planning**

The sorting and selection routine is able to perform long- and mid-term planning of all the active proposals and calibration and engineering tasks in order to obtain a queue of tasks sorted following predefined criteria. Tasks are evaluated with the same algorithm used for the dynamic task scheduling, although only execution constraints that have a predictable variation are considered.

A mid-term scheduling approach is used to develop specific plans to be carried out over a night, whereas long-term scheduling is intended for use on longer time intervals (from two nights to several months).

Inputs

Tasks are taken from the general observatory database and only the pending ones are considered. The routine reads the input data in the form of OB execution constraints for later execution of the sorting function. OBs can be associated to: targets and extra calibration images included in the submitted proposals, nominal calibration images, maintenance and engineering tasks included in the observatory maintenance plan.

Task

This routine executes the scheduler algorithm for each target as a function of time, and sorts the pending tasks based on the output values given by the algorithm. Those factors that have a dependence on environment and housekeeping status are not used. Some selection constraints can be used with a different parameterization for long-term planning purposes (see the algorithm description).

The static scheduler runs when is triggered by the following events:

- **Long-term planning:** It is executed by the user when requiring a long-term planning between two specific dates through the web-based user interface. This process is not required in the unattended operation control.
- **Mid-term planning:** The system event DF_SCHgettime (see Sect. 5.4) triggers its execution.

Outputs

The output of these planning cycles is a sorted list of tasks distributed in consecutive time slots. Oversubscription is permitted in order to ensure the complete use of available time.

➤ **Dynamic scheduling, for short-term planning**

The sorting and selection routine is able to perform a real-time scheduling of tasks (in order of seconds) according to the current conditions. This kind of task planner is known as a dispatch scheduler (Granzer 2004) and it is the one used by the short-term planning of the ISROCS.

Input

Short-term planning is done by taking the list of tasks previously queued by the mid-term process as the input data. Process execution is faster by using this list instead of using all the pending tasks stored in the database.

Task

The selection of the task to execute is done within the list of tasks computed by the mid-term scheduler. Only pending ones are considered. Sorting is based on the dynamic and static constraints, both cross-correlated with the observation constraints required by the user.

The short-term scheduler runs on the following situations that trigger the same event (DF_SCHgettask, Sect. 5.4):

- **Change of constraints:** It is executed every time the system state changes due to a variation of the housekeeping parameters.
- **Ending of the execution process:** It is executed at the end of every OB execution and processing routines giving the possibility to consider the last executed OB in case it did not achieve the required data quality.

Output

The OB with the highest merit is selected and delivered to the control system for execution.

9.3.3.3 Accessing the task specification parameters

The scheduler reads the specification parameters for the execution of each OB. Values are taken from the Projects database level when computing the long and mid-term ranking of tasks and from the night sorted list of tasks when computing the short-term ranking.

9.3.3.4 Operator privileges

In order to allow for special unanticipated observing situations, the scheduling system permits the telescope operator to override its decisions when working on manual or interactive planning modes. These include:

- Selecting an alternative OB to be executed.
- Setting the telescope in manual mode.
- Creating on-the-fly OBs that represent “targets of opportunity”.

9.3.4 Architecture overview

The scheduler routine is related with several modules and applications of the control system (Fig. 9.3):

- **OpenROCS:** The scheduler must provide the system with the new task to execute and with enough details to configure the system, for the dynamic response. Housekeeping data must be accessed by the scheduler for that purpose. It must also compute the static list of tasks when required by OpenROCS (mid- and long-term purposes).
- **ICAT data processing:** This module is responsible for updating the OBs status in the list of pending tasks considered by the dynamic scheduler and according to the data quality obtained after image acquisition. It does not have a direct interface with the scheduler.
- **DDROCS database:** The scheduler interfaces with the observatory database where all the proposal data and OB execution constraints are stored. It must select the tasks suitable to be scheduled for static planning from the pool of pending proposals.

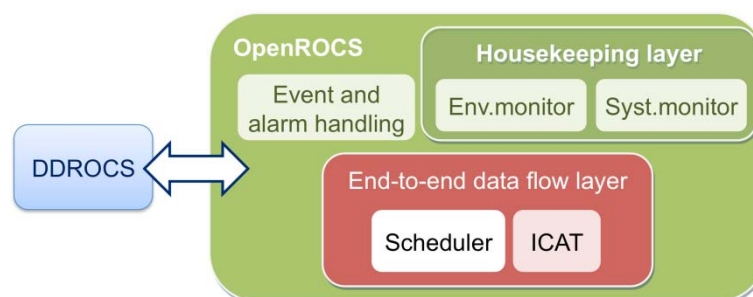


Fig. 9.3: Control system modules related with the scheduler

The scheduler architecture is composed by two main modules (see Fig. 9.4):

- **Sorting and selection service:** This service implements a function that is executed by different events related with the planning modes and cycles. It interfaces with OpenROCS and the scheduling algorithm. Any time a specific event triggers an action to be performed by the scheduler, the sorting and selection service acts with a different predefined response depending on the event case.
 - **Short-term sorting and selection:** Any time a new task can be performed or the system state changes, the system inquires the scheduler for the best task to execute. The scheduler algorithm is then executed by using the mid-term list of tasks as input, and a ranking of tasks is obtained for a short interval of time. The task at the highest position is scheduled first. Oversubscription is not permitted.
 - **Mid-term sorting:** A list of sorted tasks for the whole night must be obtained to determine the starting time for the observatory initialization process executed by the control system. The scheduler algorithm is executed by using the pending OBs archived in the observatory database, and a ranking of tasks is obtained for a full night period of time. Oversubscription is enabled.
 - **Long-term sorting:** A list of ordered tasks for several nights (ranging from 2 days to several months) must be obtained when required by a user (TAC member or external user). A ranking of tasks is obtained for the specific period of time. Oversubscription is enabled.
- **Scheduling algorithm:** The scheduler algorithm is an analytical function used to evaluate the suitability of executing a specific task as a function of time and under specific conditions. Each particular factor contribution to the ISROCS algorithm is parameterized and weighted to optimize the scheduler performance. The OB rating index computation is tuned by modifying these parameters and weight coefficients. Some factors in the algorithm are not considered for mid- and long-term sorting purposes: environment conditions, system health and change of system configuration (i.e., telescope or dome slew-time).

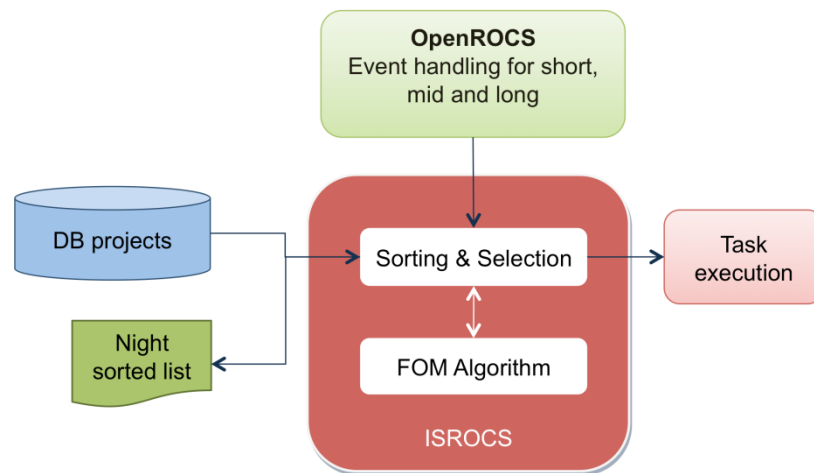


Fig. 9.4: Scheduler architecture

9.4 Scheduler detailed design

9.4.1 Scheduling function

Scheduling tasks for astronomical observatories involves computing the best planning process while introducing new tasks or new constraints and / or a change in the status of the initial constraints.

Among the possible solutions, the one we chose for the TJO-OAdM is based on dispatch scheduling as described in Granzer (2004) and used in the STELLA observatory.

Dispatch scheduling provides a fast and effective approach for dynamic prioritization of tasks based on current conditions. It is not a high consumption algorithm in terms of computer CPU as it only scales with the number of tasks, but it is difficult to be used to do predictions into the future. Dispatch scheduling fulfils our requirements, although other approaches to solve the task prioritization problem may be used.

We propose a modified dispatch scheduling solution to improve the short-term planning and to provide also the mid- and long-term perspective.

9.4.1.1 Dispatch scheduling

The observing conditions for a specific target or for a specific astronomical event can be modeled as a time dependent function. The routine to determine the best point in time to execute an astronomical observation can be based, then, on multiple terms describing the time-varying constraints. So, a time function with a suitable definition and parameterization can be used to compute the task priority and to determine the time when this priority reaches its maximum value.

Dispatch scheduling is based on the Figure of Merit (FOM) computation and its modelization as a time dependent function. FOM considers the priorities for a set of multiple tasks at a specific time using the observation constraints and the current conditions. The selection of the task to execute at the observatory is done by comparing the obtained priorities at that time. The complexity of this task allocation problem arises for samples containing large number of tasks that must be scheduled at a specific period of time and taking into account the long-term perspective. It solves the prioritization problem based on a unique criterion.

A general definition of a merit function is (Eq. 9.1):

$$m(t) = \sum_i \omega_i \cdot f_i(t)$$

Eq. 9.1: Figure of merit

where $f_i(t)$ functions and their associated weights (ω_i) describe the different effects that contribute to increase or reduce the global merit $m(t)$ per object. In order to add long-term perspective to the schedule, a slightly modified function is used at the STELLA observatory (Eq. 9.2):

$$m(t) = \prod_i v_i \cdot s_i(t) \cdot \sum_j \omega_j \cdot g_j(t)$$

Eq. 9.2: Figure of merit with long-term contribution

where the s_i 's (time-slots) incorporate the long-term behaviour of target selection (i.e., over several nights), while the g_j 's (gains) are mainly used for short-term scheduling, i.e. over the course of a given night. A suitable selection of the merit functions (s_i and g_j) and weights (v_i and w_j) is required to compute the merit for each task.

9.4.1.2 Modified dispatch scheduling

The scheduling routine we propose is also based on the FOM (Eq. 9.2). We use this algorithm to perform a prioritization sequence that takes into account also the long-term contribution when doing the dynamic task selection. The sequence to execute is composed by the following steps:

➤ **First step: Single task with time varying conditions for mid- and long-term planning**

Task allocation in time slots is done first by computing the maximum values of the FOM function for every different task as a function of time.

Time scales used when resolving these values determine the resolution of the solutions found: time scale of minutes is used to compute the priority all over a single night used for mid-term scheduling, whereas a time scale of hours only shows daily dependencies required for long-term scheduling.

Repeating this process for all the active tasks at the observatory populates the time slots (minute range for mid-term and hour-range for long-term) with overlapping of tasks.

A suitable selection of terms contributing to the FOM must be done to obtain the desired results, and task execution constraints must also be well defined. Only task observing conditions (i.e., target coordinates, time scheduling pattern, or minimum distance to the Moon) that are predictable a priori and without any dependence on the initial conditions of the system (telescope or dome configuration) are taken into account. The FOM used for this purpose is called Static FOM (SFOM) as it is based on the static constraints.

The output resulting from this process is a list of pairs ($SFOM_{i_{max}}$, t_i) corresponding to the local maximum values for a particular task. Multiple local maximum points can be found, as illustrated in Fig. 9.5 where m_{Task_i} and t_i pairs are given for points of maximum merit. These values are used to

define the execution time slots for every task for mid- and long-term planning. However, they are also important when computing the dynamic selection of tasks, as described in the following step.

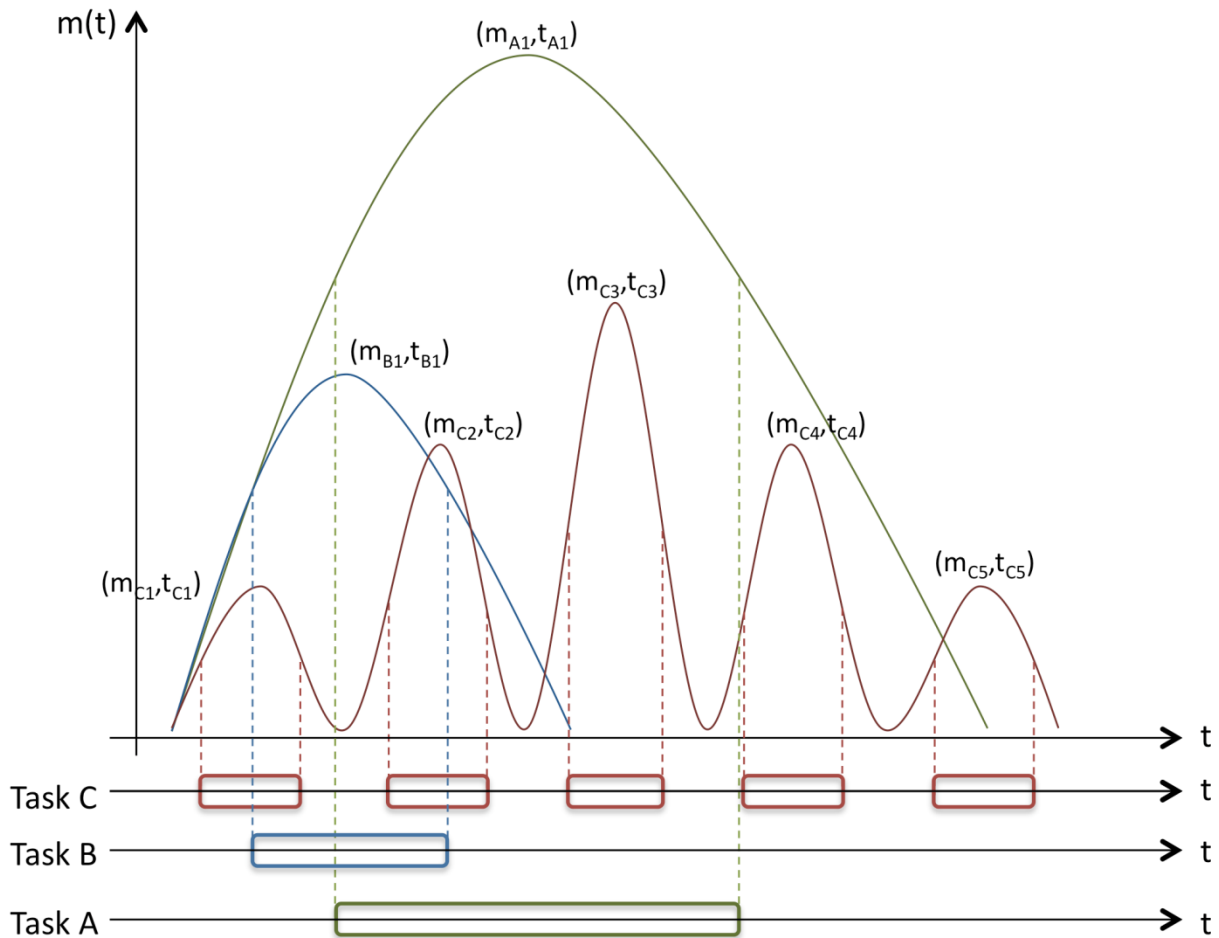


Fig. 9.5: Merit function for three different tasks (top) and time intervals (indicated with boxes) at each merit maximum value where Peak Closeness merit promotes the task selection (bottom)

➤ **Second step: Multiple task prioritization under a particular set of non varying conditions for short-term planning**

Computing the FOM at specific initial conditions, also considering the system housekeeping data, is used to dynamically prioritize the pending tasks. FOM used for this purpose is called Dynamic FOM (DFOM), and it is an extended version of the SFOM where dynamic constraints are added.

Tasks used to compute this short-term prioritization and selection are only those included in the list obtained in the mid-term planning for the current night in order to minimize the execution time.

Using the DFOM has a major drawback: long-term perspective is not taken into account. The task found to have the highest priority, for instance, could be very far from its optimum execution time (that is: close to a maximum of the FOM).

We propose a generic solution to this problem by including the mid-term perspective in the FOM when computing the dynamic selection: a new multiplicative term is added to the FOM function to modulate its value by using the distance to the $SFOM_{lmax}$ points that were computed in the first step of this process. This new term contributes by increasing the merit value as it gets closer to or goes away from a local maximum. It describes how the FOM approaches its maximum value and is used to define an area around it where the task priority has to be maximized.

This new term is based on the SFOM second time derivative that represents the slope sign and gradient in the neighborhood of the SFOM local maximum. We have called it the Peak Closeness Merit that can be modeled with the following merit function:

$$m_{PC}(t) = \tau \cdot \left| \frac{1}{\ddot{m}} \right|^\varphi; \text{ where } \tau = \begin{cases} 0, & |SFOM_{imax}(t_i) - m(t_0)| < \alpha \\ \Delta, & |SFOM_{imax}(t_i) - m(t_0)| \geq \alpha \end{cases}$$

Eq. 9.3: Peak Closeness merit function

φ is used to define the steepness of this merit and Δ is a constant time interval that is scaled to higher or lower values depending on the SFOM function gradient. \ddot{m} is the second time derivative and α is a fixed value that determines how close the FOM must be from its maximum value to consider this as a schedulable task. Fig. 9.5 illustrates how this merit behaves for three tasks with different shape: it promotes the selection of each task when time has entered into the interval (indicated with round-shaped boxes) given by the $m_{PC}(t)$ merit.

The FOM we designed for the TJO-OAdM to provide the dynamic task selection is shown at Eq. 9.4.

$$m_M(t) = DFOM \cdot m_{PC}(t); \text{ where } DFOM = SFOM + (\text{dynamic constraints})$$

Eq. 9.4: Modified Figure of Merit for dynamic task selection

9.4.1.3 Sorting and selection service

This service is in charge of running the sequence of processes described in last section. It is mainly devoted to sorting the pending tasks for mid- and long-term planning and selecting the task to execute for short term. It involves several processes: reading the parameters describing the task execution constraints for all the pending tasks and executing the scheduler algorithm to compute the priority (merit) of each task under predefined conditions (static or dynamic).

The FOM function used (SFOM or $m_M(t)$) depends on the scheduling cycle (long-, mid- or short-term).

Different calls to the algorithm and different outputs are defined for this service. Calls can be described with the following parameters to configure the execution of the algorithm:

- FOM: It defines the FOM function to use and the list of constraints.
- $t_i, t_f, \Delta t$: These are the parameters that describe the time variation for mid- and long-term computation. t_i, t_f correspond to the time interval limits, whereas Δt is the time scale used.

➤ Long-term cycle

Long-term scheduling is triggered manually through the scheduler user interface. The service performs a time varying query to the scheduler algorithm to compute the FOM for a specific task and with the following parameters:

- **Configuration of the algorithm call:**
 - FOM = SFOM
 - $\Delta t = 1$ hour
 - t_i, t_f = Time interval defined by the user
- **Output:** It stores the local SFOM maximum value/s for the complete time interval and for all the required tasks.

➤ Mid-term cycle

The mid-term scheduling is triggered by the control system that also configures its execution. In order to define the best execution time for each task for the next night, a 24h time periode is evaluated. The service performs a time varying query to the scheduler algorithm to compute the FOM with the following parameters:

- **Configuration of the algorithm call:**
 - FOM = SFOM
 - $\Delta t = n$ minutes (n value is subject to calibration at the implementation phase)
 - $t_f - t_i = 24h$, where t_i corresponds to the time of execution
- **Output:** It stores the local SFOM maximum values for the complete time interval and for all the pending tasks, both specified by the control system.

➤ Short-term cycle

The short-term scheduling is triggered by the control system that also configures its execution. The service performs an instantaneous calculation of the FOM for all the tasks included in night tasks selected by the mid-term execution. An iterative process is carried out to check if there is a task with higher priority during its execution time.

- **Configuration of the algorithm call:**
 - $FOM = m_M(t)$
 - $\Delta t = 0$
 - $t_f = t_i$
- **Output:** It composes a sorted list of tasks and delivers the task at the highest position to the control system for its execution.
- **Process iteration:**
 - The service calls the algorithm that returns the selected task with an execution time of t_s seconds.
 - It checks if there is a different task with higher priority between t_i and t_i+t_s . It uses the values obtained by the mid-term cycle process for this purpose, where maximum merit values per task and as a function of time are computed and stored.
 - If there is a task with higher priority, the first selected task is temporarily discarded and iteration starts again to select a different task.

9.4.2 Scheduler algorithm

The scheduler algorithm we designed to use at the TJO-OAdM is based on the per-target merit calculation described in Granzer (2004), but it is modified to add new contributions to the merit function. Basic and new terms are included in the design description given here. The relation between the merit terms and the execution constraints is also presented.

The execution constraints for the TJO-OAdM observatory are detailed first as a short introduction before describing the algorithm.

9.4.2.1 Execution constraints

All the execution constraints are derived from the contents of the scheduler analysis and design section and are mainly a consequence of the observatory operational design specification (see Sect. 3.1). The constraints we considered to compute the per-target merit are grouped into static and dynamic contributions.

➤ *Static contribution*

The following list corresponds to those execution constraints that have a predictable value or variation:

- Operation mode (MEIA, ARES).
- Target position, computed from the Equatorial coordinates or the orbital elements for solar system objects.
- OB execution pattern (image type, exposure time, mode, etc.).
- Priority index.
- Time scheduling mode (flexible, time- or phase-critical).
- UT value, for ephemeris dependent proposals.
- Elapsed UT since previous execution (for time- or phase critical proposals).
- OB and TG interdependencies.
- Breakpoint reached or released.
- Proposal execution status (started, approaching completion, currently on the telescope).
- Source visibility and remaining visibility time.
- Airmass.
- Distance to Sun and Moon, Moon phase and sky darkness, in terms of Sun and Moon height.

➤ *Dynamic contribution*

Unexpected events that can change the task scheduling defined a priori are included here:

- Hardware availability and status, provided by the control system and the image quality control
 - Instrument (MEIA, with CCD camera, filters and flats lamp; or ARES).
 - Dome (shutter status and azimuth).
 - Telescope pointing position (RA, DEC).
 - Dome flats lamp status.
- Environment conditions that should abort any telescope operation or pauperize the execution conditions:
 - Weather (rain, lightning, clouds, humidity inside and outside the dome, wind).
 - Air stability (measured in terms of seeing).
 - Atmospheric transmission.

9.4.2.2 Merits

We present here the original concept for the algorithm given by Granzer (2004) and the modifications we have made to adapt it to the TJO-OAdM observatory and to add more functionality. The latter includes those effects that are particular to this telescope and those added according to the modified dispatch scheduler design discussed before.

The effects considered in the original concept of Granzer and applied to the TJO-OAdM are:

- **Execution conditions:** altitude above the horizon, time remaining to observe the target in the current night, distance from a celestial object (Moon, Sun, etc.), best observation time in case the observation is required not at the meridian crossing, required period between iterations, distance from previous target (telescope slew-time), dependencies on former tasks.
- **Environment conditions** (Boolean constraints): weather, seeing, atmospheric transmission.
- **Hardware availability for each observation mode** (Boolean constraints): MEIA and photometric filters, ARES and wavelength windows at the TJO-OAdM.
- **Proposal priority:** term to ensure a balance between the observation time within different users or groups.

And the new effects to add for its application to TJO-OAdM are:

- **Execution conditions:** sky darkness in terms of Sun altitude and Moon phase and altitude, time cost to change the system configuration from previous task (initialization processes, instrument change, time to open the dome shutter or to turn on the flats lamp, and time to slew the dome to the new pointing azimuth).
- **Environment conditions** (non-Boolean constraints): seeing, atmospheric transmission.
- **Proposal priority**

The main contributions to the global merit for the TJO-OAdM are summarized in Table 9.1, where merit contributions are related with the execution constraints specified by the users and stored in the Projects database level. Each merit contribution is included in a particular merit function (SFOM and/or DFOM) depending on the scheduling cycle. New effects added are labelled with an (N). Merits are grouped following the criteria used by Granzer (2004) for easy comparison:

Merit groups	Merit contribution	Execution constraints	Scheduling cycle (long, mid, short)
Execution conditions – Target position and system state	Window	Target coordinates & maximum airmass	SFOM & DFOM
	Airmass	Minimum & maximum airmass	SFOM & DFOM
	Avoidance	Distance to celestial object (Sun, Moon, etc.) and Moon phase	SFOM & DFOM
	(N) Sky darkness	Sky darkness, Sun and Moon altitude	SFOM & DFOM
	Telescope slew-time	Target coordinates	DFOM
	(N) Dome slew-time	Target coordinates	DFOM
	(N) Merits for the state transition processes	Type of task (calibration, science)	DFOM
	(N) Instrument change-time	Instrument specification	DFOM
Execution conditions – Previous executions	After-pick	Time scheduling, OB or TG interdependencies	SFOM & DFOM
	Forced time-out	Time scheduling	SFOM & DFOM
	Maximum number of observations	OBs & TGs specification	SFOM & DFOM
	Phase-coherent	Time scheduling: time or phase critical task	SFOM & DFOM
Execution conditions – Restricting pick times	At-time	Time scheduling: ephemeris task	SFOM & DFOM

	Not-before and not-after	Time scheduling	SFOM & DFOM
	Terminal	Time scheduling	SFOM & DFOM
Proposal priority	User fairness	Granted time to users	SFOM & DFOM
	(N) Proposal priority	TAC assigned priority	SFOM & DFOM
	(N) TG & OB priority	User assigned priorities	SFOM & DFOM
Environment conditions	Weather	Weather	DFOM
	Seeing	Seeing range	DFOM
	Atmospheric transmission	Atmospheric transmission range	DFOM
Hardware availability	MEIA and filters status	Instrument specification	SFOM & DFOM
	ARES status	Instrument specification	SFOM & DFOM

Table 9.1: Contributions to the merit function

A detailed description of each one of these effects, including the expression used to evaluate them, is given below grouped by static and dynamic contributions. Most of them are parameterized as defined in Granzer (2004), except for those newly added effects.

➤ **Static contributions to the FOM**

Window merit

When the time remaining to observe a target in the current night drops this merit rises. It is evaluated according to:

$$m_w(t) = \max\left(0, 1 - a \cdot t_r + b \frac{1}{1 + c \cdot t_r}\right); \text{ with } t_r(t) = \frac{t_{r:\text{target}}}{t_{r:\text{night}}}$$

Eq. 9.5: Window merit

$t_{r:\text{target}}$ is the time remaining that the target can be observed and $t_{r:\text{night}}$ is the night-time remaining, respectively. a , b and c determine the shape of the merit. Fig. 9.6 illustrates the function with $c = 10$ and different values for b and a .

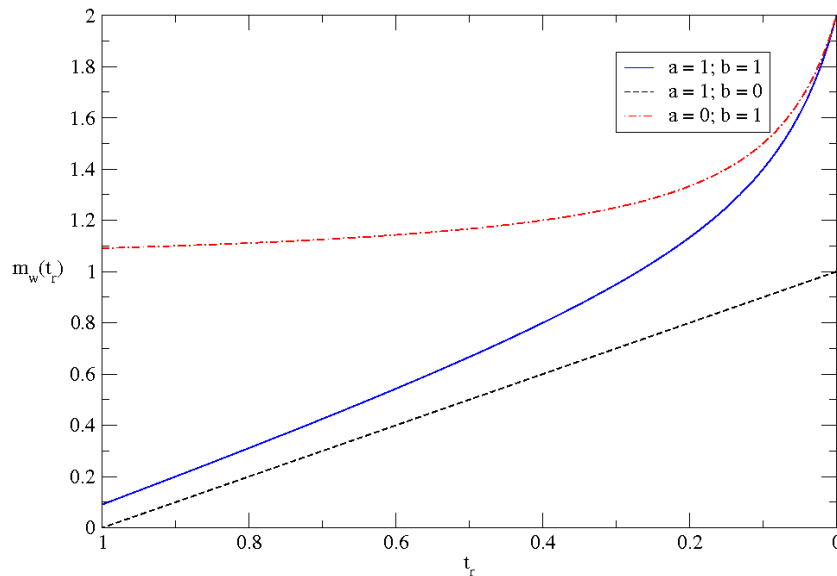


Fig. 9.6: Window merit

Airmass merit

The observation is scheduled at an airmass smaller than the maximum airmass specified. This merit maximizes target selection at low airmass values:

$$m_A(t) = \frac{1}{\left(\chi(t)/\chi_{max}\right)^{\alpha_A}}$$

Eq. 9.6: Airmass merit

X is the target airmass, X_{max} corresponds to the maximum value of the target airmass and α_A is a configuration parameter defining the steepness of the merit. The shape of this merit is shown in Fig. 9.7, with $X_{max}=4$ and different values for α_A .

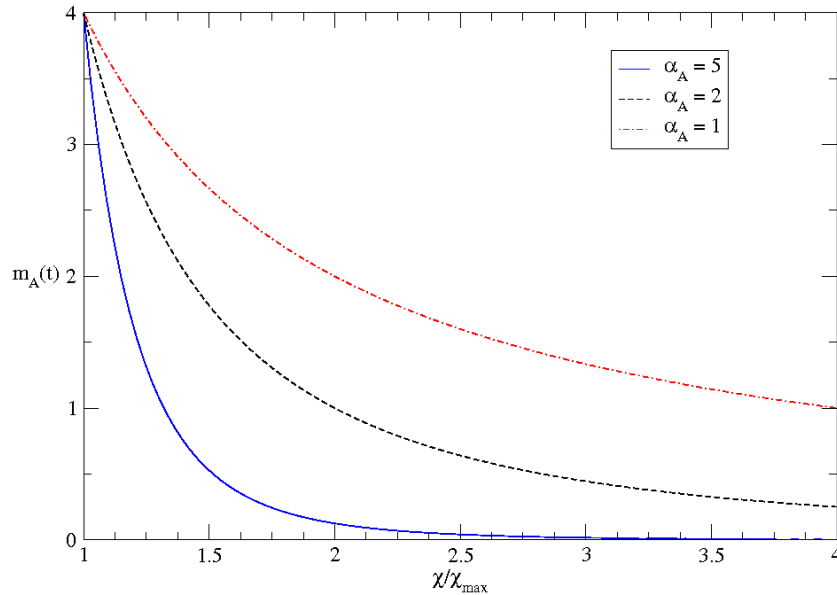


Fig. 9.7: Airmass merit

Avoidance merit

This merit is used to avoid those regions close to another celestial object that could perturb the science observations (Moon, Sun, Jupiter, etc.).

$$m_{Av}(t) = \left(\frac{d(t)}{\Pi}\right)^s$$

Eq. 9.7: Avoidance merit

d is the distance of the target to the avoidance region and s defines the steepness of the merit.

The effects of the Moon at different phases are correlated with the distance: objects close to the Moon at a low phase state are much less perturbed than objects far from the Moon at a high illumination percentage. When evaluating the avoidance merit considering the Moon as a perturbative object, its phase (P_{Moon}) status is considered:

$$m_{AvMoon}(t) = \left(\frac{d_{Moon}(t)}{P_{Moon}}\right)^{s_{moon}} ; \text{ with } 0 \leq P_{Moon} \leq 1$$

Eq. 9.8: Avoidance merit for the Moon

Sky darkness merit

This merit is used to avoid sky conditions with high brightness due to the Sun or Moon altitude. Users provide minimum Sun and Moon altitude to execute their observations. They can specify science observations to be done during twilight and it must be reflected in the merit definition.

$$m_{hSun} = \begin{cases} 1, & h_{0-Sun} < h_{Task-Sun} \\ 0, & h_{0-Sun} \geq h_{Task-Sun} \end{cases}$$

$$m_{hMoon} = \begin{cases} 1, & h_{0-Moon} < h_{Task-Moon} \\ 0, & h_{0-Moon} \geq h_{Task-Moon} \end{cases}$$

Eq. 9.9: Sky darkness merit

After-pick merit

This merit grows after a successful observation of a target until it reaches a certain peak, from where it drops to zero. The merit is modelled with the following quadratic function, which is defined by two parameters: the maximum and zero values.

$$m_{After}(t) = \frac{(t - t_0)^2 - 2m(t - t_0)}{z(2m - z)} + 1; \text{ with } \begin{cases} t_0 < t < t_0 + z \\ 2m < z \end{cases}$$

Eq. 9.10: After-pick merit

The observation time of the precursor is t_0 , the time from t_0 to the peak of the merit is m , while z denotes the time it takes until the merit drops to zero. The merit drops continuously if m is negative. Note that the precursor task can be the observation of a different target or the target itself. Fig. 9.8 illustrates the variation of this merit with time.

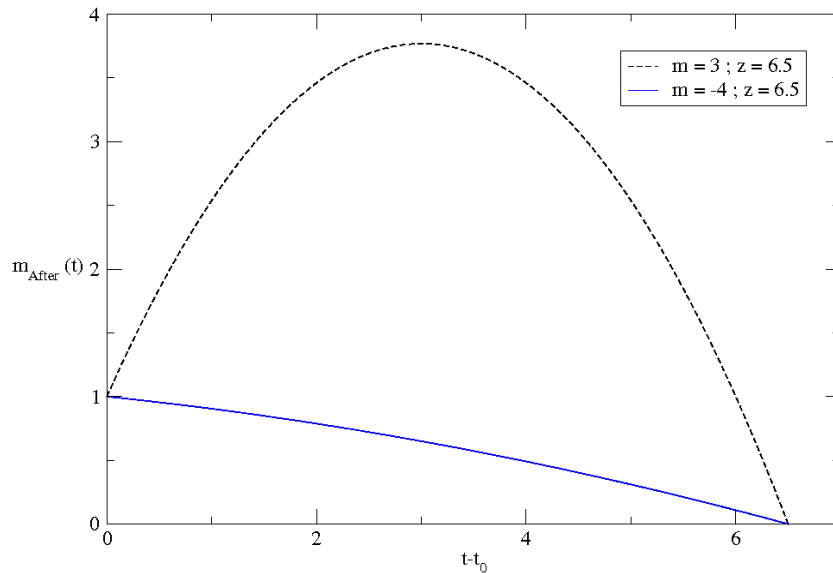


Fig. 9.8: After-pick merit

Forced time-out merit

This merit drops to zero after a task has been executed successfully. It remains at zero for a certain amount of time, until which it regains immediately its full value. It follows a time-out pattern shape.

$$m_{TOut}(t) = \begin{cases} 1, & t < t_1 \\ 0, & t_{i-1} < t < t_i \\ 1, & t_i < t < t_{i+1} \\ 0, & t_{i+1} < t < t_{i+2} \\ 1, & t > t_k \end{cases}$$

Eq. 9.11: Forced time-out merit

Maximum number of observations

This merit (m_{nummax}) drops to zero as soon as the required number of successful observations of the target has occurred.

Phase-coherent merit

This merit allows target selection at certain intervals in its phase space.

$$m_{Phase}(t) = \sum_i \kappa_i \cdot e^{-\frac{(x-\mu_i)^2}{\sigma_p}} ;$$

$$with \mu_i = \frac{i}{N_a}$$

$$\kappa_i = 0, or 1 + \beta \frac{N_a}{N_a - N}$$

$$x(t) = \frac{mod(t - t_0, p)}{p} ; for |x| \leq 0.5$$

Eq. 9.12: Phase-coherent merit

The period p of the observations and the number of desired observations within a period, N_a , are supplied by the user. The restrictiveness to stick to certain phases is defined with σ_p , small values here lead to more restrictive picking close to the projected phase-times. The κ_i 's are zero if an observation has been successfully completed close to the appropriate phase, or else evaluated according to the equation. The parameter β can be used to boost the merit as the observing program reaches completeness. The zero-point of the observation, t_0 , is either specified or set equal to the first successful observation of the target. N is the actual number of successful observations of this target. The individual merit peaks drop to zero as an observation has been completed successfully close to the requested phase. Fig. 9.9 illustrates the variation of this merit vs. $x(t)$ for different values of σ_p and with $\beta = 0$.

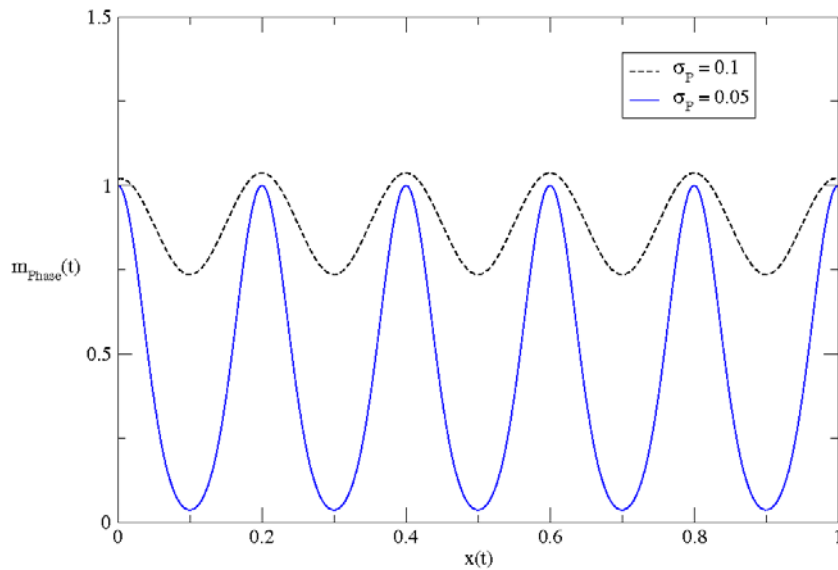


Fig. 9.9: Phase-coherent merit

At-time merit

This merit is a gauss-shaped modelization with a peak at a predefined time t_0 , provided as an execution constraint.

$$m_{At}(t) = e^{-\frac{(t-t_0)^2}{\sigma_A}}$$

Eq. 9.13: At-time merit

σ_A determines how fast the merit drops if the current time (t) is not close to t_0 .

Not-before and not-after merits

This merit is used to block the task selection prior ($m_{NotB}(t)$) or after ($m_{NotA}(t)$) time t_0 .

$$m_{NotB,A}(t) = \max \left(0, \ln \left(\pm \frac{t - t_0}{\sigma_{NotB,A}} + 1 \right) \right)$$

Eq. 9.14: Not-before and not-after merit

$\sigma_{NotB,A}$ defines the steepness of the merit and the sign defines the direction. In Fig. 9.10 the not-before merit vs. the time elapsed from t_0 is shown.

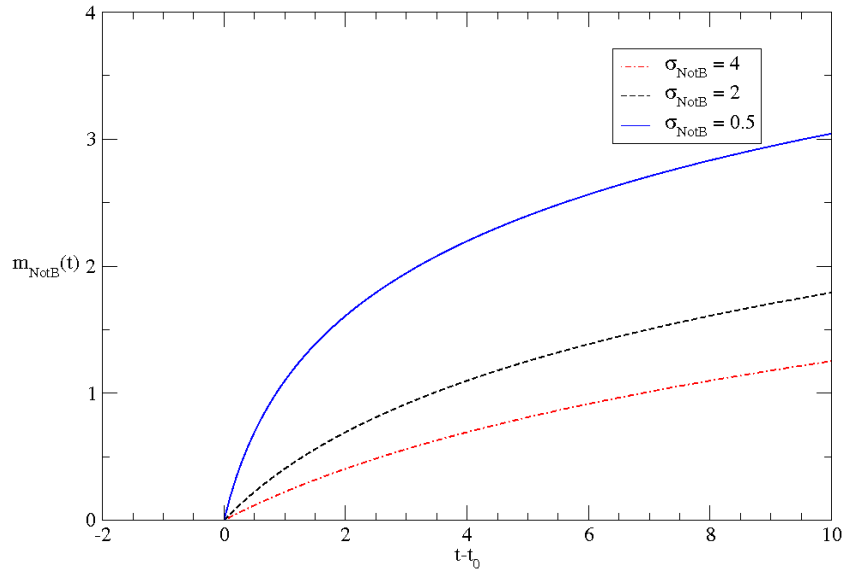


Fig. 9.10: Not-before merit

Terminal merit

The following expression can be used to drop the merit to zero within a specified time:

$$m_T(t) = \frac{1}{2} \left[1 - \tanh \left(\frac{t - t_0}{\sigma_T} \right) \right]$$

Eq. 9.15: Terminal merit

m_T is equal 1 when $t < t_0 - \sigma_T$ and zero for $t > t_0 + \sigma_T$. In Fig. 9.11 the merit variation with $t-t_0$ is shown for two different values of σ_T .

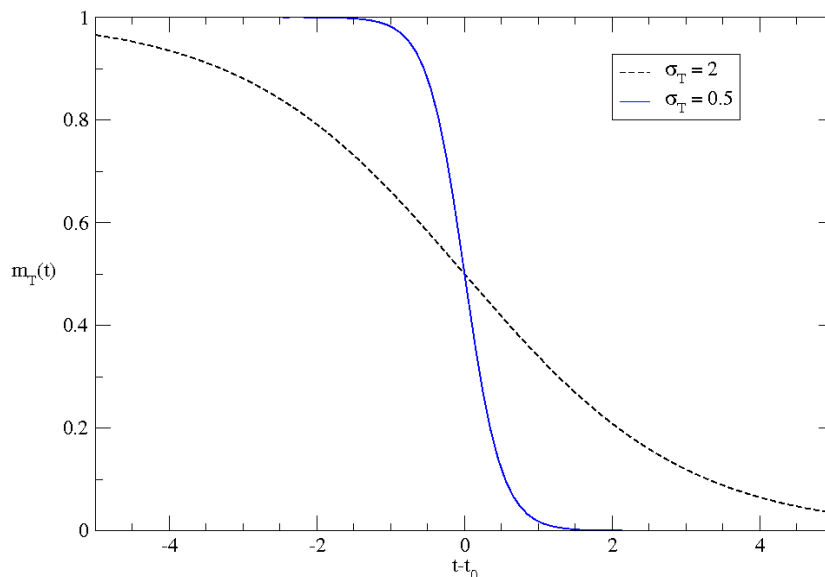


Fig. 9.11: Terminal merit

User fairness

This merit is used to favour the selection of a particular proposal or user when the associated OBs are not selected on a random basis. In case an executive priority is defined (see Sect. 3.1.4), this merit is also used to compare the granted time with the actual observation time for a specific group of users.

$$m_{User}(t) = \frac{(t_{granted} - t_{actual})^{\alpha_u}}{t_{granted}^{\alpha_w}}$$

Eq. 9.16: User fairness or executive priority merit

α_u and α_w defines the rigidity of this merit.

Proposal, TG and OB priorities

The priority merit ($m_{Priority}$) results from the priority values defined by the TAC and the user that submits the proposal.

$$m_{Priority} = \frac{\alpha_P P_P + \alpha_{TG} P_{TG} + \alpha_{OB} P_{OB}}{\alpha_P + \alpha_{TG} + \alpha_{OB}}$$

Eq. 9.17: Priority merit

α_P , α_{TG} , α_{OB} , coefficients are used to modulate the weight of each priority.

➤ Dynamic contributions to the FOM

Merits for state change processes

The system state transitions have to be taken into account in some situations. The control system has to change the observatory configuration depending on the kind of tasks (calibration or science) the scheduler selects and the current state (see Sect. 5.4 for more details on the state change processes). However, changing the configuration involves some processes that require some time to be done:

- Initializing the instrument is required for any kind of task.
- The initializing of science devices (telescope and dome) is done for Flat calibration tasks or science observations.
- The dome flats lamp is turned on to perform dome flat fields.
- Opening the dome shutter and thermalizing the air temperature are necessary processes for science observations.

An evaluation of the static merit (SFOM) function is required to predict the merit on different cases. SFOM is computed for a time later (t_{init} , t_{calib} , t_{flats} , and $t_{thermal}$, defined in Sect. 3.1) and it has a value equal to 1 only if the merit value is greater than a threshold (ρ_x). Finally, this term is introduced into the DFOM algorithm and the values obtained are compared to select the task for execution among those tasks that meet this condition.

The priorities for the science tasks are specified in a way that a science observation is always scheduled if there is at least one science task fulfilling the previous condition. This selection criterion maximizes the execution of science tasks. In addition, the selection of a calibration task is favoured when conditions for science observations are not met.

The merit terms described here are introduced to perform this modulation of the system state transition.

- **Instrument initialization merit:** A task is selected for execution if any of the pending ones has an SFOM bigger than a predefined threshold (ρ_{InitI}) at time $t + t_{Calib}$.

$$m_{InitI}(t) = \begin{cases} 1, & SFOM(t + t_{calib}) > \rho_{InitI} \\ 0, & SFOM(t + t_{calib}) \leq \rho_{InitI} \end{cases}$$

Eq. 9.18: Instrument initialization merit

- **Telescope and dome initialization merit**

$$m_{InitTD}(t) = \begin{cases} 1, & SFOM(t + t_{Init}) > \rho_{InitTD}; \text{ for Flat - field calibration or science tasks} \\ 0, & SFOM(t + t_{Init}) \leq \rho_{InitTD} \end{cases}$$

$$m_{InitTD}(t) = 1; \text{ for Bias or Dark calibration tasks}$$

Eq. 9.19: Telescope and dome initialization merit

- **Dome shutter open merit:** Opening and closing the shutter involves $t_{Shutter}$ minutes, and $t_{Thermal}$ minutes are spent thermalizing the dome after the shutter aperture.

$$m_{SO}(t) = \begin{cases} 1, & SFOM(t + t_{Sh}) > \rho_{SO}; \text{ for science tasks} \\ 0, & SFOM(t + t_{Sh}) \leq \rho_{SO} \end{cases}$$

$$\text{with } t_{Sh} = t_{Thermal} + t_{Shutter}$$

$$m_{SO}(t) = 1; \text{ for calibration tasks}$$

Eq. 9.20: Dome shutter open merit

- **Dome flats lamp merit:** The system selects the best moment to perform dome flat fielding images aswell. This task requires a specific system state (Flat) that basically means that it turns the flat lamp on and waits for its stabilization. This process adds a similar restriction to the opening of the dome shutter. An evaluation of the static FOM based on the warming up time t_{Wup} of the flats lamp is done every time a calibration task is evaluated. Other calibration images (Bias, Dark) do not require this change on the state configuration.

$$m_{Flat}(t) = \begin{cases} 1, & SFOM(t + t_{Wup}) > \rho_{Flat}; \text{ for Flat - field calibration task} \\ 0, & SFOM(t + t_{Wup}) \leq \rho_{Flat} \end{cases}$$

$$m_{Flat}(t) = 1; \text{ for Bias, Dark calibration and science tasks}$$

Eq. 9.21: Dome flats lamp merit

Telescope and dome slew-time merit

Short slew-time optimization for telescope and dome is optimized by this merit, modelled with the equation:

$$m_{Slew}(t) = \min \left[1, \max \left(0, 1 + \frac{p_t - t_s}{a_t} \right) \right]; \text{ where } t_s = \max[t_{s:tel}, t_{s:dome}]$$

Eq. 9.22: Slew-time merit

$t_{s:tel}$ and $t_{s:dome}$ are the expected slew-times from the current telescope and dome position, respectively, to the position of the next target. At the TJO-OAdM, this value corresponds to the maximum value of the comparison between telescope and dome slew-time. p_t specifies a plateau time to define a threshold after which the merit should become active. This may be the read-out time of the instrument. a_t defines the time after which the merit drops to zero. Fig. 9.12 shows the shape of this merit when the telescope or dome slew-time grows, and for a two different combinations of values for p_t and a_t .

Instrument change-time merit

This merit can be expressed with the same modellization as the slew-time merit. For the instrument change-time merit, t_{ci} is the expected time necessary to change from one instrument to another (MEIA to ARES), p_{ii} specifies a threshold that could be the read-out time, as the slew-time case ($p_t = p_{ii}$). And a_{ii} has the same meaning than before. Using this merit for the scheduler is foreseen for the time when ARES is installed.

$$m_{Instrc}(t) = \min \left[1, \max \left(0, 1 + \frac{p_{ti} - t_{ci}}{a_{ti}} \right) \right]$$

Eq. 9.23: Instrument change-time merit

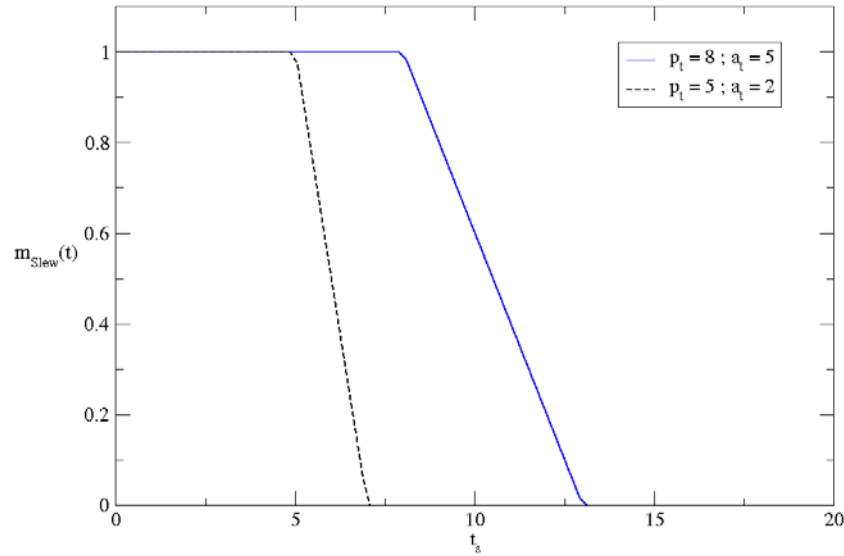


Fig. 9.12: Slew-time merit

Weather merit

Weather conditions are used as a Boolean condition. Merit ($m_{Weather}$) is equal to zero if weather conditions do not fulfill the requirements. Indoor data are used to check the suitability of executing calibration tasks that do not require opening the dome shutter, whereas outdoor data are used for science observations.

$$m_{Weather} = \begin{cases} 1, & \text{no weather event} \\ 0, & \text{weather event} \end{cases}$$

Eq. 9.24: Weather merit

Seeing merit

Seeing merit function drops as seeing conditions move away from to the required value. Its maximum merit is a function of the maximum seeing required by the user (S_0). This favors the selection of tasks that require more demanding conditions when these conditions appear.

$$m_{Seeing}(Seeing) = \frac{1}{S_0} \left[1 - \tanh \left(\frac{Seeing}{\sigma_{Seeing}} - 10 \cdot (S_0 + S_{Tol}) \right) \right]$$

Eq. 9.25: Seeing merit

In Fig. 9.13 the course of the merit with measured seeing for three different tasks requiring very good (< 1 arcsec), good (< 2 arcsec) and poor (< 3 arcsec) conditions is shown. σ_{Seeing} defines the steepness of the drop, whereas S_{Tol} is the tolerance given to select a task when the current seeing is above the S_0 . $\sigma_{Seeing} = 0.1$ and $S_{Tol} = 0.1$ are used in the figure.

Atmospheric transmission merit

The contribution to the merit function due to atmospheric transmission (m_{Atm}) follows the same modelling than the seeing constraint.

MEIA and filters and ARES status

Instrument availability status is used as a Boolean condition. Merit (m_{Instr}) is equal to zero if the required instrument is not available.

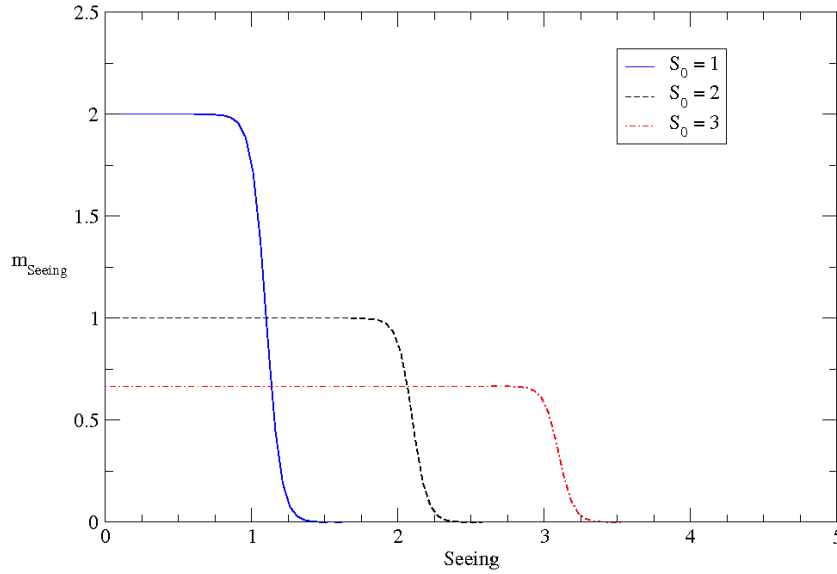


Fig. 9.13: Seeing merit

9.4.2.3 Algorithm equation

The figure of merit algorithm for the TJO-OAdM is based on the merit contributions described in last section and the equations Eq. 9.2 and Eq. 9.4. It is finally written as a combination of additive and multiplicative terms for each of the main components (static and dynamic). Those terms that do not represent a Boolean condition are weighted with specific coefficients (v_i and ω_i) that are used to calibrate the relevance of each effect.

➤ Static component

The equation used for mid- and long-term scheduling is finally written as follows:

$$\begin{aligned}
 SFOM &= SFOM_{\Pi} \cdot SFOM_{\Sigma} = \prod_i v_i \cdot s_i(t) \cdot \sum_j \omega_j \cdot g_j(t) \\
 &= v_{After} \cdot m_{After}(t) \cdot v_{Phase} \cdot m_{Phase}(t) \cdot v_{At} \cdot m_{At}(t) \cdot v_{NotBA} \cdot m_{NotBA}(t) \cdot v_T \cdot m_T(t) \\
 &\quad \cdot v_{TOut} \cdot m_{TOut}(t) \cdot m_{hSun} \cdot m_{hMoon} \cdot m_{nummax} \cdot (\omega_W \cdot m_W(t) + \omega_A \cdot m_A(t) + \omega_{Av} \cdot m_{Av}(t) \\
 &\quad + \omega_{AvMoon} \cdot m_{AvMoon}(t) + \omega_{User} \cdot m_{User}(t) + \omega_{Priority} \cdot m_{Priority})
 \end{aligned}$$

Eq. 9.26: SFOM, algorithm for mid- and long-term scheduling

➤ Dynamic component

The equation used for short-term scheduling is finally written as follows:

$$\begin{aligned}
 m_M(t) &= DFOM \cdot m_{PC}(t) = SFOM_{\Pi} \cdot DFOM_{\Pi} \cdot (SFOM_{\Sigma} + DFOM_{\Sigma}) \cdot m_{PC}(t) \\
 &= SFOM_{\Pi} \cdot v_{Seeing} \cdot m_{Seeing}(seeing) \cdot v_{Atm} \cdot m_{Atm}(atm. transmission) \cdot m_{InitI}(t) \\
 &\quad \cdot m_{InitTD}(t) \cdot m_{SO}(t) \cdot m_{Flat}(t) \cdot m_{Weather} \cdot m_{Instr} \\
 &\quad \cdot (SFOM_{\Sigma} + \omega_{Slew} \cdot m_{Slew}(t) + \omega_{InstrC} \cdot m_{InstrC}(t)) \cdot m_{PC}(t)
 \end{aligned}$$

Eq. 9.27: Modified dispatch scheduling algorithm for short-term planning

Where $m_{PC}(t)$ is a function of (\ddot{m}) , with m corresponding to the SFOM function, but excluding the Boolean terms (m_{hSun} , m_{hMoon} , m_{nummax}).

9.4.2.4 Scheduler algorithm calibration

The scheduler algorithm has tens of parameters that are used to tune the selection process by giving more emphasis to a merit contribution or reducing the accent to a different one. Adjusting their values is a task planned to be done after implementation and during a specific commissioning of the ISROCS application. A simulation tool will be used for this purpose, based on two main components:

- **Test sample:** A sample of tasks composed by a wide spectra of realistic science and execution cases to be fully representative of the needs of the observatory.

- **Simulation platform:** A software platform to simulate the system behaviour in terms of the system configuration, the system health and the environment conditions.

Simulations will also determine if we have to add a new term with a random behaviour to avoid the systematic selection of a specific kind of targets (i.e., targets at higher altitudes).

A summary of the parameters that have to be calibrated is shown in Table 9.2 (grey boxes correspond to merit terms with no parameter to calibrate).

	Merit contribution	Weight	Parameters
Static terms	Window merit	ω_W	a, b, c
	Airmass merit	ω_A	α_A
	Avoidance merit	$\omega_{AV}, \omega_{AVMoon}$	S, S_{moon}
	Sky darkness		
	After-pick merit	V_{After}	m, z
	Forced time-out merit	V_{TOut}	
	Maximum number of observations		
	Phase-coherent merit	V_{Phase}	σ_p, β
	At-time merit	V_{At}	σ_A
	Not-before and not-after merits	V_{NotBA}	$\sigma_{NotB,A}$
	Terminal merit	V_T	σ_T
	User fairness	ω_{User}	α_u, α_w
	Proposal priority, TG & OB priority	$\omega_{Priority}$	$\alpha_p, \alpha_{TG}, \alpha_{OB}$
Dynamic terms	Merits for the state transition processes		$\rho_{Initl}, \rho_{InitTD}, \rho_{SO}, \rho_{Flat}$
	Telescope and dome slew-time merit	ω_{Slew}	a_t
	Instrument change-time merit	ω_{InstrC}	a_{ti}
	Weather		
	Seeing	V_{Seeing}	σ_{Seeing}
	Atmospheric transmission	V_{Atm}	σ_{Atm}
	MEIA and/or ARES		

Table 9.2: Algorithm parameters to calibrate

9.4.3 Scheduler interfaces

The scheduler has well defined interfaces with the OpenROCS and the database of the TJO-OAdM. These were described in Sect. 5.4.3. Input and output data are given in more detail in this section.

9.4.3.1 ISROCS input

There are two main data sets used as input for the scheduler:

- **Task execution constraints:** The parameters specifying the execution constraints for each OB are taken from the Projects database level of the DDROCS database. They were given at Table 9.1.
- **Housekeeping data:** Environment (weather, seeing and atmospheric transmission) and hardware and system status are provided by OpenROCS

It is important to mention that the current seeing is determined from two inputs: the on-image seeing measured from the mean FWHM of stars in the previous images, and the direct seeing measurements obtained from the Robotic Differential Image Motion Monitor (RoboDIMM, Aceituno 2004), located at the observatory. The RoboDIMM does not have a permanent and automatic building and real-time data will not be available until it can be operated in a fully unattended way. Atmospheric transmission, on the other hand, is obtained by the on-the-fly data processing described in Chapter 11 that provides a determination of the current photometric sky quality from the previous images.

9.4.3.2 ISROCS output

The scheduler outputs depend on the planning cycle, as already described in Sect. 9.4.1.3:

- Mid- and long-term scheduling returns a list with all the evaluated tasks. SFOM maximum value/s ($SFOM_{i_{max}, t_i}$) for the complete time interval are added to each task registry.
- Short-term planning output is composed by the selected task.

The output data is provided to the OpenROCS software suite.

9.5 Scheduler implementation

ISROCS implementation based on the presented design is currently under process with the collaboration of the engineering team. Its completion is planned for mid-2012.

An alternate and simple queue scheduler (as defined in Granzer 2004) was designed and implemented to start the robotic operations at the observatory, while the dispatch scheduling function is underway.

9.5.1 Development strategy

9.5.1.1 Development framework

The ISROCS software package is developed as a tool for the TJO-OAdM and other robotic observatories. It will be used to manage astronomical tasks to schedule the observatory operations, with the goal of maximizing the system efficiency. ISROCS was designed to be easily adapted for other observatories and this approach is considered as a hard constraint during its development. Among the features the software must have, we highlight some points here: its configuration must be doable in a straightforward manner, small number of changes shall be done to the interface packages during installation, and it must be easy to change a module to adjust ISROCS to the general control system and the instrument control software.

We are developing in parallel other scheduling solutions for two more facilities and we are minimizing the effort involved by using similar, or the same, code implementation by taking profit of the mentioned software modularity and easy configuration. The first one is the CARMENES instrument (Quirrenbach et al. 2010), whose requirements for task planning are very close to the TJO-OAdM project. This spectrograph will be installed at the 3.5m telescope of the Centro Astronómico Hispano Alemán (CAHA) observatory and will be dedicated to exoplanet detection. The second one, the CTA observatory (CTA Consortium 2010), has an inherent complexity that will probably lead to a slightly different solution. This scheduler will be used to schedule tasks for more than eighty telescopes on the same site working on sub-array configurations. A dispatch scheduling algorithm could be insufficient for this observatory and is being compared with other solutions (GTD and IEEC, 2011). It is foreseen that the efficiency of both projects will be significantly increased by using a task scheduler, as already proved by the experience for other telescopes (i.e., STELLA – Granzer et al. 2010, LT – Steele et al. 2004).

We are responsible to develop scheduler applications for three projects in the coming years (TJO-OAdM, CARMENES and CTA). This represents a clear benefit from and to the TJO-OAdM observatory.

9.5.1.2 Modules

The system modularity is reached by defining software components to be easily replaceable without changing other software components. Three main modules have to be developed for the ISROCS implementation: the sorting and selection service, the FOM algorithm and the interfaces.

➤ *Sorting and selection service*

We will implement this service as an independent application running under the control of the OpenROCS software suite (mid- and short-term planning) or executed by a user through the user interface for long-term planning.

This module includes the communication with the external interfaces. It is therefore very important to implement its features in a standard way to use ISROCS on other observatories. This will enable the exchange of the environment where this application is used in a transparent way.

The implementation of this module will be done in a suitable language providing a fast and reliable execution.

➤ **FOM algorithm**

Astronomical calculations

There are several astronomical values that the scheduler needs to calculate for static and dynamic planning. Specific libraries or routines will be used to compute the following data:

- Sidereal time.
- Sun altitude.
- Coordinates of the Moon (RA, DEC) and Moon altitude.
- Moon phase (% of illumination).
- Zenith distance and airmass of the target.
- Angular distance in degrees from specific coordinates to the actual pointing of the telescope and dome.
- Angular distance to Sun and Moon.
- Time remaining while the target is visible.
- Night-time remaining.
- Transform orbital parameters to Equatorial coordinates for Solar System objects.

New tools will be developed when no existing routines can be used or when using them reduces the software homogeneity to a critical level.

Calibration and simulation platform

The weights and parameters for the different terms of the FOM will be configured by means of simulations. These simulations will represent the observations as close as possible to the real case over several years, with a realistic target sample. Furthermore, the simulations will account for the accurate length of the nights, the position of the Moon for a given date, and will make use of the weather statistics at the TJO-OAdM of the past years.

➤ **Interfaces**

External interfaces with the control system (OpenROCS) and the database (DDROCS) will be done following the design given in Chapter 5 and Chapter 7. A standard communication protocol will be used.

Internal interfaces, mainly devoted to exchange data between the sorting and selection service and the algorithm, will guarantee that the system modularity is not compromised.

9.5.2 Queue scheduler

We designed and developed a queue scheduler in collaboration with the engineering staff to compute the observation night planning before the beginning of the night. A summary of the work done is presented here.

9.5.2.1 Design

➤ **Scheduler workflow**

The queue scheduler workflow (Fig. 9.14) starts reading the pending tasks from the Projects level database and provides a list of tasks to OpenROCS after selecting and sorting them. This list is intended to cover a full night operations of the TJO-OAdM telescope and is used by OpenROCS to execute the complete sequence unless non-routine events appear and abort it.

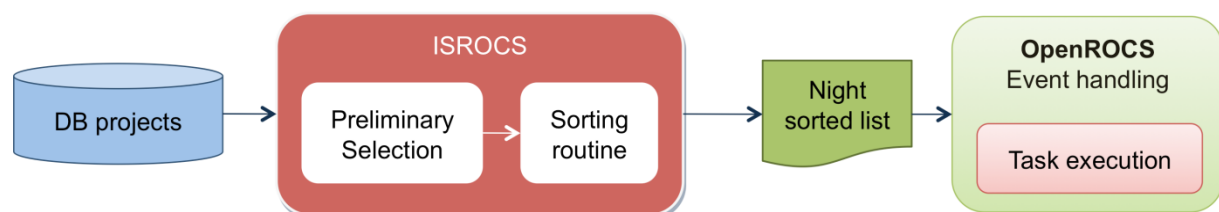


Fig. 9.14: Queue scheduler workflow

The aforementioned processes perform two main actions: a preliminary selection of targets and a sorting routine based on several criteria.

➤ *Preliminary task selection*

The DDROCS database (Projects level table) contains all TGs split in OBs, including both completed and pending ones. Those TGs that have not been completed are selected by the scheduler and the rest are skipped.

The calibration frames are organized into a single proposal with three TGs, one for each image type (darks, bias, and flats in each filter). This proposal is always active and defines the standard set of calibration images to acquire at the beginning and end of the night. This proposal was already specified in Section 8.3.3. Additional calibration frames required by a particular proposal are treated in a different manner, as they are considered together with the associated science targets of such proposal.

Science targets are organized into individual TGs based on project and OB definition (filter, exposure time, number of repetitions, etc.). Targets having fixed observing times that fall between sunset and sunrise are selected by the scheduler, as well as all the Flexible targets (regardless of their visibility). Once selected, the list of targets is sorted to have a static queue that is used by OpenROCS to know the task to execute at any time.

➤ *Sorting factors and process*

In order to compute the queue schedule, the tasks are first grouped by priority. The routine applies, then, several criteria in order to sort the tasks with the same priority:

- **Observation window:** This parameter is the first one computed for the selection of OBs. It is computed for those proposals that require monitoring programs, based on time or phase intervals. Two different scenarios are considered:
 - Targets with fixed starting times are programmed first and checks are performed to see if they have the required altitude and do not conflict with other fixed targets with the same priority. If there are conflicts, the first programmed target is given priority and the conflicting target is discarded.
 - Next, targets with flexible starting times fill in the gaps between the fixed targets. This second step starts at sunset and steps up in 1 minute intervals, determining if any targets can be observed at the given time and if they fit in the gaps between fixed targets. There are 10 priority levels and targets in level 10 are considered first.
- **Airmass:** The airmass of each target, computed in terms of its altitude, is calculated at the current time step and at the end of its observation period. The target is considered to be setting if the starting altitude is higher than the setting altitude. Thus, targets which are setting at the beginning of the night are programmed first. Targets which are rising at the beginning of the night are considered setting when they reach their maximum altitude. Thus by considering setting targets first, most of the targets are programmed at their optimal altitudes.
- **Distance to Moon and sky brightness:** Next, the distance between the Moon and the target is calculated and a determination of sky brightness is done based on lunar phase and altitude. A decision whether to consider the target is made based on its altitude, lunar distance and sky brightness compared to the user-defined requirements.

Once a target is considered, an attempt is made to insert it into the gaps between the already programmed observations. If it is inserted successfully, the scheduler goes on with the next target. If it cannot be inserted at the given time, an attempt is made on all consecutive time steps while the user defined observing requirements are still met. If the current time steps up to sunrise and the target still cannot be included, it is discarded.

After all the targets have been considered and the night observing list has been determined, a file containing the observation sequence is generated fulfilling the TALON standard format and it is used to guide the unattended operation. If there is enough time remaining before the start of the first target, the calibration images are programmed first. The science targets are programmed next, followed by another round of calibration images at the end of the night.

9.5.2.2 Implementation at TJO

The queue scheduler action is manually triggered some time before sunset. The subsequent action consists of three basic steps executed by a Python based script (scheduler.py):

- The first step is reading the MySQL database using SQL queries.
- The second step is the target selection and the queue scheduling computation. A second Python routine, containing various astronomical and time-keeping calculations (astrocalc.py), is called many times by the scheduler routine to do the required calculations.

- Finally, once the target list has been created, the scheduler routine calls the `slsMaker2.3.c` program that creates the `telrun.sls` file, containing the sequence to execute during the whole night in a TALON standard format. This file is the input needed by TALON to run in batch mode. After the scheduler has finished, OpenROCS detects a new `telrun.sls` has been created and proceeds to execute the night observations.

➤ **Short-term scheduling**

At the present time, short-term scheduling is not implemented. A simple approach is based on the same queue scheduling routine: if for any reason the observations are interrupted throughout the night (due to weather, power outputs, etc.), the queue scheduler routine is executed again when OpenROCS determines that conditions are right to resume observations. The MySQL database is then consulted and the rest of the night is planned based on the selection and sorting criteria explained before. This works fine at present, but has important limitations: the high time consumption to compute the new schedule or the non-consideration of the present environment and system status.

Chapter 10

Task Execution Control

The execution of tasks at the TJO-OAdM involves different scenarios where different workflows are defined. It is done in a completely unattended manner when nominal operations (science and calibration) are performed, although some maintenance or test operations may also be done automatically. Task execution mainly involves the observatory start up and power down routines, and the hardware layer control to perform data acquisition. We summarize here the main processes and applications running this important step in the data flow control layer.

10.1 Analysis and design

Most of the analysis and design of the task execution module was already discussed in previous chapters (Chapter 3 and Chapter 5) when defining the requirements and design for the other modules of the observatory control system (task preparation, database, scheduler, data process).

10.1.1 List of requirements

The task execution module was designed to fulfill the general requirements presented in Section 2.3 and the observatory task execution requirements given in the system Operational Design (Chapter 3). It follows a compilation of those ones considered to specify the features of this package:

10.1.1.1 Functional requirements

➤ *Operation mode*

The execution of tasks is done at the observatory working on the MEIA operation mode.

➤ *Operation tasks*

The operation tasks supported by the execution control are: science observations, calibration tasks, and engineering tasks (periodic maintenance and special testing). Science observations and calibration tasks (bias, dark and flat frames) have to be executed following the workflows given in Chapter 3.

➤ *System efficiency and reliability*

The operation efficiency is increased by reducing the response time required by the system to perform any action and to notify its result. The communication time between the task execution module and all the components it controls has to be minimized.

System reliability is achieved by using highly reliable communication channels and protocols. It is required to reduce the possibility of undesired or unexpected responses to the triggered actions.

➤ *Execution modes*

Two execution modes are required: **robotic**, where all the processes are executed without human interaction; and **manual**, to permit an experienced operator attain control of all the execution processes.

➤ Execution of tasks

The execution of tasks is carried out by controlling the hardware layer components to perform the observatory operations involved in data acquisition.

➤ Execution users

Two kinds of users have to be considered to enable automatic and manual control according to the user profiles defined in the general requirements: robotic (or ROCS), for unattended operation, and superuser, for engineering mode.

10.1.1.2 Boundary conditions

- **Input:** Execution task constraints are used to configure the hardware setup to perform the acquisition process.
- **Output:** The hardware operation feedback and the resulting image acquired are the outputs of this module.

10.1.1.3 Interface requirements

The task execution module has to interface with the OpenROCS software according to the control system design described in Chapter 5.

10.1.1.4 Non-Functional Requirements

At the analysis level, the non-functional requirements given in Section 2.3 were considered as recommendations.

10.1.2 Architecture overview

The task execution module is related with modules of the control system and of the Hardware layer (Fig. 10.1):

- **OpenROCS:** Execution actions are triggered by the event handling module of OpenROCS. Execution constraints of the task selected by the scheduler are delivered to this module to perform the image acquisition process according to the desired setup configuration. Housekeeping data are also provided to this module as they are used to define the focus axis position. The system response is transferred to OpenROCS for feedback and is used to process the image with the ICAT routine.
- **Hardware layer:** Scientific instruments (telescope, dome and instrument) are controlled using TALON, while embedded software and network protocols (SNMP) are used to communicate with the support devices necessary to power up and down the motors and electronics of this equipment.

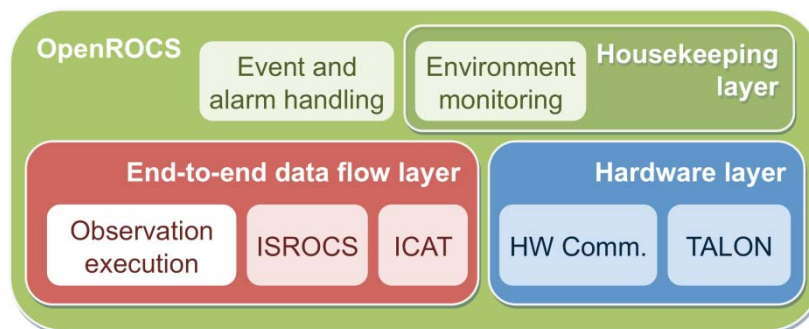


Fig. 10.1: Task execution module architecture

The task execution module is composed by the OpenROCS commands that trigger the actions at the Hardware level to perform all the required actions. The latter being devoted to:

- Change the system configuration state.
- Move the telescope, filter, focus and dome to the specified position, necessary for dome Flat field and science images, and acquire the images.

10.2 Implementation

The task execution implementation is based on the event-action management done by OpenROCS and the applications and low level software necessary to communicate with the scientific instruments (telescope, dome and instrument) and the support devices (computers and power control equipment – PDU, UPS) described in Chapter 4. They enable the system for performing the required actions on the Hardware components to:

- Initiate observatory devices.
- Shut-down devices.
- Control the task execution: telescope and dome pointing and tracking, positioning of the telescope primary mirror covers, filter, focus, dome shutter, and, finally, acquiring the image.
- Stopping or aborting a task.
- Monitoring the status.

Event-action pairs related with the task execution workflow control were extendedly described in Chapter 5. They were defined for the OpenROCS software to manage the execution module. It follows a description of the system configuration states for task execution, the actions triggered by OpenROCS to run the required processes and, finally, the applications in charge of executing them.

10.2.1 States for task execution

System configuration for each one of the possible states in which the system can be for its nominal operation was described for the ROCS design (Sect. 5.4). Science and calibration tasks, in particular, can only be executed with the system being on Calib, Flat or Science states. The configuration for these three states is extracted and shown in Table 10.1.

State	Calib	Flat	Science
Telescope			
Power (motors, electronics)	On	On	On
Electronics (CSIMC)	Ready	Ready	Ready
HA, DEC and Focus	On	Home & Pointing	Home & Tracking
Filter wheel	On	At position	At position
Mirror covers	On	Open	Open
Control software	On	On	On
Dome			
Power (electronics)	Off	On	On
Dome status	Off	Home & Closed & Pointing	Home & Open & Tracking
Control software	Off	On	On
Instrument			
Power (instrument)	On	On	On
Instrument status	Ready	Ready	Ready
Control software	On	On	On
Flats lamp	Off	On	Off
Control system			
Data flow SW	ToO	On/Off	On/Off
	DB	On	On
	ISROCS	On	On
	ICAT	On/Off	On/Off
	TALON	Off	On
	Camera	On	On
HK SW & HW	UPSs & PDUs		
		Dome redund.	Dome redund.
	Environment and System Monitors		
OpenROCS	On	On	On

SUB-ESTALL	On	On	On
SUB-REBEI (SUB-RUBIES)	On	On	On
SUB-SARGA	On	On	On
SUB-ALIS	On	On	On
SUB-AGER	On	On	On
Environment			
Indoor conditions	Ok	Ok	Ok
Outdoor conditions			Ok

Table 10.1: System configuration for the states where tasks can be executed.

Three kinds of calibration tasks are considered: Bias, Darks and dome Flats. Bias and Darks can be done on any of these three states as the required configuration is compatible with all of them. Flat calibration frames and science observations, instead, can only be done on Flat and Science states, respectively, which are mutually excluding due to the dome shutter and the flats lamp status.

10.2.2 Event-action pairs for task execution management

Event-action pairs related with task execution are copied here. Software modules used to execute the actions triggered are also specified and are described in more detail in the following section.

Event	Actions	Event code	Modules involved
New task required	Send query to OpenROCS for a new task	DF_SCHgettask	OpenROCS commands
New task received	Execute state transition, if necessary	DF_EXsettask	OpenROCS commands
	Perform task execution		
System state transition	Power up/down device/s by using the PDU outlets (telescope, dome, instrument, flat lamp, electronics)	DF_EXsetstate	HW Communication (SNMP)
	Start/stop control software		OpenROCS commands
	Initialize/park telescope, dome and instrument		TALON
	Open/close dome shutter		TALON
	Shutdown computer		OpenROCS commands
Perform task execution	Get the required environment data for telescope and focus positioning	DF_EXruncalib DF_EXrunscience	OpenROCS commands
	Point telescope and dome to specified position		TALON
	Move filter and focus to the specified position		TALON
	Acquire image with the instrument and store relevant HK data in the image header		TALON (Camera)

Table 10.2: Event-action pairs for task execution management.

10.2.3 Action control software

Actions triggered by the OpenROCS control system to acquire a calibration or science image are based on three different software tools that are subsequently described.

10.2.3.1 OpenROCS commands

System commands are commonly used to perform computer shutdown or to start/stop applications. Other commands are mainly based on the OpenROCS programming language. All the commands used to interface with the system modules are described in Chapter 6.

Manual execution of OpenROCS can be done by using the shell commands it incorporates. A superuser can interface with TALON and the power supply units by using the OpenROCS commands.

10.2.3.2 TALON

The TALON software is used to control the telescope, dome and instrument devices (Clear Sky Institute 2000). The main processes done with these Hardware devices follow specific sequences. Initialization (homing and warming-up processes), pointing and positioning are carried out by TALON. Some of the routines involved were modified or added by the engineering staff to improve the software performance. These modifications (Sanchez et al. 2011) derived into a new version of the software with higher reliability and stability that will be released following the GNU/GPL licensing policies.

A new interface module (Fig. 10.2) was developed also by the electronic and software engineers to control the operations with the science devices done by TALON. The solution found is based on interfacing directly to the communication channel (fifo) used by this software. This solution was adopted also to enable a reliable and quick feedback to OpenROCS. Having real-time information on the Hardware status is an important improvement of the control system.

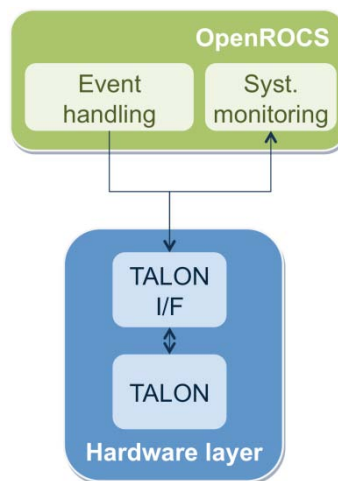


Fig. 10.2: TALON interface module

For a further step forward, an improved communication system with these devices will be implemented using INDI¹. INDI supports control, automation, data acquisition, and exchange among hardware devices and software frontends, providing a framework that decouples low level hardware drivers from high level front end clients.

10.2.3.3 SNMP commands

PdUs used to power up and down Hardware devices are accessed using SNMP protocol. SNMP (Simple Network Management Protocol) is an Internet-standard protocol for managing devices on IP networks.

¹ <http://www.indilib.org>

Chapter 11

ICAT for Data Processing

For an efficient operation, a reliable and fast data calibration and analysis software is crucial. This is needed to have an automatic treatment of the vast amount of images produced in a robotic observatory for scientific studies. The IEEC Calibration and Analysis Tool (ICAT) software described here was developed as a data processing tool for robotic observatories with the objective of managing astronomical images to extract good quality and relevant scientific information in real time. Because of its modularity and design, the software can be easily incorporated in the frame of a general observatory control system, under either batch or direct user control.

The main goal of ICAT is to contribute to the end-to-end data flow control layer of the TJO-OAdM with the extraction of high quality and high precision photometry, conforming on- and off-site processing and data storage. With that purpose, we define the image calibration and reduction procedure sequences, together with a quality control routine and other complementary packages that handle data storage and data archiving. The selected processing options and the internal and external interfaces with the rest of the control system are included in this package.

A description of the data processing steps and key aspects is given below.

11.1 Data processing analysis

The initial analysis performed to define the data processing architecture included the requirement specifications and the definition of the possible flux of processes for each planned scenario.

11.1.1 List of requirements

We designed the data processing package to fulfill the observatory requirements in terms of data extraction applied to the science images acquired. The list of general requirements specified for this package is presented below.

11.1.1.1 Operation mode

Data processing is specified to process the images obtained at the observatory working on the MEIA operation mode.

11.1.1.2 Type of images

It is required that ICAT can process calibration and science images.

11.1.1.3 Quality control

Image quality has to be checked to ensure compliance with the requirements of the user who submitted the proposal. The quality control results have to be compared also with a minimum quality threshold defined by the observatory. Those images not achieving the mentioned quality conditions has to be stored accordingly. The minimum set of parameters to verify for each science and calibration frames are:

- **File quality:** It has to include the verification of the image file name and the directory structure, the file size and image header keywords. Size is checked considering the detector definition and the image binning. Some keywords missing could make other quality checks impossible to be done.
- Calibration images – **Photometric quality:** The signal level and distribution throughout the detector chip has to be compared with the expected values.
- Science images – **Photometric and astrometric quality** for the targets identified in the image frame: The instrumental magnitudes have to be compared with the expected magnitudes (estimated according to the environment conditions), target shape, target and central image coordinates.
- **Observation pattern and environment quality:** A comparison of the real observation conditions with those required by the user has to be done.

11.1.1.4 Image calibration and analysis

A pipeline has to be developed for the reduction of calibration and science images and for the analysis of the calibrated science images, with the extraction of high precision photometric and astrometric data.

11.1.1.5 Image and data storage, data archiving

The data processing routine has to define and implement file naming and directory structure policies for image and data storage. Data archiving has also to be designed and developed compliant with the Virtual Observatory standards and compatible with the processes for the data retrieval to the end user.

11.1.1.6 Compression and backup

Compression and backup policies have to be defined and implemented in order to ensure high storage capacity. The images and the data have to be stored for a minimum period of time of two years.

11.1.1.7 Execution modes

Two modes of execution have to be considered:

- **On-the-fly:** A “quasi” real-time execution is necessary to serve as an input for the dynamic scheduling routine task.
- **Off-line:** A final execution using all the calibration images obtained has to be performed upon finishing the night sequence. The overall process is executed for all the images, which implies reducing the science images using more accurate master calibration frames.

11.1.1.8 Execution users

Two kinds of users have to be considered to enable automatic and manual control according to the user profiles defined in the general requirements: robotic (or ROCS), for unattended operation, and superuser, for engineering mode. The latter has the capability to adjust the parameters of the image reduction and analysis.

11.1.1.9 Boundary conditions

Data processing has the following inputs and outputs:

- **Inputs:** image files, observation constraints defined by the user, control system setup configuration.
- **Outputs:** calibrated images, extracted data in ASCII table files, data quality results, data archiving in the observatory database, backup storage, logging for all the processes.

11.1.1.10 Interfaces

Data processing has to have the following interfaces defined:

- **OpenROCS** software infrastructure: The execution of the package, including logging, has to be integrated into the observatory software infrastructure in order to contribute to the end-to-end data flow scheme. Processing routines have to be triggered from the event and alarm management included in the OpenROCS suite (DF_PRset, Sect. 5.4.4).
- **Database for output data archiving:** It includes the archiving of the image acquisition descriptors (selected keyword) and the extracted data with the generation of a new register into the database, at the Observations level. Data and images have to be stored to be subsequently retrieved by the user.

- **Database for proposal data input:** The observation constraints defined by the user have to be accessed by the data processing routines to evaluate if the observation scheduling requirements were fulfilled.
- **System file architecture** for file storage on- and off-site: The interface with the on- and off-site file architecture has to be defined and implemented to manage file storage.
- **User interface:** Superusers should be able to execute an off-line image reduction and analysis to test other reduction and analysis configurations (different calibration files or analysis software or analysis parameters). This interface could be made accessible off-site to the general users.

11.1.1.11 Development

- **Modularity:** The data processing package has to be programmed in a way to be used with different acquisition hardware and setups (camera detector, control software, etc.) in a straightforward manner.
- **Programming languages:** Languages ensuring minimum execution times have to be chosen.
- Use of **existing applications:** Existing software applications, with a well-known performance and under Open Source license, have to be used to execute the image reduction and analysis. Some applications are considered: IRAF, SExtractor, CFITSIO.
- **Multithreading:** Parallel processing of different image frames has to be possible.

11.1.1.12 Performance

- **Execution time:** Execution time has to be minimized as the data processing results have to be used as a feedback control tool for the data flow routine. It is especially relevant for on-the-fly execution mode. The reduction and analysis of a science image has to be doable in less than 5 seconds. The reduction of the calibration images could take longer.
- **Calibration goal:** Master calibration frames have to ensure that photometric time-series can be obtained with a RMS value smaller than 10^{-3} . This will permit the extraction of the highest precision photometry for the data collected by TJO-OAdM.

11.1.2 Flow diagrams

Data processing is executed each time a new image is acquired, resulting from the observation execution process. Different scenarios are evaluated by the data processing routines depending on the type of image acquired and the observatory state at the acquisition time:

1. A single image frame is obtained:
 - 1.1. Calibration image: The system executes the quality control, the storage and the archiving processes.
 - 1.2. Science image: The system executes the image calibration and analysis and the quality control, the storage and the archiving processes.
2. The single calibration frame is the last image of a calibration image sequence:
 - 2.1. A calibration sequence is executed before the night starts (end of astronomical twilight): The system executes the quality control, archiving and storage processes for the single frame and starts the reduction of all the calibration images to obtain the master calibration frames. At the end of such process, master frames are also evaluated to check their quality and stored for subsequent use.
 - 2.2. A calibration sequence is executed after the end of the astronomical twilight in the morning: The same process described in 2.1 is executed. When the master frames are obtained, the system starts the final reduction, analysis and quality check for all the science images acquired during the night.

The schemes below (Fig. 11.1) describe the work flow for each of these scenarios:

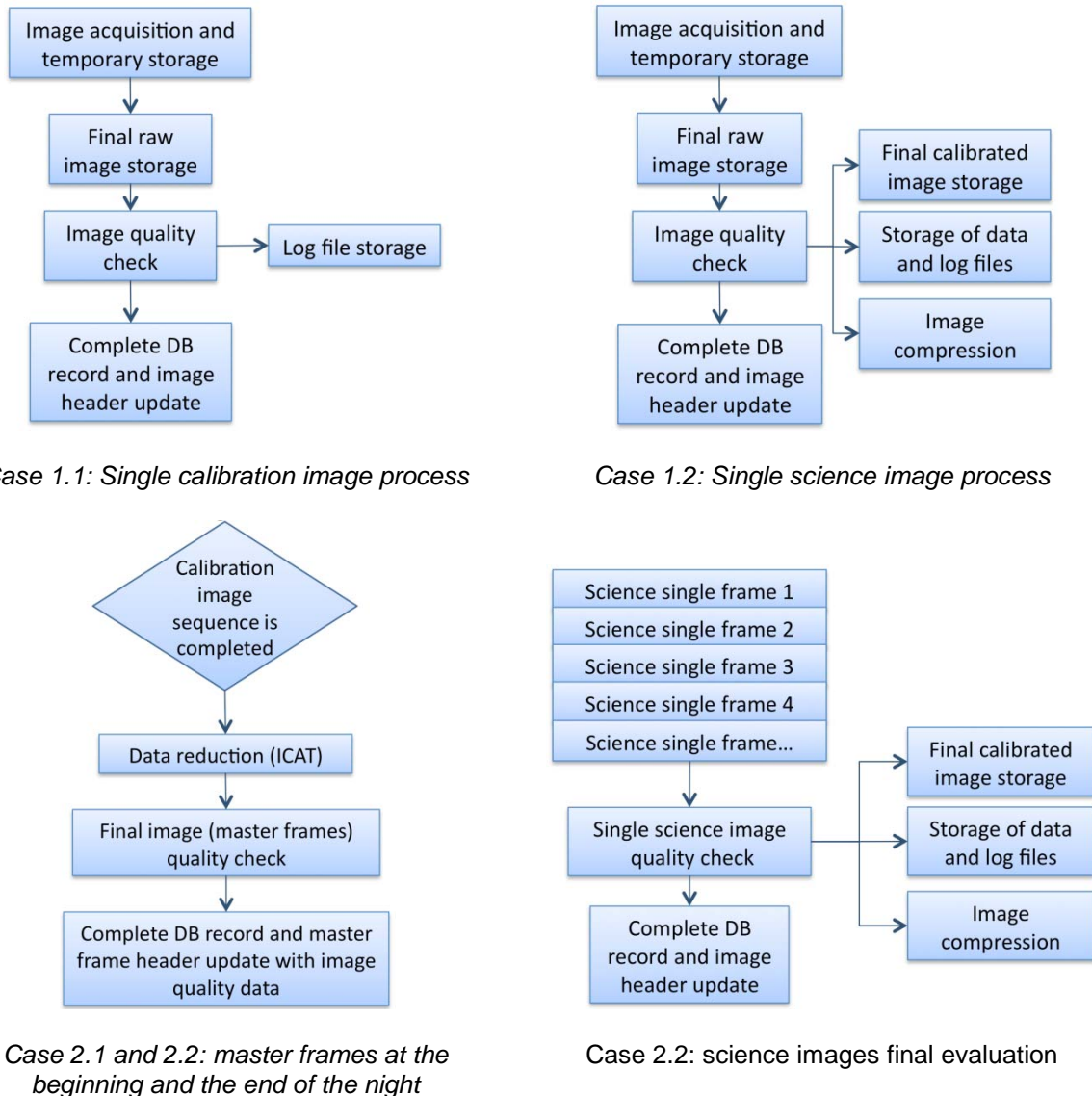


Fig. 11.1: Data processing work flow schemes

11.2 Data processing design

We designed the data processing package based on the list of requirements and the work flow scenarios described.

11.2.1 Automatic execution

11.2.1.1 Multi-observatory

Data processing is easily adapted to work on a different instrument set-up, just with the modification of a configuration file at the time of installation. Few parameters are to be set up according to the system: system configuration (file system architecture, detector, header keywords, etc.), minimum number of calibration images required for the reduction process, analysis parameters (analysis software...), default image quality ranges, etc. After that, automatic execution is possible.

11.2.1.2 Execution modes

This package executes automatically the on-the-fly and the off-line reductions described in Sect. 11.1.1. Although automatic execution is its main functionality, we designed it to be run by manual control, enabling the user to modify a number of parameters to adjust the reduction process to different criteria.

11.2.2 Data processing use cases

Several use cases were considered for the development of the data processing package and were the basis for its implementation. In some cases, two different responses were defined depending on the type of user (manual or automatic) that is executing the process:

11.2.2.1 System initialization

This use case is designed for the selection of parameters to define the execution process. The main goal of this use case is to enable the processing routines to run with different system configurations in a transparent way. The parameters to configure were already mentioned in Sect. 11.2.1.1. A default configuration is set for the automatic and manual execution, although it can be changed interactively in the latter case.

11.2.2.2 Image reduction

The calibration and science image reduction use case is divided into several sub-cases:

- **Number of calibration images and generation of list of images to reduce:** It revises the number of available calibration images (bias, dark and flat per each filter) in the input directory. This process also creates lists of images grouped by the type of image which are used by several processes. The software proceeds as follows when the number of calibration images is insufficient:
 - Manual: The software warns but does not stop the process if the number of images is less than a specified value. It aborts the process when any of bias, dark or flat-field calibration images are missing.
 - Automatic: The software copies the final master frames from a previous night if the number of images is lower than a configured value. It starts the search in the immediately previous night and it goes back until it finds a night with all the necessary reduced calibration images (bias, dark and flat-field). It copies all the master calibration frames and it does not use the calibration images of the current night for the reduction of the astronomical images. When not all the necessary reduced calibration images are found in past night directories, it uses the individual calibrated images instead of the master images until it reaches the minimum number required.
- **Reduction of calibration images to obtain master calibration frames:**
 - Manual: User can choose the images to reduce and the use of the dark current correction.
 - Automatic: Calibration images available for the current night directory are used to obtain master calibration frames. Only good quality images are considered. The process is executed just after the acquisition of the last calibration image in both evening and morning calibration runs. The output files resulting from the reduction are: master bias frame, master dark frame, and master flat frames. Master flat frames are obtained for all the filters used.
- **Checking the presence of master calibration frames:** The software checks for the existence of master calibration frames in the output directory before starting the reduction of the astronomical images.
 - Manual: It does not continue the execution if they do not exist.
 - Automatic: It looks for them in past nights directories.
- **Checking the presence of reduced astronomical images:** The software deletes the reduced science images from past processes if they exist in the output directory in order to avoid interferences. It is only applied when the new process is going to generate new files with the same name. If new images have different names, the old ones are kept.
- **Reduction of science images:** This use case involves the reduction of the science images using the reduced calibration master frames.
 - Manual: The user selects the images to reduce and decides if the dark current correction has to be applied or not.
 - Automatic: Dark current correction is used or not depending on its contribution to the image. Therefore, an exposure time threshold is preconfigured to define when it has to be applied. The instrument commissioning will determine this time threshold.

11.2.2.3 Image analysis

This use case is defined for the science image analysis to obtain high quality astrometric and photometric data with minimum execution time. The process uses different analysis software packages (DAOPHOT/IRAF, SExtractor, etc.) and measuring techniques (PSF or aperture

photometry). It returns the targets identified in the image, together with their Equatorial coordinates and instrumental magnitudes. The analysis output file format is the same regardless of the process configuration used. The stored data and their format enable an easy quality check and delivery to the end user.

- Manual: The user can select the analysis configuration and, then, execute the process.
- Automatic: A preconfigured analysis is executed.

Two sub-cases are considered:

- **Astrometric calibration:** This process is executed in two phases. The first one is run before the analysis of the science calibrated image. It executes a routine (called WCS) to calculate the position of the image field on the sky and to create new entries in the image header describing the corrections to apply to find the RA and DEC sky coordinates. The second phase is executed when the photometric analysis has finished. It applies the correction obtained in the first phase to transform the X and Y CCD coordinates to RA and DEC sky coordinates.
- **Crowded area analysis:** This sub-case is only considered for the automatic execution. A first analysis is done to identify high crowding areas in the field. A second analysis iteration is performed on these areas by using PSF photometry to improve the analysis results.

11.2.2.4 Image extra-analysis

This use case is defined to extract relevant scientific information from the data obtained with the image analysis. It performs a supplementary treatment of the photometric and astrometric results (i.e., differential time-series photometry).

11.2.2.5 Quality check

This use case is defined for the automatic evaluation of the image quality. It is divided into several steps to check different quality parameters according to the requirements given in Sect. 11.1.1.3: file quality, photometric quality, astrometric quality and observation quality.

It is performed on every calibration and science image. The results obtained are used as a feedback for the control system and, in particular, by the scheduling application. Calibration images identified as good quality frames are then used to compute the master calibration files.

The reduction and analysis use case for the science images is a child process of the quality control routine. It is executed before the photometric and astrometric quality control. The reduction is done using the available master calibration images in the output directory.

11.2.2.6 On-the-fly and off-line processes

On-the-fly and off-line processes are triggered by the automatic data processing routine to perform the image quality check. The control system generates the configuration files containing all the parameter values required for the execution.

11.2.2.7 Data archiving

Data processing also executes the archiving of the extracted photometric and astrometric data in ASCII table files in the observatory database. Archiving is done following the Virtual Observatory standards.

11.2.2.8 File compression and backup

The image files are compressed just after the final analysis.

On-site and off-site backup is also included in this use case. Raw images and final data are stored with sufficient information to recover the same final data when executing the data processing routine at any time.

11.2.3 Architecture overview

The architecture design of the ICAT data processing package is shown in the following scheme (Fig. 11.2):

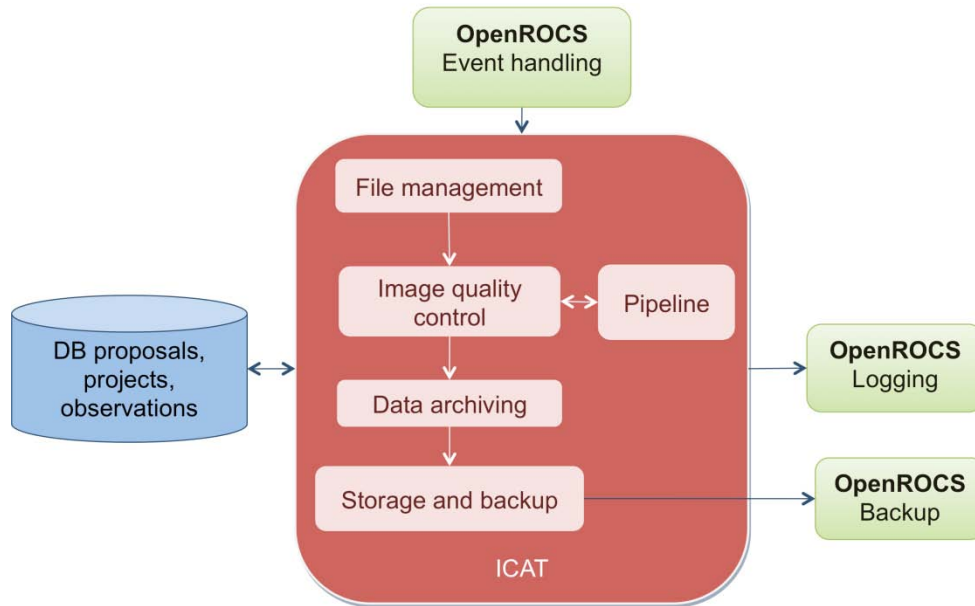


Fig. 11.2: ICAT architecture

The ICAT service package is divided in the following modules:

- **File management:** Module in charge of the definition of data types and levels, naming policies and header keywords.
- **Image quality control:** It includes the routines to evaluate the quality of each frame acquired in terms of file parameters, photometric and astrometric values and conditions (acquisition pattern and environment).
- **Image calibration and analysis pipeline:** This module performs the science and calibration image reduction and science image analysis.
- **Data archiving:** The archiving of data and the generation of new registries into the observatory database and into the logging system for all the images is handled with this module. It also updates the OB status in the observatory database.
- **Storage and backup:** It is responsible for the file storage process on- and off-site according to the predefined policies for image and data backup.

We describe each module design in the following sections.

11.2.3.1 Interfaces

The modules interfacing with the data processing service are shown in Fig. 11.3.

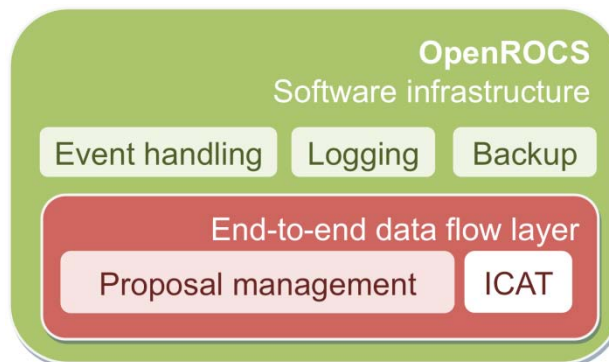


Fig. 11.3: ICAT interfacing modules

➤ Interface with OpenROCS

Data processing interfaces mainly with the software infrastructure of the observatory, the OpenROCS software suite. OpenROCS is in charge of executing the data processing complete run every time a new image is acquired, by producing a triggering event (DF_PRset). ICAT notifies OpenROCS the end

of the data process execution and a new event is generated (DF_PREnd) to update the task status in the observatory database and to perform the image backup.

➤ **Interface with Database**

ICAT interfaces also with the database (DDROCS). Data in the database are necessary to compare the acquisition conditions (execution pattern and environment) with those required by the user or by the observatory staff. The interface with the database is also necessary to archive the obtained data and to update the OB and its corresponding proposal status.

11.3 ICAT detailed design

11.3.1 Process scheme

The image data process is described in Fig. 11.4.

Once a new image is acquired, the on-the-fly data processing starts and follows the sequence given here:

- **Final raw image storage:** The image is acquired by the task execution control subsystem (SUB-ESTALL) and, then, it is moved to the final storage subsystem (SUB-ALIS).
- **Start image processing routine:** A control system (OpenROCS) event triggers the general workflow execution using a new thread. Multiple threads can coexist.
- **Read observatory configuration parameters:** The general configuration parameters are read from a single configuration file at the beginning of the process. All the subroutines access the same configuration file to avoid inconsistencies.
- **File quality check:** The file name and storage directory are initially checked. Keywords in the image header, file size and format are evaluated. In case some relevant keywords are not there, the process is aborted as it makes further image quality process unnecessary.
- **Photometric quality check:** Different scenarios are defined when evaluating the photometric quality of a particular image:
 - Calibration images: Level and distribution of counts recorded on the image frame are computed and compared with the configured ranges or with reference images.
 - Science images: The images are calibrated and analyzed with the pipeline. The obtained photometric data are evaluated.
- **Astrometric quality check for science images:** Astrometric data resulting from the image analysis performed by the pipeline are evaluated.
- **Pipeline:** The pipeline is called to calibrate, analyze and compress the science images or to compute the master calibration frames. It is executed just before the photometric and astrometric quality check of the science images.
- **Observation pattern and environment quality check:** Image time and environment conditions at the time of acquisition are read from the image header.
 - Comparison with observatory and user specified ranges: The range of values defined by the user (specified in the proposal submission) and the default values considered at the observatory are compared with the real acquisition conditions. The constraints defined by the user are read from the observatory database.
- **Determination of the image quality keywords:** The quality keywords are determined. They summarize the quality of the image based on each of the evaluation steps.
 - Update image header: The quality keywords are added to the image header.
- **Finish image data process:** The last step for the data processing routine is to update the OB status and produce a new register with the main image parameters in the Observations level of the database. This register is used in the end as a notification to the final user and a way to provide access to the data.
 - A query to the database is executed to insert the new register.
 - A query to the database is executed to update the OB and the proposal status. The sorted list of tasks for the night associated to the mid-term scheduling (see Chapter 9) is updated.

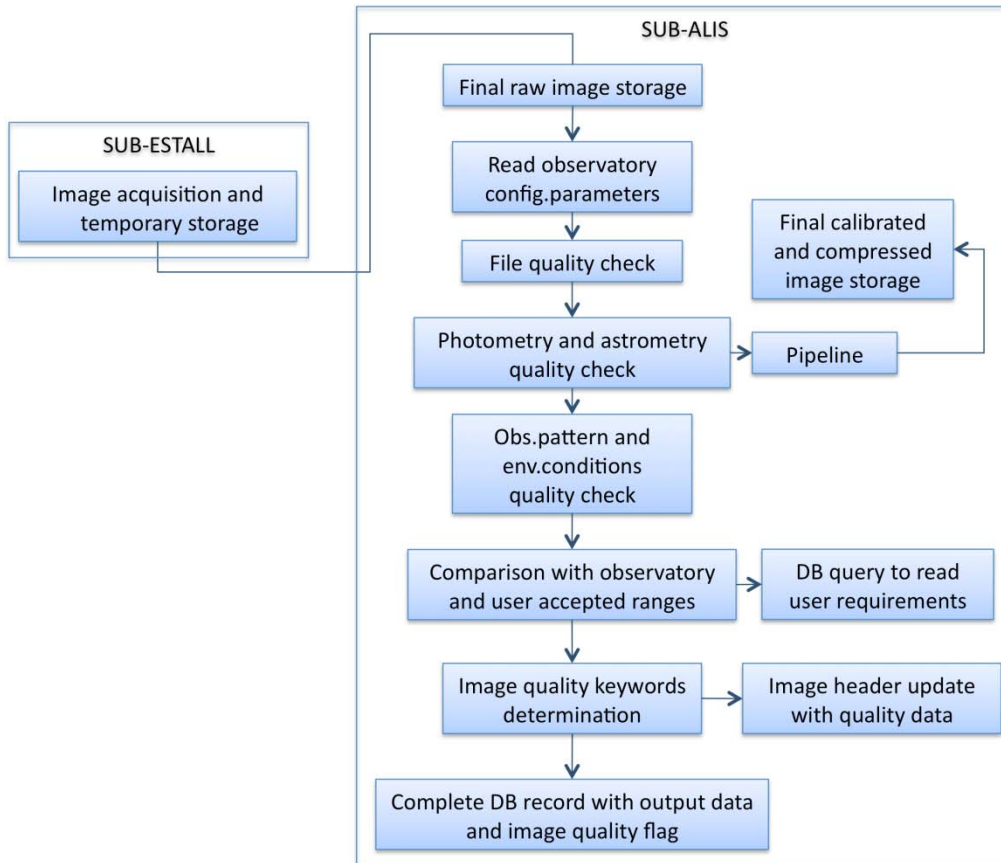


Fig. 11.4: Image data process work flow

Alternative workflows are designed in case the image under process is the last image of a complete calibration sequence (the image frame is flagged accordingly). Different scenarios are considered according to the acquisition time, as described in Sect. 11.1.2:

- **Evening:** The preliminary master calibration frames are obtained with the pipeline and their quality is evaluated.
- **Morning:** The final master calibration frames (based on evening and morning calibration images) are obtained with the pipeline and processed. Finally, the off-line processing of all the science images is executed.

Each of these process steps is described in detail in the following sections.

11.3.2 File management module

Several aspects of file configuration and management are specified to establish the way in which the entire processing package extracts and uses the relevant information from the images taken.

11.3.2.1 Data types and data levels

Several types and levels of data can be distinguished depending on the information they contribute to the data processing. The definitions given here specify where the process takes place and how the data are stored.

There are four types of data at the TJO-OAdM observatory:

- **Science:** They include the FITS images and photometric and astrometric data of the selected sky fields.
- **Calibration:** The calibration FITS images are used for science image calibration.
- **Housekeeping:** They include the environment data useful to determine image quality and to extract possible trends that could affect photometric precision.
- **Logging:** Log message files are produced with science and calibration image processing details.

In order to optimize the data storage capacity, different levels are defined (Table 11.1). Each one has an associated policy for its storage and its use during the on-the-fly or final processing:

Type	Level	Description	Process	Final storage
Science	0	raw images	on site	on / off site
	1	calibrated images		off site
	2	extracted data		on / off site
Calibration	0	raw images	on site	on / off site
	1	corrected images		none
	2	master frames		on / off site
Housekeeping	0	registered data	none	on / off site
Logging	0	registered data	none	on / off site

Table 11.1: Data types and levels

Image processing routines modify the image size: the original FITS files produced at the observatory working with the MEIA mode are changed from 16 bits to 32 bits, thus roughly doubling their size. For this reason, it is important to choose accurately which images are stored, and which can be deleted, to make later reprocessing possible with guarantee to reproduce the results. Log files contain the trace of every process. Science calibrated images, for instance, are obtained to extract photometric and astrometric data on site, but they do not need to be stored there because they are not used further and they can be easily reproduced.

Master calibration frames are kept on site because they are used during the entire night observation period to evaluate the system stability. In case of high stability, initial calibration frames could be eliminated gradually. Intermediate images to obtain the master frames do not need to be stored locally.

11.3.2.2 FITS images and headers

The TJO-OAdM images are stored in FITS format (Wells et al. 1981, Pence et al. 2010), supported for most common image reduction software, like IRAF¹.

The FITS image header represents a key element for data processing. The keywords included in the header that are required for the TJO-OAdM are specified in this section following the NOAO standards (NOAO FITS Keyword Dictionary, Version 1.0 – January 2000²). This is a common standard format for most astronomical observatories.

Header keywords at the TJO-OAdM are mainly grouped in eight typologies to describe all the characteristics of the image:

- **Image:** type, axis, bits per pixel, etc.
- **Site:** location, environment, sky description
- **Telescope:** telescope, filter, dome
- **Instrument:** detector, name, gain, readout noise, pixel scale
- **Observer** (or proposer)
- **Observation:** type of image, proposal code, target group identifier and observation block identifier
- **Object:** coordinates, pointing correction
- **Processing:** pipeline, quality

According to the NOAO dictionary, there are five fields to describe a keyword entry:

- **Keyword:** name of the keyword
- **Logical name**
- **Values:** range of values that the keyword can take
- **Description:** short description of the keyword

The value formats in the value entry have the form "%W.DC", where W is the field width, D is the number of decimal places or the number of digits of precision, and C is the format code. The W and D fields are optional. The format codes are as follows (Table 11.2):

¹ <http://iraf.noao.edu/>

² <http://iraf.noao.edu/projects/ccdmosaic/imagedef/fitsdic.html>

b	boolean (T or F)
c	single character (c or '\c' or '\0nnn')
d	decimal integer
e	exponential format (D specifies the precision)
f	fixed format (D specifies the number of decimal places)
g	general format (D specifies the precision)
h	hms format (hh:mm:ss.ss, D = no. decimal places)
H	h format with value divided by 15
m	minutes, seconds (or hours, minutes) (mm:ss.ss)
M	m format with value divided by 15
o	octal integer
s	string (D field specifies max chars to print)
u	unsigned decimal integer
x	hexadecimal integer
z	complex format (r,r) (D = precision)

Table 11.2: Keyword format codes

Conventions for W (field width) specification are given here:

- n: Right justify in field of n characters, blank fill
- n: Left justify in field of n characters, blank fill
- 0n: Zero fill at left (only if right justified)
- absent, 0: Use as much space as needed (D field sets precision)

The particular mapping applied at TJO-OAdM is shown in Table 11.3 with the list of the required keywords to use. Rows with grey color correspond to those keywords not originally included by the acquisition software (TALON). A description of how these new keywords are added in the image header is given in the ICAT implementation section (Sect. 11.4.9) and an example of header is shown in Sect. 11.4.11.

NOAO Keyword	Logical name	Type	TJO-OAdM keyword	Values	Description
SIMPLE	Image.simple	%b	SIMPLE	T	Standard FITS
BITPIX	Image.bitpix	%d	BITPIX	16	Bits per pixel
NAXIS	Image.naxis	%d	NAXIS	2	Number of dimensions
NAXISi	Image.naxis[i]	%d	NAXIS1	Positive Number (2048)	Number of columns
			NAXIS2	Positive Number (2048)	Number of rows
BZERO	Image.bzero	%g	BZERO	32768	Pixel zero factor (Real = Pixel*BSCALE+BZERO)
BSCALE	Image.bscale	%g	BSCALE	1	Pixel scale factor
			OFFSET1	0	Camera upper left frame x
			OFFSET2	0	Camera upper left frame y
			XFACTOR	1	Camera x binning factor
			YFACTOR	1	Camera y binning factor
EXPTIME	Detector.Ccd[n].Amp[n].Exp. exptime	%g	EXPTIME	Any	Exposure time, seconds
OBSERVAT	Site.observatory	%4s	ORIGIN	'OAdM'	Name of the observatory
TELESCOP	Telescope.name	%5s	TELESCOP	'TJO'	Name of the

					telescope
OBSERVER	Observer.name[1]	%s	OBSERVER	'ROCS', 'Superuser', 'Operator'	System operator(s)
PROPID	Observer.proposalid	%5d	PROPCODE	Positive integer	The unique observatory proposal identification
		%d	TARGETID	Positive integer	TG ID. The unique target group identification.
OBSID	Observation.obsid	%d	BLOCKID	Positive integer	OB ID. The unique observation identification. This serves to identify all data from the same observation.
OBJNAME or OBJECT	Object[1].name	%s	OBJECT	'Obj.Name' 'Bias frame' 'Dark frame' 'Flat frame'	IAU standard object name or calibration image type
IMAGEID	Observation.imageid	%d	IMAGEID	'Any'	The image identification when there are multiple images within an observation block
OBSTYPE	Observation.Obstype.type	%s	OBSTYPE	'SCIENCE' 'BIAS' 'DARK' 'FLAT'	Image type
		%d	PRIORITY	0	Scheduling priority
INSTRUME	Instrument.name	%s	INSTRUME	'MEIA', 'ARES'	Name of the instrument
PIXSCALEi	Detector.Projection.pixscale[i]	%16.13g	CDELTA1	Any	RA step right, degrees/pixel
			CDELTA2	Any	Dec step down, degrees/pixel
GAIN	Detector.Ccd[n].Amp[n]. Controller.gain	%g	GAIN	Any	Amplifier gain in electrons per analog unit
RDNOISE	Detector.Ccd[n].Amp[n]. Controller.readnoise	%g	RDNOISE	Any	CCD readout noise in RMS electrons
MJDEND	Detector.Ccd[n].Amp[n].Exp.Expend.mjd	%16.8g	JD	Any	Time at end of exposure
DATE-OBS	Detector.Ccd[n].Amp[n].Exp.Expstart.date	%10s	DATE-OBS	'CCYY-MMDD'	UTC Date
TIME-OBS	Detector.Ccd[n].Amp[n].Exp.Expstart.time	%h	TIME-OBS	'HH:MM:SS.ss'	UTC Time
LST	Detector.Ccd[n].Amp[n].Exp.Expstart.lst	%h	LST	'HH:MM:SS.ss'	Local sidereal time
CAMPANn	Camera.Sensors.posangle[n]	%g	POSANGLE	'±DD:MM:SS'	Position angle, degrees, +W
		%g	LATITUDE	' 42:03:06'	Site Latitude, degrees +N
		%g	LONGITUD	' 0:43:47'	Site Longitude, degrees +E
TELALT	Telescope.Altaz.altitude	%0h	ELEVATIO	' DD:MM:SS'	Degrees above horizon
TELAZ	Telescope.Altaz.azimuth	%0h	AZIMUTH	'DDD:MM:SS'	Degrees E of N
HA	Telescope.hourangle	%h	HA	'±HH:MM:SS.ss'	Local Hour Angle
		%h	RAEOD	'HH:MM:SS.ss'	Nominal center Apparent RA
		%h	DECEOD	'±DD:MM:SS.ss'	Nominal center Apparent Dec
RA	Coordinate.ra	%h	RA	'HH:MM:SS.ss'	Nominal center J2000 RA

DEC	Coordinate.dec	%h	DEC	'±DD:MM:SS.ss	Nominal center J2000 Dec
OBJRA or RA	Object[1]. Coordinate.ra	%h	OBJRA	'HH:MM:SS.ss'	Target center J2000 RA
OBJDEC or DEC	Object[1].Coordinate.dec	%h	OBJDEC	'±DD:MM:SS.ss	Target center J2000 Dec
EPOCH or MJD-OBS	Coordinate.epoch	%g	EPOCH	2000	RA/Dec epoch, years (obsolete)
EQUINOX 2000.0	Coordinate.equinox	%g	EQUINOX	2000	RA/Dec equinox, years
CAMTEMnn	Camera.Sensors.temperature[n]	%3.0g	CAMTEMP	±deg C	Camera temp, C
		%10.8g	RAWHENC	Any	HA Encoder, rads from home
		%10.8g	RAWDENC	Any	Dec Encoder, rads from home
		%10.7g	RAWOSTP	Any	Focus Motor, rads from home
TELFOCUS	Telescope.focus	%9.3g	FOCUSPOS	Any	Focus position from home, μm
TELSTAT	Telescope.status	%s	TELSTAT	'Home', 'Tracking', 'Guiding'	Telescope status
FILTERnn	Filter[n].name	%8s	FILTER	'U', 'B', 'V', 'R', 'I'	Filter code (or filter wheel position if empty)
FILPOSnn	Filter[n].position	%9.7g	RAWISTP	Any	Filter Motor, rads from home
DOMSTAT	Dome.status	%s	DOMESTAT	'Open', 'Closed', 'Tracking'	Dome status
AIRMASSn	Observation.airmass[n]	%g	AIRMASS	Any	The computed airmass
WEATHER	Site.weather	%s	WEATHER	'Clear', 'Partly cloudy', 'Cloudy', 'Raining'	Description of the weather condition, from cloud sensor and rain detector data.
PHOTOMET	Site.photometric	%s	PHOTOMET	'Phot', 'Non-phot'	Atmospheric transmission conditions
SEEINGnn	Site.seeing[n]	%4.2g	SEEING	Any	Seeing estimates specified as the stellar full-width at half maximum in arc seconds
ENVTEMnn	Site.Environment.temperature[n]	%5.2g	WXTEMPn	Any	Air temperature (sensor n, C)
ENVPREnn	Site.Environment.pressure[n]	%7.3g	WXPRESn	Positive Number	Atm. pressure (sensor n, mB)
ENWINnn	Site.Environment.windspeed[n]	%g	WXWINDSPDn	Positive Number	Wind speed (sensor n, kph)
ENVDIRnn	Site.Environment.winddir[n]	%3.0g	WXWINDDIRn	Any	Wind dir. (sensor n, degs E of N)
ENVHUMnn	Site.Environment.humidity[n]	%3.0g	WXHUMIDn	Positive Number	Rel. humidity, (sensor n, %)
CTYPEi	Detector.Ccd[n].Amp[n].Wcs.ctype[i]	%s	CTYPE1	'RA_TAN'	Coordinate type for image world coordinates (WCS). Columns are RA
			CTYPE2	'DEC_TAN'	Coordinate type for image world coordinates (WCS). Rows are Dec
CRVALi	Detector.Ccd[n].Amp[n].Wcs.crval[i]	%11.7g	CRVAL1	Positive Number	Reference RA coordinate corresponding to the image reference pixel coordinate (WCS). RA at CRPIX1, degrees

		%12.8g	CRVAL2	Any	Reference DEC coordinate corresponding to the image reference pixel coordinate (WCS). Dec at CRPIX2, degrees
CRPIXi	Detector.Ccd[n].Amp[n].Wcs.crpix[i]	%g	CRPIX1	Positive Number (1024)	RA reference pixel index, 1-based (WCS)
			CRPIX2	Positive Number (1024)	Dec reference pixel index, 1-based (WCS)
CROTAx (IAU Standard)		%g	CROTA1	0	Rotation angle, in degrees, of actual axis n of the array from the coordinate type given by CTYPE _n (WCS)
			CROTA2	Any	Rotation N through E, degrees (WCS)
		%g	AMDX0	Any	Astrometric parameter alpha_c (radians) (WCS)
		%e	AMDX1	Any	Astrometric parameter a1 (WCS)
		%g	AMDY0	Any	Astrometric parameter delta_c (radians) (WCS)
		%e	AMDY1	Any	Astrometric parameter b1 (WCS)
PIPELINE	Processing.pipeline.name	%s	PIPELINE	'ICAT1.0'	Name of processing pipeline applied
		%d	IMGQS	See Sect. 11.3.3	Image quality: summary (4 digits for science and 3 for calibration images)
		%s	IMGQF	See Sect. 11.3.3	File quality parameters
		%s	IMGQP	See Sect. 11.3.3	Photometric quality parameters
		%s	IMGQA	See Sect. 11.3.3	Astrometric quality parameters
		%s	IMGQC	See Sect. 11.3.3	Execution pattern and environment conditions parameters
		%s	IMGQE	See Sect. 11.3.3	Image quality: errors
COMMENT		%s	COMMENT		
END		%g	END		

Table 11.3: Detailed list of keywords

11.3.2.3 File-naming system

A naming convention for all files is applied in the data process package in order to have a standardized naming format. These files are usually FITS format, and thus take the .fits extension at the end of the file name. The name itself is broken into three parts, separated by underscore symbols:

TJO[date]_[R]_[sfx].[fits or dat]

A description of each part is given here:

➤ **TJO[date]**

It contains the acquisition date and time of the observation.

- Values: fixed term (TJO) and JD number (format: floating point with 5 decimal digits to achieve 1 second precision)

➤ **[R]**

It describes the type of register according to the storage and backup policies (Sect. 11.3.6).

- Values:
 - R: on-the-fly copy
 - B: on-site backup copy
 - O: off-site backup copy

➤ **[sfx]**

This identifier corresponds to the image type.

- Values:
 - imr: raw science image
 - imd: dark corrected image
 - imb: bias corrected image
 - imf: flat corrected image
 - imk: bias and dark corrected image
 - iml: bias, dark and flat corrected image
 - imm: bias and flat corrected image
 - imt: preliminary (on-the-fly) reduction
 - imc: final (off-line) reduction
 - cbs: bias
 - cth: dark
 - cfX: flat, X=filter
 - trl: trailer file, output data, ASCII tables
 - pdq: post observation summary and data quality comment

➤ **File name extension**

- Values: “.fits” extension for image files and “.dat” extension for trl and pdq files.

11.3.3 Image quality control module

The imaging performance of the optical system is optimized during the instrument manufacturing and installation phases, but the quality of an image can be severely reduced depending on other multiple effects. A list of sources causing these effects includes:

- **Optical elements**: The misalignment of the optical elements (mirrors, filters, CCD) can modify the PSF at the focal plane.
- **Instrument**: Some control issues concerning the CCD (temperature not stabilized, shutter blocked, etc.) may derive in faulty processes during the image acquisition: image is not downloaded or read correctly, image is not stored at the correct directory, etc.
- **Observation pattern**: The user defines a pattern to observe proposal targets (observation window, moon phase and distance, photometric and seeing conditions, etc.), but it might be not accomplished for some reason.
- **Environment**: The environment conditions (atmospheric transmission, seeing and weather) can reduce the image quality. They have to fulfill the user or observatory specifications.

Some of these effects can be minimized with an exhaustive maintenance plan and a widely tested scheduler application. The most common ones are usually linked to poor local environment conditions, and these can be minimized with an exhaustive monitoring policy and a generation of events to optimize the system response. We have already included these mitigation processes in the control system design and development, already described in the previous chapters.

However, it is also necessary to implement automatic techniques to detect poor image quality associated with the unexpected faulty behaviour of the system hardware or software elements. We

consider that this automatic image quality check is mandatory for a robotic observatory, in order to decide if an image passes the quality level and does not need to be repeated. It is a key process to maximize the system efficiency, once the other mitigation actions mentioned in the last paragraph have been considered.

A non-compliant image is reschedulable.

11.3.3.1 Images to qualify

Data processing routines are applied to all kind of images acquired with the MEIA instrument. The images that are processed to identify their quality level are:

- **Bias current calibration images:** single and master frames
- **Dark current calibration images:** single and master frames
- **Flat field calibration images:** single and master frames
- **Science images**

A similar approach will be used in the future when the ARES instrument is installed, but the quality check for this instrument is out of the scope of this Thesis.

11.3.3.2 Four step evaluation

The image quality check process executes an evaluation of four different aspects for each image file: the file quality, the image photometric quality, the image astrometric quality and the observation pattern and environment quality. It follows a description of them and a description of those parameters that are used to compute each qualification feature.

➤ *File quality*

The first step represents a basic analysis of the image file. This evaluation starts once the file is moved to its final storage location. It verifies the following aspects and parameters:

- Expected **size and format:** These depend on the instrument configuration (binning).
- Header has all the **required keywords:** A minimum set of keywords is mandatory to execute the full data process (i.e., the image type is used to select the processing work flow). Keywords are specific to the detector control software and are defined in the system configuration data file.
- **File location and image name:** They depend on the observatory file management configuration.

➤ *Photometric quality*

The second step is focused on the image photometric quality. It checks science and calibration images after the reduction and analysis are performed.

- For calibration images, the system checks the **counts per pixel level and distribution** for:
 - Single calibration images: The signal level is computed and compared with an accepted range, depending on the image type (bias, dark and flat). Flat images are also compared with a good quality one to detect a possible non-homogeneous illumination pattern. If it has the expected quality, the image is used to compute the master frame.
 - Master calibration frames: They are compared with good quality master frames before using them in the reduction processes of the science images.
- For science images, the evaluation involves:
 - **Star shape:** This evaluation is based on the computation of the median ellipticity and the standard deviation of the position angle distribution. The main goal is the detection of errors on the PSF profile, produced by, i.e., bad telescope tracking. High ellipticity and coherent position angle are obtained in case of faulty tracking, whereas in galaxy fields one might have high ellipticity but incoherent position angle.
 - **Difference of the measured instrumental magnitude with catalogue values:** The difference is computed to detect any photometric issue. The process is done by comparing the expected and the measured fluxes for the brightest stars in the field. These targets are cross correlated with the same targets in catalogues, according to their catalogue magnitude and the observing conditions at the observatory. The instrumental magnitude difference is considered significant if it is larger than the specified tolerance.

- **Median FWHM:** It is computed to provide an estimation of the seeing conditions. The value is then compared with the minimum required environment conditions specified by the user and the observatory.

➤ **Astrometric quality**

The third step checks the astrometric image quality. It is only applied to science images. Two separate checks are performed:

- **WCS solution:** The module checks if the WCS solution is successful. If objects in the field are crosscorrelated with a catalogue, sky coordinates for each object and the pointing error of the field's center are obtained. Sometimes, the WCS solution is not resolved because there are too few stars in the field. In this case, the astrometric quality cannot be evaluated.
- **Difference between the image center and the target coordinates.** The coordinates at the image center are known if the WCS solution is obtained. In this case, the pointing offset (difference between expected and obtained image center coordinates) is compared with the user required pointing tolerance to decide if the image is sufficiently centered.

➤ **Observation pattern and environment quality**

The last step checks if the user observation pattern, defined at the proposal submission phase, has been accomplished. Warnings to be considered for the image evaluation are also checked here. This evaluation is split into three fields:

- **Observation pattern aspects:** lunar distance, airmass, observation window, etc.
- **Weather warning flags:** This happens when weather conditions are out of the desired range.
- **System warning flags:** These warnings are triggered when the state of any element is suitable to produce a reduction of the image quality (CCD working temperature value and stability, dome status, weather data disagreement among different sensors, etc).

11.3.3.3 Process outputs

➤ **Image quality flag and process description keywords**

Each one of the steps described before is summarized in one digit and all together gives a four digit numerical code that is included in the image header as an image quality descriptor keyword. This code is then used for several applications of the OpenROCS system (scheduler, etc.). Other keywords are also added to the image header, which include a more extended description of this quality process and the errors found. It follows a description of the related keywords (Table 11.4):

Description	Keyword	Information content
Image quality summary	IMGQS	'mnpq' ('mnq' for calibration images), where m, n, p, q are integers describing the results of the quality control (see below). 0 means a good quality result
File quality parameters	IMGQF	<ul style="list-style-type: none"> • Missing keywords • File size, only if not equal to detector configuration values (default file size and binning) • File name and location errors Example: IMGQF = 'Miss.keywords: PROPCODE, TARGETID, F.size: 8395200'
Photometric quality parameters	IMGQP	Calibration images (single and master frames): <ul style="list-style-type: none"> • Counts per pixel level: mean value for bias current, dark current and normalized flat field images • Counts per pixel distribution: standard deviation for normalized flat field images
		Science images: <ul style="list-style-type: none"> • Ellipticity: mean measured value, maximum accepted value (i.e., 1.11, 1.5) • Position angle: measured distribution sigma value, minimum accepted value (i.e., 56.86, 5.0) • FWHM (arcsec): mean measured value, and user and observatory maximum values (i.e., 1.31, 1.5, 2.0) • Magnitude difference with catalogue: measured mean

		value, maximum accepted value (i.e., 0.34, 1.0) Example: IMGQP = '1.11, 1.5, 56.86, 5.0, 1.31, 2.0, 0.34, 1.0'
Astrometric quality parameters (only applies for science images)	IMGQA	<ul style="list-style-type: none"> WCS solution result: yes/no Image center pointing difference (arcmin): measured value, observatory maximum accepted value (i.e., 3.36, 1.0) Example: IMGQA = 'yes, 3.36, 1.0'
Obs. pattern and environment conditions parameters	IMGQC	<ul style="list-style-type: none"> Airmass, lunar distance, etc.: measured values, and user and observatory accepted ranges (i.e., airmass: 1.32, 2.0, 3.86) Instrumentation warnings: CCD temperature, etc. Weather warnings: high humidity, etc. Example: IMGQC = '1.32, 3.86, none, none'
Image quality process errors	IMGQE	QF{xyz} QP{xyz} (science images) / QP{xy} (calib.images) QA{xy} QC{xyz} x, y and z are equal to 1 if the quality process returns an error when computing the related parameters (see below). 0 means no error occurred. Example: IMGQE = 'QF:000,QP:000,QA:00,QC:100'

Table 11.4: Image quality keywords description

We give in the following subsections a list of possible values returned by the quality control process. An example of header, which includes these keywords, is shown in Sect. 11.4.9.

➤ **Quantification values**

The value of the image quality flag (IMGQS) depends on the parameters computed and whether they fill in the accepted range or not. A zero is given for those that have a correct response. The following table (Table 11.5) gives the assigned integers when the parameter is out of the accepted range:

Quality feature	Evaluation parameter	Value
File ('m')	Size & Format	1
	Header keywords	2
	Location & Name	4
Photometry ('n')	Counts per pixel level (calib. images)	1
	Counts per pixel distribution (calib. images)	2
	Shape (science images)	1
	Catalogue magnitude difference (science images)	2
	FWHM (science images)	4
Astrometry ('p')	WCS	1
	Pointing difference	2
Obs.Pattern and Environment ('q')	Obs. Pattern	1
	Weather warning	2
	Instrument warning	4

Table 11.5: Quality descriptor values

The values are summed so that a unique and non degenerated number will result as the flag entry in the image quality keyword. '0000' corresponds, for instance, to a good quality science image, whereas '2006' is a science image taken under some weather and system warning conditions that also has an error on the header keywords.

The parameters range depends on the observatory defined constraints or on the user constraints, when they are stricter. The constraints defined by the observatory are specified in a following section.

➤ **Error keyword descriptor**

Each process can be aborted due to several errors. The list of possible errors is given below. The ICAT logs the exact description of the occurred errors into the process summary file (file named with [sfx] = pdq, according to the file name convention given in Sect. 11.3.2.3), whereas the error code is included in the image header, as shown before (Table 11.4).

File quality (QF)

- File size (x): File is corrupted, quality process has not permission to read file, size cannot be evaluated, etc.
- Header keywords (y): Header cannot be read (i.e., software used to read it does not respond), etc.
- File name and location (z): File name and location cannot be evaluated

Photometric quality (QP)

- Calibration images (x, y): (mean value, standard deviation) for the counts per pixel distribution cannot be computed, reference image used to normalize flat field frame not found, etc.
- Science images, shape (x): Ellipticity and/or position angle are not given in the image analysis output data file.
- Science images, magnitude difference (y): Catalogues are not accessible (i.e., there is no internet connection), magnitudes are not given in the image analysis output data file, etc.
- Science images, FWHM (z): FWHM for each star is not given in the image analysis output data file, user requirements cannot be read (i.e., database is not accessible), etc.

Astrometric quality (QA)

- WCS (x): Pipeline returns an error during the analysis process, etc.
- Pointing difference (y): User requirements cannot be read (i.e., database is not accessible), etc.

Obs.Pattern and Environment Conditions quality (QC)

- Observation pattern (x): Database is not accessible, proposal and/or project identifier keyword is not resolved, etc.
- Instrumentation warning (y): Instrumentation data is not available, etc.
- Weather warning (z): Weather data is not available, etc.

➤ **Quality control configuration**

The image quality check returns a flag when the evaluation parameters are found to be out of the specified range. The observatory accepted ranges used by default are shown at Table 11.6. Some values are missing because they are still under evaluation. User requirements are also checked for some of them.

Parameter		Value / Range	Tolerance
File			
Minimum set of keywords required	Proposal information	PROPCODE, TARGETID, BLOCKID	N/A
	Detector	Pixel scale, gain, readout noise, CCD temp.	
	Image	Image type, exposure time, JD.	
	Telescope	Target coord. (RA&DEC), nominal coord. (RA&DEC), filter, airmass (or elevation), focus position.	
	Environment	Weather data (temperature, wind speed, humidity).	
	Other	Dome status.	
Photometry			
Calibration images,	Bias and dark	> 0	

counts per pixel mean value	current images		
	Flat field images	[10000, 45000]	2000
Flat field images, counts per pixel std. deviation		Under definition	Under definition
Ellipticity, maximum average value		0.5	0.1
Position angle, minimum std. deviation		5	0.1
Mean FWHM (arcsec)	Very good seeing	< 1	0.1
	Good seeing	[1, 2]	
	Poor seeing	> 2	
Mean catalog magnitude difference (mag)		1	0.2
Astrometry			
Image center pointing difference (arcmin)		2	0.5
Obs.pattern and environment			
Observation pattern	Airmass	< 4 (altitude > 14.5 deg)	0.2
	Others	Under definition	
Instrumentation warnings		CCD temperature, etc	N/A
Weather warnings		High humidity, etc	N/A

Table 11.6: Quality control configuration ranges

11.3.4 Pipeline module

The main goal of the pipeline is the calibration and analysis of the science images. It is executed by the data processing routine when performing the automatic photometric and astrometric quality control of a science image, but also to obtain the master calibration frames. Manual execution is also considered.

Two main blocks are defined while designing the pipeline:

- **Calibration:** It includes the processes for the reduction of the calibration images to obtain the master calibration frames, and the processes for the reduction of the science images using the master calibration frames.
- **Photometric and astrometric analysis:** It implements the analysis routines to identify the objects in the science image frame and to extract precise photometric and astrometric data for each object.

11.3.4.1 Calibration

Those effects that reduce the image quality are: bias current, dark current, flat fielding, shutter effect, and bad pixels. Most of them are due to the characteristics of the MEIA mode of operation and the main CCD camera. They were introduced in Section 3.1.4.3 as well as their acquisition policies.

The pipeline executes a calibration routine devoted to extract these contributing effects. It is based on the NOAO-IRAF software package and it is the first step to obtain the photometry with the required precision.

➤ Sequences for the reduction of calibration images

A basic set of calibration images (bias, dark and flat) is acquired, at least, twice per night during twilight hours. All these contributions to the image noise are treated separately by executing a sequence to obtain the master calibration images that are then used for the reduction of the science images.

The reduction sequence for the calibration images is detailed here and it is triggered by the last image acquired when executing the sequence of calibration images:

- **Bias master frame**
 1. Acquisition of a set of bias images.
 2. Combine the bias images.
 3. Fit a two dimension function to the bias to obtain the master bias frame.
- **Dark master frame**
 1. Acquisition of a set of long exposure raw dark current images.
 2. Subtract bias master frame from each dark image.

3. Combine dark images scaled with exposure time.
4. Master dark frame per unit time is obtained.
- **Flat master frame**
 1. Acquisition of a set of raw flat field images.
 2. Subtract bias master frame from each flat image.
 3. Dark current correction is not necessary. Long integration time is foreseen for flat-fields.
 4. Combine the flat images.
 5. Normalize last image to obtain the flat master frame.

Once a master frame is obtained it passes a quality control before being stored and used. If it does not obtain clearance, it is discarded. Quality control ensures the achievement of the calibration goal (Sect. 11.1.1.12).

In case no master frame could be obtained, due to a lack of calibration images taken during the current night, master frames are taken from past nights. System stability is still being evaluated in order to decide the suitability for this approach and the look-back time interval.

➤ **Sequence for science image reduction**

Once a set of master calibration images is obtained, science images can be reduced. This means that they can be corrected from the main effects: bias and dark current and flat-fielding.

The pipeline is structured in separate blocks to manage calibration and science images for all the reduction steps. The reduction sequence of science images is presented here:

1. Subtract bias master frame.
2. Apply dark current correction if necessary, depending on the image exposure time. When dark current correction is recommended (detector commissioning must determine the minimum time exposure), time scaling for the dark master frame is carried out to match the integration time of the science image.
3. Select a flat master frame to match the configuration of the science image and divide them.
4. Apply bad pixel mask.
5. Apply shutter correction.

The on-the-fly reduction is executed after the acquisition of each science image and it uses the calibration master frames obtained during sunset twilight. When the second set of calibration images is obtained, both evening and morning calibration sets are used to compute the master calibration frames and a final off-line reduction of all the science frames is executed.

11.3.4.2 Photometric and astrometric analysis

Image photometric and astrometric analysis is performed by the pipeline in several steps: computation of the sky pointing coordinates and the correction coefficients, object identification, photometric and astrometric computation for each object, transformation of coordinates to obtain sky (RA and DEC) coordinates, archiving the data in an ASCII file. The routines designed to implement those are described below in more detail.

➤ **WCS**

The pipeline uses, as a first step, a routine to crossmatch targets in the FOV with the USNO or GSC catalogues. The WCS routine is used for that purpose. With this routine, a six order polynomial fit is employed to transform the pixel coordinates into absolute RA and DEC coordinates. The routine modifies the image header to include the corrections to apply to the image centre coordinates.

➤ **Object identification**

Identification of sources in the image has to be done in a reliable way and using existing software tools developed for this purpose. Two packages are selected: DAOPHOT (running under IRAF) and Source Extractor (SExtractor). Both packages run convolution routines to detect objects and select those that are above the background by a certain threshold.

➤ **Aperture and PSF photometry**

Both types of photometry are implemented in the pipeline to be used when running a calibration process using the automatic control mode or the manual operation through a user interface. The pipeline uses the DAOPHOT and the SExtractor packages. DAOPHOT carries out a semi-empirical calculation of the PSF function, which has proven to be more adaptable to any kind of field (either dominated by bright or faint stars). SExtractor executes an improved aperture photometry, choosing the best aperture shape and radius depending on the object found.

Automatic execution

On-the-fly image processing was initially designed to be done with the SExtractor package. Final processing was foreseen to be splitted into two steps to improve the analysis in crowded areas of the FOV. An initial analysis with SExtractor was envisioned to define a density function to identify crowded areas in the field, and, then, a second analysis of these areas with DAOPHOT would be done using PSF photometry calculation.

Tests using both software packages on fields with different crowding showed that the differences on data quality were not significant enough. SExtractor proved to run extremely fast in comparison with DAOPHOT (see Sect. 11.3.4.4 for more details). SExtractor was then selected to be used on both the on-the-fly and final analysis processes. SExtractor is executed in two iterations: the first run uses default configuration parameters and the second run uses the mean values of the star FWHM and background threshold obtained from the results.

The software used for image analysis is highly configurable due to its modularity. Therefore, it could be modified in the future in case the instrument configuration was modified or better software applications appeared.

Manual execution

Manual execution using different analysis configuration and software was considered to let the users run their own data extraction preferences. These are the options given to the user through a web-based interface:

- SExtractor: analysis using SExtractor with configuration parameters as specified by the user.
- DAOPHOT-BASIC: basic astrometric and photometric analysis using DAOPHOT (PSF photometry).
- DAOPHOT-APERTURE: basic astrometric and photometric analysis using DAOPHOT (aperture photometry).
- DAOPHOT-PLUS: optimised astrometric and photometric analysis using DAOPHOT (PSF photometry). Bright targets used to compute the PSF function are taken from a list provided by the user or they are selected among the brightest and non-contaminated by neighbouring targets given by a SExtractor analysis.

➤ **XY2SKY**

A second step on astrometry calculation is done based on the XY2SKY routine. This routine transforms the pixel coordinates of all identified targets in the FOV into absolute RA and DEC coordinates by using the correction keywords provided by the WCS routine and stored in the image header.

➤ **Archiving format**

A final process is executed to homogenize the output data format to be archived in an ASCII table file independently of the software package used when analysing the image (DAOPHOT or SExtractor). When executed under automatic control, the output file includes ellipticity and position angle for each target so that the mean stellar shape can be used for image quality control.

11.3.4.3 Processes for the evaluation of execution

Several checks are also designed to be carried out during execution to improve reliability:

- **Reduction of calibration images:** Master frames are not produced if the number of calibration images (bias, dark and flats) is smaller than a configured value in order to guarantee good photometric accuracy.
- **Reduction of science images:** When no master frames are detected at the current night directory, master calibration frames from past nights are copied and used accordingly. Consequences on data quality when using calibration images from other nights, which depend on system stability, have not been determined yet.
- **Logging:** Pipeline results are logged in order to let the user know the steps performed and the calibration images used to extract the data. The user is able, then, to reproduce the calibration and the analysis at any time.

In all these cases, the creation of every new image is evaluated and possible errors are reported.

11.3.4.4 DAOPHOT vs SExtractor performance tests

DAOPHOT and SExtractor software packages were tested using a crowded and a non-crowded field of stars to check if there were significant differences and to detect hypothetical weaknesses. Other software packages (i.e., DoPHOT - Schechter et al. 1993) were discarded as good performance or direct integration into unattended execution routines did not seem to be better than DAOPHOT or SExtractor.

➤ Test sample

Two fields with different star crowding levels were chosen to test both software packages. Images used (Fig. 11.5) are shortly described in Table 11.7.

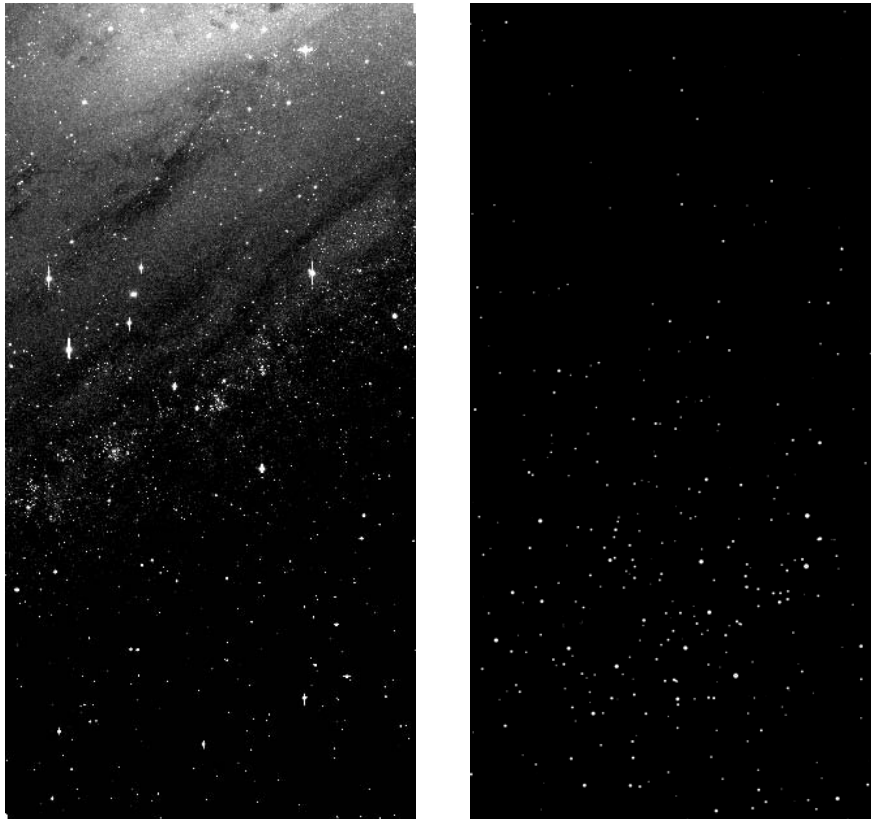


Fig. 11.5: Images used to test the analysis software packages. Crowded (M31, left) and sparse (m67, right) fields were selected.

Parameter	Crowded field	Uncrowded field
Target	M31	M67
Coordinates (J2000.0)	RA: 00:42:44.33 DEC: +41:16:07.50	RA: 08:51:00.62 DEC: +11:49:00.01
JD	2451433.644383	2451578.6346290
Telescope-Instrument	INT (ORM) – WFC	INT (ORM) – WFC
Detector	A5506-4 (CCDSEC 2048x4096 pix ²)	A5382-1-7 (CCDSEC 2048x4096 pix ²)
Arcsec/pix (FOV)	0.333 (11.4x22.7 arcmin ²)	0.333 (11.4x22.7 arcmin ²)
Filter	V	Strömgren Y
Binning	1x1	1x1
Exposure time (s)	899	15.07
CCD Mean value	2723.686	16.011
CCD Processing	Done	Done

Table 11.7: Description of images used to test the analysis software packages

➤ **Comparison of photometric and astrometric results**

Instrumental magnitudes and position coordinates were compared. The analysis was carried out based on aperture photometry calculation to compute the instrumental magnitude. This configuration gives the results to best compare the software packages as it represents computing the magnitudes in an analogue way, although in both cases this does not correspond to an optimal configuration (see software packages documentation for more details on their implementation: SExtractor - Bertin 1998; DAOPHOT - Stetson 1987).

Only objects with extracted instrumental magnitude with an error smaller than 0.05 magnitudes were taken into account for the software evaluation. The objects were cross-identified using position coordinates on the image by selecting only those objects closer than 0.2 pixels on both axes.

Results are summarized in Table 11.8.

Analysis details		M31			M67		
SExtractor	Aperture diameters (pixels)	6	10	14	16.3	19.2	22.4
	Detection threshold (σ above background)	2			2		
	Background	A global background map is computed for the entire image using a base area of 64 pixels (BACK_SIZE parameter)					
	# objects identified	13230			590		
DAOPHOT	Aperture diameters (pixels)	6	10	14	16.3	19.2	22.4
	Detection threshold (σ above background)	5			5		
	Background (annulus diameter, width) in pixels	11.25, 10			20.25, 10		
	# objects identified	28883			522		
Comparison results		M31			M67		
# cross-identified targets (targets with low photon noise)		797 (610)			319 (296)		
RMS (Δm_v)	Complete sample	0.040	0.104	3.888	0.010	0.013	0.018
	M31: mag < -12.5; M67: mag < -11	0.006	0.014	0.030	0.001	0.002	0.002

Table 11.8: Results obtained to compare the analysis software packages using aperture photometry

Fig. 11.6 and Fig. 11.7 show the magnitude difference distribution for the cross-identified targets. Fainter stars present a bigger background noise contribution, which is more significant for the crowded field (M31). The computed magnitudes for bright stars are very similar on both fields when using aperture photometry. In addition, no differences are seen when comparing targets identification and photometric values through the image frame.

The results show that both software packages produce similar results when performing aperture photometry. The differences between them increase in crowded fields. Further tests will be carried out to check the differences between PSF photometric data extracted with DAOPHOT and the improved aperture photometry computed by SExtractor. These will be based on an artificially produced field and a variation of its crowding to determine how it affects the photometric and astrometric results obtained with each analysis routine.

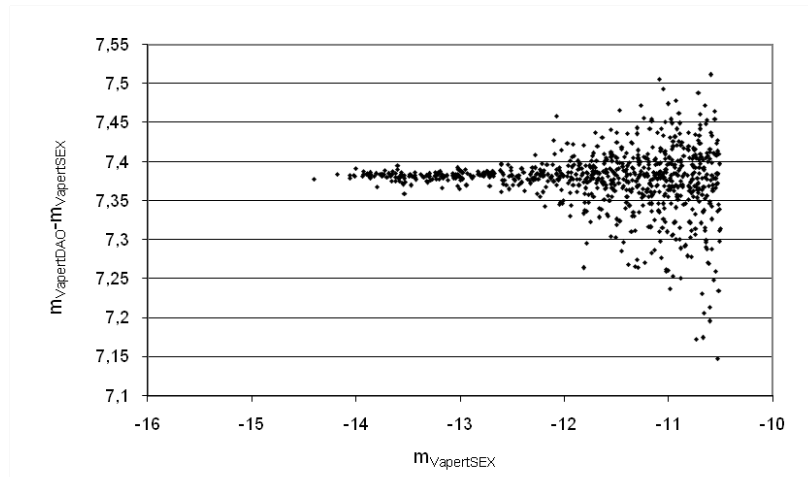


Fig. 11.6: Difference of aperture instrumental magnitude measured in the M31 field

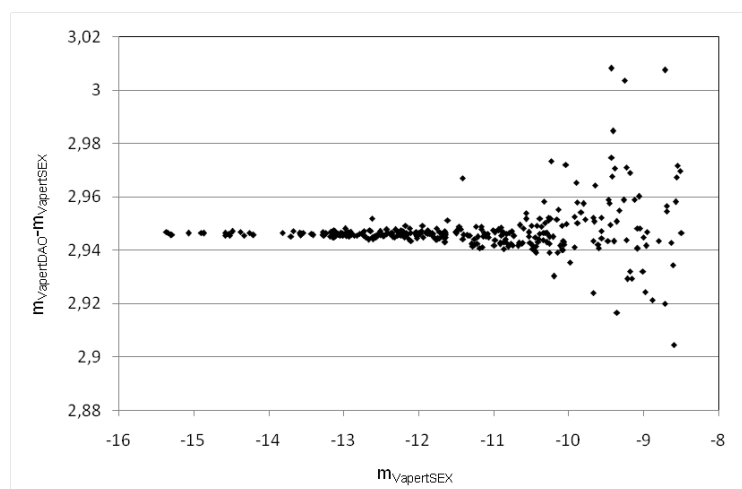


Fig. 11.7: Difference of aperture instrumental magnitude measured in the M67 field

➤ **Comparison of modularity and easy integration into the main execution flow of the pipeline**

DAOPHOT is a highly configurable software and new subpackages can be developed based on the implemented functions. Error detection when running the software integrated into a pipeline is extremely difficult and lengthens bug tracking during pipeline testing. The execution of the program requires also a specific environment (directory structure, configuration of variables) that must be present any time it runs.

The configuration of SExtractor is limited and it can be executed only as an external package, without enabling any modification or selection of the algorithms used to perform the actions. The execution of the program, instead, can be done in a straightforward manner and the configuration is easy to implement into a pipeline. No errors are normally produced when executing the package as a consequence of the low modularity and limited configuration possibilities.

➤ **Comparison of running time**

DAOPHOT is relatively slow when performing image analysis, whereas SExtractor is significantly faster making it more useful for massive or time constrained processes. These have been tested with the aforementioned fields, but also with other images.

➤ **Conclusions**

DAOPHOT and SExtractor were proven to be suitable to be used for the analysis of crowded and uncrowded fields with the aim to extract high quality photometric and astrometric data. Small differences were detected on data results obtained from an uncrowded field. The suitability of SExtractor for automatic and fast analysis of astronomical images was proven to be higher, although further tests on photometric data results will help to further characterize the accuracy of the extracted results.

SExtractor was chosen to implement the photometric and astrometric analysis in the pipeline module of ICAT. DAOPHOT was discarded for the automatic execution, but it was kept for the manual operation.

11.3.5 Archiving and database update module

11.3.5.1 Archiving

Automatic data extracted from each science observation frame is archived in an ASCII table file containing the following data for each target:

- RA, DEC, EPOCH: object sky coordinates
- XCOORD, YCOORD: object coordinates on the CCD chip
- MAG, ERRMAG: instrumental magnitude and error
- ELLIP, POSANGLE: object shape ellipticity and position angle
- AIRMASS: field airmass, computed using the image zenith angle
- JD: Julian Date of the image acquisition
- FWHM: target signal amplitude
- SEXFLAG: possible errors detected by SExtractor
- SW: software used for the analysis (i.e., sex)

A tracking file is also stored at the observatory archive (on- and off-site storage), where all the actions resulting from the data processing routines can be found. This tracking file is used in case image reprocessing or process evaluation is required by the observatory staff or by the end-user.

11.3.5.2 Database update

A new register in the observatory database is created after image processing. All the data involved in the evaluation made by the data processing routine is added to this new register, uniquely identified to be cross-matched with the corresponding OB task specification.

The OB identifiers and the image quality summarized with the image quality keyword are used to update the associated task stored in the database. The task, project and proposal completion level is then checked and updated (modifying the status descriptor fields) to be lately considered by the scheduler routines.

11.3.6 Storage and backup module

The storage and backup of image and data files represents a critical issue for unattended observatories. A large amount of data is usually produced and this must be stored in a way to optimize file management and processing routines.

Images at the TJO-OAdM are obtained with the subsystem controlling the full task execution process (SUB-ESTALL). They are immediately moved to SUB-ALIS, where they are processed and stored so that CPU-intensive processes do not compromise hardware control. The directory structure and backup configuration detailed next apply to the final on- and off-site storage subsystems.

11.3.6.1 Directory structure

The directory structure is designed as an element to organize information and data for processing routines, and to easily manage data storage and backup.

For TJO-OAdM, data types and levels defined above (Sect. 11.3.2.1) fix the directory tree. There are different directories for different data levels. Raw science and calibration images, for example, are stored in a different directory than reduced images and extracted data. A Julian Date (JD) classification is used as a final organization step. Housekeeping and logging data are stored in a different directory. And housekeeping data are also kept in the observatory database. A diagram of the directory structure follows (Fig. 11.8):

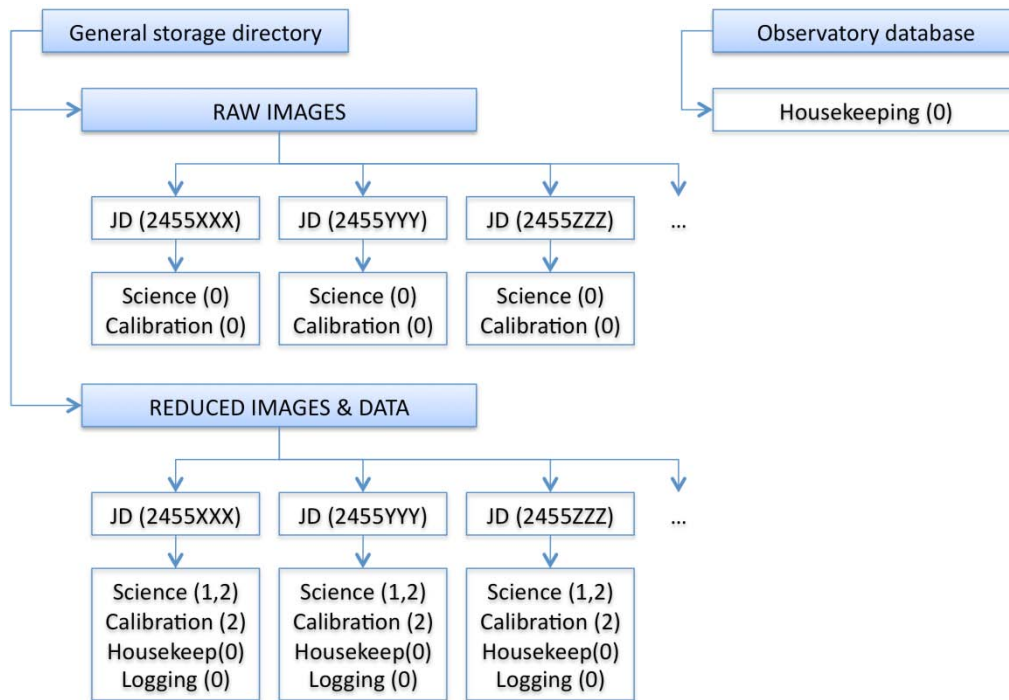


Fig. 11.8: Directory structure design

11.3.6.2 Data storage

Images and data files are stored on- and off-site (see file system architecture in Fig. 11.9) following the criteria defined above. And an incremental backup is executed by the system to copy the files to an off-site server. The latter is implemented on the SUB-AGER subsystem, also used as the observatory web server in order to minimize the time to deliver the images and data to the end-user.

A major constraint to decide which files are kept is the available storage space. According to the main CCD camera features (frame dimensions, image read-out time, etc.) and the number of images foreseen, depending on night time duration in different seasons, a total of 1 to 6 GB of raw images are expected every night at the TJO-OAdM. Additional data (system logs, housekeeping, etc.) are produced, but these have a much lower impact on the data storage needs.

➤ **Storage media**

There are two main storage devices at the on-site storage subsystem (SUB-ALIS): a RAID5 and portable magnetic storage tapes. The capacity of each medium is:

- RAID5: 276 GB, implemented on four hard drives.
- Portable tape: 200 GB

An external unit (SUB-AGER), located outside the observatory, is in charge of making backups off-site. The location of this unit is at the IEEC facilities and its features are:

- RAID6: an LVM over a RAID6 that provides double redundancy, implemented on four hard drives of 500 GB each connected to the computer using eSATA boards.

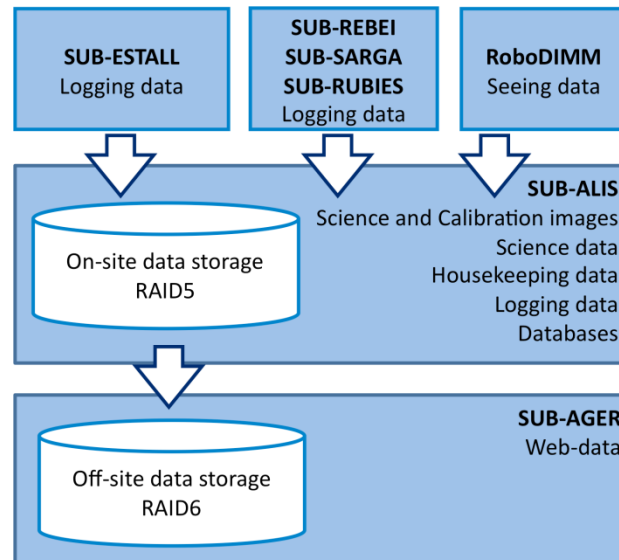


Fig. 11.9: Storage configuration at TJO-OAdM

➤ *Managing the Data Repository*

- **On-line:** Raw images are copied on-line from the acquisition system control computer (SUB-ESTALL) to the final backup unit. All the rest of data to backup (see below) is generated at the time of processing in the backup unit (i.e., master calibration frames or data files). They are kept in this backup unit until the full storage capacity reserved for this purpose is used. A separated division of the hard disk space is used for images and the rest, in order to avoid blocking the store capacity with, for instance, log files.
- **Off-line:** The files selected were planned to be copied on a portable tape every day, although this option was not implemented as it added another level of redundancy that was finally considered unnecessary. Once a portable tape is full, human intervention is required to change it for a new one, which is a clear drawback for unattended control. Data storage robots avoid this problem, although they are extremely expensive.
- **Off-site:** An off-site copy of all the files selected is done during the backup window defined below.

11.3.6.3 Selection, extraction and manipulation of data

➤ *Selection and extraction of file data*

The files selected for the backup copy are: raw and reduced astronomical images, raw calibration images, master calibration images (bias, dark, flat_x_filter), ASCII table files with the resulting data from image processing and files with a post observation summary and data quality comments.

➤ *Selection and extraction of live data*

Another block of files to include in the backup copy is the one related with the control system and the housekeeping applications. They store the relevant information in the observatory database and in several log files, which are included in the backup copy. These files are continuously updated.

➤ *Manipulation of data*

The on-line and off-site backup copies are done by the server unit SUB-ALIS, which contains all the data to copy. The routines and protocols to move the data to the server, when required, are as follows:

- Images taken by the SUB-ESTALL subsystem are automatically stored in the local server SUB-ALIS via an ICAT routine that is executed any time a new image is acquired. This routine is managed by OpenROCS with a triggering event and a predefined action.
- Log files are produced by OpenROCS in the local server (SUB-ALIS). Those not produced by OpenROCS are centralized also in the local server.
- Most of the housekeeping data and database contents are directly generated for the server applications and do not need to be moved. Those data files that, in the future, will have to be moved from other subsystems (i.e., seeing data) are transferred once per day using secure transfer protocols.

The off-site backup copy is also done using secure transfer protocols.

11.3.6.4 Automatic compression procedure

A lossless compression algorithm is used to increase the amount of data that the system is able to store. The compression procedure defines the instance an image is de/compressed at the processing sequence, and the criteria is a balance between processing speed (use of CPU) and amount of data that has to be stored. It is, then, important to decide whether the file de/compression is executed before/after the file processing or at the end of the whole process. A choice also needs to be made regarding the data compression algorithm to use (compression factor vs. speed).

➤ *Files to compress*

Only FITS image files are compressed at the TJO-OAdM observatory. Other data files are stored without compression due to their small size.

➤ *Compression scheme*

The compression scheme depends on each data type and level:

- **Science images:** Raw images (level 0) are compressed upon completion of the data processing routine executed just after the image acquisition. That is, as the last step of the image quality check procedure. Subsequent image processing (i.e., the final image reduction performed during daytime) decompresses raw science images first. Calibrated images (level 1) are compressed at the end of the process executed by the pipeline and they are then moved to the off-site storage facility by the backup program.
- **Calibration images:** Raw images (level 0) are compressed upon completion of the first data processing routine, as described for science images case. Decompression is required before the reduction of the calibration images to obtain the calibration master frames. Once the master frames are obtained, the images are compressed again for their storage. Master frames (level 2) are never compressed instead, because they are used continuously during the entire night for science images reduction.

According to this description, image files are compressed just after the image quality check. This procedure minimizes the response to update the OB status and, therefore, the time needed to provide feedback to the scheduler.

➤ *Compression algorithm*

Some compression algorithms were considered: GZIP, BZIP2, Rice, etc. Compression rate and speed were compared. An algorithm providing higher compression rate is preferred at the TJO-OAdM as data storage is a major constraint than the use of CPU. Therefore, Rice is the program chosen to be used as it provides a better performance according to these criteria (Pence et al. 2009, see Fig. 11.10).

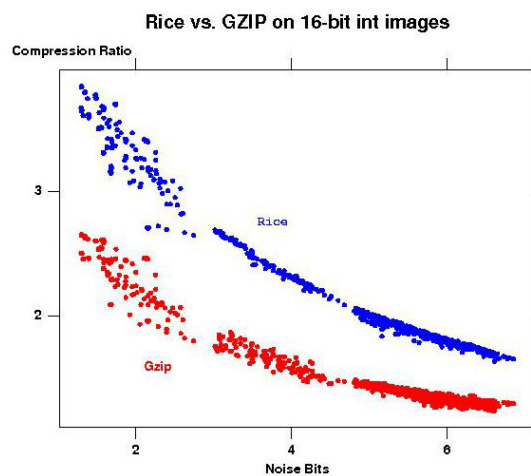


Fig. 11.10: Comparison of compression ratio for Rice and GZIP algorithms (CFITSIO – NASA webpage)

Tests regarding compression speed are continuously conducted and the algorithm used can be easily changed thanks to the implementation modularity. A short description of the Rice algorithm is given here summarizing its main features:

- Origin and license: Rice is a free and open source lossless program and data compression algorithm implemented in the NASA FITS input/output library (FITSIO).
- The Rice tiled-image compression is superior to all the other methods (BZIP2, GZIP or ZIP algorithms). Rice compressed images are 1.4 times smaller than GZIP. Applied to an 8 MB FITS image, it can reach a factor 2 in compression.
- With 16-bit integer images, Rice image compression is more than 3 times faster than GZIP or Hcompress.
- Rice excels even more with 32-bit integer images, giving compression speeds about 5 times faster than Hcompress or GZIP, and producing files 1.6 times smaller than GZIP. Lossless compression of floating point images is generally ineffective.
- Headers for all extensions are still visible in compressed images.
- Compression works transparently and absolutely lossless for integer data (some truncation happens for floats - below the noise level if properly applied).
- Saves large amounts of bandwidth and archive space.
- CFITSIO routines `fpack` and `funpack` can be used to compress and uncompress images using Rice.

11.3.6.5 Managing the backup process

➤ *Limitations*

- **Backup time windows:** The off-site backup window is opened the whole day to shorten the time between image acquisition and delivery to the end-user. In such case, the system copies the files just after creation. The on-line copy is done in processing time.
- **Network Bandwidth:** For the off-line process, an amount of data of 5 to 10 GB is transferred. The required bandwidth needs to provide the capability to transfer such data volume in a reasonable time without blocking completely the link connection for security reasons.

➤ *Measuring the process*

A specific policy to minimize data management errors and loss of information is required.

A validation process is done for those images copied from the acquisition system (SUB-ESTALL) to the backup unit (SUB-ALIS). Initial files in the acquisition subsystem unit are deleted after validation. Validation is also required to check the backup accuracy.

Process log files are created after every backup.

11.4 ICAT implementation details

The ICAT software package was developed as a tool for the TJO-OAdM and other robotic observatories with the objective of managing astronomical images and extracting relevant scientific information in real time. ICAT was designed and implemented to be easily adapted and used in various observatories. Just a few interface packages have to be changed during installation to adjust ICAT to the general control system and the instrument control software. Most of its features have been pointed out in several steps of the data processing design. A description of its implementation is given here.

11.4.1 Implementation architecture

The aforementioned modules (File Management, Image Quality Control, Image Calibration and Analysis Pipeline, Data Archiving and Storage and Backup, see Sect. 11.3) are implemented using four routines to manage the data processing execution.

- **Fitscopy** (Fig. 11.11): This routine moves the raw image to the final storage subsystem. It partially implements the features of the File Management and Storage and Backup modules presented in the design description.
- **Dataprocessing** (Fig. 11.12): It executes the general data processing routine, including quality control, database update and image final storage. It implements the File Management, Image Quality Control, Data Archiving and Storage and Backup modules. Its execution is triggered by the Fitscopy routine when the image is stored in SUB-ALIS.
- **Pipeline** (Fig. 11.13): The calibration and analysis of the images is executed by this routine when required by the Dataprocessing. It implements the Image Calibration and Analysis Pipeline module and some features of the File Management and Storage and Backup modules.

- **Backup** (Fig. 11.14): It executes the off-site backup of a subset of files, as defined in the Storage and Backup module design.

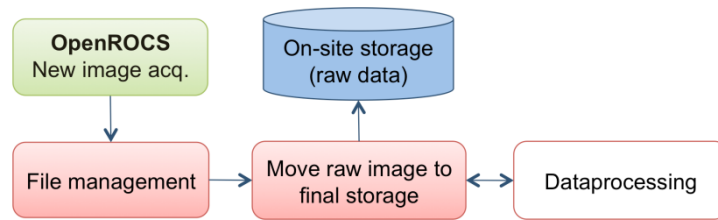


Fig. 11.11: Fitscopy routine work flow

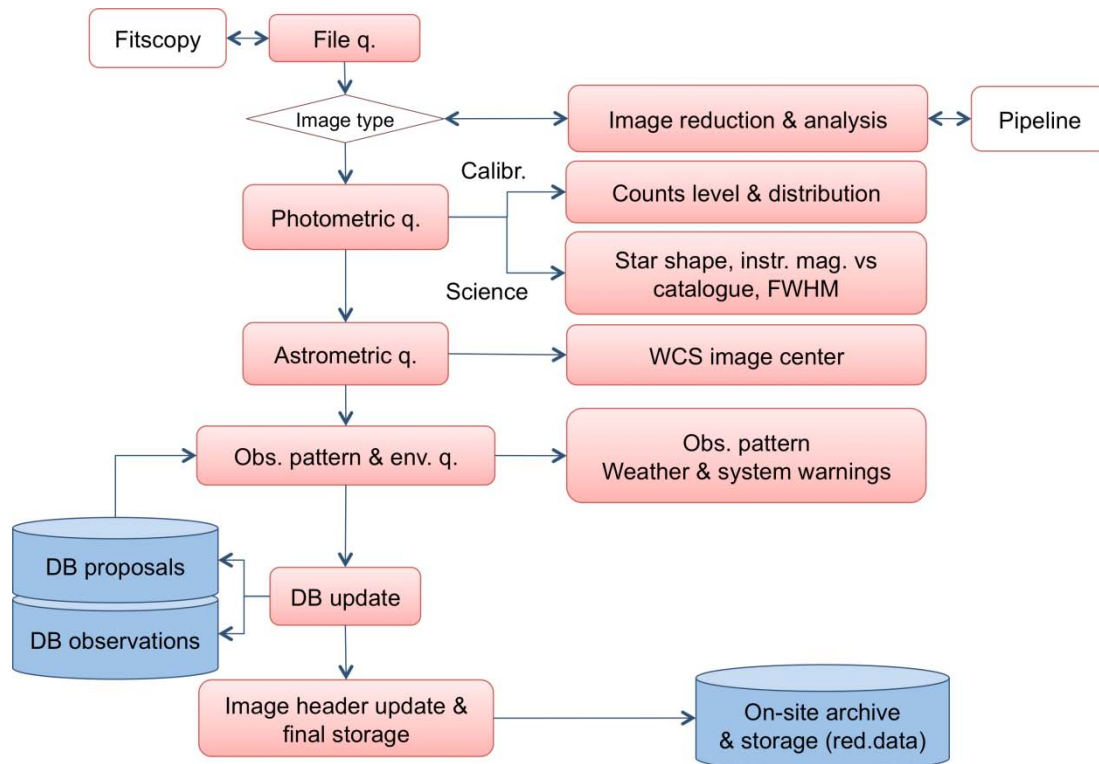


Fig. 11.12: Dataprocessing routine work flow

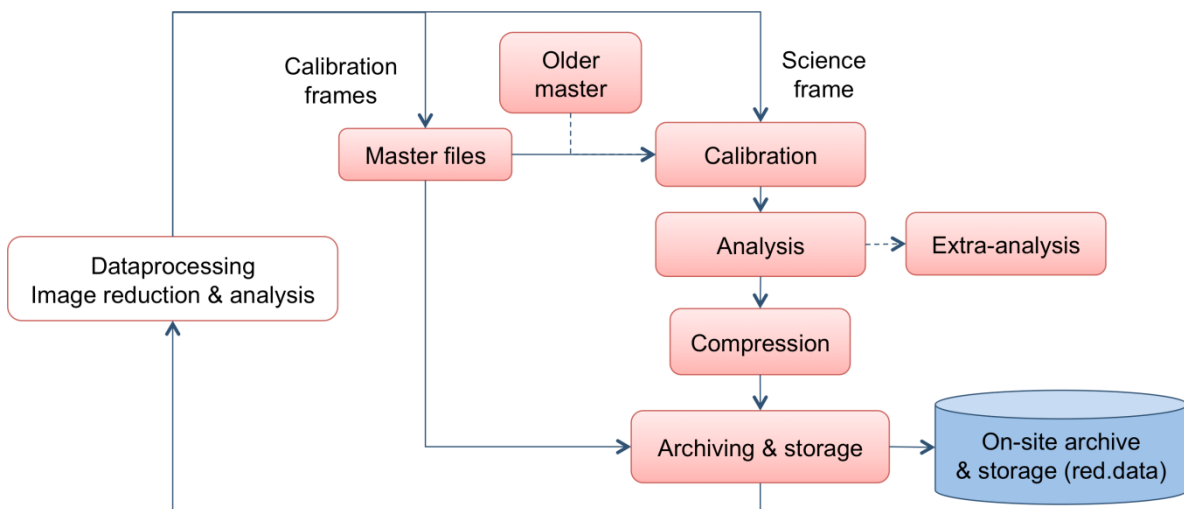


Fig. 11.13: Pipeline routine work flow



Fig. 11.14: Backup routine work flow

The complete task processing sequence is executed after the acquisition of a new image and involves the Fitscopy, Dataprocessing and Pipeline routines. This is currently triggered by the MEIA control software (TALON) when the image is read from the CCD camera. This process will be controlled by OpenROCS in the near future with the event-action management capability (see the associated event specified in Sect. 5.4.4). The execution of the Backup routine is carried out by system synchronization software that performs periodic and incremental backup. This will be also controlled by the software infrastructure (OpenROCS) as specified in the same section.

The configuration parameters are taken at the beginning of each routine execution from a configuration file that defines the data processing steps and some system characteristics.

All the possible options and scenarios that the package implements are based on the use cases discussed in previous sections. Below follows a description of the developed programs and packages that ICAT uses.

11.4.2 Generalities

The ICAT package is based on scripting languages (Python, Perl) to control program flux and manage image files. It is executed together with UNIX shell scripts, NOAO-IRAF and Source Extractor software packages. CFITSIO library routines are also used to implement several utilities (management of header keywords, file compression, etc.).

The general characteristics of ICAT are: automatic management and treatment of FITS images, high accuracy photometric and astrometric data extraction and storage and real time execution. It can be used in other observatories just changing its configuration when performing the installation (the configuration files are described below). After that, the execution is automatic.

Although the automatic execution is its main functionality, user-controlled execution of the pipeline is also available, using a web interface (written in PHP) and enabling the user to modify a number of parameters to adjust the reduction process to different criteria.

11.4.3 Source code organization

The source code of ICAT is divided into four repositories, one for each aforementioned routine:

- **Fitscopy:** It contains the base system to move the acquired images to the final storage subsystem. It includes those programs required to read the image header keywords so that the acquisition time can be obtained to store the image at the corresponding directory and to create the storage directory tree if it does not exist.
- **Dataprocessing:** It contains the system and configuration file to execute the data processing routine. The source tree includes the following folders:
 - **cfg:** It contains the general configuration file where all the parameters are specified.
 - **bin:** It contains the subroutines of the main process.
 - **lib:** It contains the binary executable and the source code of the programs used by the main process.
 - **src:** It contains the image files used as good flat field quality patterns.
- **Pipeline:** It includes the system and configuration files to execute the pipeline routine. The source tree is composed of the following folders:
 - **par:** It contains the general configuration files where all the parameters are specified.
 - **bin:** It contains the subroutines, the binary executable and the source code of the programs used by the main process.
 - **src:** It contains the template for the configuration files and for some procedures to be used by the main external software packages (IRAF, Source Extractor).
 - **GUI:** It contains the files for the web-based user interface.

- **Backup:** This routine is based on system software commands for remote synchronization of computer directories.

11.4.4 Implementation tools

This section describes the tools used to develop the ICAT routines.

11.4.4.1 Software development tools

- Development language: Python (version 2.6.5) and Perl (version 5.10.0).
- CFITSIO libraries¹: C and Fortran subroutines for reading and writing data files in FITS (Flexible Image Transport System) data format. These are also used for image compression. The version in the current ICAT implementation is 3.270.
- SExtractor (Bertin et al. 1996): Image analysis programme, revision 2.5.0.
- NOAO PC-IRAF²: Image processing programme, revision 2.11.3.
- WCS (standalone version extracted from WCSTools³, Mink 1996) and XY2SKY (from wcstools_3.7.2) routines: Catalogue cross-matching algorithm and frame coordinate transformation to RA and DEC.
- MySql: Database interface language, version 5.0.77.
- UNIX: shell scripts (csh) to execute NOAO PC-IRAF and SExtractor from command line, and system synchronization commands.
- Pipeline GUI interface: Apache server software, PHP, MySql.

➤ *Reliable and fast data calibration and analysis software*

Three packages for astronomical image processing are used for the pipeline implementation:

- **NOAO-IRAF**, a software for image calibration and analysis
- **DAOPHOT**, a reliable PSF photometry package
- **SExtractor**, a fast and good precision analysis software, using flexible aperture photometry.

These are used for different purposes according to their capabilities:

- **Calibration:** NOAO-IRAF packages are used to compute the master calibration frames by combining the different calibration images (bias, dark and flat) and to calibrate the raw science images with the master calibration frames. Shutter effect correction is also applied in the science image calibration.
- **Analysis:** The automatic image analysis is performed using SExtractor with a double iteration process, as described in Sect. 11.3.4.2.

11.4.4.2 Modules

ICAT is structured in software modules providing the required modularity that simplifies the software maintenance and its modification to use it in other observatories. A list of the software inputs, outputs and the modules developed are described in this section.

➤ *Inputs*

- List of calibration images: Bias images, Dark images (optional) and Flat images.
- List of science images.
- General program paths and parameters for quality control, reduction and analysis.
- User parameters for manual calibration and analysis.
- Instrument definition parameters and images: list of header keywords, shutter map image, main calibration frames used as quality reference images, bad pixel mask file. Each one of them is associated to a particular instrument.
- Configuration files of the software packages (NOAO-IRAF, SExtractor, WCS) and the ICAT routines.
- Task execution constraints, taken from the DDROCS database.
- Backup configuration: directory structure, computers and periodicity of the backup synchronization.

¹ <http://heasarc.gsfc.nasa.gov/fitsio/>

² <http://iraf.noao.edu/>

³ <http://tdc-www.harvard.edu/wcstools/>

➤ *Processing modules*

Table 11.9 summarizes the modules developed for each routine.

Routine	Module	Description
Fitscopy	fitscopy.py	General managing program
	readkeyword.c	Other tools
Dataprocessing	dataprocessing.py	General managing program
	config.py	Reading configuration parameters
	calibframeq.py fitscheck.py qualitycheck.py analyze.c	Quality control
	ICATpipeline.py	Running the pipeline, preparing the input files required by the pipeline and checking the process
Pipeline	loggen.py readkeyword.c imarith.c imstat_stddev.c imstat.c insertkeyword.c	Other tools
	icat.pl	General managing program
	movefrom.pl	Modification of image file names and directory structure
	process.pl process*.csh	Reduction of calibration and astronomical images
	check_calimg.pl	Check number of calibration images to obtain master calibration frames
	analysis.pl analysis_extra.pl 16bpp.csh andao.csh ansex.csh wcs.csh xy2sky.csh	Analysis of astronomical images
compress.pl fpack.c	Image compression	
	log.pl jd.pl readheader.pl modhead.c readkeyword.c insertkeyword.c	Other tools

Table 11.9: ICAT processing modules

➤ *Outputs*

- Reduced calibration images and calibrated astronomical images. FITS format.
- Photometric and astrometric data extracted from the image file for all the objects in the field of view.
- Archiving the extracted data in the database and the raw and calibrated images in the storage subsystems.
- Log file with the execution details.

11.4.4.3 Documentation

Software implementation is self-documented using plain text comments. As a further improvement of the software code, Doxygen will be used. Doxygen is a software documentation generator with GPL license.

11.4.5 Fitscopy

11.4.5.1 Programs

➤ *fitscopy.py*

- Routine: Fitscopy
- Location: SUB-ESTALL/icat/fitscopy
- Language: Python

This script is used to copy the image from SUB-ESTALL to SUB-ALIS just after its acquisition. It also creates the storage directory in the destination subsystem if it does not exist by following the directory structure policies. The latter is done executing the binary executable *readkeyword*.

The script executes the Dataprocessing routine at the destination subsystem and deletes the image in the original subsystem after verifying that the image is moved correctly. Dataprocessing execution is done by calling the routine and specifying the name of the file to process. It also specifies if the image to process is the last one of the image calibration set in order to inform the Dataprocessing routine if master calibration frames have to be computed.

Once the Dataprocessing routine starts, it returns a notification string (“START”) to the Fitscopy routine that ends its process.

The script is currently triggered by the program executing the camera control (Camera routine in TALON) at the end of an image acquisition. It will be modified in the near future to be called by the OpenROCS control software as an action to perform every time the event of a new image acquisition is produced.

➤ *Programs based on CFITSIO library*

- Routine: Fitscopy
- Location: SUB-ESTALL/icat/fitscopy
- Language: C

This binary executable is based on the CFITSIO libraries:

- *readkeyword*: Program to read a specific keyword value stored at the image header.

11.4.6 Dataprocessing, main ICAT process

11.4.6.1 Programs

➤ *dataprocess.py*

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing
- Language: Python

It is the main script to globally manage the image processing steps involved in the data process flow.

It initially reads the system configuration and selects the processing option to execute among several possibilities:

- **Single frame process** (science or calibration image): Calibration and science images are processed on the fly, just after image acquisition. For science images, it produces a temporary reduction and analysis. Some input data are given to this module: directory name, file name, process configuration (on-the-fly processing is the default mode).
- **Master calibration frame computation and processing**: Master frames are obtained and processed just after the last calibration image is acquired. The required input data includes: directory name, and process configuration.
- **Processing the full set of science images acquired in a particular night** (final processing mode): Final reduction process for the full set of science images is done when last calibration image is acquired at the morning astronomical twilight. The input data for each science image are: directory name, file name, process configuration (final processing).

The process always starts with a single frame evaluation sequence. It may continue, then, with any of the two other possibilities, or both, if a string (“LAST”) is provided as a process configuration value indicating last calibration image has been acquired. In any of these cases, the pipeline routine is called to obtain the master frames, the science reduced images and data files.

Single image frame quality evaluation is done following the design specification given before: file quality, photometric quality, astrometric quality (only for science frames), and observation conditions suitability. It determines the associated quality keywords (IMGQS, IMGQF, IMGQP, IMGQA, IMGQC, and IMGQE) and updates the image header. During the process, it queries the database to read each particular task execution constraints and to add each new image processing results. Finally, it produces a log file for each run.

Data processing execution requires other scripts, compiled programs and source files, located in several directories (/lib, /bin, /src and /cfg).

The script is currently triggered by the program fitscopy.py every time it moves an image file from the acquisition subsystem (SUB-ESTALL) to the final storage location (SUB-ALIS). It will be modified in the near future to be called by the OpenROCS control software as an action to perform every time the event of a new raw image appearing in the final store system is produced.

➤ **config.py**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing
- Language: Python

This module is a parser to read the configuration parameters required for ICAT execution. The values obtained are returned to the parent process. It is used by all the scripts described in this section.

➤ **fitscheck.py**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/bin
- Language: Python

This script computes the parameters required for the file quality control. The values obtained are returned to the parent process (dataprocess.py).

The script checks for: the FITS file keywords in the image header, according to a list included in the configuration file; the image size, as a function of the applied binning and the image type (master or single frames); and, finally, the file and directory names according to the TJO-OAdM policies.

➤ **calibframeq.py**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/bin
- Language: Python

This script computes the parameters required for photometric quality control of the calibration images (single and master frames). The values obtained are returned to the parent process (dataprocess.py).

Photometric analysis

Photometric quality is evaluated in terms of mean counts per pixel level and standard deviation of the counts distribution. Flat field images are previously normalized using a good quality flat image obtained with the same filter.

➤ **qualitycheck.py**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/bin
- Language: Python

The parameters required for photometric and astrometric quality control of the science images are computed with this script. The values obtained are returned to the parent process (dataprocess.py).

Photometric analysis

Photometric quality is evaluated in terms of: star shape, difference of magnitudes with the NOMAD1 catalogue¹ data and mean FWHM value:

- **The PSF shape** is checked to detect tracking errors. The PSF shape is evaluated based on the ellipticity and position angle values given for each target in the image frame as a result of the image analysis with SExtractor package.

¹ <http://www.nofs.navy.mil/nomad/>

- **The photometric values** are checked using the standard magnitudes of a star catalogue to compute the magnitude difference. The NOMAD catalogue is used for such purpose. Stars in the same field are selected from the catalogue using the findnomad1 program included in the CDSCLIENT package¹. findnomad1 allows to query the NOMAD1 catalogue that merges USNO-B1, UCAC2, Tycho-2, 2MASS catalogues and contains over 1.1 billion sources. Expected instrumental magnitudes of the star catalogue are computed using the image exposure time, the seeing conditions and the empirical Exposure Time Calculator of the TJO-OAdM (see Sect. 8.2.3.2). The targets are cross-matched based on sky coordinates (if resolved by the pipeline) and on instrumental magnitudes. The script finally computes the mean difference of magnitudes for the cross-matched targets.
- **The mean FWHM** is computed taking the image data after the analysis process.

Astrometric analysis

The astrometric quality is evaluated, first, by checking if WCS provides the sky coordinates of the image center. The pointing accuracy is obtained by comparing the nominal central image coordinates (WCS solution) with target coordinates (user specification). This value is compared with the pointing tolerance ranges specified by the user and the observatory.

➤ **ICATpipeline.py**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/bin
- Language: Python

This script calls the pipeline routine when required for image reduction (science and calibration images) and photometric and astrometric analysis (science images). Pipeline execution outcome (fail, success) is returned to the parent process (dataprocess.py).

Preparation actions to be done for the pipeline execution are carried out: creation of a configuration file based on a template file using ICAT configuration parameters; creation of lists containing the image file names to process. The list of calibration images used to compute the master frames is obtained considering only the good quality images, information extracted from the image header and the keyword with the image quality summary.

➤ **loggen.py**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/bin
- Language: Python

This script generates a log file where the results of all the actions included in the ICAT processing routine are stored. A different log file is created for every specific image process. The parent process is the dataprocess.py script.

➤ **analyze.c**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/lib
- Language: C

This program computes the parameters required for photometric and astrometric quality control of the science images. The values obtained are returned to the parent process (qualitycheck.py).

➤ **Programs based on CFITSIO library**

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/lib
- Language: C

These are the binary executables based on the CFITSIO libraries and programmed to perform different actions:

- *insertkeyword*: Program used to insert a new keyword or to update the keyword value.
- *readkeyword*: Program used to read a specific keyword value stored in the image header.
- *imstat*: It computes statistics of pixels in the input image.

¹ <http://cdsarc.u-strasbg.fr/doc/cdsclient.html>

- *imstat_stddev*: The same as *imstat*, but standard deviation of pixel counts is added as it is necessary for the evaluation of the photometric quality of calibration images.
- *imarith*: Program to perform operations between image files. Used to normalize flat calibration images.

11.4.6.2 Source files

The files included here are used as a template to specify the configuration of every image processing action and as a pattern of good quality images.

➤ *Flat field images*

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/src
- Language: FITS images

Flat field images for different filters are used as good quality patterns when evaluating new flat calibration frames. Images for different instruments could be included here.

11.4.6.3 Configuration files

➤ *dataprocess.cfg*

- Routine: Dataprocessing
- Location: SUB-ALIS/icat/dataprocessing/cfg
- Language: ASCII text

This file contains the configuration parameters grouped into six main blocks: path declaration, quality parameter ranges and keywords, specification of variables for photometric and astrometric quality analysis, detector name and parameters associated to each detector, database connection variables. The process that reads the contents of this file is the config.py script. An example of the configuration file is shown below.

```
# TJO-OAdM - ICAT
# Parameters for the Data Processing routines

[Path]
# Data processing script execution directory (not
# necessary if declared at /usr/local/bin)
# at alis.oadm.cat
dir_bin= /opt/dataprocessing/
# Raw images directory
# at alis.oadm.cat
dir_rawdata= /mnt/storage/rawdata/
# Reduced images directory
# at alis.oadm.cat
dir_reddata= /mnt/storage/reddata/

[Quality]
#Global quality parameters:
#Maximum difference between image centre and
# target coords (arcmins)
sepLimit=1.0
#Maximum allowed seeing
seeingLimit=2.0
#Maximum average star ellipticity
ellipLimit=0.5
#Minimum allowed star orientation angle STD
stdThetaLimit=5.0
#Maximum difference between expected and actual
# photometric values (in magnitudes)
diffLimit=1.0
#Quality keywords:
#Image quality process description
kwimgqtjo= IMGQTJO
#Image quality summary keyword
kwimgqs= IMGQS
#Keyword for the list of image quality parameters
```

```
kwimgqf= IMGQF
kwimgqp= IMGQP
kwimgqa= IMGQA
kwimgqc= IMGQC
#Keyword for the list of image quality error
kwimgqe= IMGQE
#Quality parameter ranges (default observatory
# values)
biasmnvalue = 0
biasmntol = 0
darkmnvalue = 0
darkmntol = 0
flatmnvalue = 10000
flatmxvalue = 45000
flatmtol = 2000
flatsgvalue = 50
flatsgtol = 20
flatmastermnvalue = 0.5
flatmastermxvalue = 1.5
flatmastermtol = 0.2
flatmastersgvalue = 0.01
flatmastersgtol = 0.005

[Analyze]
#NOMAD1 search radius for comparison stars
# (arcmins)
nomadrad = 7
#Verbose (0 for silent, 1 for minimal, 2 for lots, 3 for
# everything):
verbose=0
#Minimum instrumental magnitude:
instmin=-10
#Search radius for star matches (radians)
searchrad=0.00001
#Maximum multiple matches per star allowed:
```

```

maxmatch=10
#Maximum total star matches to catalogue:
maxfound=20

[Detector]
#Detector identifier. Uncomment according to the
  observatory setup:
detector= FLI4240
#detector= CCD??

[Detect-FLI4240]
# Keywords for the detector used
#Pixel scale
kwpixscale=CDELTA1
#Observation type keyword
kwobstype=OBSTYPE
#Filter
kwfilter=FILTER
#Exposure time
kwexptime=EXPTIME
#Focus position from home
kwfocus = FOCUSPOS
#Julian date
kwJD = JD
#Nominal Ra J2000
kwranom=RA
#Nominal Dec J2000
kwdecnom=DEC
#Target Ra J2000
kwratarg=OBJRA
#Target Dec J2000
kwdectarg=OBJDEC
#Target Elevation
kwelev = ELEVATIO
kwairmass = AIRMASS
#Observation type keyword values
bias = bias
dark = dark
flat = flat
science = science
#Proposal and project ID
kwpropID = PROPCODE
kwTGID = TARGETID
kwOBID = BLOCKID
#CCD data
kwgain = GAIN
valgain = 1.21
commentgain = 'CCD\ Gain'
kwrdnoise = RDNOISE
valrdnoise = 8.67
commentrdnoise = 'CCD\ Readout\ noise'
kwCCDtemp = CAMTEMP
kwbinnx=XFACTOR
kwbinny=YFACTOR
kwdomestat = DOMESTAT
#Environment data (temp, wind speed, humidity)
kwtemp = WXTEMP
kwwindspd = WXWINDSPD
kwhum = WXHUMID

#ICAT pipeline detector name
detectICAT= TALON-FLI4240

```

```

#Quality parameters
#Filesize
filesize=8395200
masterfilesize=16784640
#Filesize tolerance (Bytes)
filesizetol=10000
masterfilesizetol=10000
#Good quality flat image frame to compare
flatflat=flatFLI4240.fits

[Detect-??]
# Keywords for a different detector
...

[Pipeline]
#Pipeline data
configfile = icat_TPL.par
pipelogfile = _ICATprocess.log
#Execution directory
# at alis.oadm.cat
dir_icat= /opt/pipelineICAT/

#Pipeline resolved RA
kwcrval1=CRVAL1
#Pipeline resolved Dec
kwcrval2=CRVAL2

#Output calibration main frames
masterbias=bias-comb-fit
masterdark=darkb-comb
masterflatnd=flatbs-comb-norm_
masterflatd=flatbds-comb-norm_

# Automatic execution configuration
# bias=1
# dark=0
# flat=1
# calib=0
# astro=1
# ana=1
# @extra=(0,"none")
custom="no"
mode=0
#final=yes
# @ins=("filenomodif","TALON-ccd1")
compress=0
compresscience=0
#Minimum time exposure to consider dark current
  (seconds)
minexptime=30

[DB]
#Access to the DB
dbhost= localhost
dbusername = oadm_user
dbuserpass = *****
dbname = oadm_data
observdb = observ_db
projectsdb = projects_db

```


11.4.7 Pipeline for image reduction and analysis

11.4.7.1 Programs

➤ *icat.pl*

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT
- Language: Perl

This is the main script to globally manage the process to obtain: master calibration frames, calibrated science images, and data files with the extracted astrometric and photometric values for the identified objects.

Multi-thread execution is implemented to let the pipeline process different images simultaneously. A particular configuration files specifying the execution process are created for each thread.

The execution of this script requires other scripts, compiled programs and source files, separated in two directories (/bin and /src). It also uses parameter or data files, in the /par directory, to set general and user processing requirements: mode.par and config.par.

The script is triggered by the dataprocess.py program for the automatic execution, and by a PHP command when running under manual control through the pipeline GUI.

➤ *movefrom.pl*

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script is used to perform the configuration of files names. Different cases are taken into account initially, but more subroutines could be generated if this routine was used for other instruments or observatories. Execution success is returned to the parent process (icat.pl). Only used for manual execution.

➤ *process.pl*

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script processes the calibration and astronomical images. It uses several shell scripts, based on UNIX syntax, to execute NOAO-IRAF image treatment software and several of its subpackages. The outcome of the execution (fail, success) is returned to the parent process (icat.pl).

The shell scripts for the reduction of calibration images are:

- process_b.csh: It combines the Bias images and fits a 2-D function to the final image.
- process_bd-f_0.csh: It subtracts the final master Bias frame to Dark or Flat images.
- process_bd_1.csh: It computes the final master Dark frame.
- process_bdf_2.csh: It subtracts the master Dark frame to Flat images resulting from Bias calibration, scaling master Dark frame according to the Flat time exposure. This process is deprecated in the ICAT last version, as it increases the image noise.
- process_bdf_3.csh: It computes the final master Flat frame for each filter.
- process_bf_1.csh: It lists of the Flat images with Bias subtracted and the master Flat frames for each filter.
- process_bf_2.csh: It computes final master Flat frame and the mean counts level to normalize it.
- process_fnorm.csh: It executes the normalization process for the master Flat frame

And these are the shell scripts for the reduction of the science images:

- process_astro.csh: It processes the astronomical images using specified calibration images.
- process_astrond.csh: It processes the astronomical images using specified calibration images, when no Dark subtraction is applied.

➤ *check_calimg.pl*

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin

- Language: Perl

This script is used to check the number of calibration images during image processing. It also obtains lists of calibration images which are used during the reduction process and it checks the availability of master calibration images to apply to the science images. The outcome of the execution is returned to the parent process (icat.pl).

For automatic execution, master frames from previous nights are used in the reduction process in case no master calibration frames are found in the raw science image directory. When manual execution is performed, only a warning message is returned to the user in case a small number of calibration images are available. Supplementary checks are made to avoid any conflict during the execution of the reduction when, for instance, old processed images are stored in the output directory.

➤ **analysis.pl**

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script is used to analyze astronomical images with different analysis software packages according to the execution configuration. Intermediate steps are also executed when necessary. The outcome of the execution is returned to the parent process (icat.pl).

Sub-processes involved in the image analysis are described here following the execution flow:

- **Astrometric calibration, first step (doWCS)**: It manages the process to execute the WCS routine.
 - 16bpp.csh: This shell script changes the number of bits per pixel for the FITS images using NOAO-IRAF packages. The output image obtained is a sixteen bits per pixel image, required for the execution of the WCS routine.
 - wcs.csh: This shell script executes the WCS routine to identify the position of the field on the sky. The WCS routine cross matches targets in the FOV with the targets in the USNO or GSC catalogues. It then modifies the image header to include the WCS coefficients (keywords labelled accordingly in the list given in Sect. 11.3.2.2). These coefficients are used to convert the frame coordinates (X,Y) of the identified objects to Equatorial coordinates (RA and Dec, see doXY2SKY).
- **Object identification in the image, and photometric and astrometric analysis**:
 - Automatic execution process: Two analysis iterations are performed using SExtractor (doANSEX). Mean FWHM and sky background standard deviation values are obtained (doSTAT) from the first iteration output data and they are used in the second iteration to improve the extracted data accuracy. Crowding evaluation (doANCROWD) will be implemented in the future to improve the photometric accuracy in those areas with high density of stars. In these areas DAOPHOT PSF photometry will be applied.
 - Manual execution process: Several options are provided (doANSEX, doANDAO_BASIC, doANDAO_PLUS) with manual configuration of several input parameters through the pipeline GUI.
- **Astrometric calibration, second step (doXY2SKY)**: It executes the XY2SKY routine that computes the RA and DEC coordinates for all those targets identified in the image frame once their relative positions are determined. It is executed only when the first step (doWCS) has been successful, which means that the identification of the field has been accomplished.
- **Output format homogenization**: It modifies the output data provided by the different analysis software packages (DAOPHOT and SExtractor) in order to have the same file format. The data is stored into ASCII table files with the same fields.
- **Subroutines used**:
 - doSTAT: It calculates the mean values for the FWHM and the standard deviation of the sky background.
 - doANSEX: It executes SExtractor for the analysis of an astronomical image.
 - ansex.csh: This shell script runs the software package.
 - doANDAO_BASIC: It executes DAOPHOT using basic PSF or aperture photometry.
 - andao.csh: This shell script runs NOAO-IRAF packages and also the andao routine programmed to execute the image analysis using DAOPHOT. A file template (andao_basic_TPL.cl or andao_apertur_TPL.cl) is used to adjust some parameters included in this routine for each particular execution.
 - doANDAO_PLUS: It runs an improved execution of DAOPHOT using an optimized selection of stars to compute the PSF function for the photometric data extraction. The

optimized selection is done using SExtractor and several selection criteria. The user can also specify a file containing the list of stars to use for such purpose. A minimum number of stars to compute the PSF function is required.

- `andao.csh`: Idem. File templates (`andao_plus1_TPL.cl` and `andao_plus2_TPL.cl`) are also used.
- `doANCROWD`: It is used to define a grid to calculate a density function over the image. Not completely implemented.

➤ ***analysis_extra.pl***

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script is used to execute specific analysis of the photometric or astrometric data extracted. A subroutine (`doextrANA`) manages the execution of several Perl commands to extract relevant information from the data obtained during the analysis. The calculation of the differential magnitude between two stars at specified sky coordinates in the field is one of the examples developed.

➤ ***readheader.pl***

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script reads the header keyword values according to the system configuration (instrument keywords). It calls a program (`readkeyword`) that returns the value of a particular header keyword.

➤ ***compress.pl***

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script manages the execution of the file compression using the program `fpack`.

➤ ***jd.pl***

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script is used to convert a specific date to Julian Day. Julian Day number is a continuous count of days and fractions thereof from the beginning of the year -4712. By convention the Julian Day begins at Greenwich mean noon, that is 12h Universal Time. The outcome of the execution (fail, success) is returned to the parent process (`icat.pl`).

➤ ***log.pl***

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/bin
- Language: Perl

This script writes the result of any process executed during the pipeline workflow in a specific logging file. The pipeline log file contents are then moved to the ICAT log file when the automatic processing is executed, or it is shown through the web-based interface.

➤ ***Programs based on CFITSIO library***

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/lib
- Language: C

Several binary executables are used. They are based on the CFITSIO libraries and are programmed to perform different actions:

- `insertkeyword`: This program inserts a new keyword or updates a keyword value in the image header.
- `readkeyword`: This program reads a specific keyword value stored in the image header.

- *fpack*: This program compresses an image file. Rice is the default algorithm used, although other algorithms are possible.
- *modhead*: This program writes or modifies the value of a header keyword.

11.4.7.2 Source files

Files included here are those used as a template to specify a different configuration for each image processing.

➤ *icat_TPL.par*

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/src
- Language: ASCII text

This file is a template of the ICAT configuration file. It is modified by the automatic ICAT process or by the web-based user interface. It includes: execution control mode, final or on-the-fly reduction option (implemented only for automatic processes), relative path location of images to reduce (optional for automatic processes, mandatory for user processes), relative path location of output images and files (optional just for user processes, not used under automatic control processes), instrument used, compression options, calibration and analysis process to apply, and initial theoretical values to use for the image analysis (FWHM (pixels), threshold (sigma) and sky background (pixels)).

➤ *wcs_TPL.cfg*

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/src
- Language: ASCII text

This template file includes the WCS configuration data. It is modified accordingly during image processing. It includes the calculation of the brightest catalog star to use (in magnitudes) depending on exposure time and airmass. This star magnitude is one of the parameters the WCS package uses.

➤ **Template files for the NOAO-IRAF package configuration**

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/src
- Language: ASCII text

The list of template files for the NOAO-IRAF package configuration is shown here. They are modified accordingly during image processing and copied to the directory where NOAO-IRAF is executed.

- *login_TPL.c*: General NOAO-IRAF configuration file. It is required in the execution directory every time NOAO-IRAF packages are run.
- *andao_apertur_TPL.cl*: It defines a basic analysis using the DAOPHOT package, which includes two analysis functions (*daofind* and *phot*) to calculate magnitudes using aperture synthesis, and file managing (*psort* and *txdump*).
- *andao_basic_TPL.cl*: It defines a basic analysis using the DAOPHOT package, which includes several analysis functions (*daofind*, *phot*, *pstselect*, *psf*, *allstar*) and file managing (*psort* and *txdump*). The selection of stars to define the PSF is done just by brightness criteria (25 brightest stars are selected).
- *andao_plus1_TPL.cl*: It defines an improved analysis using the DAOPHOT package, which includes analysis (*daofind* and *phot*) and formatting (*txdump*) functions. The output file of such process is used for a star selection subroutine (using SExtractor) that improves the calculation of the PSF.
- *andao_plus2_TPL.cl*: It defines the final step for the improved analysis using the DAOPHOT package. It includes the analysis (*psf* and *allstar*) and formatting (*psort* and *txdump*) functions.

➤ **Template files for the SExtractor package configuration**

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/src
- Language: ASCII text

The template files for the SExtractor package configuration are included here (see Bertin 1998 for more information). They are modified accordingly during image processing and copied to the directory where SExtractor is executed.

- default_TPL.sex: SExtractor configuration file template.
- default.param: This file specifies the parameters SExtractor prints in the output file.
- starselect.param: This file has the parameters used when executing DAOPHOT.
- default.nnw: SExtractor classifies the objects detected in stars or galaxies. A "Neural-Network-Weights" file is used to improve this star/galaxy separation on images with Moffat-like PSFs.
- default.conv, gauss(n×m).conv, tophat(n×m).conv, etc.: Several convolution masks are used for optimal detection (pyramidal function, Gaussian function, top-hat function, etc.). Only masks with Gaussian functions are used by the pipeline. These are more appropriate for images with seeing FWHMs between 1.5 and 5 pixels (best for faint object detection).

11.4.7.3 Configuration files

Parameters or data files are located in the /par directory. Instrument shutter maps and bad pixel masks for image calibration are also stored in this directory.

➤ ICATconfig.par

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/cfg
- Language: ASCII text

This file contains the general configuration parameters of the pipeline, used on both automatic and user control processes. The parameters included are: pipeline description data, path specification, WCS global variables, file names used by the pipeline to enable multithreading execution, MEIA characterization data that must be used during image calibration (i.e., minimum exposure time to correct for dark current or minimum number of calibration images to use). An example of the configuration file is shown below.

```
# ICAT Pipeline Configuration
#

# Pipeline data
$pipeiname = "ICAT";
$pipeversion = "1.0";
$pipeinamekwd = "PIPELINE";
$pipeinamekwdcomment = "\"Pipeline name and
version\"";
$pipeprockwd = "PROCCODE";
$pipeprockwdcomment = "\"Code to identify pipeline
process log\"";
$configfile = "icat_TPL.par";
# Execution directory
$dir_icat = "/opt/pipelineICAT/";

# Partial path to input/output files directory
# Path to basic directory, in case access to image
directory fails:
$inoutdir = "/mnt/storage/rawdata/";
# Path to input (it contents night directories) and
output (it contents reduction output data) image
directories, to be used in combination with the GUI
based on PHP:
$imagein = "/mnt/storage/rawdata/";
$imageout = "/mnt/storage/reddata/icat_result/";

# List of detectors: name of detectors and file names
for the header keywords file, shutter map and bad
pixel mask. Format: ("id1: detector1_name:
detector1_headerkwd.par:
detector1_shuttermap.fits:
detector1_badpixmask.par", "id2: detector2_name:
detector2_headerkwd.par:
detector2_shuttermap.fits:
detector2_badpixmask.par",...).
@detector = ("TALON-FLI4240:TALON-
FLIProLine:FLI4240_headerkwd.par:FLI4240_shutt
ermap.fits:FLI4240_badpixmask.par", "IAC80-
```

```
SI600CCD:IAC80-
SI600CCD:ccdSI600IAC80_headerkwd.par:ccdSI60
0IAC80_shuttermap.fits:ccdSI600IAC80_badpixmas
k.par", "TALON-ApogeeAltaUUSB:TALON-
ApogeeAlta-
UUSB:ApogeeAltaUUSB_headerkwd.par:ApogeeAlt
aUUSB_shuttermap.fits:ApogeeAltaUUSB_badpixm
ask.par");

# WCS Environment variables and parameter
definitions:
$wcsradius = 1;
$WCSpath = "/opt/wcs";
$telhome = "setenv TELHOME ${WCSpath}";
$ldpath = "setenv LD_LIBRARY_PATH
${WCSpath}/libWCS:/usr/local/lib";

$path = "setenv PATH ${WCSpath}/wcs";

# File names
$ICATlogfile = "ICATprocess.log";
$GUIlogfile = "GUIprocess.log";
$img = "img.dat";
$imgformat = "img_format.dat";
$imganalysis = "img_analysis.dat";
$imgextranalysis = "img_extranalysis.dat";
$imgcalib = "img_calib.dat";
$imgcalibf = "img_calib_format.dat";
$imgcompress = "img_compress.dat";
$biaslist = "bias-list";
$flatlist = "flat-list";
$flatmean = "flat-mean";
$darktmp = "darktmp-list";
$darklist = "dark-list";
$darktimelist = "darktime-list";
$darkblist = "darkb-list";
$tmp = "tmp-list";
$filterlist = "filter-list";
$cfilterlist = "cfilter-list";
```

```

$prefilterlist="prefilter-list";
$checkwcslist="checkwcs-list";

# Minimum time exposure (seconds) for a Dark image
to be used during the reduction process (just a
warning if reduction is executed under user control)
$MDTIME = 30;

# Minimum number of calibration images to use
(Images from other days are copied to the execution
directory if the number of images is smaller than
these values; but just a warning if reduction is
executed under user control)

```

```

$MINBIAS = 4;
$MINDARK = 4;
$MINFLAT = 4;

# Analysis parameters.
# Values can be changed at the GUI configuration
page:
$theorwhm = 5 ; # Default value for FWHM (pixels)
$thres = 5 ; # Default value for Threshold (sigmas)
$theorsigma = 15 ; # Default value for stdev sky
background (counts)

```

➤ **img.dat and img_calib.dat**

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/cfg
- Language: ASCII text

These files contain the lists of astronomical (img.dat) and calibration (img_calib.dat) images to process. They are automatically created by the dataprocessing routine for the unattended control or by the web-base interface for the manual execution.

➤ **[detector]_headerkwd.par**

- Routine: Pipeline
- Location: SUB-ALIS/icat/pipelineICAT/cfg
- Language: ASCII text

The list of keywords included in the image header is provided in this file. The list of keywords for each detector is saved in a different file. Keywords missing in the image header may be included in these files giving them ad-hoc values that will be used by the pipeline. They are used when the pipeline is executed manually. The automatic execution process uses the keywords defined at dataprocess.cfg (11.4.6.3).

11.4.8 Archiving and database

The ICAT routines described above implement the archiving of extracted data and images and also the database update following the design specifications presented in previous sections. Some general implementation details are given here.

11.4.8.1 TJO-OAdM archive

Final archive has the following components for each image:

- **Image FITS files**: raw and reduced science images, raw calibration images and master calibration frames. They are all provided with a set of header keywords where the relevant data are accessible to enable image retrieval based on different criteria: proposal codes, sky coordinates, image quality, image type, operation mode, etc. Acquisition parameters and housekeeping conditions are also stored in header keywords.
- **Data file**: each science image is provided with an ASCII table file containing the photometric and astrometric data following the design specifications. A homogeneous format is given whatever the analysis software package used.
- **Log file**: ICAT provides a plain text file containing a post observation summary and data quality comments.

These files are finally stored in the off-site server (SUB-AGER) and users can retrieve them using the TJO-OAdM web page.

➤ **VO archiving**

The mentioned files are also copied periodically to the Spanish Virtual Observatory site, where archiving is done following the VO standards. This process is executed with a system synchronization command. An automatic inspection of the images is done and the major descriptors (keywords) of the image are inserted in the VO archive. A web-based interface is available in the VO server, where public and private access is provided according to the data privacy policies of the TJO-OAdM.

This work was done by the Spanish Virtual Observatory staff in collaboration with the TJO-OAdM team, following an agreement signed among both institutions.

11.4.8.2 Database

Three main actions require the interaction of the ICAT with the TJO-OAdM DDROCS database.

➤ *Extraction of the required execution constraints*

A query is executed during the image quality control performed by the dataprocessing routine to read the execution constraints for the specific task associated with the image under process. The task is identified in the database with the proposal and observation codes.

The parameters read at the Projects database level (lunar distance, airmass, sky brightness, etc.) are used to evaluate the proper execution of the task in comparison with the established requirements.

The implementation is done in the Dataprocessing routine (dataprocess.py) using MySQL queries. Access data to the database is defined in the routine configuration file (config.py).

➤ *Archiving the task execution results*

A query is executed at the end of the Dataprocessing routine workflow to create a new register into the database with the most relevant parameters of the image. The stored parameters, most of them also included in the image header, are listed in Sect. 7.3.1.1.

The implementation is done in Dataprocessing routine (dataprocess.py) using MySQL queries. Access to the database is defined in the routine configuration file (config.py).

➤ *Scheduling blocks status update*

Each time a new image is processed, the associated scheduling block status is updated according to the quality control result. The task status update is propagated in real-time to the list of tasks scheduled for the current night and to the generic list contained in the TJO-OAdM database.

The implementation of this action is still pending as it is included in the ISROCS development.

11.4.9 File management

File management matters are implemented by the routines already described in this section. However, in terms of the required header keywords for the acquired images, some of them were not included by the instrument control software (TALON). These keywords have to be added by modifying the software or using other applications: ICAT, TALON and ISROCS.

The list of new keywords included in the image header is given in Table 11.10, specifying also the software package in charge of adding them.

Keyword	Description	Package
PROPCODE	Proposal ID, a five digit code	ISROCS & TALON
TARGETID	Target ID, a three digit code	ISROCS & TALON
BLOCKID	Observation block number	ISROCS & TALON
IMAGEID	Image number within its observation block	ISROCS & TALON
OBSTYPE	The type of image (Science, Dark, Bias, Flat)	ISROCS & TALON
GAIN	Amplifier gain in electrons per analog unit	ICAT
RDNOISE	CCD readout noise in RMS electrons	ICAT
TELSTAT	Telescope status (Home, Tracking, Guiding)	TALON (pending)
DOMESTAT	The dome status (Open, Closed, etc.)	TALON
AIRMASS	Airmass	ICAT (pending)
WEATHER	Weather conditions (Clear, Partly cloudy, Cloudy, Raining)	ICAT (pending)
PHOTOMET	Atmospheric transmission	ICAT (pending)
SEEING	Seeing value from the mean FWHM measured in the image	ICAT (pending)
PIPELINE	Pipeline name and version	ICAT

IMGQS	Image quality: summary (4 digits for science and 3 digits for calibration)	ICAT
IMGQF	Image quality: file quality parameters	ICAT
IMGQP	Image quality: photometric quality parameters	ICAT
IMGQA	Image quality: astrometric quality parameters (only for science images)	ICAT
IMGQC	Image quality: execution pattern and environment conditions parameters	ICAT
IMGQE	Image quality: errors detected during image quality evaluation	ICAT

Table 11.10: List of new keywords added in the image header

An example of image header is given below for the image TJO2455825.41112_R_Imc.fits taken the 20th of September, 2011. New keywords are highlighted in bold text.

<pre> SIMPLE = T / Standard FITS BITPIX = 16 / Bits per pixel NAXIS = 2 / Number of dimensions NAXIS1 = 2048 / Number of columns NAXIS2 = 2048 / Number of rows BZERO = 32768 / Real = Pixel*BSCALE + BZERO BSCALE = 1 / Pixel scale factor OFFSET1 = 0 / Camera upper left frame x OFFSET2 = 0 / Camera upper left frame y XFACTOR = 1 / Camera x binning factor YFACTOR = 1 / Camera y binning factor EXPTIME = 60 / Exposure time, seconds ORIGIN = 'Montsec AO' TELESCOP= 'Montsec 80-cm telescope' COMMENT 11008 COMMENT 9 OBSERVER= '4 ' / Investigator(s) OBJECT = 'PG1633+099' / Object name PRIORITY= 100 / Scheduling priority INSTRUME= 'MEIA ' CDELTA1 = -0.0001011701987 / RA step right, degrees/pixel CDELTA2 = -0.0001011791615 / Dec step down, degrees/pixel JD = 2455825.411127769 / Time at end of exposure DATE-OBS= '2011-09-20' / UTC CCYY-MM-DD TIME-OBS= '21:52:01.43' / UTC HH:MM:SS.ss PROPCODE= '11008 ' / Proposal code TARGETID= 'PG1633+099' / Target group ID BLOCKID = '4 ' / Observation block ID IMAGEID = '9 ' / Image ID DOMESTAT= 'Open ' / Dome status OBSTYPE = 'SCIENCE ' / Observation type LST = '21:52:42' / Local sidereal time POSANGLE= ' 48:53:41' / Position angle, degrees, +W LATITUDE= ' 42:03:06' / Site Latitude, degrees +N LONGITUD= ' 0:43:47' / Site Longitude, degrees +E ELEVATIO= ' 14:36:58' / Degrees above horizon AZIMUTH = '270:05:33' / Degrees E of N HA = ' 5:16:33.42' / Local Hour Angle RAEOD = ' 16:36:07.94' / Nominal center Apparent RA DECEOD = ' 9:45:53.3' / Nominal center Apparent Dec RA = ' 16:35:33.87' / Nominal center J2000 RA DEC = ' 9:47:07.8' / Nominal center J2000 Dec OBJRA = ' 16:35:34.0' / Target center J2000 RA </pre>	<pre> OBJDEC = ' 9:47:08.0' / Target center J2000 Dec EPOCH = 2000 / RA/Dec epoch, years (obsolete) EQUINOX = 2000 / RA/Dec equinox, years FILTER = 'I ' / Filter code CAMTEMP = -30 / Camera temp, C RAWHENC = -1.4011602 / HA Encoder, rads from home RAWDENC = 0.16036297 / Dec Encoder, rads from home RAWOSTP = -0.18070441 / Focus Motor, rads from home FOCUSPOS= -2961.4747 / Focus pos from home, um RAWISTP = 2.003268 / Filter Motor, rads from home COMMENT Weather at UT Tue Sep 20 21:52:01 2011 WXTEMP = 13.3 / Ambient air temp, C WXPRES = 848.565 / Atm pressure, mB WXWNSPD= 0 / Wind speed, kph WXWNDDIR= 194 / Wind dir, degs E of N WXHUMID = 56 / Outdoor humidity, percent RDNOISE = '8.67 ' / CCD Readout noise GAIN = '1.21 ' / CCD Gain IMGQTJO = 'Image quality (IMGQ) at TJO-OAdM' IMGQS = '0020 ' / IMGQ summary IMGQF = 'Miss.keywords: none' / IMGQ File IMGQP = '1.11,1.5,56.86,5.0,1.31,2.0,0.34,1.0' / IMGQ Photometry IMGQA = 'yes,3.36,1.0' / IMGQ Astrometry IMGQC = '1.32, none, none' / IMGQ Env.Cond. IMGQE = 'QF:000,QP:000,QA:00,QC:100' / IMGQ Errors PIPELINE= 'ICAT1.0 ' / Pipeline name CTYPE1 = 'RA---TAN' / Columns are RA CRVAL1 = 248.8772149 / RA at CRPIX1, degrees CRPIX1 = 1024 / RA reference pixel index, 1-based CROTA1 = 0 CTYPE2 = 'DEC--TAN' / Rows are Dec CRVAL2 = 9.817027249 / Dec at CRPIX2, degrees CRPIX2 = 1024 / Dec reference pixel index, 1-based CROTA2 = -0.7926462821 / Rotation N through E, degrees AMDY0 = 4.343726833 / astrometric parameter alpha_c (radians) AMDY0 = 0.1713394483 / astrometric parameter delta_c (radians) AMDY1 = -1.765487315E-06 / astrometric parameter a1 AMDY1 = -1.763656809E-06 / astrometric parameter b1 </pre>
---	--

AMDX2 = -2.769771008E-08 / astrometric parameter a2	AMDY6 = -1.483371971E-12 / astrometric parameter b6
AMDY2 = 2.443462366E-08 / astrometric parameter b2	AMDX7 = 0 / astrometric parameter a7
AMDX3 = 0.001834818237 / astrometric parameter a3	AMDY7 = 0 / astrometric parameter b7
AMDY3 = 0.001782500458 / astrometric parameter b3	AMDX8 = 0 / astrometric parameter a8
AMDX4 = 3.530708892E-13 / astrometric parameter a4	AMDY8 = 0 / astrometric parameter b8
AMDY4 = -2.097948351E-12 / astrometric parameter b4	AMDX9 = 0 / astrometric parameter a9
AMDX5 = -7.228609111E-13 / astrometric parameter a5	AMDY9 = 0 / astrometric parameter b9
AMDY5 = 1.596671188E-12 / astrometric parameter b5	AMDX10 = 0 / astrometric parameter a10
AMDX6 = 1.340390535E-13 / astrometric parameter a6	AMDY10 = 0 / astrometric parameter b10
	AMDX11 = 0 / astrometric parameter a11
	AMDY11 = 0 / astrometric parameter b11
	AMDX12 = 0 / astrometric parameter a12
	AMDY12 = 0 / astrometric parameter b12
	AMDX13 = 0 / astrometric parameter a13
	AMDY13 = 0 / astrometric parameter b13
	END

11.4.10 Storage and backup implementation

11.4.10.1 Data levels and storage

Data storage distribution at the TJO-OAdM follows the specifications given in Sect. 11.3.2.1. It is implemented as summarized in Table 11.11.

Type	Level	Description	Final storage
Science	0	raw images	on / off site
	2	extracted data	on / off site
Calibration	0	raw images	on / off site
	2	master frames	on / off site
Housekeeping	0	registered data	on / off site
Logging	0	registered data	on / off site

Table 11.11: Data storage configuration

Each data level storing is implemented in a different way, as described below.

➤ **Science and calibration data**

Implementation of the final storage for science, calibration and their processing log files was described in the Fitscopy, Dataprocessing and Pipeline sections. They are stored following the directory structure policy specified in Sect. 11.3.6.1 and this structure is maintained also for the off-site storage backup.

➤ **Housekeeping**

Housekeeping data are stored in the TJO-OAdM database. A script querying the environment monitor receives and parses all the environment data before storing them into the database located in the SUB-ALIS server. Other data (i.e., weather data from the EM-SMC weather station) are also stored. The database is mirrored in SUB-AGER every time a change is detected, as described before. Mirroring is done using MySQL configuration features and by defining databases in SUB-ALIS and SUB-AGER as masters, in order to avoid any loss or corruption of the stored information. So, housekeeping data are kept on both sites almost in real time.

➤ **Logging**

Logging data for all the control system is centralized in SUB-ALIS. All packages running in TJO-OAdM report the result of every process they execute and this is included in the general logging system.

11.4.10.2 Backup Implementation

Two backup systems are implemented in TJO-OAdM: one in SUB-ALIS, for the on-site backup, and the second one in SUB-AGER, for the off-site backup. Their main storage capacity features were already described and the backup process implementation is given below.

➤ *On-site backup*

SUB-ALIS storage media is based on a RAID5 configuration, implemented on four hard disks. All data levels stored on-line in this subsystem take profit of this storage architecture with a fault tolerance of one disk.

➤ *Off-site backup*

The external unit (SUB-AGER), located outside the observatory, is in charge of making backups off-site. The storage media is based on a RAID6 configuration, implemented on four hard disks and with a fault tolerance of two disks.

A job scheduler is used to execute the off-site backup routine without human intervention. Secure copy processes are run for the backup copies. Copying the selected files for science, calibration and logging data levels and the whole database to the final storage media requires automatic authentication which is configured in a secure fashion.

File backup implementation is based on the “rsync” routine, a Unix command to perform incremental backups by synchronizing directories from on-site and off-site storage centres. Synchronization is configured for execution once every 30 min by the crontab job manager.

➤ *OpenROCS event for backup management*

Synchronization of the two storage media is currently made with no management of possible unexpected conditions that could interfere the optimum transference of data to the off-site storage subsystem. No prioritization is done for the available data levels, and that could be necessary if any of them should be transferred first for any reason (observatory status monitoring, proposal status update, user quick-look to the acquired data, etc.). Letting OpenROCS manage the backup process would improve the system performance in that sense. A definition of events and actions to implement this functionality is foreseen for the near future.

11.4.11 ICAT performance

ICAT has been running at the TJO-OAdM since mid 2010 with no major problems detected. Although no conclusive statistical measurements have been made to know the package performance in the multiple possible configurations (on-the-fly or final processing, single image or multiple image processing, calibration or science image frames, master calibration frames), on-the-fly processing of a single astronomical image has been proved to take less than 5 seconds to be fully completed, with almost 50% of the time spent on the Pipeline execution (tests carried out with an Intel Pentium-4 3.0 GHz 478 Bus 800 CPU).

A more thorough study is being carried out, in the frame of the global system performance test, to have a clear estimation of the time spent to execute every critical step included in the full process, and to identify those whose execution time could be improved. An improvement was already performed during the ICAT unit testing phase when the program used to read the image header, originally based on the NOAO-IRAF package, was changed to the readkeyword.c program based on the CFITSIO library. Such modification represented an important reduction of the execution time.

11.4.11.1 ICAT Pipeline routine example

A particular example of the use of the ICAT pipeline routine is shown here, with an automatic processing of nearly 400 images taken for the same target and on the same night. Execution was done at the end of the night. Extra-analysis was performed in this case to obtain differential photometry to derive the light curve of an exoplanet transiting in front of its host star. No image quality control was done as it was not still implemented.

Fig. 11.15 shows the transit light curve of the HAT-P-1b exoplanet obtained. 360 images were taken on November 30, 2007 at the TJO-OAdM, with the Johnson V filter. Calibration images were also taken and used for image correction. This figure also shows the instrumentation quality: a photometric dispersion of 0.4% was obtained in the best parts of the curve, which is expected to be reduced up to 0.2% with further improvement of calibration system. The increasing noise at the end of the observation is because the low altitude of the target. The gap during the transit was produced by an interruption of the observation sequence.

Execution conditions, process configuration and execution time are summarized here:

- Source Data
 - Name: HAT-P-1b
 - Coordinates: RA (22:57:46.82), Dec (+38:40:29.8)

- Apparent Mag. V (star): 10.4
- Planetary transit depth: 0.015 mag
- Transit signal period: 4.46529 d
- Observation setup
 - TJO-OAdM 0.8m telescope
 - Filter (J-C): V
 - Instrument MEIA (CCD FLI ProLine): 2048×2048 xip, 12.5'×12.5' FOV
 - Exposure time: 10 s
 - Observation window (LST): 23:07 – 04:02
- Process configuration and input data
 - Type of execution: automatic
 - Type of reduction: final
 - Acquisition SW and CCD: TALON - FLI ProLine
 - File names format: standard
 - Reduction: Calibration images (dark image application), Astronomical images
 - Analysis: 2 iterations of SExtractor
 - Extra-analysis: HAT-P-1b (differential magnitude)
 - Input astronomical images: 360 images (30/11/2007, JD 2454435)
 - Input calibration images: 10 Bias, 5 Dark, 5 Flat V
- Output data
 - Calibration images: Bias, Dark and Flat main frames (FITS)
 - Astronomical calibrated images: Bias, Dark, Flat and shutter effect correction (FITS)
 - Analysis using SExtractor, 2 iterations (first iteration to obtain real median FWHM and sigma values): object identification and photometry (instrumental mag.) (ASCII tables)
 - Extra-analysis: differential magnitude light curve (planetary transit curve)
- Execution time with an Intel Pentium-4 3.0 GHz 478 Bus 800 CPU
 - Total process: 1902.60 s/CPU (if 1 iteration of SExtractor: 1545.95 s/CPU)
 - File modification: 97.00 s/CPU
 - Master calibration frames computation: 27.82 s/CPU
 - Astronomical image reduction: 985.46 s/CPU
 - Astronomical image analysis (2 iterations): 769.86 s/CPU (1 iteration: 339.44 s/CPU)
 - Extra-analysis: 22.46 s/CPU

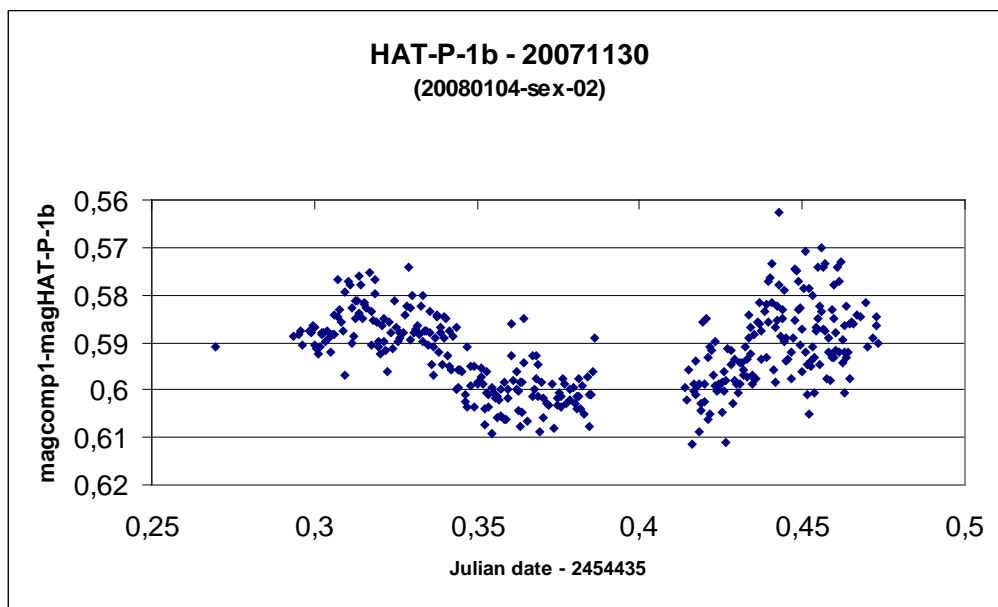


Fig. 11.15: Transit light curve of the HAT-P-1b exoplanet obtained using ICAT

Chapter 12

Observatory User interface

The observatory's graphical user interface (GUI) is mainly devoted to provide the external user with an effective and easy way to handle the proposal submitting and data retrieval processes. A second motivation, but not less important, is to provide an interface between the observatory control system and the superuser for remote or local monitoring of the observatory state and health and for its easy configuration and control. The main requirements, design and development aspects are discussed in this chapter.

12.1 User interface analysis and general design

During the system analysis, several requirements for the user interface were identified. These are listed in Section 2.3 and specified in more detail here.

12.1.1 List of requirements

12.1.1.1 Functional requirements

➤ *User interface*

The user interface is mainly used to submit tasks to be executed at the TJO-OAdM, but other functionalities are foreseen: monitoring of the observatory status at different levels (housekeeping and data flow parameters), configuring and controlling the system and, finally, executing some of the applications independently and off-line. The latter is useful, for instance, to check the performance of the data processing or scheduler routines.

➤ *Proposal management tool requirements*

- **Database access:** The user interface has to provide ways to access the database and manage the list of tasks submitted by external users or superusers.
- **Data handling:** Science programs and observation preparation details are entered using this interface and following the observation preparation pattern (see Chapter 8). Phase 1 general data and Phase 2 detailed execution constraints associated to a specific proposal have to be considered.
- **Data retrieval:** Data retrieval has to be managed through the same interface access procedure to provide a clear identification among the active proposals and the data acquired for every user.
- **TAC proposal evaluation:** TAC members should be able to use this proposal management tool to access the proposals for their evaluation and prioritization.
- **Superuser data validation:** Validation of Phase 2 data by the observatory staff (superuser) is done with this interface as the last step to enable the task selection by the scheduler.

➤ *Observatory operation and control interface requirements*

- **Status overview:** Monitoring of the system components status at hardware and software levels have to be done using the user interface. Active events and alarms for every subsystem

have also to be displayed, together with the observatory state and configuration (i.e., telescope pointing coordinates or PDU outlet status).

- **Observatory configuration and control:** The user interface has to be used by the superuser to modify the system configuration in a straightforward manner, including the observatory control system and all the associated applications. Switching among the control modes (robotic, interactive, manual and simulation) has to be doable using the user interface options. The interactive mode of operation has to let the superuser modify the observatory state, the task selected by the scheduler or simply insert a new configuration for a task.

➤ *Off-line execution of applications*

Some applications require a user interface for their execution off-line. It is mainly useful during the commissioning phase or for test and maintenance tasks, when direct control of some parameters for the configuration of the applications is necessary. It can also be useful, i.e., to let the external users simulate the schedule for their tasks or reprocess the images with different data processing configurations. The latter requires having the applications and the interface installed off-site for external user access. Applications included in this category are: ISROCS for task scheduling, TALON for task execution (telescope, dome and instrument control) and ICAT for task processing.

12.1.1.2 Development requirements

➤ *Programming language and architecture*

Different implementations could be used for each of the interface levels identified. A unique tool acting as a central node and based on web development programming languages is preferred.

➤ *User login*

A login / logout functionality should be used to enable different access levels according to user permissions.

12.1.2 Architecture overview

Different access levels are designed to provide an interface to the user with the observatory, devoted to the proposal data management, the observatory operation and the off-line application access following the requirements given above. A unique web portal is conceived for that purpose, as described below.

A scheme of the architecture general design is show in Fig. 12.1.

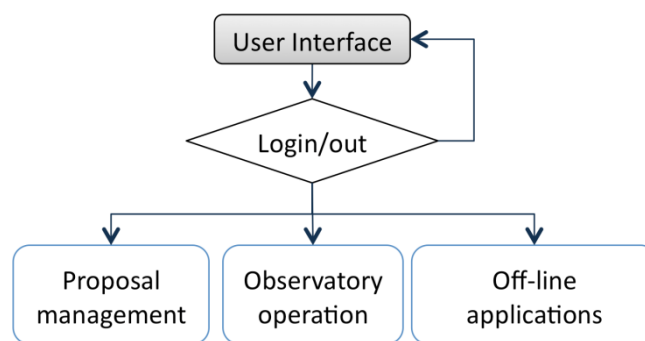


Fig. 12.1: User interface architecture overview

Each one of the modules in Fig. 12.1 and their interaction with other modules of the control system are designed as explained below.

12.1.2.1 Log in system and website

The observatory interface access is designed to be centralized in a single website where general information of the TJO-OAdM observatory can be found. A login control system is used to provide user access to the interface modules based on the user permissions.

The system efficiency is higher if external users, and also the superuser or TAC members, have a unique and specific communication channel with the observatory. The observatory website can play this role, becoming a framework for the development of all the interface modules. And it is also important for outreach activities.

12.1.2.2 Proposal management

➤ **Actors**

There are three main actors using this interface module:

- **External user:** Proposal submission and data retrieval is done manually by the external user.
- **Superuser:** Superuser can insert new tasks into the database to be considered by the scheduler for calibration and engineering purposes. The superuser also validates the task constraints specified by external users. Both processes are done using the user interface.
- **TAC:** Proposal validation is done by the TAC. Phase 1 related data are accessed by the TAC members that can select a prioritization index to be used by the scheduler. A member of the TAC may act as a chairman and does the final prioritization check.

➤ **Tasks**

The tasks can be defined as a group of operations the interface executes on the observatory database. These operations depend on the profile of the logged in user:

- **External user:**
 - Insert new proposals (phase 1 and phase 2 data).
 - Modify data already introduced. After the TAC and superuser evaluation, these data are frozen and cannot be changed any longer by this user.
 - Data retrieval is also done using this interface level.
 - Database levels that can be accessed are: Proposals, Projects and Observations.
- **Superuser:**
 - Can perform the same actions as external users, but with global permissions to see and modify all the proposal data at any time.
 - Database levels accessible: Proposals, Projects and Observations.
- **TAC members:**
 - Access the phase 1 data of those proposals with pending evaluation.
 - Add a prioritization index and comments to the evaluated proposal.
 - TAC chairman resolves the final prioritization index according to the TAC member evaluation.
 - Proposals database level is the one accessed by this process.

➤ **Interface design**

The proposal management interface is designed as shown in Fig. 12.2.

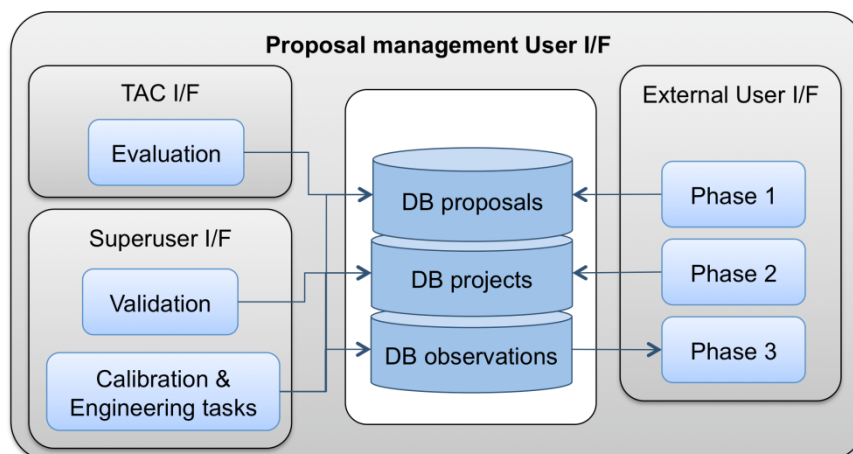


Fig. 12.2: Proposal management interface design

Proposal phases

The data submitted at each phase (1 and 2) include the general proposal description and the OB task execution specifications, as described in Chapter 7 (database level configuration) and in Chapter 8 (task preparation). The interface enables the introduction of this information in a structured, scalable and secure way for its subsequent storage in the database. Multiple investigators and multiple targets with multiple execution configurations can be associated to each proposal. Unique codes are assigned to the proposal and to the associated tasks in order to trace and recover all the information and related

data. Observations made and the extracted data can be easily retrieved by the investigators using these codes.

12.1.2.3 Observatory operation

The observatory operation interface provides a friendly environment to the superuser to perform several tasks that require human interaction with the control system. At the TJO-OAdM, it is translated to an interface with OpenROCS.

➤ **Actors**

- **Superuser:** The Superuser is the only user authorized to use this interface both remotely or locally.

➤ **Tasks**

This operation interface is mainly devoted to perform the following actions:

- **Remote or local monitoring** of the observatory, including hardware and software component status.
- **Modify the control system configuration** to adjust it to the observatory hardware and software. OpenROCS event management is also configured through this interface. This process is only made when installing the control system or when making significant changes (i.e., installing a new instrument).
- **Change the execution configuration** for a specific task (i.e., pointing coordinates), enter a new task not previously stored into the database or change the task selected by the scheduler for execution. It provides the interactive mode of operation used for engineering tasks.

➤ **Interface design**

The monitoring and control capability is provided to the superuser with an interface with the OpenROCS servers located at each subsystem. Fig. 12.3 illustrates the interaction among OpenROCS and the other layers of the observatory. Interfaces with the main applications running at the Hardware and Data flow layers are shown (ICAT, ISROCS, and TALON). These interfaces are included in the control system architecture.

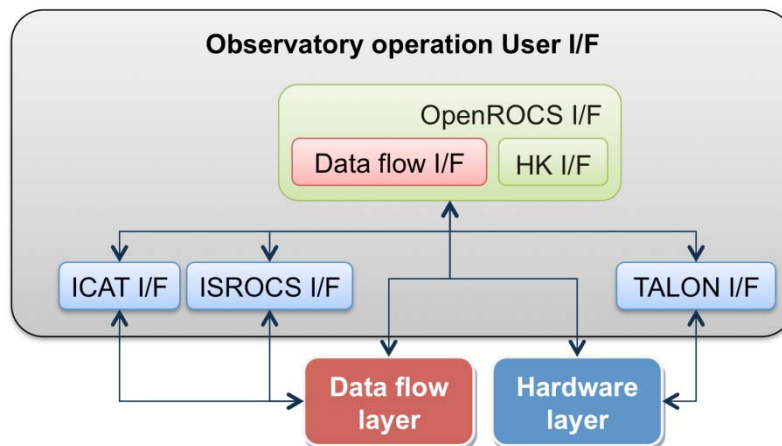


Fig. 12.3: Observatory operation interface design

12.1.2.4 Off-line access to applications

Three applications require a direct interface with a user: ISROCS, TALON and ICAT. They were developed as standalone software tools, whose execution could be necessary to perform independent operations, not linked to the observatory workflow activity.

A module specific to develop the web contents was added at this level, although it was not originally included in the requirements.

➤ **Actors**

Three actors were defined to run this applications:

- **External user:** This user with reduced permissions can execute the task scheduler (ISROCS) and the ICAT applications.

- **Superuser:** Can execute the three mentioned applications with no limitations.
- **TAC:** The committee members could use the long-term scheduler (ISROCS) running on simulation mode when evaluating the proposals and when defining their scientific priority.

➤ Tasks

The tasks the user can perform with each one of these applications are defined by the user profile:

- **External user:**
 - Can execute the scheduler running in simulation mode. He/she can run the long-term planning to check the feasibility or the appropriate definition of the execution constraints of a new proposal during the submission process.
 - A reprocessing of the acquired images is allowed by using the ICAT application. Some parameters can be selected to check the differences when using different software configurations, mainly for the image analysis. Output files are stored in a temporary directory to avoid overuse of disk storage capacity.
 - Both ISROCS and ICAT applications are run on the off-site subsystem (SUB-AGER).
- **Superuser:**
 - Can execute ISROCS running in simulation mode for the three planning terms (long, mid and short) and using all the active proposals.
 - Can execute the data processing application on all the images acquired by selecting the configuration parameters. Output files are stored at a temporary directory.
 - Can execute TALON on- and off-site to run tasks with the observatory science devices (telescope, dome and instrument) without using the OpenROCS software infrastructure.
 - Can use it to develop the web contents.
 - Can execute ISROCS and ICAT on- and off-site.
- **TAC:**
 - This user can execute ISROCS running in simulation mode for the long-term planning and using all the proposals with pending evaluation.

➤ Interface design

Logged in users can select among the three mentioned applications and the web development module, and use them to execute the tasks described above, as illustrated in Fig. 12.4. ICAT, ISROCS and the web development module use the information stored in the observatory database.

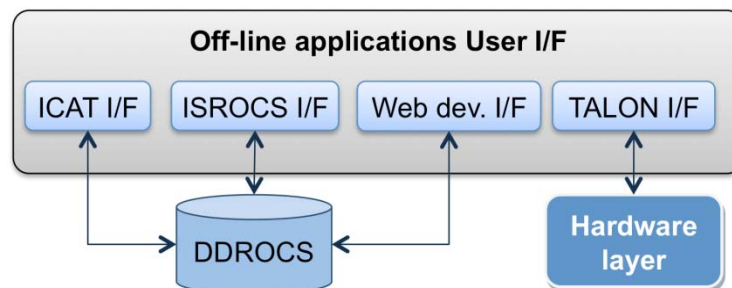


Fig. 12.4: Off-line access application interface design

12.2 User interface detailed design and implementation

The detailed design of the interface modules is based on a web configuration. This kind of implementation provides multiplatform compatibility and the capability to be run by any user without the need of a local installation. Those developed in the framework of this Thesis are described in detail: the observatory web site, the proposal management tool and the ICAT interface.

12.2.1 Observatory website

The observatory website was the first software interface designed and developed, although its use was initially only devoted to outreach purposes (Fig. 12.5). It became a fundamental tool for the robotic operation of the observatory when the proposal management and remote monitoring tools (webcams and environment data) were added as features. A description is provided here to give the framework in which all the interfaces are developed.



Fig. 12.5: OAdM website home page

12.2.1.1 Implementation architecture

The website architecture is structured in sections focused on the following areas:

- **General observatory information.**
- **Technical information:** Description of the instrumentation.
- **Observing at the TJO-OAdM:** Science case description and Proposal management tool interface.
- **News and publications:** Press releases, proceedings and technical notes.
- **OAdM environment:** Weather data and live webcams for remote monitoring.
- **Images:** Images of the observatory and celestial objects.
- **IntraOAdM:** Internal area.

Contents for most of these sections are generated from the observatory database (Fig. 12.6). So, a suitable interface between the webpage and the database was defined with such purpose.

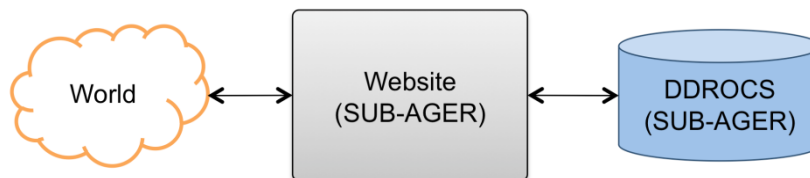


Fig. 12.6: General website architecture

The database levels are related with the web sections as follows (Table 12.1):

Web section	Web subsection	DB level	DB sublevel
Observing at the TJO-OAdM	Service-observing request – Phase 1	End-to-end data flow	Proposals
	Service-observing request – Phase 2		Projects
	Status and results (Phase 3)		Observations
News and publications	OAdM news	Web page	Docs
	Publications		Docs
OAdM environment	Current data	Housekeeping	Environment
Images	OAdM images	Web page	Docs
	Astronomy		Docs
IntraOAdM	Access (restricted)	Web page	Users

Table 12.1: Website sections and connection with database levels

Database levels are configured to enable performing the required tasks.

12.2.1.2 Implementation details

All mentioned sections are based on a simple structure (Fig. 12.7) running in combination with style sheets. Thus, maintenance is very simple.

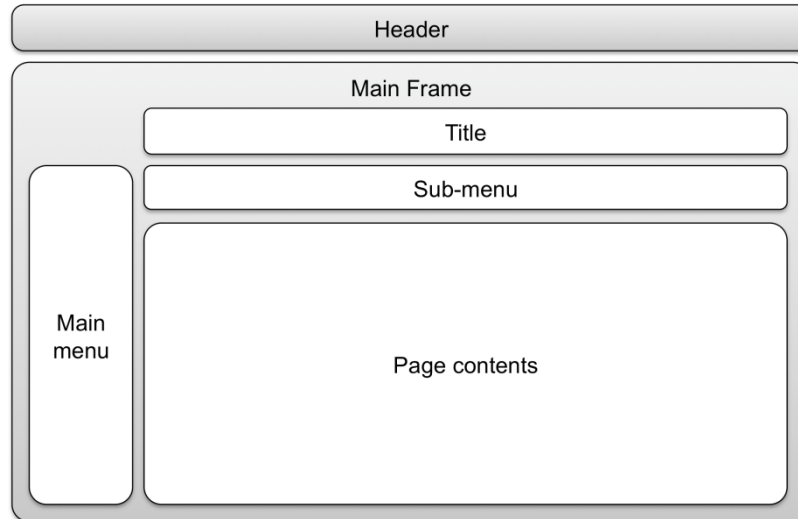


Fig. 12.7: Web page basic structure

When the web section is selected by a user in the Main menu block, a specific string of characters is used to select the contents for the subsequent blocks to generate the web page at the server side. This kind of implementation was chosen because it simplifies any change on style or web structure in the future.

➤ **Source code organization**

The source code is mainly structured in a different directory for each language (Catalan, Spanish and English), and two more directories for documents and for uploaded files. The same structure is repeated for each directory for a specific language:

- File with Main menu contents.
- File with Sub-menu contents.
- Files with Section contents (one file per section).
- File with Header contents.

➤ **Software development tools**

The website was developed with a combination of PHP, Javascript and MySQL commands for database interfacing. And the server is running with the Apache software.

12.2.1.3 Upgraded version

A new website is currently under development. It will centralize the access of the external users and the superusers to all the interface modules by using the same login system. This new approach will enable different user access by allowing different functionalities (module interfaces) depending on the user permissions, easier management of the proposals and website development, and having a more user-friendly interface tool.

The login system was not needed in the first stages of the project. The access of the external users was handled with the initial website configuration. In addition, an intranet area was developed for superuser access control only. At this restricted website area, links to devices (UPSs, PDUs, etc.) or applications (ICAT) with web-based interfaces were available. This simpler approach has several drawbacks in developing the user interface, being the arduous browsing among proposals its major problem. The need to identify superuser or TAC member users, with specific permissions, and to use the same website for different purposes motivated the upgrade to this new implementation.

12.2.2 Proposal management tool

This application is fundamental for the unattended operation of the observatory. It handles and stores all the information the scheduler needs to perform the task selection. It was developed in the website framework once the robotization project was mature enough to start offering service time using internal calls for proposals.

12.2.2.1 Implementation architecture

The general implementation architecture is based on the scheme shown for the observatory website (Fig. 12.6). The sequence followed by the application to manage the proposal process is described in Fig. 12.8.

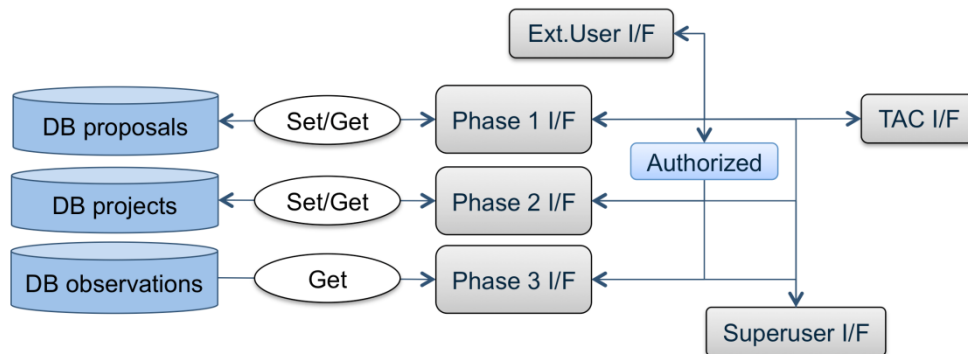


Fig. 12.8: Upgraded proposal management implementation architecture

This scheme illustrates the tasks and user privileges discussed in the general design. It is important to mention that the set/get commands are enabled to external users until data are frozen by the TAC (for phase 1) or by the superuser (phase 2). TAC members can also add information at the Proposals database level with the result and comments of their evaluation.

The currently implemented architecture is shown in Fig. 12.9, where a less flexible interface is described. External users cannot modify the proposal data once submitted. In addition, Superuser interacts with the database contents using phpMyAdmin¹, a free web software tool developed for this purpose. TAC members are provided with the proposal data by specific file reports.

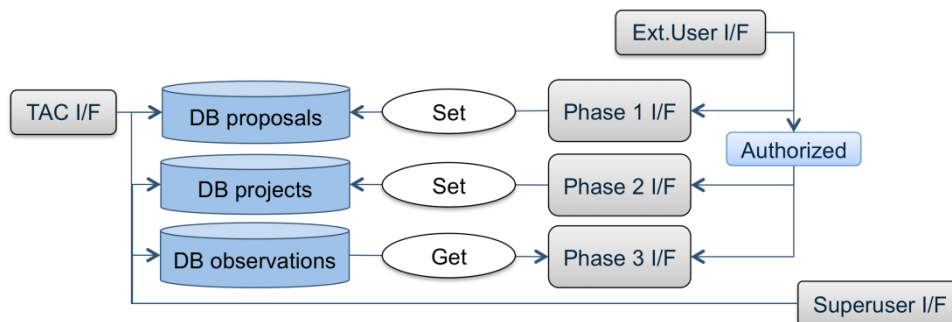


Fig. 12.9: Proposal management implemented architecture

12.2.2.2 Implementation details

This interface uses the complete database structure to store the submitted data and the observation results. Each proposal is identified with a unique code, which is used in the submission Phases (Fig. 12.10). The investigators can also track their proposals and retrieve the data by entering into a private area in the website, which require authentication with this code and a password (Fig. 12.11).

¹ <http://www.phpmyadmin.net>

Select one of the two phases according to your proposal submission status. Use 'test' Code and 'test' Password to access Phase 2 on a demo version.

Phase 1
New Proposal

Phase 2
Accepted Proposal

Code:

Password:

Fig. 12.10: Proposal submission interface (Phase 1 and Phase 2)

OAdM - Montsec Astronomical Observatory

[Service-observing request](#) |
 [Help Ph1 & Help Ph2](#) |
 [Status and results](#) |
 [Science Cases](#)

Status and Results

Proposal PI	Title	Code	Submission Date	Status	Results (password)
Staff TJO-OAdM	Calibration images	10001	2010-03-19	Ready	<input type="text"/> <input type="button" value="↩"/>
Xavier Luri	Astrometry of spacecrafts at L2 from ground	10014	2010-10-22	Phase 2 Submitted	<input type="text"/> <input type="button" value="↩"/>
Bosch,J.	Cossos Menors del Sistema Solar amb el TJO	10015	2010-10-27	Accepted	<input type="text"/> <input type="button" value="↩"/>
Balaguer-Nuñez, L.	Variability in the corona of M67	10016	2010-10-29	Phase 2 Submitted	<input type="text"/> <input type="button" value="↩"/>
Mongiñó,M.	Tracing the Perseus Arm at the anticenter	10017	2010-10-29	Phase 2 Submitted	<input type="text"/> <input type="button" value="↩"/>

Fig. 12.11: Proposal data retrieval (Phase 3)

➤ **Source code organization**

The source code is included in the website English version directory. The proposal management tool was only developed in this language given the likely profile of the expected user community. It also simplifies software maintenance.

➤ **Software development tools**

The web pages are developed using PHP and Javascript, providing a dynamic generation of the required form fields to fit the proposal definition interface to the user constraints. MySQL commands are used to interact with the database.

➤ **Modules**

Modules to implement the proposal management tasks are included in the PHP files as webforms that allow a user to enter data that are sent to the server for processing. They are basically used to perform the data collection and data storage or data retrieval processes to/from the different database levels. A pair of snapshots are shown in Fig. 12.12 and Fig. 12.13 to illustrate the implementation of the Phase 2 submission tool.

Service-observing request | Help Ph1 & Help Ph2 | Status and results | Science Cases

Service-observing Request - Phase 2

In the current form you must transform your accepted Phase 1 proposal into a list of observations with the detailed information needed for implementation and scheduling of your science program.

* = required fields

General Information

Project Code test
Project title Test
Observation science case Exoplanets

Investigators

Principal Investigator IP Name
 Affiliation

Observing Conditions

Sky brightness Bright
Seeing Average
Photometric Non-Photometric

Timing

Type of observing time Flexible

Observation data *

Target Altair (00:00:00, 00:00:00) **Target Group ID [1]**

Fig. 12.12: Phase 2 proposal submission I/F snapshot – General information

Observation data *

Target Altair (00:00:00, 00:00:00) **Target Group ID [1]**

Instrument CCD FLI ProLine
Priority 5

Observation Block

Exp.T.(s)	Filter	Binning	Defocus	Repeats	Priority
	V	1x1	No	1	5

iterations 1

Scheduling requirements

No Constraints
 Constraints

BETWEEN

Time interval			Phase interval		
Start	End	Toler	Start	End	Toler
dd/mm/yyyy	dd/mm/yyyy				

AFTER OBSERVATION BY

Time	Phase
±	±

If requirements based on phase, enter:

Period **Zero-Phase (HJD)**

Lunar distance > 0 deg

Airmass Min. 2.0 Max. 1.0

Calibration images

Extra-Bias N 0
 Extra-Dark N 0 Exp.Time 0
 Standards

Add new Target Group

Fig. 12.13: Phase 2 proposal submission I/F snapshot – Task execution constraints

12.2.2.3 Upgraded version

A new proposal management tool is under development and this is based on the architecture shown in Fig. 12.8. The new implementation is based on an AJAX-like web development method, combining PHP, MySQL and JavaScript used on the client side to create a most interactive web application. It is being integrated into the new website version.

12.2.3 ICAT off-line access interface

The ICAT interface is devoted to the image reduction and analysis pipeline. It provides an off-line access to this routine, mainly to configure different execution parameters, according to the description given in Chapter 11 for manual operation. Image quality control or archiving are not yet accessible by this application interface and will be added in the future. In case a full processing (pipeline, quality control and archiving) for a particular set of images is to be done, it can be run by a superuser using shell commands.

12.2.3.1 Implementation architecture

The ICAT interface is based on web programming languages. The general implementation architecture is based on the scheme shown for the observatory website (Fig. 12.6) as it was developed in such framework.

The user interface implements the manual execution of the ICAT pipeline included in the data processing specifications. The processes performed by this off-line access are very similar to those executed by the automatic task processing run, indicated with dashed lines in Fig. 12.14. They mainly build the configuration files and trigger the ICAT pipeline script (icat.pl).

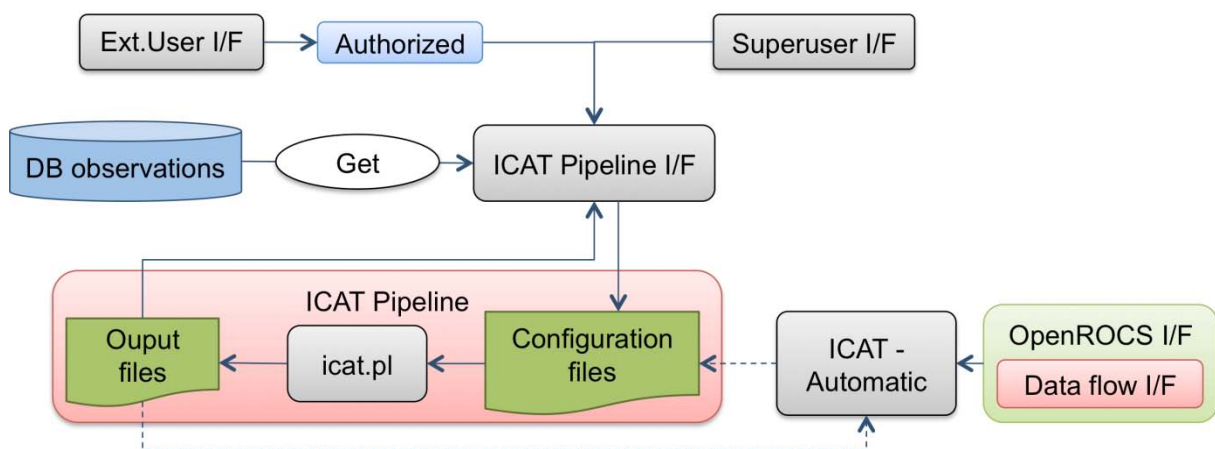


Fig. 12.14: ICAT pipeline implementation architecture

The web interface has an input data section that is divided into two main blocks:

- **Configuration:** The first block is used to select the execution process parameters that are stored in the configuration files (ICATconfig.par, img.dat and img_calib.dat), described in Section 11.4.
- **Images:** The second block enables the dynamic selection of images to use from the Observations level of the DDROCS database, where raw image data are stored.

A submission button starts the form contents verification and the execution of the icat.pl script. This initial page is linked to a help page.

When the script execution has finished, the output log file contents are displayed, together with the input parameters used. The user can then execute a new processing routine.

Output file storage is done at a preconfigured output directory. An automatic name is generated for this directory. The user can specify a different name, but cannot change the absolute path where the output directories are created to avoid a lack of security.

12.2.3.2 Implementation details

➤ Source code organization

According to the architecture scheme (Fig. 12.14), this application has to be installed on the same computer server where the database and the images are stored. The latter reduces the system complexity and the process execution time. It is currently installed on the off-site subsystem computer (SUB-AGER), where it uses the mirrored database, for external user and superuser access. It is also installed on SUB-ALIS only for superuser on-site access.

This application has permissions to access the database, execute the icat.pl script and create directories and files. It is composed of several files that are located at the same directory and linked to the web server structure.

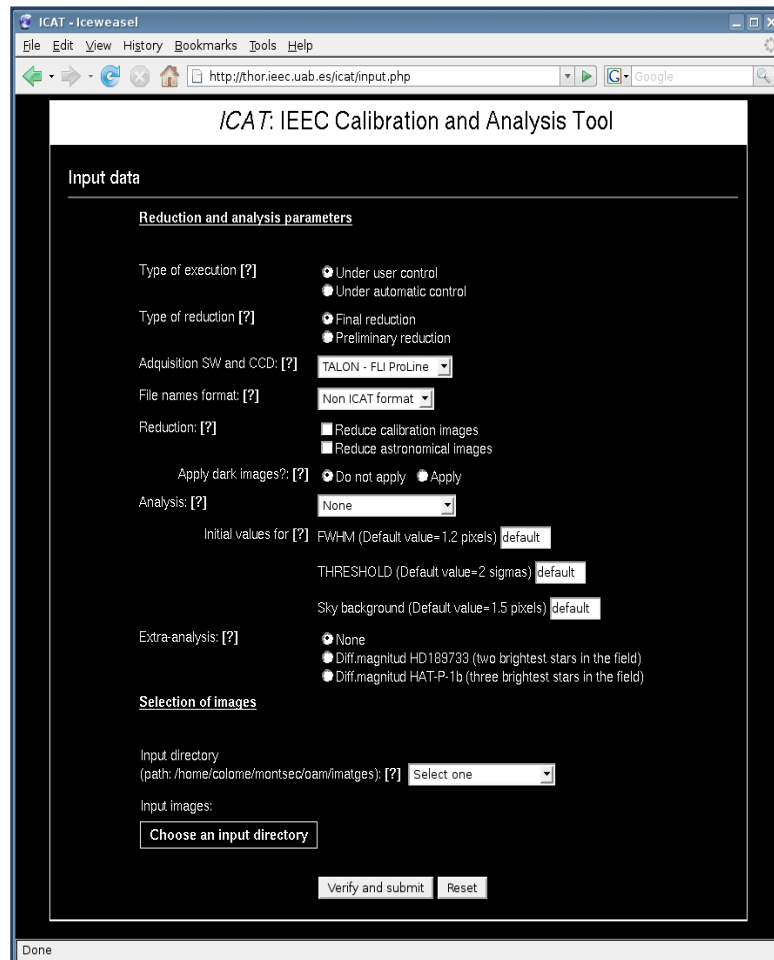


Fig. 12.15: First implemented interface of ICAT

➤ Software development

PHP is used to program the ICAT interface. This interface includes a webform to configure the pipeline execution (Fig. 12.15). PHP also manages form submission, creates the ICAT input data files, and gets data from the observatory database by using MySQL queries. A help page for user information is also available.

AJAX (Asynchronous JavaScript And XML), a combination of PHP, MySQL and Javascript, is also used to dynamically update the interface contents when selecting images to use in the process.

Apache is used as the web server software to run this application.

12.2.3.3 Upgraded version

The ICAT application interface will be redesigned and moved into the general website to be accessed following the same user access control, as all the interface modules. Some missing features will be added: a filter to let the user select only those images associated to his proposals, a new section to download the data obtained after the reduction process, etc.

12.2.4 TALON off-line access interface

TALON is provided with its own user interface that is executed to monitor and control the telescope, dome and instrument camera. Several GUIs are used: XObservatory (Fig. 12.16), for general operation control and system monitoring; Camera, for CCD operation and image inspection; Shm, for shared memory data monitoring; etc. They are used for local or remote control. Thus, real-time remote operation and monitoring can be performed using the network connection (Clear Sky Institute 2000).

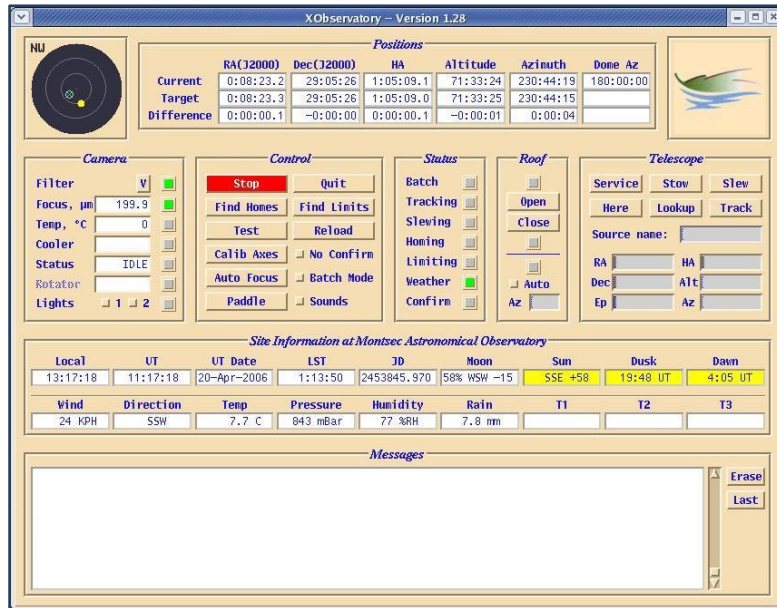


Fig. 12.16: TALON XObservatory interface

Network data flow can be highly compromised when running the system remotely depending on the required data and image transference rate. In addition, it is not possible to integrate this control tool on a web-based interface. These main drawbacks motivated discarding these interfaces and adding science instrument operation to the OpenROCS user interface, as described below.

12.2.5 New user interface modules

Some interface modules are under development by the engineering team. They all are planned to be integrated into the observatory website and accessed using the login system control.

12.2.5.1 Observatory operation

The observatory operation interface module is basically an interface for the OpenROCS software. Three main submodules are being designed to perform: the control system configuration, the system monitoring and the control of the system operations. Fig. 12.17 illustrates a pre-design of the OpenROCS interface covering the monitor and control features. Each box shows the parameters related with the different system layers (OpenROCS (green), Data flow (red) and Hardware (blue)) grouped into services, hardware devices and applications. Set and Get options indicate that these parameters can be just monitored for system supervision, or modified for direct superuser control on the engineering operation mode. The latter option is activated when the superuser changes the general system state from Robotic to Engineering.

The OpenROCS communication among its services and the external client (shell or GUI) involves a minimum data rate exchange. Thus, a low communication data flow is ensured.

➤ Configuration

OpenROCS has multiple variables that have to be configured to adjust the system behaviour to the user requirements: housekeeping data ranges for environment and system parameters, network IP addresses to computers and devices, etc. Furthermore, each specific application (ISROCS, ICAT and TALON) has its own configuration parameters. The global system setup may be configured with a user friendly interface. A new interface module will be used to specify the full set of parameters.

➤ Monitoring

OpenROCS is always monitoring all the parameters that describe the system health and the environment data. Showing these data through the web user interface is doable in a straightforward manner. This functionality is very important for remote or local monitoring of the system.

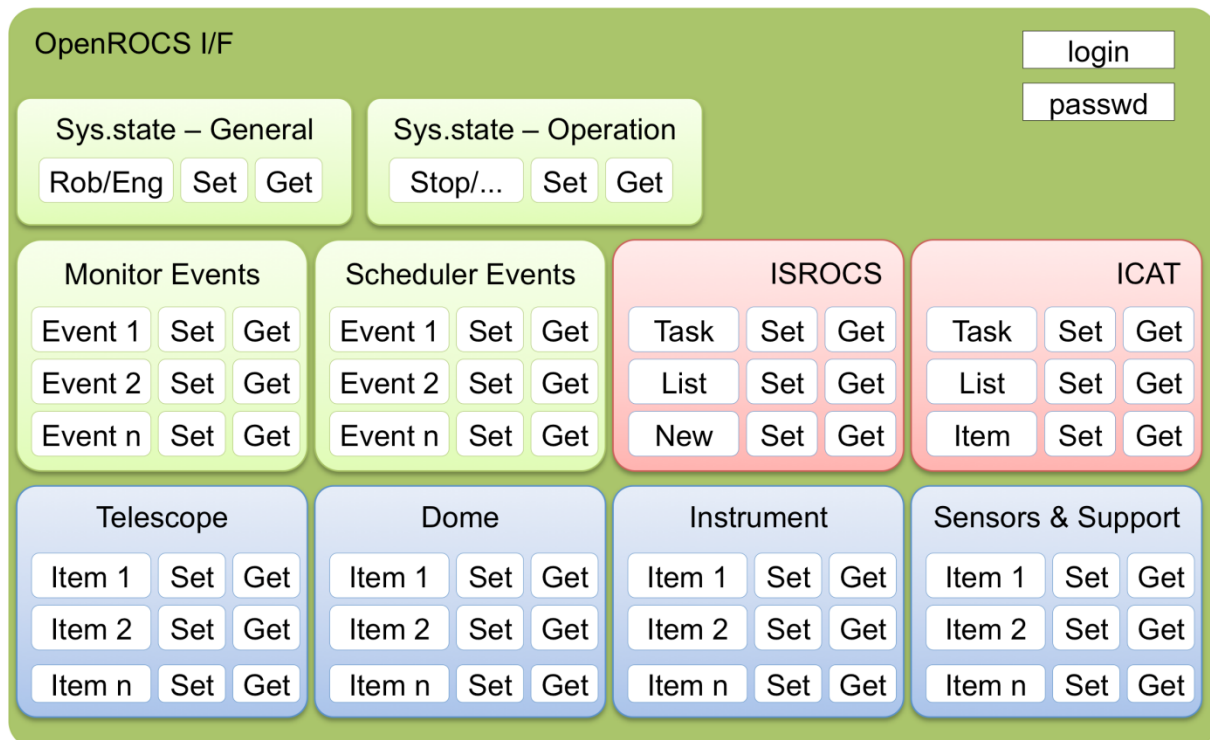


Fig. 12.17: OpenROCS interface general design

➤ **Operation control**

Letting the superuser execute control operations using OpenROCS is the preferred approach at this stage. The software is already prepared to communicate and execute actions on software applications or on hardware devices at the TJO-OAdM for the automatic operation control. The control system software interfaces were designed and developed to achieve this purpose and can be used to provide interactive operation through the user interface.

A different control feature is the capability to change the system operation mode and state. That is, moving the system from any of the states (except Shutdown) to a different one. A first system state change from Robotic to Engineering is required to enable any other state transitions (i.e., Stop to Science) performed by the superuser.

12.2.5.2 Off-line access to applications

➤ **ISROCS, ICAT and TALON**

The user interface for the ISROCS, the ICAT and TALON applications must include the tasks specified in the general design architecture given in the Section 12.2.5.1. Two different approaches for their implementation could be considered: developing a different interface for each application or adding all functionalities to the OpenROCS interface. The latter is the preferred solution as it would represent having a central access to all the control system levels, although it could increase the complexity of this interface. A pre-design for this integral interface is shown in Fig. 12.17. The TALON interface is not maintained as a compact unit in this scheme as all its functionalities are distributed in a slightly different logical structure.

➤ **Web development interface**

A new version of the website is under development based on the CakePHP¹ framework. This new implementation is based on a separation of the web programming in three different layers: design, structure and contents. Web contents are stored in the database and an ad-hoc interface editor is used to modify them and publish the modification afterwards. It makes web development and maintenance easier for the superuser and increases the system flexibility and efficiency by providing different interfaces to the same database as a function of the user profile.

¹ cakephp.org

Chapter 13

Quality Assurance

The system robustness and the quality of the overall operations performed by the control system are especially critical for the robotic operation of an astronomical observatory. These features were introduced in the specification of requirements of the TJO-OAdM (Sect. 2.3) and were taken into consideration during the design and development phases.

We have applied quality assurance (QA) practices during all the project phases. We have made an important effort to define and implement such practices as a fundamental ingredient of this Thesis. The main goal is to ensure that the final implementation meets the project needs. These practices are described in this chapter grouped by: risk analysis, verification control, testing, and configuration management.

The role played at the TJO-OAdM by the author of this Thesis has been strongly associated to Project Management (PM) and QA tasks, mainly since the developing phase of the robotization project (2008) and in collaboration with Dr. Ignasi Ribas. This role has been central to elaborate the work described so far without losing the global perspective of the project and to better define the correction actions for achieving the desired quality.

13.1 Quality assurance and qualified robotic operation

Unattended observatories have an inherent high level of risk. This risk can be minimized by designing and developing physical and logical architectures that ensure a high level of robustness and reliability. Systems engineering¹ practices are recommended to achieve this goal. These deal with work-processes to handle a project development over its life cycle, and its main objective is to ensure the quality of the implemented system. QA activities are the fundamental tool for Systems engineering to monitor and evaluate that the system developed is suitable for the intended purpose and it performs as expected.

We did not develop a formal QA plan that could be used as an integral tool to follow the project from the very beginning. Instead of this, a complete definition of the observatory science case and science requirements and a well defined operational concept (see Chapter 2) were used. But some QA processes became more and more important as the project complexity grew.

The QA processes applied at the TJO-OAdM are composed by:

- **Risk management:** The aim of Risk management is to ensure that project risks are identified and covered by actions to eliminate or reduce them, bringing the residual risk to a level that is acceptable.
- **Verification control:** Verification control is defined to confirm that the specified requirements have been fulfilled and that the implemented solutions perform correctly.
- **Testing:** It includes the testing activities needed for the verification control.
- **Configuration management:** It is focused on the activities done to handle all types of documents, hardware devices, software code and test procedures.

¹ <http://www.incose.org>

A description of each one of these parts is given in the following sections.

13.2 Risk management

A simple risk analysis was done during the pre-robotization project without a specific formalism that helps planning the response or designing mitigation actions. Only several issues were considered mainly focused on reducing the risk of unexpected conditions during the nominal routine operations (i.e., direct lightning strikes or power outage).

The main objective of this Thesis was the reduction of the number of SPFs or the risk that these represent in order to ensure the system robustness. A deeper analysis was then performed.

Risks associated to personnel injuries have not been taken into consideration at any of the project stages, although some preventive protocols have been always used to minimize them during the development phase. A specific security plan is currently being designed to establish the procedures to access and work at the observatory avoiding any risk or to activate contingency actions in case of occurrence. This plan is mainly devoted to maintenance and engineering work that requires people being physically present at the observatory. It is out of the scope of this Thesis and not included here.

13.2.1 Risk analysis

In this section we identify the risks for the observatory operation that could produce severe damage to the system in terms of financial losses in case of occurrence.

Lessons learned during the project development phases were used to recognize the possible improvements to implement. They were applied to define the corrective, mitigation and contingency actions that are discussed in this section.

It should be mentioned that it is not possible to identify all risks, nor is it possible to eliminate them. The remaining risk, called residual risk, requires human intervention in case there is no automatic palliative solution implemented.

The assessment and analysis of each risk includes:

- **The occurrence likelihood:** It is the probability that a risk event occurs. This value (from 1 – low – to 5 – high) is difficult to estimate and the number given in Table 13.1 is based on the past experience. Poor statistics on some events is available due to the low occurrence frequency.
- **Risk level:** It is the quantification of the impact in terms of damage severity. The lower level (1) indicates only minor damage to instrumentation. Severity increases up to level 5, where the instrument is lost (it is not recovered at reasonable cost) in case of occurrence.

We define a classification of the risks based on these parameters by obtaining a score index by multiplying the risk level and the occurrence likelihood. Two different categories are defined depending on this value:

- **Unacceptable risk:** For any risk classified at this category, correction actions have to be defined. The goal is to reduce the risk to an acceptable level. Risks with score values between 10 and 25 fall in this category.
- **Acceptable risk:** For the acceptable risks, mitigation and contingency actions are defined and implemented in order to restrict the likelihood and to reduce the potential impact in case of occurrence. The goal is to increase the system safety. Risks with score values between 1 and 9 fall in this category.

The analysis performed is split into two steps. The first one corresponds to those unacceptable risks evaluated during the pre-robotization and the robotization projects that have been downgraded to acceptable risks thanks to correction actions. The work done in that sense was conducted in collaboration with the engineering team or by some subcontracted industries. And the second one contains the risks according to the final implementation, as presented in this Thesis. The increase in system robustness is evident when comparing both analyses.

13.2.1.1 Unacceptable risks

Several risks were initially present at the observatory having a high level of damage severity and occurrence likelihood. They are summarized in Table 13.1.

Code	Risk component	Risk name	Description	Effects	Probability	Risk level	Risk score
U.1	Housekeeping	Damage caused by direct lightning strike or induced power surges	Two lightning rods, a deep connection to ground and the storm detector for lightning protection were installed, but they proved to be insufficient to avoid the risk.	Damage to computer hardware and/or to electronics.	5	5	25
U.2	Dome shutter	Dome shutter control failure	Electronics to control the dome shutter movement were unreliable.	Loss of dome shutter control. Damage to electronics and hardware inside the dome if failure happens at the open position and in case of bad weather conditions.	4	4	16
U.3	Housekeeping	Power supply system failure	The system unreliability was very high while running with the power generator and a set of batteries. Power supply was frequently lost when high humidity conditions affected the electronics of the battery transformers.	Loss of control. Critical devices not stopped at a secure position (dome shutter, telescope, instrument). Damage to electronics and hardware inside the dome in case of bad weather conditions.	4	4	16
U.4	Housekeeping	Critical environment data not available	Communication link to the weather station was unstable and unreliable.	System not moved to a safe configuration in bad weather conditions. Damage to electronics and hardware could be produced.	3	4	12
U.5	Communications	Communications failure	Frequent loss of external communications mainly due to the radiolink fiber transceivers.	No external alarm communicated in case human reaction is required.	3	4	12

Table 13.1: Unacceptable risks

Corrective actions performed on these risks were mentioned in Chapter 4 and are summarized here:

- Risk code: U.1
- Correction: Electric insulation was used for any communication wiring and protections were installed in the electrical wiring. New lightning strikes have occurred after implementing these changes and the probability of damage caused by them has been reduced to acceptable levels. Severity has been also reduced after protecting all the critical devices. Installation of new equipment is always done following the same protocol: it is installed with proper screening by the storm detector, it has electric protections of different level depending on the sensitivity to power fluctuations of its electronics, filters or optical connections are used for the communication wiring (RS232, USB, etc.). Minor damage could still occur, but they have been minimized.
- New risk score: 3 (likelihood) × 2 (severity) = 6
- Risk code: U.2
- Correction: A redundant control system was implemented at the electronics and software control levels and unreliable electronic components were removed by redesigning and simplifying the system. Risk on mechanics and electric components is still present, although the failure likelihood is much smaller (see Table 13.2).
- New risk score: 1 (likelihood) × 4 (severity) = 4
- Risk code: U.3
- Correction: A complete redefinition of the power supply system was done and implemented. The electrical wiring of the entire building, the power generator, the batteries and the transformers were revised and modified to increase their reliability. Support equipment (UPS, PDU) was also installed for the system to stop always on safe mode.
- New risk score: 1 (likelihood) × 4 (severity) = 4
- Risk code: U.4
- Correction: The original serial connection was discarded. A new weather station was installed based on wireless technology. No connection cuts have occurred since then.
- New risk score: 1 (likelihood) × 4 (severity) = 4
- Risk code: U.5
- Correction: This situation had a very high likelihood of occurrence because of the reliability of the power supply system (risk U.3). A new external connection system was recently installed based on fiber optics and providing a wider bandwidth and higher reliability. The radiolink will be only used as a backup system.
- New risk score: 1 (likelihood) × 4 (severity) = 4

13.2.1.2 Acceptable risks

The main risks identified in the current system configuration are shown in Table 13.2 grouped by observatory components. Grey background cells correspond to SPF risks. Mitigation and contingency actions are specified for each case to describe how they are managed. Only those risks with a risk score value larger than 3 are discussed, because lower values do not involve critical situations for the system safety. Failures on telescope movement enter into this category, as it cannot suffer any collision or damage thanks to the telescope and dome dimensions and their configuration.

Code	Risk component	Risk name	Effects	Probability	Risk level	Risk score	Mitigation	Contingency
A.1	Dome shutter	Shutter mechanics (motors, gears) or power electronics failure.	Damage to electronics and hardware inside the dome if failure happens at the open position and in case of bad weather conditions.	1	4	4	Robust and reliable implementation, status monitored by the control system. Maintenance policies.	Error is detected and communicated to observatory staff. There is no alternative action . Urgent human response is required (<120 min).
A.2	Dome shutter	FSC1 control electronics failure.		2	4	8		
A.3	Dome shutter	Failure of the UPS-Dome or the dome power line surge protection.		2	4	8		
A.4	Dome shutter	Nominal control failure (SUB-ESTALL, fiber optics transceivers, CSIMC boards).		2	4	8		
A.5	HK	Rain detector failure.	System not moved to a safe configuration on bad weather conditions. Damage to electronics and hardware could be produced.	1	5	5		No redundancy is available. There is no alternative action .
A.6	HK	Weather sensor failure (humidity, wind speed), unavailable (SUB-REBEI HW or SW failure) or providing unreliable data.		1	4	4		Data are taken from a redundant weather station. Also used to detect sensor calibration lost. Components are restarted trying to recover them.
A.7	HK	Failure of UPS-CPU, CPUs power line surge protections or PDU IP-switch 1.	Shutdown computers, sensors and telescope communication.	1	4	4		Watchdog functionality at FSC1 closes the dome shutter. No feedback is possible. No-urgent (>120 min) human response is required.
A.8	HK	Internal communications failure.	No communication among computers nor housekeeping data available.	1	4	4		System is moved to Stop state. No urgent (>120 min) human response is required.

Table 13.2: Acceptable risks

This analysis proves the improvement of the system safety and robustness thanks to the changes made, most of them carried out in the framework of this Thesis.

13.2.2 Single Points of Failure

Contingency actions described in Table 13.2 show that there are four risk situations with a SPF: A.1, A.2, and A.3 for the dome shutter; and A.5 for the housekeeping. They are highlighted with a grey cell background. An occurrence of any of these situations could be fatal for the system if it happened

during a storm. So, special care has been taken to ensure that their occurrence likelihood is minimized by using robust equipment and periodic maintenance. They are all evaluated here:

- Risk code: A.1
- Remarks: Shutter mechanics and power electronics were manufactured by an experienced company that has installed domes at sites with worse environment conditions than those at the OAdM. A component of the shutter mechanism used to control the open-close movement broke early in 2011 more than seven years after the dome installation and with almost no maintenance performed. This mechanical component was replaced and there has been no more failures of this sort. Maintenance tasks for all the dome shutter components are currently being defined and included in the maintenance plan of the observatory.
- Risk code: A.2
- Remarks: FSC1 control electronics was developed using electronic components of high robustness and its performance was exhaustively tested before the installation at the observatory. No failures have happened since its installation in February 2010. Maintenance tasks for this electronic board mainly involve checking the response of the microchip and the embedded software. These processes will be included in the maintenance plan of the observatory.
- Risk code: A.3
- Remarks: Power is supplied to the dome using a dedicated and overdimensioned UPS unit. The UPS installed is based on on-line double-conversion technology providing the maximum level of protection. The UPS load, battery charge level and battery status are automatically monitored and communicated by the control system to the observatory staff in case of failure. A self-test of the batteries is routinely performed and the UPS is included in the maintenance operations plan.
- Risk code: A.5
- Remarks: The current rain detector is based on photodiode technology. Its simplicity and robustness ensures the low likelihood of this fault. Errors have never happened since its installation in 2005. It communicates its signal to the environment monitor at SUB-REBEL by using serial connection, representing a point of weakness due to the lower robustness of the electronic components. The latter was corrected when FSC1 board was installed and the rain detector signal communication was hardwired to it. Rain detection signal triggers the shutter close movement independent of any high level software.

It is nevertheless clear that redundant detection should be added to the system to avoid this SPF. The FSC1 and CRRD boards were developed to have a second rain detector connected and to combine signals coming from both detectors using an OR logical gate. Redundancy, then, would be achieved by installing a second detector. However, a different solution was finally chosen: the TFRM project installed a third weather station on site in early 2011 equipped with a similar detector. Network access to data was provided to the TJO-OAdM control system. Current plans are focused on integrating these data into the environment monitoring performed by OpenROCS to have the required duplicity of rain detection sensors.

13.3 Verification and validation

The verification and validation processes are mainly intended to prove that the final implementation meets the project needs.

According to IEEE-STD-610 standard¹ (applied to software development), verification is the process of evaluating the system to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase (does it perform correctly?). And validation is done to evaluate the system during or at the end of the development process to determine whether it satisfies specified requirements (does it satisfy the user needs?).

A general discussion on the high level requirements and how they have been translated into the overall design and implementation is presented first. And the system verification process is described next.

¹ <http://standards.ieee.org/>

13.3.1 Robustness, reliability, efficiency and quality

The design and implementation of the TJO-OAdM system for the unattended operation was done by ensuring its robustness, reliability, and efficiency at both software and hardware levels. The quality of the system operations measured in terms of the quality of the acquired data was also a goal of this work.

13.3.1.1 Robustness and reliability

The risk management presented before proves that the system **robustness** has been achieved at a high level of confidence. The system design was based on different subsystems and modules with a reduced number of interdependencies that improves the verification process and, therefore, the detection of weaknesses. In addition, its implementation was done by choosing technology with proved robustness, especially for the critical elements (dome shutter control, housekeeping monitoring, etc.) and at different levels: electronics, computers, support equipment, software applications and OS, etc. For instance, almost all the software acquired and developed is running on Linux platforms owing to the better stability and reliability when compared with other operating systems; and extremely robust computers are used for especially critical subsystems (SUB-REBEI and SUB-SARGA). Special care was also taken to protect the system against the unfavourable environment conditions at the Montsec mountain. An example of that is the use of power surge filters or fiber optics for communications.

Reliability has also been pursued at all the system levels. Commercial hardware selection, and new hardware and software design and manufacturing have been done with this feature in mind.

The system verification helps proving the system robustness and reliability with specific tests to check the correct performance for each functionality. The tests defined with this purpose are detailed in a test procedure given in Sect. 13.4.

➤ **Robustness vs. number of subsystems for a distributed control**

A multi-service system can be implemented based on different approaches to reduce the risk of general failure and increase robustness: use a different computer for each service in order to lose just one service if the computer fails; or join the maximum number of services into a single computer, reduce the number of computers and use extremely robust machines.

The solution adopted at the TJO-OAdM is a compromise between them: the number of subsystems was minimized joining as many services as possible in the same machine, but using different computers for redundant services or for services that had to be accessed by other instruments at the observatory site. The latter is illustrated in Fig. 13.1, where AllSky represents the instrument for the detection of meteoroids and bolides and TFRM is another telescope at the OAdM as introduced in Sect. 1.2.1.

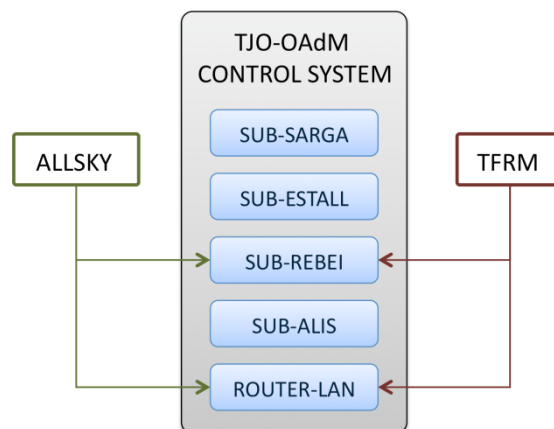


Fig. 13.1: Services with external access

The justification of using up to five subsystems is due to the impossibility of merging some of them in the same computer.

- **SUB-ESTALL**: Its main functionality is the operation control of the instrumentation at the TJO-OAdM. It must be very stable, safe and reliable, because its malfunctioning could have a direct impact on hardware maintenance and safety. It must be then under the supervision of a computer with a very restricted access.

- **SUB-ALIS.** It must have a high processing and storage capacity, and it must be very robust. It must have the configuration of a server, in terms of stability and security, as it provides external access for remote supervision and backup functionalities. It supports the data flow control processes.
- **SUB-REBEI** and **SUB-SARGA:** These are the most critical subsystems and it is a strong requirement that they work with high level of confidence. They are responsible of the most sensitive issues of the housekeeping layer. SUB-REBEI is the central server of weather data and is accessed by all the facilities at the observatory site. Therefore, it must run independently of the TJO-OAdM operation status, including Shutdown. These subsystems use computers prepared to work under severe conditions.
- **SUB-AGER:** It executes the off-site supervision of the observatory.

13.3.1.2 Efficiency

End to end data flow and housekeeping layers were introduced and designed with the purpose of enhancing the system efficiency. Maximizing the system feedback throughout was derived to have an efficient response in real-time under the responsibility of the OpenROCS software and its event-action management. Finally, a proposal management tool, a dynamic database (DDROCS), an intelligent scheduler (ISROCS), and a data processing application (ICAT) were specially designed to run on-the-fly and provide this real-time feedback applied to the scientific tasks and the acquired data.

13.3.1.3 Quality

A goal of the work done in this Thesis was focused on having a system running in a fully unattended manner and providing optimal operation quality (robustness, reliability and efficiency) and highest data quality (data quality control, telescope optical and mechanical performance). In this sense, the development life cycle followed the traditional approach that is oriented to guarantee the quality of the system. It was based on requirements, design, implementation, and testing phases; although it was combined with a more flexible approach, where any of the mentioned phases was modified or adapted to changing conditions (requirements, staff, etc.).

The quality achieved will be derived from the results obtained by testing the system performance, as described in Sect. 13.4.

13.3.2 Requirements traceability to module

We present Table 13.3 that describes how to trace the implementation of each specific requirement specified in Sect. 2.3.2 to a system module (hardware or software) at the design level. Its use is to validate that all the project needs have been included in the final system.

Some descriptors are used when the complete implementation of a specific module is not available or when the requirement imposes a generic constraint that applies to the full system design and cannot be traced. The following descriptors are used:

- **Ongoing:** The module has been designed and is under development.
- **Pending:** No module has been designed to implement this requirement.
- **N/A** (not applicable): The requirement is generic for the whole system design.

Req.Code	Req.Title	Prty.	Module
FUN-1	Observatory operation	5	Physical architecture
FUN-2	Scientific return	5	Physical architecture
PER-1	System efficiency	5	ISROCS
PER-2	Operation overheads	5	End-to-end data flow packages (ISROCS, ICAT)
QUA-1	Housekeeping control	5	OpenROCS Not all the SPFs have redundant equipment, but risk has been minimized to acceptable levels (see Sect. 13.2.2).
QUA-2	End-to-end data flow	5	ICAT
QUA-3	Science data quality	5	N/A
QUA-4	Qualified operation	5	OpenROCS, ICAT

QUA-5	Efficiency, reliability and robustness	5	N/A (see Sect. 13.3.1)
QUA-6	Maintainability	5	Physical architecture (see Sect. 13.5)
OP-1	Observatory operation	5	N/A (mainly included in the ROCS design)
OP-2	Time to repair	5	OpenROCS. Time requirement not traceable.
OP-3	Observatory modes	5	N/A (inherent to all modules)
OP-4	Observatory users	5	N/A (inherent to all modules)
OP-5	Human interaction with the observatory	5	Observation preparation and Proposal management tool (Sect. 12.2.2), ROCS
OP-6	Observation control processes	5	ROCS
OP-7	Environment	5	Physical architecture
CTR-1	Global management program	5	OpenROCS
CTR-2	TALON usage	5	Physical architecture (telescope modifications)
CTR-3	End-to-end data flow control applications	5	End-to-end data flow layer modules (DDROCS, Proposal management, ISROCS, TALON, OpenROCS, ICAT)
CTR-4	End-to-end data flow processes	5	End-to-end data flow layer modules (DDROCS, Proposal management, ISROCS, TALON, OpenROCS, ICAT)
CTR-5	Events and alarms generation and response system	5	ROCS, OpenROCS
CTR-6	Housekeeping	5	ROCS, OpenROCS
CTR-7	Scheduling	5	ISROCS
CTR-8	Data processing	5	ICAT
CTR-9	VO compliance	5	ICAT (Archiving)
BOU-1	Observatory control	5	ROCS, OpenROCS, ICAT
BOU-2	Observation constraints	5	Observation preparation and Proposal management tool (Sect. 12.2.2)
BOU-3	Observatory output	5	ROCS, OpenROCS
USR-1	Proposal management tool	5	Proposal management tool (Sect. 12.2.2)
USR-2	Observatory status overview	3	Observatory user interface (Observatory operation)
USR-3	Observatory configuration	3	Observatory user interface (Observatory operation)
USR-4	Pipeline interface	2	Observatory user interface (Off-line access to applications)
DEV-1	Initial configuration	5	Physical architecture
DEV-2	Stray light	5	Physical architecture (telescope modifications)
DEV-3	Exposure of critical elements to open environment	5	Physical architecture (telescope modifications)
DEV-4	Counterweighing	5	Physical architecture (telescope modifications)
DEV-5	Mirror covers	4	Physical architecture (telescope modifications)

DEV-6	Telescope pointing	3	Pending
DEV-7	Focus temperature dependence	3	Ongoing: Commissioning task, see Sect. 13.4.1.
DEV-8	Telescope finder	1	Pending
DEV-9	Reliability of control electronics	5	Physical architecture (dome)
DEV-10	Encoder	5	Physical architecture (dome)
DEV-11	Integration of the dome control system	5	Physical architecture (telescope modifications)
DEV-12	Redundant control	5	Physical architecture (dome, SUB-SARGA)
DEV-13	Vignetting of the CCD FLI4240	5	Physical architecture (MEIA instrument modifications)
DEV-14	Flat fielding system	5	Physical architecture (Calibration – Dome flats). Ongoing: trade-off to compare sky and dome flat performance.
DEV-15	Characterization of CCD	4	On-going: Commissioning task, see Sect. 13.4.1.
DEV-16	Characterization of UBVRT filter set	3	Commissioning task, see Sect. 13.4.1.
DEV-17	Use of the Apogee Alta U16	2	Pending: OpenROCS modularity provides the capability to add new instruments (i.e., ARES).
DEV-18	Environment control system I	5	Physical architecture (housekeeping modifications, SUB-REBEI), ROCS, OpenROCS
DEV-19	Power supply and communications	5	Physical architecture (housekeeping modifications)
DEV-20	Remote monitoring	4	Physical architecture (housekeeping)
DEV-21	Security	3	Pending: A security system is installed, but not integrated in the control system.
DEV-22	Environment control system II	2	Pending
DEV-23	Power control	5	Physical architecture (housekeeping modifications) and OpenROCS
DEV-24	System modularity	5	N/A
DEV-25	Software infrastructure	5	OpenROCS
DEV-26	User interface development	3	Observatory user interface
NONFUN-1	Open source and platform compatibility	5	N/A
NONFUN-2	Software programming	4	N/A
NONFUN-3	Development tools	4	SVN, Git, TRAC (see Sect. 13.6)
NONFUN-4	Documentation	4	N/A
NONFUN-5	Interoperability / Compatibility	4	N/A
NONFUN-6	Modifiability / Maintainability / Extensibility / Scalability	4	N/A
NONFUN-7	Testability	4	N/A
NONFUN-8	Operability	3	ROCS, OpenROCS

NONFUN-9	Quality	3	Pending
NONFUN-10	Usability	3	Pending

Table 13.3: Requirements traceability to module

The results of this traceability analysis show that the high priority requirements (priority = 5) have been implemented at the current stage. Some improvements are still under development as it pointed out in the description of the applications or the control system given in different chapters of this Thesis.

The implementation of the requirements of lower priority is planned for further improvements of the observatory, since these are not critical to achieve the robotization control envisioned for the TJO-OAdM.

13.3.3 Verification control

The verification of the system is done at different levels and based on different test cases. A testing policy is used to check the correct behaviour of the modules running independently (unit testing) or running integrated within the global system.

Unit testing has been performed on every software and hardware component without a formal procedure specification, whereas verification of the global system has been formalized with a test procedure. The latter defines the protocol to execute when evaluating the agreement between the expected and the real response of the system under specific conditions, and it is described in the next section.

13.3.3.1 AIV phase

The AIV phase was done based on a gradual integration of new subsystems, hardware components or software applications at the observatory. The level of human independence, robustness, efficiency or quality of the system was increased each time a new element was added. The modular design and a clear definition of the interfaces provide the flexibility and integrity required for this kind of project development.

It is important to emphasize that this approach has important drawbacks, as it increases the risk on the project management and on guaranteeing the system integrity. It has introduced delays in some development stages of the TJO-OAdM project, due to inconsistencies at the interface layer that arised during the integration. This was fixed by defining a more rigorous unit testing, running in combination with a platform that simulates the response of the rest of the system.

A monolithic AIV process is only foreseen when reviewing the system performance at some development stage. A test procedure for the system verification is required. This was applied to the System Acceptance Review, held in April 2010, and it will also be considered for the System Operation Review, planned for mid-2012.

13.4 Testing

Two main verification test procedures are defined in order to check the system performance: unit tests for individual software applications or hardware components, and a global test for the overall control operation.

Unit testing is not described in detail here, as it was done without following any formalism. Unit failures could compromise the reliability of the system, but tests have usually been extensive enough to detect any bug on the component prior to its integration. The specification of the requirements for most of the critical modules (see the analysis section in several chapters of this Thesis) helps avoiding such failures.

The telescope and instrument characterization is done following a different approach, because their performance is the most critical issue regarding the quality of the acquired data. A classical astronomical commissioning process was defined. It could be considered a unit test procedure to check the mechanical and optical performance of the scientific devices, with the exception of the dome. The quality of the site is also included in the test in order to evaluate the sky conditions for astronomical observations. This gives a general picture of the capability of the system to collect high precision data with the available scientific hardware. The astronomical commissioning process is described in the following section.

The global test designed to check the operation control is defined as a formal test procedure and it is described below in more detail. It is routinely executed as an integral verification process of the system devoted to detect implementation errors and to collect data for a statistical analysis of the system response.

13.4.1 Summary of the astronomical commissioning results

The procedure for the classical commissioning was designed and executed by the engineering team and can be seen in detail in Casteels et al. (2010). The results obtained are summarized here. The performance of the telescope, instrument and site proved to be reasonably good. More extensive results and conclusions are expected from a second commissioning phase under process, thanks to the changes applied to the telescope configuration (optics, electronics, etc.), as already described in Chapter 4.

13.4.1.1 First commissioning results

A first commissioning process was designed and executed in parallel with the start of the robotization project. The results discussed here correspond to this commissioning phase.

➤ *Differential and absolute photometry*

In order to check the precision in the differential photometry, fields with two or more Landolt standard stars were observed following a predefined sequence. The results obtained had a precision (RMS) of 10^{-3} mag, which proved the good system performance. This was confirmed with the first scientific results published, as illustrated in Fig. 13.2 (Herrero et al. 2011) where the differential-photometry light-curve of the transiting exoplanet WASP-33 b (HD 15082) is shown. These data correspond to the observations made in the R J-C band with the TJO-OAdM telescope the 28th of September, 2010, and have a 2.1 mmag (RMS) precision.

Concerning the absolute photometry, a mean precision of $\sigma \sim 0.02$ mag was obtained for all Johnson-Cousins filters and colors.

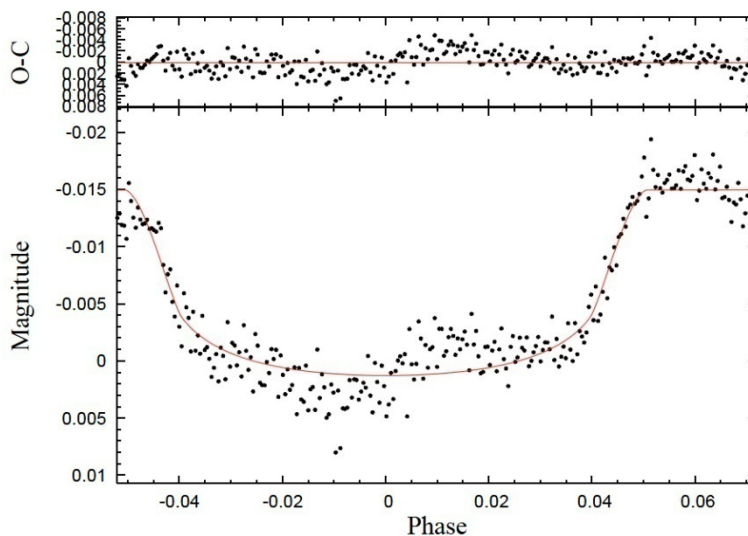


Fig. 13.2: Light-curve of WASP-33 b. The top panel shows the residuals of the best fit.

➤ *Astrometric precision*

Fields in the Astrometric Calibration Regions (ACR) were observed, containing 10^6 stars in the range $10.0 < V < 18.3$. These have accurate positions and proper motions that permit the determination of the current positions of the stars. Equatorial coordinates were derived for the ACR stars detected in the images using WCS routines.

The results show that there is no error dependence on chip position or on instrumental magnitude (as faint as $V > 11$). Averaging several images, astrometric precisions of $\sigma_{RA} \sim \sigma_{Dec} \sim 0.04''$ were obtained.

Concerning the mechanical behaviour of the telescope, a typical pointing error of $\sigma_{HA} \sim 20''$ and $\sigma_{Dec} \sim 16''$ was obtained. And tracking errors were found to limit the exposure times to 5-10 minutes with the adopted pointing model.

➤ Site testing

Between 1999 and May 2009 several campaigns were carried out to determine the sky quality at Montsec. The results obtained concerning light pollution, weather conditions, seeing, and atmospheric extinction proved the good quality of the sky on the site.

Light pollution

The main conclusion of the various observing campaigns is that the OAdM location has very little artificial light pollution. The sky background brightness at the zenith in moonless nights averages $22.0 \text{ mag}\cdot\text{arcsec}^{-2}$. Thus, the background sky is basically in its natural state, with the atmospheric glow caused mostly by stars in our galaxy and zodiacal light. This value indicates that this is a particularly dark site with night-sky brightness comparable to other observatories, like Calar Alto ($22.01 \text{ mag}\cdot\text{arcsec}^{-2}$), La Palma ($21.9 \text{ mag}\cdot\text{arcsec}^{-2}$) or Mauna Kea ($21.9 \text{ mag}\cdot\text{arcsec}^{-2}$), as shown in Sánchez et al. (2007).

Weather conditions

After several years of continuous monitoring, weather conditions at the OAdM show that over 70% of the nights have more than 4 hours with good weather for astronomical observations. Specifically, the humidity was found to be below 80% (90%) 64% (77%) of the time. In terms of temperature, the minimum night temperature in such series was -10°C and the maximum day time temperature was between $20\text{-}25^{\circ}\text{C}$ (the maximum night temperatures were around 20°C). As for the wind speed, recorded gusts averages did not exceed $15 \text{ m}\cdot\text{s}^{-1}$, with peaks on some nights up to $25 \text{ m}\cdot\text{s}^{-1}$. Therefore, the wind does not lead to dome closure except on very rare windy nights.

Another important point to consider is cloud coverage. The automatic weather stations measured the average solar illumination between 10-14 UT every day, and compare such values to the theoretical illumination without atmosphere. The ratio of the observed and theoretical values is defined as the clarity index, and was found to be above 0.5 during 72% of the time. This indicates that good clarity conditions were measured during more than 70% of the time. Note that the night cloud coverage could be slightly different.

Seeing and atmospheric extinction

The median seeing at the OAdM was obtained using a RoboDIMM (Aceituno 2004) and was found to be $\sim 0.86''$ (Fig. 13.3).

Atmospheric extinction coefficients were derived by fitting absolute photometry equations with Landolt standard stars, observed during several nights. First- and second-order extinction coefficients in the V band were derived giving average values of ~ 0.10 . 289 observations of stars between $6 < V < 16$ over 8 nights were used.

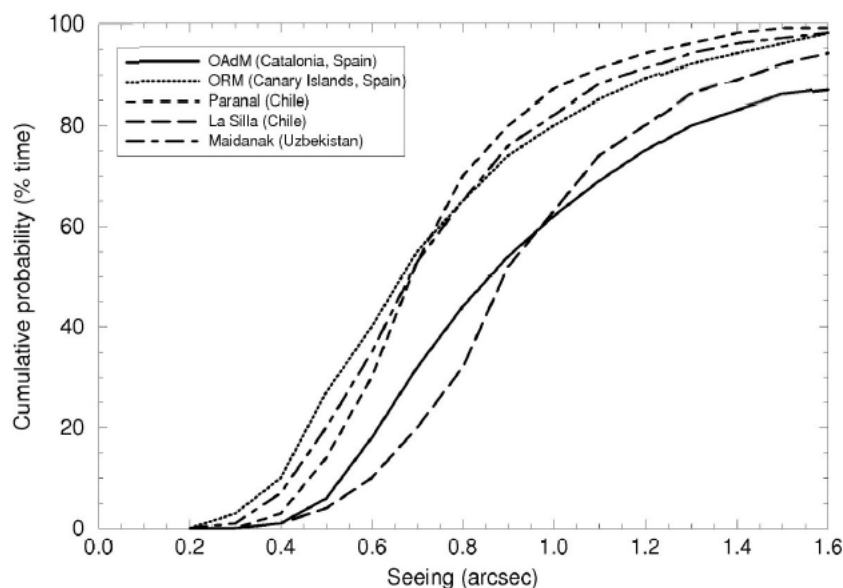


Fig. 13.3: Seeing at OAdM and at important observatory sites

➤ *Data processing routine*

All the observations done for the commissioning were reduced and analyzed using the ICAT package running in a default configuration using calibration images (Bias, Dark and Flat frames) and Source Extractor for the data extraction.

13.4.1.2 Expected improvements for a second phase for the astronomical commissioning

The results obtained in the first phase are expected to improve after the changes done on the telescope optics, mechanics and electronics (i.e., new baffle configuration). New weather data are also available to the OAdM provided by the SMC and they are currently under evaluation. A new commissioning phase started in late 2010 and results will be published in the near future.

13.4.2 Operation control tests

We defined a test procedure at the end of the robotization project in order to validate the robotic operation of the TJO-OAdM (Ribas 2010). It is summarized in this section.

The operations run at the observatory since the end of the robotization project (end 2010) were focused on testing the control system. The tests showed a good performance of the robotic operation. However, these tests enabled the detection of some shortcomings in the optics, mechanics, low level software and electronics of the telescope and an important fraction of the time was invested by the engineering team in solving these issues in the last months. A small number of tests could be finally carried out.

The quality of the obtained data is now at optimal level and we are currently executing the test procedure in a systematic way to obtain sufficient data to evaluate the control system performance. This will continue until the System Operation Review, scheduled for mid-2012.

13.4.2.1 Test procedure

The test procedure is intended to demonstrate that each of the observatory subsystems performs as defined by its requirements and that the entire system will be able to operate in a robust fashion for an extended period of time. It describes the test campaign necessary to validate the robotic operation of the TJO-OAdM through the testing of the individual subsystems and the simulation of different scenarios: routine observation in good weather conditions, incidences arising from bad weather conditions, system failures, etc. The control system is designed to take decisions in real time to maximize the science output from the observatory while preserving the integrity of the system, the latter being the highest priority. Image processing and data control quality need also to be tested to ensure the quality of the observations.

➤ *Test objectives*

A positive report from the test certifies that the TJO-OAdM is compliant with the robotic operation needs although, necessarily, further testing will take place during the current stage of robotic operations (with non-interactive human attendance) to ensure that all risks are mitigated. The tests described here address a subset of all possible incidences and thus validate proper behaviour in such cases. Hardware failures are much more difficult to foresee and to properly account for and the first stage of robotic operations should help to identify the most common failures so that they can be monitored. The design of the system should allow for such hardware monitoring to be fine tuned as needed.

Robotization requirements

A full description of the requirements for the robotic operation of the TJO-OAdM can be found in Sect. 2.3. The points below list the main aspects of the requirements that this test procedure addresses:

- The TJO is required to operate in a fully unattended mode. This includes the programming and execution of the observations in an optimized fashion.
- The system comprises the full data flow from the acquisition of an image to its automated quality control system and subsequent storage. This includes the feedback to the scheduling system to ensure that a given observation is only complete when its quality is sufficient (as defined by both the system and user).
- In the case of incidences affecting normal operations, the system generates alarms (environment, hardware, software) and performs the subsequent specified actions described in Sect. 5.4.4.4.

➤ **Test of individual subsystems**

Each individual subsystem within the control system has to be tested for normal operation and for its reaction to environment, hardware and software incidences. The test makes assumptions regarding the performance of some subsystems (barring hardware incidences that the system shall monitor) that are beyond the scope of the robotization project as they were already present in the initial state:

- Telescope: Normal operation of the telescope (i.e., pointing, focusing and filter changes) is assumed.
- Dome: Normal operation of the dome (i.e., flap and shutter opening and closing, and azimuth pointing) is assumed.
- Computers: Normal operations of the control computers and OS are assumed.
- Sensors (weather stations, rain sensor, cloud sensor, GPS) performing nominally.
- Auxiliary electronic devices (fiber converters, IP switches, rain sensor electronics) performing nominally.

The individual subsystems that the test monitors are:

- Dome control electronics with hardwired rain sensor feed, watchdog feature and redundant control interface.
- System SUB-ESTALL according to the tasks defined in Sect. 3.3.2.
- System SUB-ALIS according to the tasks defined in Sect. 3.3.2.
- System SUB-REBEL according to the tasks defined in Sect. 3.3.2.
- System SUB-SARGA according to the tasks defined in Sect. 3.3.2.

Each individual event and action in Sect. 5.4.4.4 has to be tested by reproducing the conditions that the system is monitoring. An alarm generator simulator is employed to emulate those conditions that are difficult to reproduce (either too time consuming or driven by sensor measurements).

➤ **Test of the robotic operations**

To verify that the requirements of the system are met, four test procedures have been defined to address the different aspects of robotic operations: routine observing, observing with the advent of weather alarms, observing with the appearance of system alarms, data processing workflow. The full test procedures are described below.

Routine operations

The first test is the simulation of one night of observations fully unattended without inducing any alarm, so the system acts only if the contour conditions are out of the security ranges. This test is carried out during the day, so the mid-term (static) scheduling is programmed to operate during the day.

This part of the test evaluates how the control system defines and executes a typical observing timeline. The timeline is simplified to contain three basic states: S1 – system stopped, S2 – system calibrating (dome closed) and S3 – system performing science operations or thermalizing (dome open). The test monitors the following steps:

1. (S1) Night planning: In the current situation of static scheduling, the system should plan the night by generating a sorted list of tasks with all calibration and science images in an optimized way (i.e., user restrictions, minimum hour angle, moon distance, etc).
2. (S2) Start up the system: At the starting time determined by the scheduler, the ROCS powers up the telescope and dome electronics and drives from the PDUs, launches the TALON software, opening the telescope covers, powers up the instrument CCD, homes the telescope, unparks and homes the dome.
3. (S2) Start calibration images: ROCS starts executing the calibration tasks. This is controlled by OpenROCS, which provides the task sequence to TALON. It points the telescope to the location of the flat screen, turns on the flats lamp when needed. Images should be stored in the proper location and pipeline reduction of all images should start as well as quality control and population of the database with the necessary information.
4. (S2) Stop calibration observations: ROCS turns off the flats lamp, closes the telescope mirror covers.
5. (S3) Start thermalization: ROCS opens the dome and opens the telescope mirror covers.
6. (S3) Start science observations: This is also controlled by OpenROCS, which provides the task sequence to TALON. Images should be stored in the proper location and pipeline reduction of all images should start, followed by quality control and population of the database with the necessary information.

7. (S2) Stop science observations: ROCS closes the telescope covers and closes the dome afterwards.
8. (S2) Start of calibration images: ROCS opens the telescope covers, follows the sequence of calibration tasks provided by OpenROCS, and points the telescope to the location of the flat screen, turns on the flats lamp when needed.
9. (S1) Stop calibration images: ROCS turns off the flats lamp, closes the telescope covers, and launches final reduction of the night. The latter produces science data and populates the database with all necessary information.
10. (S1) Stop the system: ROCS stows telescope, parks dome, stops TALON, and turns power off using the PDUs.

Annex 1 contains the steps described above in the form of a checklist in a table.

Operation with weather alarms

The plan is to simulate the appearance of weather alarms in different stages of routine operations. The test checks whether the control system reacts as specified in Sect. 5.4.4.4. The tests cover the following points:

1. Rain detected at different states (S2 and S3) and while the dome subsystem is busy (opening, closing and rotating).
2. Outdoor weather station humidity alarm at different states (S2 and S3).
3. Outdoor weather station wind alarm at different states (S2 and S3).
4. Indoor weather station humidity alarm at different states (S2 and S3).
5. Thunderstorm event inducing a sudden power cut on all systems at different states (S2 and S3).

Annex 1 contains the steps described above in the form of a checklist in a table.

Operation with system alarms

The plan is to simulate the appearance of system alarms at different stages of routine operations. The test checks whether the control system reacts as specified in Sect. 5.4.4.4. The tests cover the following points:

1. ISROCS application not responding
2. ICAT application not responding
3. Error accessing database
4. SUB-ALIS system not responding to pings
5. SUB-ESTALL system not responding to pings
6. SUB-SARGA system not responding to pings
7. SUB-REBEI system not responding to pings
8. SUB-AGER not responding to pings
9. Router not responding to pings
10. No access to Internet
11. Disk space low in SUB-ALIS
12. Power cut in UPS-CPU
13. Power cut in UPS-Dome
14. Power cut in UPS-Telescope
15. No environment sensor data
16. TALON application on SUB-ESTALL not running
17. Telescope hardware error
18. CCD hardware error
19. Dome hardware error (shutter/flap and azimuth)
20. GPS error
21. OpenROCS (SUB-ESTALL, SUB-SARGA, SUB-REBEI, SUB-ALIS) not responding.

Annex 1 contains the steps described above in the form of a checklist in a table.

Data reduction, storage and quality control

The plan is to evaluate the pipeline reduction of data, execution of quality control and storage of final products as performed by ICAT and the database system. The tests have to be conducted on the images taken during the normal robotic operations as detailed above. The tests cover the following points:

1. Keywords in header as specified in Sect. 11.3.2.2.
2. Images are moved to the correct directories in SUB-ALIS.

3. On-the-fly image reduction (both calibration and science images) is carried out and data files are produced as specified in Sect. 11.3.4.
4. Quality control files are generated compliant with Sect. 11.3.3.
5. Final reduction is carried out with resulting final calibration images and science products as specified in Sect. 11.3.4.
6. Final quality control files generated compliant with Sect. 11.3.3.
7. The data products are stored and the metadata are inserted in the database in compliance with Sect. 11.3.5 and Sect. 11.3.6.
8. User interface compliant with Sect. 12.2.3.
9. Proposal management tool compliant with Chapter 12 and Sect. 12.2.2.

Annex 1 contains the steps described above in the form of a checklist in a table.

➤ **Test conditions**

The environmental conditions can be arbitrary. The pipeline reduction, quality control and storage of the images obtained during the robotic procedure can only be tested if the test is carried out during the night. The test takes place at the TJO-OAdM facility.

➤ **Test report**

Test plan is executed and test results are reported using the test procedure template and the tables included in Annex 1. The test report includes:

- Title, issue and date of the document
- Configuration records. Results of the test for each step can have the following values:
 - OK: Test success, it is used when the system response is the expected one.
 - HF: Test half, it is used when the system response is compliant with most of the expected actions.
 - FA: Test failure, it is used when the majority of the expected responses fail.
 - NT: Not tested.
 - NI: Non-implemented, it is used when the feature to test is not yet implemented.
 - NA: Non-applicable, it is used when the test step is no longer applicable due to a change on the system configuration or on the system design.
- Procedure test deviations
- Procedure non-compliances
- Results summary
- Test conductor signature

➤ **Tests responsibility**

The tests are conducted by the engineering team, except for those included in any official review that have to be authorized by the QA manager or the project PI. The latter applied for the System Acceptance Review held in April 2010. All actions needed to correct the observed non-compliances will be corrected by the observatory engineering team.

13.4.3 System Acceptance Review (SAR)

A System Acceptance Review was held in April 2010. The objective was to validate that the overall system implementation at the TJO-OAdM telescope complied the Robotization Project requirements established by IEEC and accepted by the OAdM Advisory Committee (Colomé et al. 2006). The main parts of the SAR procedure are described below, extracted from Colomé (2010).

13.4.3.1 Review Outputs and Criteria

The criteria to be verified and met were the following:

- Performance assessed and verified against requirements.
- Compliance between requirements and design proven.
- All the subsystems design and their interface clearly defined and captured in the documentation.

As a final outcome, the Review Board had to assess and make recommendations on:

- Completion of the system performance.
- Readiness to authorize the start of the routine operation.

13.4.3.2 Review Process

The authority responsible for the execution of the TJO-OAdM SAR was the Review Board. The overall process consisted of the main steps described below.

1. The Review data package had to be delivered and distributed to the Board reviewers by the Board chairman.
2. The Review Board members reviewed the documentation.
3. A Presentation was given by the IEEC Project team to the Review Board on the status of the design, development and the key features and items included in the documentation. The Review Board addressed questions during and after the presentation.
4. The Review Board compiled a Review Report, including recommendations and actions for the Project team.

13.4.3.3 Review data package

A set of documents describing the different packages developed for the robotization project, a test plan and a test report was delivered to the SAR Review Board.

13.5 Maintenance

Maintenance tasks were identified as one of the main mitigation actions to reduce the risk at the TJO-OAdM. It is important to identify such tasks and establish a periodicity for their execution. This will determine the service frequency that will provide an indicator of the robotization level. This value has not yet been determined at the TJO-OAdM, although it is estimated to be between 2 and 6 months according to the system architecture. This value is mainly constrained by the maintenance tasks for the dome shutter motors and mechanics and for the general power supply system. They have proven to have long-term stability since their installation, but a less frequent service is not recommended by the manufacturer.

Maintenance tasks involve both hardware and software components, although the software layer is less prone to instability or to other issues that require periodic checking.

13.5.1 Maintenance plan

The definition of a maintenance plan has been identified as a critical task to complete the robotization of the observatory. A maintenance plan is currently under development including different parts summarized in this section.

13.5.1.1 Hardware maintenance

The maintenance tasks for the hardware components are grouped in the same manner as in the physical architecture description given in Chapter 4. Different aspects are taken into consideration when defining the priority to check each one of the components:

- **Risk score** (see a definition in Sect. 13.2), in case they constitute a risk of damaging the observatory or producing any harm to a person.
- **Impact on the system efficiency** running on nominal or non-routine mode in case of failure.
- **Reliability**, unreliable systems require a more frequent supervision.
- **Predefined consumption** (i.e., diesel storage capacity for the power generator).

The periodicity and the tasks for the maintenance of the COTS hardware components are initially based on the manufacturer specifications. The adopted processes could vary due to the harsh environment conditions at the observatory and will be tuned in the future to adjust them to the experience gained by the engineering team members.

The engineering team is in charge of defining the periodicity and tasks for the hardware components developed in-house.

13.5.1.2 Software maintenance

Software maintenance will only affect security packages. Subsystems are grouped into two main blocks according to the external access and their vulnerability to external attacks:

- SUB-ALIS, SUB-REBEI, SUB-AGER and the router are accessible from internet and require a higher level of security. A higher frequency of the system update is required for these systems.

- SUB-ESTALL and SUB-SARGA, on the other side, are only visible from the LAN. They are extremely sensitive to changes in the OS due to their interfaces with hardware components. A Long Term Support OS and a low update frequency are preferred in these cases.

13.5.1.3 Human safety and security

Although reducing the system failure is the main goal of the maintenance plan, the security protocols to carry out this maintenance or other engineering tasks are also included to reduce the risk of producing any harm to a person while performing them.

13.6 Configuration management

A configuration item may be any kind of software or hardware item, a document or a set of them. A unique identification and some level of control for their development are important as a QA issue for the project. Configuration management (CM) is devoted to these processes. An exhaustive management of the configuration items at the TJO-OAdM has not been carried out due to the small dimension of the project, but some important practices have been adopted to establish and maintain the system consistency. These are described in this section.

13.6.1 Configuration items

Some components of the observatory infrastructure are kept under configuration item management:

- Hardware components (at instrument level)
- Software modules
- Documents

Their attributes are identified at various points in time, and a systematic control of changes is performed on them with the purpose of maintaining integrity and traceability throughout the development life cycle.

Some of the attributes that are controlled are: physical location, functional attributes, and links among items.

Hardware and software items are assigned a name representative of the activities they perform. Documents are uniquely identified with a reference code and versioning issue. New code numbers or item names are always supervised by the PM, who is in charge of providing a new reference or name, by ensuring that no other item has the same assigned value, and to supervise the configuration control.

13.6.2 The project repository

Changes on software and documents are tracked using software tools for repository management.

GIT is the chosen tool for software development. The software code is self-documented and any change can be traced in the same way.

Document versioning is based on MSWord document change control, although new documentation is being produced with LaTeX editors, whose encoding can be also put under versioning control. GIT is used for such purpose aswell.

The project repository is stored on the IEEC main servers under restricted access policies. They include a strict backup policy, avoiding the possibility to lose the information stored.

13.6.2.1 Software directory tree

The software repository is structured as follows:

- ROCS ICAT: Data processing software, including the GUI.
- ROCS OpenROCS: Control system software.
- ROCS ISROCS: Scheduler software.
- ROCS TALON: Telescope, dome and instrument control software.
- ROCS Web: Website and proposal management tool software.

These directories contain the software developed. A different group of directories is used to store the configuration particular for the TJO-OAdM observatory. This structure is used to develop the applications in such a way they can be tested or implemented using different set-up configurations. This structure is used, i.e., to develop in parallel the software for the SQFT project and to test it at the

IEEC laboratories. Installing the software from this repository enables an easy and fast replication of the system.

13.6.2.2 Documentation management

All project-related documents are available (with login and password) at the robotization project web page hosted at the IEEC servers (https://trac.ice.csic.es/rocs_tjo). They are stored as PDF files. These documents include:

- Project meetings related files: presentations and minutes.
- All documents generated by the observatory work team (engineers and scientific staff).

Paper originals of all robotization project documents under configuration control are stored at least at IEEC.

13.6.3 Bug tracking

A tool to track the bug-fixing activities was found necessary since the very beginning of the robotization project. It was also required to act as a memory of the project, in order to trace at any time the motivations that led to every specific decision in the project development. This practice was initiated as a way to correct the lack of procedures that help following the project life cycle in a formal and documented way.

Its use has been maintained since the current days, although it is now used only as a bug-tracking system.

13.6.4 TRAC, media control

The activities involved in CM processes are managed using TRAC. TRAC¹ is a web-based software project management and bug/issue tracking system emphasising ease of use. It provides an integrated Wiki, an interface to version control systems, and a number of convenient ways to monitor events and changes within a project. TRAC is distributed under the modified Berkeley Software Distribution (BSD) License.

This tool is used as a bug-tracking system, as a software repository access, and as document management system.

¹ <http://trac.edgewall.org/>

Chapter 14

Conclusions

14.1 The TJO-OAdM in the context of the robotic telescopes

The proliferation of robotic telescopes at astronomical observatories has been extending since the 90's, when the first telescopes running under fully automatic operation appeared (Castro-Tirado 2010). They proved to increase the efficiency of the acquisition of science data using new technological solutions for the execution of actions with no human interaction. Initial experiences (i.e., Fairborn/TSU telescope – Henry 1995, BRT – Baruch 2007) were rapidly exported to other observatories, mostly to those devoted to restricted science cases, i.e., survey observations, follow-up or detection of transient phenomena, etc.

We can distinguish different levels of unattended control among the projects developed so far. We can group them according to the operations they perform without human interaction and using two different criteria: the completeness of the unattended operation for the data flow execution and housekeeping control and the level of dynamic response to changing conditions.

Regarding the number of processes executed with no human interaction, robotic telescopes have historically been focused mainly on automatizing the execution of tasks and the processes carried out for the control of housekeeping (i.e., monitor the environment and ensure the system safety). However, the efficiency of an astronomical facility can be much increased with its complete automatization. This includes the scheduling of tasks, their execution and, finally, processing the acquired data. This is especially important for multipurpose and multiuser observatories that handle different types of programs and require a more intelligent management.

The level of the dynamic response of a system to the changing conditions is defined by the type of feedback the system has for any of the running processes and how this feedback is used to automatically react with corrective or reactive actions. The static (or open-loop) observatories are those that do not react to changing conditions except for those actions executed to minimize the risk under bad weather (i.e., dome closure when it rains). Dynamic observatories (or closed-loop), on the other hand, use the data from the system health monitoring or the quality of the acquired data to run contingency actions to restore any malfunctioning component or to modify the schedule. Dynamic facilities are prepared to react efficiently according to the events (routine and non-routine) providing a high level of feedback.

The majority of the robotic telescopes are designed to perform a simple data flow control, mainly devoted to the execution of tasks, and are based on the static control approach. Their autonomy is quite low, but they are a good solution as they represent a simpler development.

A small number of robotic telescopes use smart (or dynamic) schedulers, on-the-fly processing routines and/or they are operated dynamically in order to achieve a high level of autonomy and efficiency. Some examples are: RAPTOR (Vestrand et al. 2002), BRT (Baruch 2007), STELLA (Granzer et al. 2010) or the LT (Steele et al. 2004). They have proved to substantially increase the system efficiency in terms of data acquisition and scientific return.

➤ *The TJO-OAdM case*

The TJO-OAdM telescope was conceived as a multipurpose telescope operated robotically, according to the robotic telescope definition given in Baruch (1993). A highly autonomous, efficient and robust control of the full facility was envisioned, mainly motivated by the site isolation, the harsh environment conditions and the generalist science case. These characteristics were hard constraints for the global system development, initially designed with a simpler approach. They impose important requirements that are not necessary for other kinds of projects. Concerning the site isolation, for instance, some robotic telescopes are installed on sites with neighbouring attended telescopes and the control system has some human interaction that is used to take critical decisions (i.e., check the dome shutter position under bad weather condition). The TJO-OAdM cannot receive such input.

The TJO-OAdM was developed based on a fully unattended and dynamic operation to maximize the system autonomy and the scientific return. It is provided with an extended management of the events and alarms produced during routine and non-routine operations and a completely unattended data-flow control. In addition, high feedback from the system enables a continuous evaluation of its performance and the execution of corrective actions at hardware and software levels. This gives the possibility, for instance, to reschedule a task if the quality of the acquired data does not fulfil the requirements.

The TJO-OAdM could be integrated in the future into a network of robotic telescopes thanks to its modular and flexible design. Some initiatives of this kind appeared a few years ago and are currently under development: RoboNet-II (Tsapras et al. 2008), in collaboration with Las Cumbres Observatory Global Telescope Network (LCOGT, Pickles et al. 2010), or the consortium for the Heterogeneous Telescope Network (HTN, White 2007). The major benefit of participating in such networks would be the possibility to exchange time with other observatories for optimal longitude coverage and ensure the execution of very high-priority observations.

14.2 Contributions of the Thesis to the TJO-OAdM robotization

We have contributed with this Thesis to develop the TJO-OAdM telescope as a facility running on qualified robotic mode. We added the *qualified* term to highlight the effort made to achieve a robust, reliable, efficient unattended operation, but also acquiring high quality data. These features are important for this telescope due to the aforementioned conditions (site, environment and science case) and give some singularity to this project. The achievement of these features was based on a few key elements, described in the following subsections.

14.2.1 Robotic concept

14.2.1.1 Operational design and requirements

We made an exhaustive evaluation of the requirements that the system has to fulfill to build a suitable design for the telescope operation, as described in Chapter 2. Some of these requirements were specified to adapt the original setup, inherited from the pre-robotization project, to the new one presented in this Thesis and maintaining the performance of the facility in terms of the science case goal.

We translated the system requirements into a global design of the telescope operation and its architecture, as summarized in Chapter 3. In this chapter, the robotic concept applied to the TJO-OAdM was established.

The operational design defines the unattended processes that the system executes in routine and non-routine conditions, the kind of tasks the system performs, and the users with system interaction privileges. We were aware that the development of a system fulfilling the project requirements would be a consequence of having a well-defined operational model. We made this exercise at the time the robotization project was defined. This has contributed to easing the validation and verification processes and to achieving the desired quality.

14.2.1.2 System architecture

The operational design was the basis to define the system architecture. The general design was structured into two main blocks:

- **Operational Hardware:** This layer is composed by instrumentation selected during the previous phase (pre-robotization) and mostly linked to the science case of the observatory. The telescope, dome and instrument are the main important ones. Other equipment was selected (COTS) or developed in-house during the robotization phase to provide the required

performance. We chose or designed most of the devices included in this second group following robustness and reliability criteria.

- **Robotic Observatory Control System (ROCS):** It is a software suite composed by two layers that execute the tasks involved in the end-to-end data flow and in the housekeeping control. This block has a modular structure and is built with some applications (ISROCS, ICAT), tools (DDROCS, Proposal management) and a software infrastructure that manage the processes used to achieve the desired operation (OpenROCS).

14.2.2 Operational hardware

The observatory system architecture evolved from the original design up to the final implemented version described in this Thesis. We are confident to have a stable and robust system running at low risk and providing nearly the best quality data that can be obtained with the facilities at the Montsec site.

We presented in Chapter 4 the modifications made to the original scientific hardware: telescope, instrument and dome. These changes were made to improve their performance and to increase the quality of the data, but also the system's reliability. Concerning the optical quality, major changes were made to the telescope stray light control. The changes to the electronics and low level software improved the telescope tracking and added functionalities, for instance, to control the mirror covers and the dome. The instrument CCD camera and the dome electronics were also modified to increase their reliability. An outstanding increment on their stability was achieved. Finally, a new system to calibrate the instrument detector was designed and installed.

We also presented the new devices added. Important actions were taken to reduce the occurrence likelihood or impact of scenarios of high risk. More protections against direct lightning strikes, a high level of electric insulation, a robust power supply system and redundancies of critical subsystems (dome shutter, environment sensors) were designed in the framework of this Thesis. The development and integration processes were done in-house by the engineering team or in collaboration with external companies. Other actions were focused on increasing the level of control and the feedback in real time that is necessary to detect the source of any malfunction as fast as possible. The majority of the new hardware components were COTS of high robustness.

14.2.3 Robotic Observatory Control System

We designed the ROCS as a software suite that handles the workflow control of a robotic astronomical observatory with minimum human interaction and maximum scientific return. A detailed list of the possible states of the observatory, the tasks the ROCS has to handle for data flow and housekeeping control and the interfaces between the different modules were specified in Chapter 5, configuring the high level design.

14.2.3.1 OpenROCS

OpenROCS is a master scheduler of the processes that the system has to execute and a monitor of the overall state of the observatory. It controls the end-to-end data flow and the housekeeping processes. The former are responsible for ensuring that the system executes the routine operation by maximizing the scientific return of the facility. While the latter consider all those anomalous situations that could put the observatory at risk and activate corrective or mitigation actions.

The OpenROCS design was based on the contents of Chapter 5, especially on the exhaustive analysis included there, where the data transferred to or by OpenROCS and the commands it executes to act as the main control software are specified.

Two versions of this software were implemented by the engineering team. The first version was based on the Ice middleware and it was highly monolithic and structured. The second one was programmed with scripting languages and was based on a more flexible structure, transferring the configuration of the control scheme to XML files. The software upgrade was motivated by the need for an easy modification and definition of the observatory configuration, providing the capability to be used at other observatories.

14.2.3.2 DDROCS and proposal management tools

External users interact with the observatory mainly through the proposal management tool described in Chapter 12. This is simply a web interface with the observatory database to let the user submit a proposal in two phases and in a clear and complete fashion. However, there is a complex design beyond this GUI and the database that is strongly related with the end-to-end data flow concept.

We described the end-to-end data flow at the TJO-OAdM in Chapter 3 as a complete process to handle the user proposal from its submission to the retrieval of data. All the steps between these two manual actions are done automatically by the ROCS. Therefore, it is very important to have a database to store the tasks associated to the proposal in a way that can be read by the control system for the automatic night planning. In addition, the acquired images and data have to be easily cross-correlated with these tasks to let the system check for the final completion of the proposal and easy retrieval by the user. This data flow design has two major consequences: there has to be a clear specification of the scheduling constraints that the user has to provide to the observatory, and there has to be a database to handle the proposal data in a consistent way. We designed the database (DDROCS) and the task preparation process to optimize this flux of data. They are described in Chapter 7 and Chapter 8, respectively.

The database also stores other data (housekeeping data, logs, etc.) that can be used for the supervision of the observatory.

14.2.3.3 ISROCS

ISROCS is the application in charge of the task scheduling process for the planning of both science observations and calibrations tasks. It is described in Chapter 9.

We designed the scheduler (ISROCS) in order to provide the observatory with static and dynamic task planning. Static scheduling provides the sorted list of tasks to execute in the following night (mid-term) or in a time period of several nights (long-term). It computes this list by only considering the execution constraints depending on static conditions that can be predicted a priori. Short-term scheduling, on the other hand, is given by the dynamic component that adds the execution constraints depending on system and environment conditions that can change in real time.

Both scheduling levels are computed using the dispatch algorithm as described by Granzer (2004). Dispatch scheduling is based on the FOM and was chosen as it provides a fast and effective approach for dynamic prioritization of tasks based on instantaneous conditions. We propose in this Thesis a modified dispatch scheduling solution to improve the short-term planning and to provide also the mid- and long-term perspective. Some contributions to the FOM specific for the TJO-OAdM telescope are also added.

The scheduler architecture was designed to integrate its execution into the process management done by OpenROCS for the overall control of the observatory.

The new (or modified) dispatch scheduler routine is under development. In order to start operating with the fully unattended control, we designed and developed a simple queue scheduler to compute the planning before the beginning of the night. This queue scheduler will be discarded once the new one is finished.

14.2.3.4 ICAT

ICAT's main goal is to contribute to the end-to-end data flow control layer of the TJO-OAdM with the extraction of high quality and high precision astronomical data. With such purpose, the image calibration and reduction procedure sequences are defined, together with a quality control routine and other complementary packages that handle data storage and data archiving. It is described in Chapter 11.

Some external applications are used to perform the image reduction and analysis. IRAF, DAOPHOT, SExtractor and CFITSIO libraries were chosen for different purposes. A trade-off was made to select the analysis package to use in order to extract photometric and astrometric data of the best possible quality.

ICAT plays an important role in maximizing the system efficiency as it is responsible of providing feedback to the control system on the quality of the acquired images.

14.2.3.5 Ancillary tools for remote supervision

Graphical interfaces (Chapter 12) were developed and two webcams were installed to enable remote monitoring of the observatory. These ancillary tools were planned for the external supervision made by the observatory staff (superusers) or to ease the maintenance and testing tasks done by the engineering team on site.

14.2.4 Quality Assurance

The system robustness and the quality of the overall operations performed by the control system are especially critical for the robotic operation of an astronomical observatory. These were specified as

high-level requirements for the development of the TJO-OAdM. Systems engineering practices, including QA processes, were applied mildly to ensure that the implemented project meets the requirements.

We defined and applied several QA practices (Chapter 13) during all the project phases: risk management ensuring the project risks are identified and covered by actions to eliminate or minimize them; verification control confirming that the specified requirements have been fulfilled and that the implemented solutions perform correctly; testing activities that are designed and executed for the verification control; and configuration management that is focused on the activities done to handle all types of documents, hardware devices, software code and test procedures.

A test procedure was designed as an important QA ingredient to demonstrate that each of the observatory subsystems performs as defined by its requirements and that the entire system operates in a robust fashion for an extended period of time. The test procedure describes the test campaign necessary to validate the robotic operation of the TJO-OAdM through the testing of the individual subsystems and the simulation of different scenarios.

The operations run at the observatory since the end of the robotization project (end 2010) were focused on testing the control system. The tests showed a good performance of the robotic operation. However, these tests enabled the detection of some shortcomings in the optics, mechanics, low level software and electronics of the TJO telescope and an important fraction of the time was invested in solving these issues in the last months. A small number of tests could be finally carried out.

The quality of the obtained data is now at their highest level and we are currently executing the test procedure in a systematic way to obtain enough information to evaluate the control system performance.

The efficiency of the system is still very low when comparing the time invested to perform science observations with the night time available. It is a consequence of the robotic supervised mode of operation used at the moment that requires personnel being at the observatory when executing tasks. This reduces the available time for operations onsite. The lack of efficiency is also caused by the execution of the test procedure that implies stopping the acquisition of science data. Once this testing process verifies the correct system response to the critical events and the unattended operations are carried out in a routine way, the efficiency should be increased at the high expected level.

14.3 On-going and future work

The overall design of the different modules that compose the TJO-OAdM telescope, the physical architecture implementation and most of the software packages and tools are already developed and running at the observatory.

The ongoing work at the TJO-OAdM is mainly focused on testing the robotic control system, repeating the astronomical commissioning of the telescope and detector, improving the software modules already implemented and adding new features that have already been designed, but not implemented. Future work is focused on the installation of new instrumentation. All this work is done by or in collaboration with the engineering team at the IEEC.

➤ **Tests**

The test procedure is executed randomly every time the observatory staff is at the observatory to run automatic observations under supervised mode. This kind of operation is foreseen for a maximum duration of six months. Robotic operations will start after this period.

Astronomical data are collected by executing specific proposals in order to repeat the commissioning analysis and evaluate the improvement of the telescope and MEIA instrument.

➤ **Improvement of software modules**

Several software modules are being refurbished to fix some bugs or to improve their implementation. The main goal is simplifying the maintenance by increasing the modularity and homogeneity of languages, and also increasing their performance. The proposal management tool and the database, for instance, are being modified to have a more flexible and complete implementation. Some effort is made to produce packages that can be easily migrated to other observatories. Such policy specially applies to OpenROCS, and ICAT. TALON is another software module that is being debugged to provide a better communication with the control system.

➤ **New features**

One of the main modules under development is the new ISROCS application for task scheduling that incorporates the dynamic response and improves the task selection process.

Another important task is the integration of the environment data collected by the TFRM weather station. This new dataset will provide the TJO-OAdM with redundancy on rain detection signal and, thus, an important SPF will be removed.

➤ **New instrumentation**

New instrumentation will be installed in the near future:

- **ARES:** A spectrograph is already installed at the observatory, but has to be integrated into the overall control system. A specific room was built in the ground floor of the TJO-OAdM building to host the instrument. The proposal management tool, the scheduler and the control system have to be modified, and a new data processing procedure is necessary to evaluate the acquired spectra.
- **Autoguiding and front-end:** The installation of ARES will have to be done in parallel of a new front-end that enables the automatic selection of the instrument and positioning the fiber for the spectrograph at the correct position. An autoguiding camera will be also installed to provide a closed-loop system that increases the guiding performance. The autoguiding capability has already been added to TALON and to the firmware of the electronic boards (CSIMC) that control the telescope motors.
- **RoboDIMM:** A robotic monitor to measure the seeing was acquired, but it does not have a dedicated enclosure. It is used only if it is installed manually outside the building during the night. It will have to be integrated into the observatory control system once it can be operated automatically. It will provide the seeing status in real time.

14.4 Gained expertise and new projects

The TJO-OAdM was developed as a small-class telescope that could be used by the astronomical community to carry out scientific programs. This kind of telescopes is very useful to address the time domain that is important for many scientific studies and that cannot be carried out with large facilities. Robotic operation provides a higher efficiency and the possibility to include this telescope on world-wide networks.

A second, but not less important motivation was to use it as a test-bench for the development of astronomical instrumentation. Learning the technology related to robotic telescopes and systems engineering processes, doing research on open issues and developing new solutions were some of the main goals. These objectives have led to the development of some control modules based on more stringent specifications than those required for this telescope. And, therefore, it may have represented an excessive effort in terms of manpower and scientific return of this facility. The dynamic scheduler is an example: the dynamic component of the scheduler represents a small optimization of the observing time when comparing the operation cost of the telescope with the cost of developing such a complex tool. However, the expertise we have gained during this research process opens the possibility of applying this know-how to other projects where it may be necessary. We provide below some examples of projects that are benefitting of our expertise.

➤ **SuperWASP Qatar Follow-up Telescope (SQFT)**

SQFT is an international project for the construction of a 1m robotic telescope equipped with a two-arm instrument and devoted to exoplanet characterisation. It was installed at the ORM observatory (Canary Islands) in early 2011 and it has a very similar architecture to the TJO-OAdM.

IIEC is participating on the low level software and the electronics of the telescope, manufactured by the same company as the TJO; it contributes with the development of the system architecture and the instrument; and it is providing the control system that manages the operations of the complete system that is based on OpenROCS. The specifications of this telescope are similar to those of the Montsec telescope and only a few modifications to the technology used at the TJO-OAdM were required.

We received an IIEC grant from an internal competitive call for projects to fund this work.

➤ **Calar Alto high-Resolution search for M dwarfs with Exoearths with Near-infrared and optical Échelle Spectrographs (CARMENES)**

This spectrograph (Quirrenbach et al. 2010) is a next-generation instrument to be built for the 3.5m telescope at the Calar Alto Observatory (Spain) by a consortium of German and Spanish institutions. It

is focused on searching for exoplanets around M dwarf stars using two channels for the simultaneous acquisition of near-infrared and optical spectra. It recently passed the PDR evaluation and the first light of the instrument is planned for early 2014.

IEEC is leading the development of the Instrument Control System, the instrument scheduler and the data quality control thanks to the experience gained at the TJO-OAdM. These three packages are analogous to the OpenROCS and the ISROCS applications and the quality control routine included within ICAT, but with more stringent specifications. A higher efficiency of CARMENES will represent a strong increase of the scientific return and, thus, a cost optimization. This can be a direct consequence of the performance of the three packages we are responsible for. Our participation is funded by the Spanish Science and Innovation Ministry.

➤ ***Cherenkov Telescope Array (CTA)***

The CTA project (CTA Consortium, 2010) is an international initiative to build the next generation ground-based very high energy gamma-ray instrument. CTA will consist of two arrays of Cherenkov telescopes composed of small-, medium-, and large-sized dishes that will make it possible to tackle different scientific problems with the same facility. More than 60 telescopes will be installed in the Northern and in the Southern sites. The project is now executing the Preparatory Phase that will last until the end of 2013, when the first operations are expected.

With its large number of individual telescopes, CTA will be prepared to be operated in a wide range of configurations, allowing, on the one hand, the in-depth study of individual objects with unprecedented sensitivity, and, on the other hand, the simultaneous monitoring of tens of potentially flaring objects, and any combination in between. This flexibility will be achieved with predefined subsets of the full array.

IEEC is participating in the work package that has to provide the array control software. We are in charge of developing the scheduler, which plays an important role in providing the large flexibility required for the global operation. The efficiency of such large facility is strongly dependent on the optimum planning of tasks, taking into account the state of the array, the state of the atmosphere and the configuration in sub-arrays. Our participation is funded by the Centre for Industrial Technological Development (CDTI), entity of the Spanish Science and Innovation Ministry. The work is carried out in collaboration with a private company (GTD S.I.).

➤ ***International Concordia Explorer (ICE-T)***

ICE-T (Strassmeier 2007) is a robotic binocular telescope for continuous wide-field high-precision photometry from the Antarctic plateau (Concordia Base, Dome C). The primary objective of ICE-T is to investigate the combined effects of exoplanets, stellar magnetic activity and non-radial pulsations on the structure and evolution of stars, as well as their internal dynamics and dynamo activity. Continuous observing at very high photometric precision is needed to detect planets and to characterize the evolution of stellar surface structures. ICE-T is currently in an early design phase for the telescope and the infrastructure.

We are in charge of the Data Processing work package (WP 7000), whose main goal is to obtain high quality and high precision photometry. A similar approach to the ICAT application will be used, although the requirements for the photometric precision and the configuration of the two detectors (they are based on 10k×10k CCD chips) will impose quite different processing options.

References

- ACEITUNO, J., 2004. "A new Robotic DIMM for Calar Alto Observatory", CAHA Newsletter 04b
- ALLAN, A., NAYLOR, T., STEELE, I., CARTER, D., JENNESS, T., ECONOMOU, F., AND ADAMSON, A., 2004. "eSTAR: Astronomers, Agents and when Robotic Telescopes aren't...", ASP Conf. Ser., v.314
Astronomical Data Analysis Software and Systems XIII
- BARTH, W., 2006. [Nagios: System and Network Monitoring], No Starch Press
- BARUCH, J. E. F., 1993. "Robots in Astronomy", *Vistas in Astronomy* 35, pp.399-438
- BARUCH, J. E. F., HEDGES, D. G., MACHELL, J., AND TALLON, C. J., 2004. "Robotic telescopes and the public", AN, v.325, pp.457-461, DOI: 10.1002/asna.200410264
- BARUCH, J. E. F., 2007. "Bradford robotic telescope: education & public outreach: journey into space", *Astronomy & Geophysics*, v.48, n.4, p. 4.27-4.28
- BERTIN, E., 1998. "SExtractor v2.5. User's guide", <http://terapix.iap.fr/IMG/pdf/seextractor.pdf>
- BERTIN, E., AND ARNOUITS, S., 1996. "SExtractor: software for source extraction", *A&A Supp.Ser.*, v.117, n.2, pp.393-404
- BISCHOFF, K., TUPAREV, G., HESSMAN, F. V., AND NIKOLOVA, I., 2006. "MONET/North: a very fast 1.2m robotic telescope", *Proc. SPIE* 6270, 62701Q, DOI:10.1117/12.671433
- CASTEELS, K., AND FERNÁNDEZ, D., 2010. "Site testing and commissioning of the TJO", Technical Note, IEEC, IEEC-OAdM-RP-0008, v1.0
- CASTRO-TIRADO, A. J., 2010. "Robotic Autonomous Observatories: an historical perspective", *Advances in Astronomy*, article id. 570489
- CASTRO-TIRADO, A. J., ET AL. 2012. "The BOOTES World-Wide Network of Robotic Telescopes". *Proceedings of the II Workshop on Robotic Autonomous Observatories*, *Bull. Astron. Soc. of India*, in press
- CHIOZZI, G., GUSTAFSSON, B., JERAM, B., PLESKO, M., SEKORANJA, M., TKACIK, G., AND ZAGAR, K., 2002. "CORBA-based Common Software for the ALMA project", *Proc. SPIE* 4848, 43, DOI: 10.1117/12.461036
- CLEAR SKY INSTITUTE, 2000. "Observatory Control and Astronomical Analysis System. Reference manual for OCAAS version 2.0", software manual
- COLOMÉ, J., CASTANDER, F. J., ISERN, J., RIBAS, I., FERNÁNDEZ, D., TORRA, J., FIGUERAS, F., GARCÍA-BERRO, E., AND LURI, X., 2003. "Disseny del cablejat Observatori Astronòmic del Montsec", Technical Note, IEEC, OAdM-CAB-000
- COLOMÉ, J., FERNÁNDEZ, D., AND RIBAS, I., 2006. "Estudi de les necessitats pendents per a la robotització de l'Observatori Astronòmic del Montsec (OAdM)", Technical Note, IEEC, IEEC-TN-2006-002, v1.0
- COLOMÉ, J., 2007. "CCD-1 Shutter Map", Technical Note, IEEC, IEEC-OAdM-RP-0001, v1.0
- COLOMÉ, J., 2010. "Requirement Specifications for the Dome Communication and Control Electronics", Technical Note, IEEC, IEEC-OAdM-RS-0004, v2.0
- COLOMÉ, J., 2010. "Procedure for the TJO-OAdM Robotization Project System Acceptance Review", Technical Note, IEEC, IEEC-OAdM-PR-0001, v1.0
- CTA CONSORTIUM, 2010. "Design Concepts for the Cherenkov Telescope Array – CTA", arXiv 1008.3703

- FRANCISCO, X., AND COLOMÉ, J., 2009. "Environment Control System", Technical Note, IEEC, IEEC-OAdM-DS-0001, v1.1
- FRANCISCO, X, 2010A. "OpenROCS General Design", Technical Note, IEEC, IEEC-OAdM-DS-0006, v0.1
- FRANCISCO, X, 2010B. "OpenROCS Implementation Details", Technical Note, IEEC, IEEC-OAdM-RP-0024, v1.0
- FRASER, S.N., AND STEELE, I. A., 2004. "Robotic telescope scheduling: The Liverpool Telescope experience", *Proc. SPIE*, v.5493, p.331-340
- GAREY, M. R., AND JOHNSON, D. S., 1979. "Computers and Intractability: A Guide to the Theory of NP-Completeness", ISBN: 0-7167-1045-5
- GÓMEZ DE CASTRO, A.I., AND YÁÑEZ, J., 2003. "Optimization of telescope scheduling: Algorithmic research and scientific policy", *A&A*, v.403, p.357-367
- GRANZER, T., 2004. "What makes and automated telescope robotic?", *AN*, v.325, n.6-8, p.513-518
- GRANZER, T., WEBER, M., AND STRASSMEIER, K. G., 2010. "Three Years of Experience with the STELLA Robotic Observatory", *Advances in Astronomy*, vol. 2010, Article ID 980182, 9 pages. DOI: 10.1155/2010/980182.
- GTD AND IEEC, 2011. "Evaluation of scheduler algorithms for CTA", in preparation.
- HENNING, M., AND SPRUIELL M., "Distributed Programming with Ice", <http://www.zeroc.com/Ice-Manual.pdf>
- HENRY, G.W., 1995. "The Fairborn/TSU Robotic Telescope Operations Model," in *Robotic Telescopes: Current Capabilities, Present Developments, and Future Prospects for Automated Astronomy*, ASP Conf. Ser. v.79, eds. G. W. Henry and J. A. Eaton (Provo: Astronomical Society of the Pacific), p. 37, 1995
- HERRERO, E., MORALES, J.C., RIBAS, I., NAVES, R., 2011. "WASP-33: the first delta Scuti exoplanet host star.", *A&A*, v.526, L10, DOI: 10.1051/0004-6361/201015875
- HOLMES, S., KOLB, U., HASWELL, C.A., BURWITZ, V., LUCAS, R.J., RODRIGUEZ, J., ROLFE, S.M., ROSTRON, J., AND BARKER, J., 2011. "PIRATE: A Remotely-Operable Telescope Facility for Research and Education", accepted for publication in *PASP*
- ISERN, J., PALAU, X., AND TORRA, J. (EDITORS), 2000. "Observatori astronòmic del Montsec i Centre d'Observació de l'Univers", Project submitted to the Generalitat de Catalunya
- JOHNSTON, M. D., 1990. "Spike: AI scheduling for NASA's Hubble Space Telescope," *Sixth Conference on Artificial Intelligence Applications*, v.1, p.184-190
- JOHNSTON, M. D., 1991, NASA, Goddard Space Flight Center, *Space Network Control Conference on Resource Allocation Concepts and Approaches*, 183
- MARCHANT, J. M., SMITH, R. J., AND STEELE, I. A., 2008. "Calibration of the Boltwood Cloud Sensor ", *Proc. SPIE* 7012, 133
- MINK, D.J., 1996. "WCSTools: Putting Image World Coordinate Systems to Use", *Sixth Annual Conference on Astronomical Data Analysis Software and Systems (ADASS 96)*, published in *Astronomical Data Analysis Software and Systems VI*, A.S.P. Conference Series, v.125, 1997, Gareth Hunt and H. E. Payne, eds., p.249
- MORA, M., AND SOLAR, M., 2010, "A survey on the dynamic scheduling problem in astronomical observations", M. Bramer (Ed.), *IFIP AI 2010, IFIP AICT 331*, pp.111-120
- PENCE, W.P., SEAMAN, R., AND WHITE, R.L., 2009. "Lossless astronomical image compression and the effects of noise", *PASP*, v.121, no.878, pp.414-427
- PENCE, W.D., CHIAPPETTI, L., PAGE, C.G., SHAW, R.A. AND STOBIE, E., 2010. "Definition of the Flexible Image Transport System (FITS), version 3.0", *A&A* v.524, A42, DOI: 10.1051/0004-6361/201015362
- PICKLES, A., ROSING, W. ET AL., 2010. "LCOGT Telescope network capabilities", *Proc. SPIE* 7733, DOI: 10.1117/12.857923
- PINEDO, M., 1995. "Scheduling: Theory, Algorithms, and Systems", Prentice Hall, Englewood Cliffs (NJ); ISBN: 0137067577
- QUIRRENBACH ET AL., 2010. "CARMENES: Calar Alto high-Resolution search for M dwarfs with Exo-earths with Near-infrared and optical Echelle Spectrographs", *Proc. SPIE* 7735, DOI: 10.1117/12.857777

- RIBAS, I., 2010. "Test procedure for the ROCS-SAR", Technical Note, IEEC, IEEC-OAdM-TP-0003, v1.0
- SANCHEZ, L., AND SANZ, J., 2011 (EXPECTED). "Modifications on TALON", Technical Note, IEEC, in preparation
- SANZ, J., 2011. "OpenROCS v2.0", Technical Note, IEEC, IEEC-OAdM-RP-0034, v1.0
- SÁNCHEZ, S. F., ACEITUNO, J., THIELE, U., PÉREZ-RAMÍREZ, D., AND ALVES, J., 2007. "The Night Sky at the Calar Alto Observatory", PASP, v.119, No.860, pp.1186-1200, DOI: 10.1086/522378
- SCHECHTER, P. L., MATEO, M., AND SAHA, A., 1993. "DOPHOT, a CCD photometry program: description and tests", PASP, v.105, p.1342
- SMITH, R. J., STEELE, I. A., MARCHANT, J. M., FRASER, S. N., AND MUCKE-HERZBERG, D., 2004. "Remote secure observing for the Faulkes Telescopes", Proc. SPIE 5496, 687; DOI:10.1117/12.551488.
- STEELE, I. A., SMITH, R. J., REES, P. C., BAKER, I. P., BATES, S. D., BODE, M. F., BOWMAN, M. K., CARTER, D., ETHERTON, J., FORD, M. J., FRASER, S. N., GOMBOC, A., LETT, R. D. J., MANSFIELD, A. G., MARCHANT, J. M., MEDRANO-CERDA, G. A., MOTTRAM, C. J., RABACK, D., SCOTT, A. B., TOMLINSON, M. D., AND ZAMANOV, R., 2004. "The Liverpool Telescope: performance and first results", Proc. SPIE 5489, 679-692
- STETSON, P.B., 1987. "DAOPHOT – A computer program for crowded-field stellar photometry", PASP v.99, p.191
- STRASSMEIER, K. G., AND HESSMAN, F. V. (EDS.), 2004. [Third Potsdam Think Shop on Robotic Astronomy], AN 325, 455
- STRASSMEIER, K. G., 2007. "Ultra-High Precision, Ultra-Wide-Field Optical Photometry", in Epchtein, N., M. Candidi & M. Swain (eds.), Large Astronomical Infrastructures at Concordia, prospects and constraints for Antarctic optical/IR Astronomy, EAS Publications Series, v.25, p.233-238, DOI: 10.1051/eas:2007100
- SWANSON, K., BRESINA, J., AND DRUMMOND, M., 1994. "Robust Telescope Scheduling", Proc. Third International Symposium on Artificial Intelligence, Robotics, and Automation for Space, JPL, 347
- TSAPRAS, Y., STREET, R., HORNE, K., SNODGRASS, C., DOMINIK, M., ALLAN, A., STEELE, I., ET AL., 2008. "RoboNet-II: Follow-up observations of microlensing events with a robotic network of telescopes", AN, v.330(1), 8
- VERDERAME, P.M. ET AL. 2010. "Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors", Ind.Eng.Chem.Res., vol.49, p.3993-4017
- VESTRAND, W. T., BOROZDIN, K. N., BRUMBY, S. P., ET AL., 2002. "The RAPTOR experiment: a system for monitoring the optical sky in real time," Proc. SPIE, 4845, p.126-136, DOI: 10.1117/12.459515
- WELLS, D.C., GREISEN, E.W., AND HARTEN, R.H., 1981. "FITS - a Flexible Image Transport System", A&A Supp.Ser. v.44, p.363
- WHITE, R.R., AND ALLAN, A., 2007. "An overview of the heterogeneous telescope network system: Concept, scalability and operation", AN, v.329, n.3, p.232-236
- WHITE, R. R., ALLAN, A., BARTHELMY, S., BLOOM, J., GRAHAM, M., HESSMAN, F. V., MARKA, S., ROTS, A., SCHOLBERG, K., SEAMAN, R., STOUGHTON, C., VESTRAND, W. T., WILLIAMS, R., AND WOZNIAK, P. R., 2006. "Astronomical network event and observation notification", AN, v.327 (8). p. 775-778. ISSN 0004-6337

Lists of Figures, Equations and Tables

List of figures

Fig. 2.1: Data flow at the TJO-OAdM	40
Fig. 3.1: Architecture conceptual design	58
Fig. 3.2: Modules involved in the single points of failure analysis (highlighted in red).....	59
Fig. 3.3: Logical architecture	60
Fig. 3.4: Physical architecture	62
Fig. 3.5: Subsystem implementation architecture	63
Fig. 4.1: TJO-OAdM areas (Author: J.Bas, from www.parcasrastronomic.cat)	65
Fig. 4.2: TJO telescope	66
Fig. 4.3: Network of CSIMC boards: current (left) and old (right) configurations	67
Fig. 4.4: Diagram connection of the initial set-up for the dome control.....	68
Fig. 4.5: Final version of the dome control block diagram.....	70
Fig. 4.6: General connection diagram	71
Fig. 4.7: FSC1 and dome electronics connection	73
Fig. 4.8: FSC1 and CSIMC board connection	73
Fig. 4.9: FSC1 board	74
Fig. 4.10: CRRD board	74
Fig. 4.11: CCD camera Quantum Efficiency (FLI, www.flicamera.com)	75
Fig. 4.12: Transmission curve of the Johnson-Cousins filters.....	75
Fig. 4.13: Flat screen mounted inside the dome	76
Fig. 4.14: Coupling system to hold the flat screen	76
Fig. 4.15: DLH200S Daylight Soft Light (Dedolight).....	77
Fig. 4.16: Transmission of the DLH200S Daylight Soft Light (Dedolight)	77
Fig. 4.17: Flat light mounted on the wall, with the diffuser surrounding the light	78
Fig. 4.18: Scheme of the electrical wiring at the TJO-OAdM	82
Fig. 4.19: Scheme of the communications wiring at the TJO-OAdM	84
Fig. 4.20: Scheme of the communications wiring at the TJO-OAdM (initial design)	84
Fig. 5.1: Control system conceptual design	87
Fig. 5.2: Control system architecture design	88
Fig. 5.3: Housekeeping control layer design	89
Fig. 5.4: Environment monitoring structure (Francisco et al. 2009)	89
Fig. 5.5: Soft (red) and hard (black) redundancy levels	91
Fig. 5.6: End-to-end data flow design.....	93
Fig. 5.7: Data flow for human supervision or attended control mode.....	94
Fig. 5.8: General states of the control system.....	97

Fig. 5.9: Housekeeping state transitions	99
Fig. 5.10: Data-flow state transitions	99
Fig. 5.11: Humidity variation, where H1 and H2 are high and critical values, respectively.....	108
Fig. 6.1: OpenROCS1.0 global architecture.....	122
Fig. 6.2: Event Monitor block diagram.....	123
Fig. 6.3: Alarm and event solving flow diagram.....	123
Fig. 6.4: Event monitor configuration XML file.....	124
Fig. 6.5: OpenROCS2.0 global architecture.....	126
Fig. 6.6: Server service structure	127
Fig. 6.7: Broadcast service structure.....	128
Fig. 6.8: Monitor service structure	129
Fig. 6.9: Scheduler service structure.....	129
Fig. 6.10: Client module structure	130
Fig. 7.1: Database architecture	141
Fig. 8.1: Preparation process for science observations	150
Fig. 8.2: Phase 1 proposal submission scheme.....	152
Fig. 8.3: Phase 2 proposal submission scheme.....	154
Fig. 9.1: Long-term planning block diagram	163
Fig. 9.2: Mid- and short-term planning block diagram.....	163
Fig. 9.3: Control system modules related with the scheduler.....	165
Fig. 9.4: Scheduler architecture	166
Fig. 9.5: Merit function for three different tasks (top) and time intervals (indicated with boxes) at each merit maximum value where Peak Closeness merit promotes the task selection (bottom)	168
Fig. 9.6: Window merit.....	172
Fig. 9.7: Airmass merit.....	173
Fig. 9.8: After-pick merit	174
Fig. 9.9: Phase-coherent merit	175
Fig. 9.10: Not-before merit.....	176
Fig. 9.11: Terminal merit.....	176
Fig. 9.12: Slew-time merit.....	179
Fig. 9.13: Seeing merit	180
Fig. 9.14: Queue scheduler workflow	183
Fig. 10.1: Task execution module architecture.....	188
Fig. 10.2: TALON interface module.....	191
Fig. 11.1: Data processing work flow schemes.....	196
Fig. 11.2: ICAT architecture.....	199
Fig. 11.3: ICAT interfacing modules	199
Fig. 11.4: Image data process work flow.....	201
Fig. 11.5: Images used to test the analysis software packages. Crowded (M31, left) and sparse (m67, right) fields were selected.	215
Fig. 11.6: Difference of aperture instrumental magnitude measured in the M31 field	217
Fig. 11.7: Difference of aperture instrumental magnitude measured in the M67 field	217
Fig. 11.8: Directory structure design	219
Fig. 11.9: Storage configuration at TJO-OAdM.....	220
Fig. 11.10: Comparison of compression ratio for Rice and GZIP algorithms (CFITSIO – NASA webpage)	221
Fig. 11.11: Fitscopy routine work flow	223
Fig. 11.12: Dataprocessing routine work flow	223
Fig. 11.13: Pipeline routine work flow.....	223
Fig. 11.14: Backup routine work flow	224

Fig. 11.15: Transit light curve of the HAT-P-1b exoplanet obtained using ICAT	242
Fig. 12.1: User interface architecture overview	244
Fig. 12.2: Proposal management interface design.....	245
Fig. 12.3: Observatory operation interface design	246
Fig. 12.4: Off-line access application interface design.....	247
Fig. 12.5: OAdM website home page	248
Fig. 12.6: General website architecture.....	248
Fig. 12.7: Web page basic structure.....	249
Fig. 12.8: Upgraded proposal management implementation architecture	250
Fig. 12.9: Proposal management implemented architecture.....	250
Fig. 12.10: Proposal submission interface (Phase 1 and Phase 2)	251
Fig. 12.11: Proposal data retrieval (Phase 3).....	251
Fig. 12.12: Phase 2 proposal submission I/F snapshot – General information.....	252
Fig. 12.13: Phase 2 proposal submission I/F snapshot – Task execution constraints	252
Fig. 12.14: ICAT pipeline implementation architecture	253
Fig. 12.15: First implemented interface of ICAT	254
Fig. 12.16: TALON XObservatory interface.....	255
Fig. 12.17: OpenROCS interface general design.....	256
Fig. 13.1: Services with external access	263
Fig. 13.2: Light-curve of WASP-33 b. The top panel shows the residuals of the best fit.	268
Fig. 13.3: Seeing at OAdM and at important observatory sites.....	269

List of equations

Eq. 9.1: Figure of merit	167
Eq. 9.2: Figure of merit with long-term contribution	167
Eq. 9.3: Peak Closeness merit function	169
Eq. 9.4: Modified Figure of Merit for dynamic task selection.....	169
Eq. 9.5: Window merit	172
Eq. 9.6: Airmass merit	173
Eq. 9.7: Avoidance merit	173
Eq. 9.8: Avoidance merit for the Moon	173
Eq. 9.9: Sky darkness merit.....	174
Eq. 9.10: After-pick merit	174
Eq. 9.11: Forced time-out merit.....	174
Eq. 9.12: Phase-coherent merit.....	175
Eq. 9.13: At-time merit.....	175
Eq. 9.14: Not-before and not-after merit.....	176
Eq. 9.15: Terminal merit	176
Eq. 9.16: User fairness or executive priority merit.....	177
Eq. 9.17: Priority merit	177
Eq. 9.18: Instrument initialization merit	177
Eq. 9.19: Telescope and dome initialization merit.....	178
Eq. 9.20: Dome shutter open merit	178
Eq. 9.21: Dome flats lamp merit	178
Eq. 9.22: Slew-time merit	178
Eq. 9.23: Instrument change-time merit	179
Eq. 9.24: Weather merit.....	179
Eq. 9.25: Seeing merit	179
Eq. 9.26: SFOM, algorithm for mid- and long-term scheduling	180

Eq. 9.27: Modified dispatch scheduling algorithm for shor-term planning	180
---	-----

List of tables

Table 3.1: Nightly nominal operation	55
Table 3.2: Nominal task run workflow	57
Table 4.1: Dome control operating conditions	69
Table 4.2: Mean exposure time per filter (flat fields)	78
Table 5.1: HW and SW status for each state	98
Table 5.2: Interface definition for environment module	101
Table 5.3: Interface definition for hardware module	101
Table 5.4: Interface definition for ToO service	103
Table 5.5: Interface definition for scheduler service	104
Table 5.6: Interface definition for execution service	105
Table 5.7: Interface definition for processing service	105
Table 5.8: List of events and alarms considered at the TJO-OAdM	107
Table 5.9: Environment monitoring events	109
Table 5.10: System monitoring events	112
Table 5.11: Task ToO events	113
Table 5.12: Task scheduling events	113
Table 5.13: Task execution events	117
Table 5.14: Task processing events	117
Table 6.1: Weather station monitors	125
Table 7.1: User access levels	139
Table 7.2: Database contents generation	141
Table 7.3: Proposal table	143
Table 7.4: Project table	144
Table 7.5: Observation table	144
Table 7.6: Environment table	145
Table 7.7: Users table	145
Table 7.8: Documents table	146
Table 8.1: Proposal Phase 1 and Phase 2 for science observations	155
Table 8.2: Proposal Phase 1 and Phase 2 for the basic set of calibration tasks	158
Table 9.1: Contributions to the merit function	172
Table 9.2: Algorithm parameters to calibrate	181
Table 10.1: System configuration for the states where tasks can be executed	190
Table 10.2: Event-action pairs for task execution management	190
Table 11.1: Data types and levels	202
Table 11.2: Keyword format codes	203
Table 11.3: Detailed list of keywords	206
Table 11.4: Image quality keywords description	210
Table 11.5: Quality descriptor values	210
Table 11.6: Quality control configuration ranges	212
Table 11.7: Description of images used to test the analysis software packages	215
Table 11.8: Results obtained to compare the analysis software packages using aperture photometry	216
Table 11.9: ICAT processing modules	226
Table 11.10: List of new keywords added in the image header	239
Table 11.11: Data storage configuration	240
Table 12.1: Website sections and connection with database levels	248

Table 13.1: Unacceptable risks 259
Table 13.2: Acceptable risks 261
Table 13.3: Requirements traceability to module 267

List of Acronyms and Abbreviations

AC	Alternating Current
ACR	Astrometric Calibration Regions
AIV	Assembly, Integration and Verification
API	Application Programming Interface
ARES	Astronomical mid-REsolution Spectrograph
BRT	Bradford Robotic Telescope
CARMENES	Calar Alto high-Resolution search for M dwarfs with Exoearths with Near-infrared and optical Échelle Spectrographs
CCD	Charge-Coupled Device
CM	Configuration Management
CdM	Consorti del Montsec
Comm	Communications
COTS	Commercial Off-The-Shelf
CRRD	Redundant Control and Rain Detector
CSIMC	Clear Sky Institute Motion Controller
CTA	Cherenkov Telescope Array
Ctrl	Control
DB	Database
DDROCS	Dynamic Database for the Robotic Observatory Control System
Dec	Declination
DF	Data Flow
EM	Environment Monitor
ETC	Exposure Time Calculator
FITS	Flexible Image Transport System
FJO	Fundació Joan Oró
FOM	Figure of Merit
FOV	Field of View
FSC1	Fiber to Signal Converter 1
FWHM	Full Width at Half Maximum

GRB	Gamma-Ray Burst
GSC	Guide Star Catalogue
GUI	Graphical User Interface
HK	Housekeeping
HW	Hardware
IAU	International Astronomical Union
ICAT	IEEC Calibration and Analysis Tool
ICE	Institut de Ciències de l'Espai (CSIC-IEEC)
Ice	Internet Communications Engine
IEEC	Institut d'Estudis Espacials de Catalunya
INDI	Instrument-Neutral-Distributed-Interface
ISROCS	Intelligent Scheduler for the Robotic Observatory Control System
J-C	Johnson-Cousins
JD	Julian Date
LAN	Local Area Network
LST	Local Sidereal Time
LT	Liverpool Telescope
MEIA	Medium-format Efficient Imager for Astronomy
N/A	Not Applicable
NC	Non Critical
NOAO	National Optical Astronomy Observatories
OAdM	Observatori Astronòmic del Montsec (Montsec Astronomical Obs.)
OB	Observation Block
OMI	Optical Mechanics Inc.
ORM	Observatorio Roque de los Muchachos
OS	Operating System
PDR	Preliminary Design Review
PDU	Power Distribution Unit
PI	Principal Investigator
PM	Project Manager
Prt	Priority
PSF	Point Spread Function
QA	Quality Assurance
RA	Right Ascension
RAPTOR	RAPid Telescopes for Optical Response
RD	Rain Detector
RF	Radio Frequency
RMS	Root Mean Square
ROCS	Robotic Observatory Control System
SAR	System Acceptance Review

SExtractor	Source Extractor
SMC	Servei de Meteorologia de Catalunya (Catalan Meteorological Service)
SNMP	Simple Network Management Protocol
SPF	Single Points of Failure
SQFT	SuperWASP Qatar Follow-up Telescope
Std	Standard
STELLA	STELLar Activity project
SW	Software
TAC	Time Allocation Committee
TFRM	Telescope Fabra-ROA at the Montsec observatory
TG	Target Group
TJO	Telescope Joan Oró
ToO	Target of Opportunity
UB	Universitat de Barcelona
UPC	Universitat Politècnica de Catalunya
UPS	Uninterruptible Power Supply
USNO	U.S. Naval Observatory
VO	Virtual Observatory
XML	Extensible Markup Language
WCS	World Coordinate System

Annex 1: Test Procedure, Running Step by Step

Routine operations

Step #	Action	Success criteria	Results	Done (UT)	Remarks
R01	Night planning	Generation of sorted list of night's tasks (calibration and science)			
R02	Start up the system	Power up the telescope and dome electronics and drives from the IP switches, launch the TALON software, open the telescope covers, power up the CCD, home the telescope, unpark and home the dome			
R03	Start calibration images	Start calibration images: start executing the calibration tasks. This is controlled by OpenROCS, which provides the task sequence to TALON. Point the telescope to the location of the flat screen, turn on the flats lamp when needed. Images should be stored in the proper location and pipeline reduction of all images should start as well as quality control.			
R04	Stop calibration observations	Turn off the flats lamp, close the telescope covers			
R05	Start thermalization	Open the dome and open the telescope covers			
R06	Start science observations	This is also controlled by OpenROCS, which provides the task sequence to TALON. Images should be stored in the proper location and pipeline reduction of all images should start, followed by quality control and population of the database with the necessary information			
R07	Stop science observations	Close the telescope covers and close the dome afterwards			
R08	Start of calibration images	Open the telescope covers, follow the sequence of calibration tasks provided by OpenROCS, point the telescope to the location of the flat screen, turn on the flats lamp when needed			
R09	Stop calibration images	Turn off the flats lamp, close the telescope covers, launch final reduction of the night. The latter will produce science data and populate the database with all necessary information			
R10	Stop the system	Stow telescope, park dome, stop talon, turn power off using the IP switches			

Operation with weather alarms

Step #	Action	Success criteria	Results	Done (UT)	Remarks
W01	Rain detected in S2	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W02	Rain detected in S3	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W03	Rain detected while the dome subsystem is opening shutter	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W04	Rain detected while the dome subsystem is closing shutter	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W05	Outside weather station humidity alarm in S2	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W06	Outside weather station humidity alarm in S3	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W07	Outside weather station wind alarm in S2	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W08	Outside weather station wind alarm in S3	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W09	Indoor weather station humidity alarm in S2	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W10	Indoor weather station humidity alarm in S3	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W11	Thunderstorm event inducing a sudden power cut on all	Response as in Sect. 5.4.4.4 for all concerned subsystems			
W12	Thunderstorm event inducing a sudden power cut on all	Response as in Sect. 5.4.4.4 for all concerned subsystems			

Operation with system alarms

Step #	Action	Success criteria	Results	Done _{UT}	Remarks
A01	ISROCS application not responding	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A02	ICAT application not responding	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A03	Error accessing database	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A04	SUB-ALIS system not responding to pings	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A05	SUB-ESTALL system not responding to pings	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A06	SUB-SARGA system not responding to pings	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A07	SUB-REBEL system not responding to pings	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A08	Router not responding to pings	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A09	No access to Internet	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A10	Disk space low in SUB-ALIS	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A11	Power outage in UPS-CPU's	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A12	Power outage in UPS-Dome	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A13	Power outage in UPS-Telescope	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A14	No environment sensor data	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A15	TALON application on SUB-ESTALL not running	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A16	Telescope hardware error	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A17	CCD hardware error	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A18	Dome hardware error (shutter/flap and azimuth)	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A19	GPS error	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A20	OpenROCS (SUB-ESTALL, SUB-SARGA, SUB-REBEL, SUB-ALIS) not responding	Response as in Sect. 5.4.4.4 for all concerned subsystems			
A21	Sensor error	Response as in Sect. 5.4.4.4 for all concerned subsystems			

Data reduction, storage and quality control

Step #	Action	Success criteria	Results	Done (UT)	Remarks
D01	Keywords in header	In compliance with Sect. 11.3.2.2			
D02	Images are copied to the correct directories in SUB-ALIS	Files in /mnt/storage/rawdata/"date", storage directory configured on this subsystem			
D03	On-the-fly image reduction (both calibration and science images)	Data files are produced as specified in Sect. 11.3.4			
D04	Quality control	Files are generated compliant with Sect. 11.3.3			
D05	Final reduction	Resulting final calibration images and science products as specified in Sect. 11.3.4			
D06	Final quality control	Files generated compliant with Sect. 11.3.3			
D07	Data products stored and metadata are ingested in the database	Files in /mnt/storage/reddata/"date" and database storage in compliance with Sect. 11.3.5 and Sect. 11.3.6			
D08	User interface	Compliant with Sect. 12.2.3			
D09	Proposal management tool	Compliant with Chapter 8 contents and Sect. 12.2.2			

