

Niko Lavonen

User centered evaluation and design for improving the usability of the quasar observation system for a radio telescope

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology
Espoo 16.7.2013

Thesis supervisor:

Docent Kalevi Kilkki

Thesis instructor:

Docent Merja Tornikoski

Author: Niko Lavonen

Title: User centered evaluation and design for improving the usability of the quasar observation system for a radio telescope

Date 16.7.2013

Language: English

Number of pages: 8+126

Department of Communications and Networking

Professorship: Network Economics

Code: ETA3003

Supervisor: Docent Kalevi Kilkki

Instructor: Docent Merja Tornikoski

This thesis presents a user centered evaluation and design process applied to improve the usability of the quasar observation system used in Aalto University Metsähovi Radio Observatory. Metsähovi Radio Observatory has monitored the variability of high frequency radio emission from quasars and other active galaxies for the last three decades. The observatory monitors active galaxies 24 hours a day, year round. Observing work is done by Metsähovi's own staff parallel with other duties, using indigenously developed observation software. Running routine observation work takes time away from research, which is why the observation system has been developed over the years to allow producing high quality data with a smaller work load. In this thesis, the development work was continued by applying methods of user centered evaluation and design.

The study was conducted in two iteration rounds. On the first iteration, usability data was collected from the system with contextual inquiry, heuristic evaluation and user feedback. The data was analyzed with the goal of recognizing the main usability problems in the system. Solutions to the problems were developed and implemented where applicable with limited resources. On the second iteration, the solutions were evaluated using in part the same methods as on the first iteration and in part by user testing.

One of the main problems in the system was associated with adding data and comments regarding observing conditions to preliminary observation results. In general, there were too many separated tools for single tasks. Access to important information on observing conditions was slow and the information was scattered over many tools. Running semi-automatic observations wasn't flexible enough. Solutions were developed for cutting down the number of tools needed to run tasks and for allowing quick access to relevant weather information.

Evaluation of the changes made to the system, points toward significantly increased efficiency and user satisfaction, while effectiveness remained on a high level. As it was only possible to implement a small portion of the suggested improvements, many solutions await implementation in the near future. To further improve the quasar observation system, it is recommended that user centered development of the system is continued in the future.

Keywords: user centered design, usability, usability evaluation, contextual inquiry, heuristic evaluation, usability test, user feedback, radio astronomy, Metsähovi radio observatory, radio telescope, quasars, observations, active galactic nuclei

Tekijä: Niko Lavonen

Työn nimi: Radioteleskoopin kvasaarimittausjärjestelmän käyttäjäkeskeinen arviointi ja -suunnittelu

Päivämäärä: 16.7.2013

Kieli: Englanti

Sivumäärä: 8+126

Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotalous

Koodi: ETA3003

Valvoja: Dosentti Kalevi Kilkki

Ohjaaja: Dosentti Merja Tornikoski

Tässä diplomityössä käydään läpi käyttäjäkeskeinen arviointi- ja suunnitteluprosessi Aalto-yliopiston Metsähovin radiotutkimusaseman kvasaarimittausjärjestelmän käytettävyyden parantamiseksi. Metsähovissa on mitattu kvasaarien ja muiden aktiivisten galaksinytimien korkeataajuisten radiosäteilyn muuttuvuutta yli kolmekymmentä vuotta. Observatorion henkilöstö tekee mittauksia vuoroissa muun työnsä ohessa itsenäisesti kehitetyllä mittausjärjestelmällä. Mittaukset jatkuvat 24 tuntia vuorokaudessa, vuoden ympäri. Rutiinimittausten tekeminen vie aikaa pois tutkimustyöstä ja mittausjärjestelmää onkin resurssien rajoissa pyritty kehittämään sellaiseksi, että korkealaatuista havaintodataa voitaisiin tuottaa pienemmällä työmäärällä. Tässä diplomityössä kehitystyötä jatkettiin soveltamalla käyttäjäkeskeisen suunnittelun menetelmiä.

Tutkimus suoritettiin kahdessa iteraatiossa. Ensimmäisellä iteraatiokierroksella kvasaarimittausjärjestelmästä kerättiin käytettävyydataa tilannesidonnaisilla haastatteluilla, heuristisella arvioinnilla sekä keräämällä käyttäjäpalautetta. Kerätty data analysoitiin ja järjestelmän suurimmat ongelmat pyrittiin tunnistamaan. Ongelmiin kehitettiin ratkaisuehdotuksia ja diplomityön ohessa toteutettavissa olevat ehdotukset toteutettiin. Toisella iteraatiokierroksella toteutetut muutokset arvioitiin soveltamalla ensimmäisellä kierroksella käytettyjen menetelmien lisäksi käytettävyydestä.

Järjestelmästä löytyneet ongelmat liittyivät pääosin heikkoon tehokkuuteen; yksittäisten tehtävien tekemiseen vaadittiin usean erillisen työkalun käyttöä. Pääsy mittausyössä tarvittavaan informaatioon oli osin hidasta ja hajallaan useassa työkalussa. Kommenttien, poistomerkintöjen ja tau-arvojen lisääminen mittausdataan oli yksi suurimmista ongelmakokonaisuuksista. Myös mittauskomentojen antamisen ja puoliautomaattisten mittauslistojen ajamisen joustavuudessa oli puutteita. Ratkaisujen toteuttaminen keskittyi mittausdatan esikäsittelyyn käytettävien yksittäisten työkalujen määrän vähentämiseen AJAX-tekniikalla. Myös datan esikäsittelyssä tarvittavien säätietojen nopeaan katseluun kehitettiin ratkaisu. Muutosten arvioinnin tulokset viittaavat selkeästi parantuneeseen tehokkuuteen ja käyttäjien tyytyväisyyteen, tuottavuuden pysyessä korkealla tasolla.

Koska havaituista ongelmista voitiin korjata vain osa ja järjestelmä on edelleen kokoelma erillisiä työkaluja, järjestelmän käyttäjäkeskeistä kehittämistä suositellaan jatkettavaksi.

Avainsanat: käyttäjäkeskeinen suunnittelu, käytettävyys, käytettävyyden arviointi, tilannesidonnainen haastattelu, heuristinen arviointi, käyttäjäpalaute, käytettävyydestä, radioastronomia, Metsähovin radiotutkimusasema, radioteleskooppi, kvasaari, aktiivinen galaksinydin

Foreword

This diploma thesis presents a usability study conducted to improve the usability of the quasar observation system used in Aalto University Metsähovi Radio Observatory. This thesis was a long time in the making. Ever since I started learning the secrets of observing work in Metsähovi radio observatory in the summer of 2009, I had been thinking about how it would be interesting to make a usability evaluation of the observation software. Being a student of network business working in a radio astronomical observatory, I feel lucky to have gotten the opportunity to write a thesis about usability. My mixed interests were met with acceptance and support both at Metsähovi and at the Department of Communications and Networking (Comnet) of Aalto School of Electrical Engineering.

I'd like to thank my instructor Docent Merja Tornikoski of Metsähovi and my supervisor Docent Kalevi Kilkki of Comnet for their good advice and support. Metsähovi has been an inspiring place to work in thanks to its relaxed, nonrestrictive atmosphere and coworkers who have been open toward developing the quasar observation system. I'd also like to specially thank all the interviewed observers who had admirable patience toward a nosy engineer pouring out an infinite stream of silly questions about their work. Many thanks also go to my colleagues Niko Kareinen, Juha Aatrokoski and Miika Oksman, who were always eager to help in creating the technical implementations needed to develop the observation system.

Special thanks to my parents, brother, grandparents and my friends for their support over the years I've spent studying in Aalto University.

Metsähovi, 15.4.2013

Niko Lavonen

Contents

Abstract	ii
Abstract (Finnish)	iii
Foreword	iv
Contents	v
Abbreviations	viii
1 Introduction	1
1.1 Background.....	1
1.2 Goals.....	2
1.3 Scope	2
1.4 Structure of the thesis	3
2 Background and starting point for the study	4
2.1 Summary.....	4
2.2 Metsähovi Radio Observatory	4
2.3 Quasar research project	5
2.4 Quasars	5
2.5 Quasar observations	5
2.5.1 Sources	6
2.5.2 Observing shifts	6
2.5.3 Observing conditions	6
2.5.4 Antenna pointing.....	6
2.6 History of the quasar observation system.....	7
2.6.1 Early years 1974-1985.....	7
2.6.2 Sleepless nights 1985-2004	7
2.6.3 Automatic lists and remote observations 2004-2010	8
2.6.4 Recent developments 2010-.....	8
2.7 Current state of the quasar observation system	9
2.7.1 The VNC desktop	10
2.7.2 Observation summary and observations delete status and comments	11
2.7.3 Real time sky view and autofile generator	13
2.7.4 Tau-tools	14
2.7.5 MB data plotting tool.....	15
2.7.6 The pointing offsets tool.....	17
2.7.7 The surveillance cameras	18
2.8 Users	19
2.9 Use context	21
2.10 Development goals and known problems.....	22
3 Theoretical background	24
3.1 Summary.....	24
3.2 Usability.....	24
3.3 User centered design	25
3.4 User interfaces	28
3.4.1 WIMP	29
3.4.2 Web user interfaces.....	29
3.4.3 User interface design.....	29

3.5	Usability evaluation.....	32
3.5.1	<i>User testing</i>	33
3.5.2	<i>User feedback</i>	33
3.5.3	<i>Questionnaires and interviews</i>	33
3.5.4	<i>Heuristic evaluation</i>	34
3.5.5	<i>Cognitive walkthrough</i>	35
3.5.6	<i>Problem severity ratings</i>	36
3.6	Task analysis.....	36
3.7	Development processes	37
3.7.1	<i>Contextual Design</i>	37
3.7.2	<i>Usability engineering lifecycle</i>	40
3.8	Evaluation and design process	41
3.9	Conclusions.....	43
4	Execution of the study	45
4.1	Summary.....	45
4.2	First iteration.....	46
4.2.1	<i>User feedback and developers' ideas</i>	46
4.2.2	<i>Contextual inquiries</i>	47
4.2.3	<i>Heuristic evaluation</i>	49
4.2.4	<i>Analysis</i>	50
4.3	Implementing modifications	51
4.4	Second iteration.....	51
4.4.1	<i>User testing</i>	52
4.4.2	<i>Heuristic evaluation</i>	54
5	Results.....	55
5.1	Summary.....	55
5.2	First iteration of studies	56
5.2.1	<i>User feedback and developers' ideas</i>	56
5.2.2	<i>Contextual inquiries</i>	59
5.2.3	<i>Heuristic evaluation</i>	68
5.3	Analysis and recommendations	72
5.3.1	<i>Choosing sources for observing</i>	74
5.3.2	<i>Running observations</i>	77
5.3.3	<i>Adding comments to the data</i>	77
5.3.4	<i>Adding and modifying tau values</i>	80
5.3.5	<i>Marking good and bad quality observations</i>	82
5.3.6	<i>Exporting data files for final reduction</i>	84
5.3.7	<i>Use of the radome heater</i>	85
5.3.8	<i>Accessing relevant weather information</i>	85
5.3.9	<i>Overall dispersion of information in the system</i>	88
5.3.10	<i>Miscellaneous</i>	88
5.4	Second iteration of studies.....	89
5.4.1	<i>User testing</i>	89
5.4.2	<i>Heuristic evaluation</i>	95
5.4.3	<i>Conclusions and recommendations</i>	97
6	Conclusions.....	100
	References	102
	Appendix A: Questions of the contextual inquiry preliminary interviews .106	
	Appendix B: Results of the contextual inquiry preliminary interviews.....107	

Appendix C: A sequence model from a contextual inquiry	108
Appendix D: Interview notes from the user tests	112
Appendix E: Test plan for the user tests	117
Appendix F: Interviewer's version of test procedures for the user tests	118
Appendix G: User's version of test procedures for the user tests	123
Appendix H: Wrap-up interview template for the user tests	125
Appendix I: SUS-questionnaire for the user tests.....	126

Abbreviations

AJAX	Asynchronous JavaScript and XML. A set of technologies for creating interactive web applications. Based on asynchronously retrieving data from the server without a page reload.
CGI	Common Gateway Interface. A technology in which an executable file or a script generates web content, e.g. an HTML document, for a web server.
CLI	Command Line Interface. A user interface that is based in completely textual input and output.
CSS	Cascading Style Sheets. A style sheet language for modifying the appearance of documents written in markup languages such as HTML.
DHTML	Dynamic HTML. A set of technologies that allow modifying the appearance of an HTML document after it has been loaded.
DOM	Document Object Model. A convention for modifying and interacting with objects in HTML, XHTML and XML documents.
GUI	Graphical User Interface. A user interface where the user interacts with the system through images and graphics.
HCI	Human Computer Interaction. A multi disciplinary field of research that studies the interaction between humans and computers.
MRO	Metsähovi Radio Observatory.
SSH	Secure Shell. An application level encrypted network protocol.
UCD	User Centered Design. An approach to designing interactive systems where systems are made usable by focusing on the users, their needs and requirements and by applying human factors/ergonomics, and usability knowledge and techniques.
VNC	Virtual Network Computing. A protocol used in remote access to a computer's graphical user interface (see GUI).
WIMP	Windows, Icons, Menus, Pointer. An interaction style used in graphical user interfaces (See GUI).

1 Introduction

1.1 Background

Today, computers have become an inseparable part of people's lives. People expect computer systems to support their work and to make their lives easier. For the last two decades, the science and practice of human computer interaction, the art of making the communication between humans and machines more fluent, has developed and matured. Designing computer systems from the point of view of the people using them has become more and more commonplace. Today, usability is an essential part of the quality of any piece of software. Good usability improves productivity, lowers development cost and improves user satisfaction. Technologies for quickly creating user interfaces have also been developed, lowering the threshold for creating highly usable computer systems.

For the last 32 years, Aalto University Metsähovi Radio Observatory has collected data from active galaxies to build up a picture of these powerful objects in the far reaches of our universe. Thanks to the data collected in Metsähovi and other observatories around the world, researchers today have a clearer view on the structure and behavior of active galaxies than ever before. In Metsähovi, quasars are monitored 24 hours a day, year round. The observatory's staff takes shifts of 1-7 days to observe with the 13.7-meter telescope. The observations are made with indigenously developed Linux software, supported by multiple web-based tools.

Observing work can be very time consuming when aiming for high quality astronomical data. Over the years, the observation system has been developed to lessen the work load and to improve the pleasantness of observing. The current observation system allows observers to conduct semi-automatic observations remotely over the internet. The telescope can be programmed to observe sources on a predetermined list for 12 hours or more. During this time the observer can go about her business, occasionally giving the observing conditions and results a glance.

In the early 1990's observers needed to stay in the observatory during observations. Their shifts lasted 24-48 hours and included manually inputting observation commands to the telescopes control computer every 20 or 30 minutes. Sleep was a rare commodity. Despite the drastically improved situation today, development must continue. Today's strict budgets and increasing bureaucratic work load on the researchers make every minute precious. Researchers need to be able to focus on science instead of routine observation work.

In this thesis, a user centered evaluation and development process is followed to improve the usability of Metsähovi's quasar observation system. User centered design is an approach to designing interactive systems, such as computer software and devices. It emphasizes the inclusion of users and their needs in the design. The main goal in the study is to lower the workload of observing while maintaining high data quality.

Today, there are several widely used frameworks (SFS-EN ISO 9241-210, 2010), processes (Beyer, Holtzblatt, 1998, Mayhew, 1999), definitions (SFS-EN ISO 9241-210, 2010, Nielsen, 1993) and methods (Nielsen 1993) that can be used when applying user centered design to specific design cases. User centered design has its roots in study of human ergonomics. Over a hundred years ago, Frederick Taylor (1911) presented time and motion studies to increase industrial efficiency. During the Second World

War, Alphonse Chapanis studied and developed the ergonomics of war plane cockpits (Horn, 2002). In the 1960's, trade unions in Scandinavia came up with the participatory design approach, including workers in the design of their own work (Bjerknes, et al. 1987). Today, human computer interaction (HCI) is its own field of science and it's sometimes also known as computer human interaction (CHI) and man machine interaction (MMI), among others. HCI is a highly multidisciplinary field, incorporating knowledge from sociology, cognitive psychology, human factors, ethnography, computer science and design. It essentially combines knowledge of the human and machine sides of human computer interaction.

1.2 Goals

The main goal of this thesis was to improve Metsähovi's quasar observation system so that work load and stress of routine observing work would be reduced. This was to be achieved by finding, and if possible, fixing, the most prominent usability problems in the observation system. User centered evaluation would be conducted on the modifications to find out if they actually had improved the system's usability. The thesis work was to produce concrete improvements in the observation system instead of only suggested improvements. To gain knowledge on the theory behind finding and fixing usability problems, a literature review would have to be conducted. The literature review would also have to give an overview of the available technologies for creating user interfaces.

The following research questions were conceived for reaching the main goal:

1. What methods and processes could be used in studying the usability of the quasar observation system?
2. What are the biggest usability problems in the quasar observation system?
 - a. How can the problems be fixed?
 - b. Have the implemented solutions improved the system's usability?

1.3 Scope

This thesis presents user centered evaluation and design of Metsähovi's quasar observation system. Focus is on the observing work and related tools that impact a large portion of Metsähovi's staff. Software and practices for final data reduction are given only little attention. These are known to be a vital part of the data production pipeline, but they mostly only affect the work load of one person. That is why full attention is given to the development of practices and tools for the final reduction only after this thesis is finished.

The thesis also does not converse on the specific details of the programming solutions developed or on software development processes. No extensive literature review is made for comparing which technologies would be ideal for implementing the user interfaces. The focus is on user centered development and usability issues. The documentation and coding work related to the programming solutions are also not considered to be in the scope of this thesis.

The study leaves out the help and documentation parts of the observation system. Most of these are presented in the Wiki pages of Metsähovi's intranet. The goal in the user interface solutions was to make the user interfaces self descriptive and intuitive so that no manual reading is required on regular basis.

This thesis does not, for the most parts, converse on observing techniques or other radio technological issues regarding the observations with Metsähovi's radio telescope. Matters having to do with e.g. integration times, antenna pointing algorithms or calibrations are not in the scope of this study. Also, no extensive attention is given to hardware solutions related to observations.

1.4 Structure of the thesis

This thesis is divided into six chapters. The first chapter gives an introduction to the subject of the thesis, presents the goals of the study and gives a brief introduction into the user centered design approach. The second chapter presents Metsähovi Radio Observatory, the quasar observation system and Metsähovi's goals for the development of the system. In the third chapter a literature review is made on the research methodology applied in the study. The most important principles and methods of user centered design are presented together with basics of user interfaces. The fourth chapter presents the practical implementation of the study. The fifth chapter presents the results of the study. The results include tables of usability problems and solutions, analysis of the observing work and presentation of implemented solutions and their evaluation. In chapter 6, conclusions are drawn of the results, the success of the study is evaluated and recommendations on continuing the observation system's development are given.

2 Background and starting point for the study

This chapter gives an introduction on Metsähovi Radio Observatory and presents the history and current state of the quasar observation system. A description of the observation system's users and use contexts is given. Also, in addition to some known problems in the observation system, the goals set by Metsähovi's staff for the development of the system are presented.

2.1 Summary

Metsähovi Radio Observatory operates a 13.7-meter radio telescope in the village on Kylmälä in Kirkkonummi, Finland. The telescope is used, among other purposes, for the monitoring of radio variability of quasars and other active galactic nuclei. The quasar observations run 24 hours a day, year round, producing time series data. The observations are conducted by Metsähovi's staff in shifts of 1-7 days. The observers pre-process the data they have produced. They then hand it over to the lead scientist who conducts the final data reduction. The finalized data is used for studying the structure and behavior of active galaxies.

The observations are conducted using a suite of mostly internally developed software. The telescope itself is controlled with Linux based software. Web-based tools, mostly based on server side Perl CGI (Common Gateway Interface) scripts, are used for selecting sources and viewing the results data. Observing conditions are also monitored with web-based tools. The whole system can be accessed from outside the observatory through a VNC (Virtual Network Computing) remote desktop. The web tools are also accessible to web browsers via an SSH (Secure Shell) proxy connection.

The system allows successful control of all the aspects of observing work. However, it is known that using the system is in many parts inefficient. This increases the work load and stress associated with observing and data processing. The software has been continually developed to increase efficiency and pleasantness of observing, but lack of resources has slowed down the development.

Goals for the development of the system included increasing the efficiency and pleasantness of routine observing work, creation of a queuing system for observing commands and the implementation of an automatic source selection system allowing automatic observations with the push of a button. Developers have also been aware of the fact that the system is too scattered. Initial prototypes have been created to implement a more centralized and integrated observing user interface. Streamlining of the high workload final reduction by better utilizing the observatory's sensor data is also on Metsähovi's development agenda.

2.2 Metsähovi Radio Observatory

Metsähovi Radio Observatory (MRO) is a separate research institute of Aalto University School of Electrical Engineering. The observatory is located about 35 km west from the Aalto University's Otaniemi campus area, in Kylmälä, Kirkkonummi. The observatory was established in 1974 and has since been active in radio astronomical research, development of radio astronomical instruments, development of methods for radio astronomical measurements, space research and education. Most important research projects include quasar research, Very Long Baseline Interferometry (VLBI),

and solar research. The largest project in Metsähovi is the quasar research project. The quasar observation system, which is the subject of the usability evaluation in this thesis, is the main set of tools used in this project. Metsähovi's primary instrument is a 13.7-meter Cassegrain type radio telescope that's enclosed in a weather proof dome. Observations are made at a frequency range of 11 to 120 GHz. Metsähovi Radio Observatory employs approximately 25 scientists, engineers, research assistants and support personnel. The institute receives its funding from Aalto University, Academy of Finland and other outside sources. (Tornikoski et.al. 2012)

2.3 Quasar research project

Since 1980, Metsähovi Radio Observatory has been involved in research and observation of quasars and other Active Galactic Nuclei (AGNs). The quasar monitoring project, which is a joint effort between Metsähovi and Turku University Tuorla Observatory, is the largest project in Metsähovi. Quasar observations take most of the telescope time of the 13.7-meter telescope which is used for observations 24 hours a day, 365 days a year. Since the beginning of the project, Metsähovi has made continuous observations of a set of approximately 85 quasars at frequencies ranging from 22 GHz to 87 GHz. The most typical observing frequency is 37 GHz. Altogether; Metsähovi has made observations of over 1000 individual sources. These high frequency long term observations provide time series data for studying the behavior and structure of quasars. Metsähovi is unique in the sense that it has been able to dedicate so much telescope time on this one project over such a long time period.

2.4 Quasars

A quasar, or a quasi-stellar (star like) radio source, is the nucleus of a distant galaxy with a super-massive black hole in its center. The name quasi-stellar radio source stems from the fact that quasars appear as point sources in the sky. Quasars emit vast amounts of electromagnetic energy in many wavelengths, including radio waves. The source of this energy is an accretion disk formed of matter falling in to the black hole, together with jets of matter shooting off from the poles of the spinning black hole. The central region of a quasar emitting excess radiation is called an Active Galactic Nucleus (AGN). The central region far outshines the radiation emitted by the host galaxy itself. As the super-massive black hole spins, jets of matter shoot away from its poles at speeds near the speed of light, also emitting vast amounts of electromagnetic radiation. Quasars are among the most distant objects known, with some being observed over 10 billion light years away. (Hovatta, 2009)

2.5 Quasar observations

In this section, aspects of conducting observations with Metsähovi's 13.7-meter telescope are introduced. The pointing offsets and operation of the dome heater are introduced. To summarize, the quality of the collected data is affected by the weather conditions and condition of the radome and the receiver. Additionally, sources need to be observed in appropriate azimuth and elevation coordinates using up to date and accurate pointing offsets.

2.5.1 Sources

In Metsähovi, a relatively fixed, although a gradually growing set of radio sources has been observed for over 30 years. The source list, along with the information on past observations, forms the basis on what sources need to be observed at any given time. The sources on Metsähovi's source list are divided into different categories, depending e.g. on how often they are supposed to be observed, whether they are a part of an observing campaign and what kind of observation technique they are supposed to be observed with. The source list forms the starting point for the observations.

2.5.2 Observing shifts

Quasar observations run 24 hours a day, year round. The quasar project's lead scientist allocates observing time to shifts of 1-7 days in co-operation with the observers. During a shift, the observer is responsible for conducting observations with the telescope. The observer is also responsible for the data produced up to the point where it's ready for final reduction.

Before submitting the observation results for final reduction, the observer goes through the raw data he/she has produced. Bad quality results such as observations that were made during rain are marked to be excluded from the final reduction. Comments are added to the data to notify the lead scientist making the reduction of any changes in observing conditions or in the telescope's hardware. The observer also has to add atmospheric opacity, or tau values to each observation result. The values are manually determined based on cloudiness and humidity conditions.

2.5.3 Observing conditions

Certain weather conditions hinder the observations, most importantly rain, thick watery clouds in the atmosphere and water, snow or ice on the radome. The radome is a spherical structure around the telescope that consists of an aluminum mesh frame covered with a plastic membrane. The radome protects the telescope itself from the elements, but also forms a platform for water, snow and ice to stick to.

To combat the ill effects of water in any of its forms, the observatory is equipped with a powerful heater that can blow warm air into the radome. The heater can raise the temperature in the dome considerably, which helps in melting snow and ice and drying out mist and rain. The heater is in relatively frequent use, especially in preventing mist forming on the dome during nights with high humidity. Excess use of the dome heater is avoided, because rise in the dome's temperature also affects the telescope's receivers which can make data reduction harder and result in low quality data.

2.5.4 Antenna pointing

When the telescope is pointed to different parts of the sky, the actual coordinates in which the telescope's beam points vary compared to the intended coordinates. This is due to mechanics of the telescope, imperfections in the pointing algorithms and gravitational and heat deformations. To counteract the inaccuracies in the telescope's pointing, so called pointing offsets are used. The offsets are parameters that are added to the coordinates in which the telescope is pointed.

To gain knowledge on what kind of offsets to use in various parts of the sky, a so called 5-point observation is needed. In a bright 5-point observation, a point source is measured by pointing the telescope's beam in a cross-like pattern on and around the intended coordinates of a source. This gives a pattern with 5 data points. By applying a

2-dimensional Gaussian fit to the pattern, it can be seen how well the telescope was pointed to the actual source. The pointing offsets for the part of the sky in question depend on how far the maximum value of the Gaussian fit is located from the intended center of the pattern.

The pointing offsets vary over time depending on weather conditions and mechanical properties of the telescope. Because of this, the offsets need to be maintained and updated regularly with the help of 5-point observations. Updating the offsets requires a high quality 5-point fit from a bright enough source observed during good and steady weather conditions.

2.6 History of the quasar observation system

In this section an overview of the quasar observation system's history is presented to give perspective on the usability evaluation performed in this thesis work. From the observer's viewpoint, the way quasar observations are made in Metsähovi has changed substantially over the years. While the end result, time series data of quasars, has remained essentially the same, the work load of the work has been reduced. The observers no longer need to stay awake and work in situ in Metsähovi during their observation shift, inputting every observation command manually. They can conduct semi-automatic remote observations from their home and usually have a good night's sleep.

2.6.1 Early years 1974-1985

In the 1970's and 1980's, in the years following the founding of Metsähovi in 1974, point source and solar observations were conducted using a Hewlett Packard 2100A computer with programs written in the HP-Basic language (Urpo, 1982). The observation programs were read either from magnetic or paper tapes. The programs were used with an interactive command line interface. The program was run and the user reacted to instructions given by the program. (Holsti et. al, 1983)

2.6.2 Sleepless nights 1985-2004

Around the mid 1980's Metsähovi's observation systems were transferred to be used on a MicroVAX II computer with the Ada language. A point-source continuum observation program called *Contobs* was developed to be used mostly for quasar observations. *Contobs* was a further development of a program called *FIVE* (Laurila, et al. 1989) that was originally used on the HP 2100-series machines. For solar observations, a program called *SMART* was conceived (Tornikoski, 1990). Like their predecessors on the Hewlett Packard computer, both of these programs relied on an interactive command line interface.

During this time, every separate observation command had to be input manually into the MicroVAX computer in the observatory. Hardware related to observing also had to be operated in situ. The hardware included the dome heater, which is used for removing moisture and snow from the dome, and the calibration diode switch. This meant that the observer had to spend her whole observing shift on location in Metsähovi. The observer could only have breaks for about half an hour at a time, the time it took for one observation to finish. This was very tiring. To ease the burden, outside help was recruited. Mostly students from Helsinki University of Technology and University of Turku were called to Metsähovi just for observing duties. Even with outside help,

Metsähovi's regular staff still handled approximately half of the observing shifts. (Tornikoski, Lähteenmäki, personal communication, 2012)

In 1991, it became possible to run pre-made observing command lists that lasted up to two hours (Tornikoski, Lähteenmäki, personal communication, 2012). This meant that the observer could have longer breaks and didn't have to input every source command into the computer before execution. Since the beginning of the quasar observations in Metsähovi in the 1970's, the observers did not decide what sources were observed, but a hand written list of sources was given to them before their shift began. In 2001, this method of selecting sources was changed to the currently used method where the observer makes the last call on what to observe (Lähteenmäki, personal communication, 2012).

2.6.3 Automatic lists and remote observations 2004-2010

In 2004 a web based system was put into use that allowed generation of observation source lists by clicking sources on a sky map (Tornikoski, Mujunen, 2005). This allowed generation of much longer list than was previously possible. Now lists exceeding 12 hours could be generated. This system, whose autofile generator user interface is presented in section 2.7.3, was still in use when this thesis was finished.

The limiting factor for the length of a list is typically the difference in the duration of a single observation on the list and in reality and the changes in weather conditions. The web based tool can't predict exactly how long an observation and the associated telescope slewing time will be. In the end of a long list the difference between the list and reality can be e.g. 30 minutes, which can affect the validity of the preselected pointing offsets. The source can also set too low or rise too high.

In December 2004 Metsähovi also upgraded the control electronics of the dome heater, which allowed remote operating of the heater. With the help of VNC remote desktop software, it was now possible to conduct remote observations away from Metsähovi. (Tornikoski, Mujunen, 2005)

2.6.4 Recent developments 2010-

Up until 2010, the observers used paper logs to document which sources they had observed, which parameters they used, what the results were and what the conditions had been like. The logs were used especially in the final reduction of the data. The reduction wasn't based on the values in the hand written logs, but the logs were a valuable document on what had happened. A picture of a typical observation log is presented in Figure 1. With the introduction of the current web based results table tools in 2010 (see section 2.7.2), the paper logs could be printed out as PDF-files and hand written paper logs were no longer needed.

Day 09/07/13		Observer ML		Frequency 37		Mode		File(s) 090713_61X		
Prog		Gain		Sp-grid		Time/beam		Subtel		
UTC	Source	Az	El	Az-off Err	El-off Err	Sp	Result	Err	t(tot)	Weather (T_ant, T_out, humidity) & Comments
	CAL	0	30				353,3	1,6	75/180	EP08041600VRES10 - 090713.5
12:53	CAL	-90	45				344,9	1,2	75/180	Baseline correction finished T _a = 18,7, T _o = 16,0, R _h = 84,2%
13:14	0552+398			5	-76		71,7	1,6	1800	
	20274	-73,6	41,0	4/-11	-73/10	X	67,6	1,7	1400	
14:55	CAL	0	45						15/180	T _a = 20,2, T _o = 19,3, R _h = 75,5%
15:39	1928+738 (uv)			8	-73				1400	
16:22	0834+710 (VEP)			14	-76				1600	
17:06	0804+490			6 ^(B-?)	-76				1600	
17:50	0917+449			5	-76				1800	
	CAL	0	30						75/180	
	552057+77			13	-74				1600	
18:20	DR21			8/56	-73/15	X	75,7	0,8	200	OFFSET?? => ET ANTENNA VIEA?
18:35	DR29			5/-2,1	-73/-27	X	74,7	0,7	400	
19:10	DR21			4/-1,0	-73/-0,8	X	76,0	0,3	1200	
19:21	CAL	0	45				338,8	0,1	1100	T _a = 20,3, T _o = 17,1, R _h = 78,5%
20:00	2109			11	-77				1800	
	2159			9	-76				1800	
22:37	N6C7027			13	-74	X			1400	
22:42	CAL	0	45						75/180	
23:31	2346+385			4	-73				1800	

Figure 1. A handwritten paper log of quasar observations from 2009-07-13.

For many years in the 2000's, a transfer away from the old MicroVAX-based system into a more modern Linux based observing system was on the agenda. Technical problems related to the observing software and a lack of resources delayed the actual launch to 2009, when the Linux based "measure" observing program was taken into use (Tornikoski et al., 2010). In 2011, the MicroVAX II computer was permanently shut down (Tornikoski, Holmberg, Uunila, 2013).

As such, the move to the new observing program didn't dramatically change the way observations are done. It did, however, open the door for redesigning the system in a much more flexible way than what was ever possible with the MicroVAX-based system. In many cases the old computer prevented developing more advanced user interfaces even though there was a need for them.

In general, the observation system hasn't been the target of comprehensive usability evaluation and development. The developers and users of the system have still worked in co-operation in lessening the work load of observing. Feedback from observers has often been the catalyst for changes made in the system. It has been known that the system could be more unified, integrated, automated and usable. Due to limited resources available, the main scope of the development of the observation system has been on the lower level tools, such as the "measure" observing program and individual web tools.

2.7 Current state of the quasar observation system

In this section the various tools that form the quasar observation system are presented. These tools are also the ones that were evaluated in the usability study. All the parts of the system are presented in the form that they were in at the beginning of the evaluation and development process. As the process progressed, changes were implemented in the user interfaces. At the beginning of the usability study, the quasar observation system included the VNC desktop with multiple command line interface terminals, and a multitude of web-based tools.

The command line programs include the “screen on daqqr” terminal for running the “measure” observation program and the terminal for running the “watch_results” program for viewing live observation results. Other command line interfaces include the “offset view” terminal where a file displaying the pointing offsets by azimuth and elevation is displayed. Another terminal is used for direct antenna control with the “antcon” program. Additionally, two terminal windows are usually displayed for semi automatic and manual operation of the radome heater, respectively. The web tools include the real time sky view, the autofile generator, observation summary, observation delete and comments tool, the tau add, tau query and tau delete tools, the pointing offset table modification tool, the MB data plotting tool, the surveillance cameras page and the all-sky image gallery.

2.7.1 The VNC desktop

A central part of the observation system is the VNC remote desktop. The VNC desktop is used for controlling the telescope and many auxiliary systems. It can also be used to view web pages in Metsähovi's intranet. If needed, observations can be run from start to finish by using only the VNC desktop. The remote desktop can be accessed from practically anywhere with the help of SSH tunneling.

The VNC session runs a Ubuntu Linux with the LXDE environment with four work spaces. Typically, one workspace houses the terminal windows needed to manually control the antenna (antcon), run the observing program (measure), watch the live observation results (watch_results), control the radome heater (autoheat.pl) and view the antenna pointing offsets. All these tools are used by command line interfaces (CLI). A typical view in the main VNC desktop is depicted in Figure 2.

Below is a typical command given to the “measure” observing program:

```
> measure --freq 37 --mode 5 --source 3C84 --az-off -0.003 --el-off
0.000 --duration 1400 --observer YOURNAME
```

In the command, “freq” stands for the observing frequency. Currently there is a choice between 22 and 37 GHz, with 37 GHz being the predominantly used frequency. “Mode” stands for the type of observation. In a 1-point observation the telescope's beam is pointed at the coordinates of the source being observed and in a 5-point observation the beam is pointed at a cross-like pattern on and around the source (see 2.5.4). “Source” is the name of the source being observed. “Az-off” and “el-off” are pointing offset values used in fine-tuning the telescope's pointing. “Duration” is the total integration time that determines for how long data is collected from the source. Typical integration times vary between 1200 and 1800 seconds.

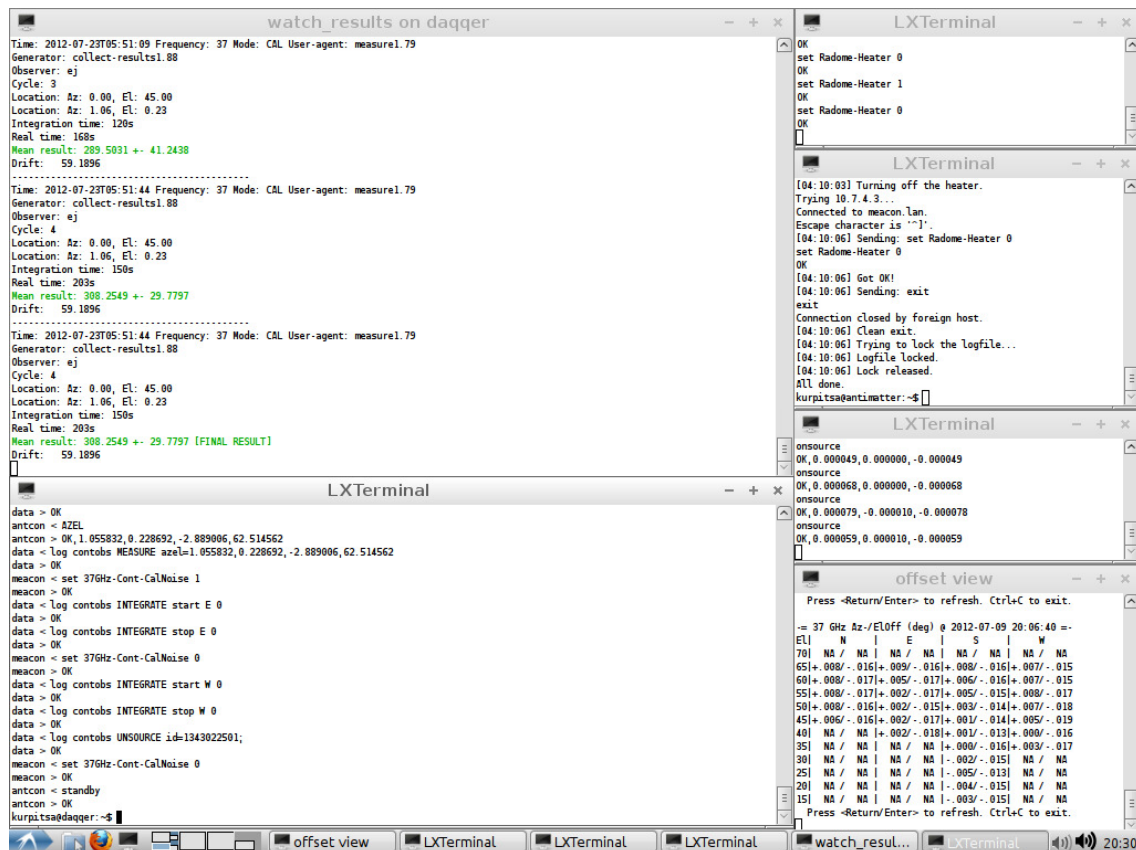


Figure 2. The main desktop of the VNC session (2012-03-07). On the top left is the terminal for watching real time observation results (`watch_results`). On the bottom left is the terminal for running the observation program (`measure`). On the right from top down are the terminals for the manual radome heater, the semi automatic radome heater script (`autoheat.pl`), the manual antenna control program (`antcon`) and the file housing the pointing offsets.

On the second workspace of the VNC desktop, a web browser is open to display weather information, observation results, sky view and autolist generator and other related tools. These web tools can be used with any web browser with access to Metsähovi's intranet, so the use of web pages in Metsähovi's intranet is not bound to the VNC desktop environment.

Because the VNC desktop has a relatively low resolution (1024x768 pixels), using some of the observation related web pages can be somewhat restricted. This is why some observers use the web browser on their local computer for viewing the web tools, especially when observing at the Metsähovi observatory site with direct access to Metsähovi's local area network. When observing remotely outside the observatory, accessing Metsähovi's network requires the use of a port forwarded SSH proxy connection to Metsähovi's intranet.

The third desktop in the VNC session is typically used for running a Matlab-based data visualization and analysis tool for solar observations, which is not considered a part of the quasar observation system. The fourth desktop has no dedicated role and it's used for doing miscellaneous tasks.

2.7.2 Observation summary and observations delete status and comments

Observation summary is a web tool that's used for displaying raw unreduced results from quasar observations together with related weather data. It also displays basic

information on what is being observed currently and who is observing. The tool is based on a server-side Perl CGI program that reads observation results from MySQL databases and prints them on an HTML table. Information related to a single observation is presented on a single row. Color coding is used to highlight e.g. high error readings and for monitoring the accuracy of the antenna pointing in 5 point observations. Observation results can be filtered by date range, source name, receiver frequency and observation ID range. The tool is used for exporting PDF and text files of the observation results that are used in final reduction of the data. The user interface of observation summary is presented in Figure 3.

The observations delete status and comments tool (obs_delcom) is used for adding comments and delete tags to observation results. It is used together with the observation summary tool. In a way, it is an interactive version of observation summary; observation summary only displays results where obs_delcom allows restricted modification of the data. Like observation summary, this tool is based on a server-side Perl CGI program. The data table presented by obs_delcom differs from observation summary's table with its higher rows and lack of cell color coding. The user interface of observation delete status and comments tool is presented in Figure 4.

Observation summary (version 1.33)

Current observation
Nobody is observing right now.

Past observations

NOTE: Empty fields don't affect search results.
Choose time range to display: and/or source: and/or frequency: and/or ID range:

ID	Start time	R	Source	az	el	aO	eO	eOC	T	Result	Err	drift	(t/rot)	CAL	Tant	Tout	RH	lan	HD	Comments	Obsv	f			
134109412	2012-06-30 23:50	X	XC345	82	52	7	-18			24.82				372.05	21.1	13.3	91.1	0.056		Tarkastelemme huolto asenn-		J	37		
134109413	2012-06-30 23:50	CAL		90	45					C	372.05	0.22	3	150	372.05	21.1	13.3	90.7	0.056			J	37		
134109414	2012-06-30 23:53	CAL		41	9	-16				-28.29	0.54	66	1206	372.05	20.7	13.4	91.1	0.056				J	37		
134109415	2012-06-30 23:55	X	Z201+315	-91	54	2	-17			13.31	0.35	77	1444	373.25	20.6	13.3	90.9	0.056				J	37		
134109416	2012-06-30 23:56	CAL		78	37		-17			37.96	0.38	51	1206	373.25	20.1	13.3	90.7	0.056				J	37		
134109417	2012-06-30 23:58	CAL		-62	64	5	-17			2.5	96.47	0.64	10	1203	373.25	20.0	13.4	90.3	0.056			J	37		
134109418	2012-06-30 23:59	CAL		90	45					C	373.25	0.21	4	151	373.25	19.9	13.3	90.3	0.056			J	37		
134109419	2012-06-30 21:27	X	2005+403	660	62	5	-17			12.74	0.73	42	1543	373.25	19.2	13.2	90.1	0.060			Seikkeytyvä.		J	37	
134109420	2012-06-30 20:55	X	DA406	32	62	6	-16			18.16	0.37	166	1445	372.85	18.4	13.5	87.1	0.060			Seikkeytyvä.		J	37	
134109421	2012-06-30 20:51	CAL		90	45					C	372.85	0.30	4	151	372.85	18.1	13.5	86.8	0.060				J	37	
134109422	2012-06-30 20:26	X	DR31	-88	52	2	-17			5	94.77	1.04	166	1205	374.83	17.6	13.6	87.5	0.060				J	37	
134109423	2012-06-30 20:20	X	CAL	-96	45					C	374.83	0.36	5	150	374.83	17.5	13.6	87.0	0.060			Kokettiin satteen jälkeen. Pöytä- 20.10+10min, ja sen jälkeen 20-40 alkaen 10 min on, 40 min off 85.20 asti.		J	37
134109424	2012-06-30 15:03	X	CAL	-0	45					C	404.16	0.41	78	60	373.69	21.5	16.9	64.9	0.060			2Sata, sett.		J	37
134109425	2012-06-30 14:32	X	X1308+326	-52	55	2	-17			9.55	1.32	186	1540	373.69	21.7	16.9	65.9	0.060					J	37	
134109426	2012-06-30 14:08	X	XC274	-38	38	0	-16			5	64.47	0.26	44	1203	373.69	21.7	17.1	63.4	0.060				J	37	
134109427	2012-06-30 14:02	X	CAL	-0	45					C	373.69	0.21	21	151	373.69	21.7	17.2	64.0	0.060				J	37	
134109428	2012-06-30 13:24	X	4C29.45	46	53	2	-17			6.50	0.60	54	1636	373.69	21.6	18.1	59.2	0.060					J	37	
134109429	2012-06-30 12:57	X	QJ287	10	56	3	-14			25.34	0.92	20	1202	375.11	21.3	17.5	65.6	0.060					J	37	
134109430	2012-06-30 12:45	CAL		90	45					C	375.11	0.41	27	151	375.11	21.4	16.8	64.8	0.058				J	37	
134109431	2012-06-30 12:21	X	3CR4	107	46	0	-16			-0.5	112.57	1.39	21	1203	375.11	21.4	16.5	66.2	0.058			Yhän ympärillä menee Zsadeptä, mutta kamerassa ei aitakaan näy sate.		J	37
134109432	2012-06-30 10:53	X	N0736+017	-5	32	-2	-15			10.79	0.43	143	1539	374.38	21.6	16.8	59.9	0.058					J	37	
134109433	2012-06-30 10:29	X	XQJ287	-42	45	1	-15			25.29	1.49	47	1204	374.38	21.5	17.1	59.3	0.058					J	37	
134109434	2012-06-30 10:04	X	3CR4	76	56	8	-17			0.5	128.38	1.10	12	1206	374.38	21.6	16.9	57.4	0.058			Yhteinen oikaisu paikannus- 2Sata. 20-20min maksaa tällä 2Sata.		J	37
134109435	2012-06-30 10:01	CAL		90	45					C	374.38	0.26	10	150	374.38	21.6	16.9	61.4	0.058				J	37	
134109436	2012-06-30 09:14	X	4C39.25	-85	56	2	-17			-0.5	39.54	0.89	12	1446	376.81	21.5	16.8	61.3	0.058				J	37	
134109437	2012-06-30 08:45	X	3CR4	55	65	7	-16			1.5	130.99	1.09	27	1203	376.81	21.3	16.7	60.5	0.058				J	37	
134109438	2012-06-30 08:42	CAL		90	45					C	376.81	0.41	3	150	376.81	21.4	16.6	58.4	0.058				J	37	
134109439	2012-06-30 08:13	X	D420-014	4	29	-2	-15			17.50	0.89	88	1396	376.81	21.1	16.4	62.1	0.058					J	37	
134109440	2012-06-30 07:45	X	TD026	9	28	-2	-15			18.39	0.78	160	1444	376.81	20.7	15.9	60.8	0.058					J	37	
134109441	2012-06-30 07:07	X	B804+499	-106	52	2	-17			5.03	1.57	54	1830	372.93	19.8	16.3	59.1	0.057			Jätkänpää & vaihtelevaa pöytä- ei kovin optimaalinen kell.		J	37	
134109442	2012-06-30 06:25	X	TA026	-13	28	-2	-15			14.37	0.41	85	1640	372.93	18.8	14.7	64.3	0.057					J	37	

Figure 3. The Observation summary tool (3.7.2012)

Observations delete status and comments (version 0.20120511)

Past observations

NOTE: Empty fields don't affect search results.
 Choose time range to display: and/or source: and/or frequency: and/or ID range:

201206300825 2012070100000

Search

ID	Start time	Source	az	el	lat	lon	freq	Result	Err	drift	dtout	f	Obsrv	D	Comments	
134100419	2012-06-30 23:55	3C345	82	52	7	-18		24.82	5.87	9	23037	JT	<input checked="" type="checkbox"/>		*Karkaussekunti hajotti antup-koneen jotenkin. -Timo* Että näin...	
134100195	2012-06-30 23:50	CAL	-90	45			C	372.05	0.22	3	15037	JT	<input type="checkbox"/>			
1341098598	2012-06-30 23:23	NGC7027	-41	69	9	-16		28.29	0.54	66	120637	JT	<input type="checkbox"/>			
1341096980	2012-06-30 22:55	2201+315	-61	52	2	-17		13.51	0.35	77	144437	JT	<input type="checkbox"/>			
1341095493	2012-06-30 22:30	BLLAC	-79	37	2	-17		37.90	0.38	51	120437	JT	<input type="checkbox"/>			
1341093805	2012-06-30 22:03	DR21	-62	64	5	1-17	2	96.47	0.64	10	120337	JT	<input type="checkbox"/>			
1341093596	2012-06-30 22:00	CAL	-90	45			C	373.25	0.21	4	15137	JT	<input checked="" type="checkbox"/>		Selkeytyvä.	
1341091560	2012-06-30 21:27	2005+403	-60	62	5	-17		12.74	0.73	42	154337	JT	<input type="checkbox"/>			
1341090645	2012-06-30 20:55	DA406	32	62	6	-16		18.16	0.37	165	144537	JT	<input checked="" type="checkbox"/>		Selkeytyvä.	
1341089452	2012-06-30 20:51	CAL	-90	45			C	372.85	0.30	4	15137	JT	<input type="checkbox"/>			
1341087998	2012-06-30 20:26	DR21	-88	52	2	1-17	-1	94.77	1.04	196	120537	JT	<input checked="" type="checkbox"/>			
1341087531	2012-06-30 20:20	CAL	-90	45			C	374.83	0.36	5	15037	JT	<input checked="" type="checkbox"/>		Kokeillaan sateen jälkeen. Poltin 20:10+10min, ja sen jälkeen 20:40 alkaen 1	
1341086872	2012-06-30 15:03	CAL	-0	45			C	404.16	21.15	78	6037	JT	<input checked="" type="checkbox"/>		Sataa, seis.	
1341086722	2012-06-30 14:32	1308+328	-52	55	2	-17		9.55	1.32	138	154037	JT	<input checked="" type="checkbox"/>			
1341085122	2012-06-30 14:06	3C274	-38	38	6	-3-16	0	64.47	2.96	46	120337	JT	<input checked="" type="checkbox"/>			
1341084907	2012-06-30 14:02	CAL	-0	45			C	373.60	2.20	21	15137	JT	<input checked="" type="checkbox"/>			
1341082618	2012-06-30 13:24	4C29.45	-48	53	2	-17		6.50	0.60	54	163637	JT	<input type="checkbox"/>			
1341081011	2012-06-30 12:57	OJ287	10	50	3	-14		23.34	0.92	20	120237	JT	<input type="checkbox"/>			
1341080283	2012-06-30 12:45	CAL	90	45			C	375.11	2.00	27	15137	JT	<input type="checkbox"/>			
1341087809	2012-06-30 12:21	3C84	107	40	6	1-16	-0	5	112.57	1.39	21	120337	JT	<input type="checkbox"/>		?ihan ympäriltä menee sadeplivä, mutta kamerassa ei ainakaan näy sade.
1341053598	2012-06-30 10:53	0736+017	-5	32	-2	-15		10.79	5.42	143	153937	JT	<input checked="" type="checkbox"/>			
1341052093	2012-06-30 10:29	OJ287	-42	45	1	-15		23.28	1.49	47	120437	JT	<input checked="" type="checkbox"/>			
1341050636	2012-06-30 10:04	3C84	78	56	8	1-17	0	5	128.38	1.10	12	120637	JT	<input type="checkbox"/>		? Pilvipelte rakoilee paikolltain, mutta Testbedin mukaan tulisi sadealue.
1341050377	2012-06-30 10:01	CAL	90	45			C	374.38	1.20	10	15037	JT	<input type="checkbox"/>		? 2	
1341047300	2012-06-30 09:14	4C39.25	-85	50	2	0-17	-0	5	39.54	0.89	12	144637	JT	<input type="checkbox"/>		
1341045914	2012-06-30 08:45	3C84	55	65	7	1-16	1	5	130.93	1.99	27	120537	JT	<input type="checkbox"/>		2
1341045670	2012-06-30 08:42	CAL	90	45			C	376.81	1.86	7	15037	JT	<input type="checkbox"/>			

Figure 4. Observations delete status and comments tool (*obs_delcom*) (24.7.2012). The tool is used in conjunction with observation summary for adding comments and delete-tags to unreduced observation results. After changes have been made to the forms on the HTML document, the changes are submitted and written to databases.

2.7.3 Real time sky view and autofile generator

The real time sky view and autofile generator are tools for displaying sources on a sky map and for creating list files of sources for automatic observations. Both of these tools are generated by the same Perl CGI server-side script. The autofile generator is depicted in Figure 5 and the real time sky view in Figure 6.

The real time sky view tool is used for displaying the current position of sources in the sky. The map automatically reloads with an updated sky map once a minute.

The autofile generator is used for creating list files containing observation commands of sources. The list files are input to a program (*auto-measure*) that gives the observation commands to the actual observation program (*measure*). Sources are added to the list by clicking them on the sky map or selecting them by name from a drop down menu. For each source added to the list, four parameters are selected: the observation type (1p or 5p), the azimuth and elevation pointing offsets and the integration time.

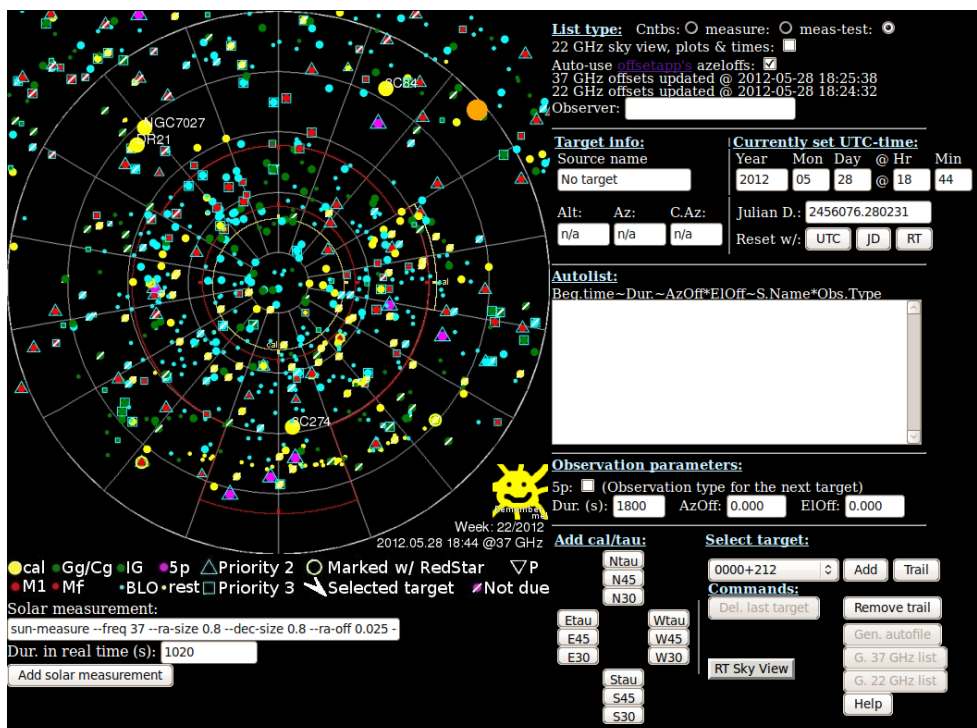


Figure 5. The autofile generator web tool (28.5.2012). The sources visible at a given moment in time are shown on the sky map on the left. When sources on the map are clicked, they are added to the list (“Autolist”) on the right. At the same time, the sky map updates to show the situation at the end of the added observation. When a list is ready for observing, a list file is exported using the “G. 37 GHz list” or “G. 22 GHz list” buttons.

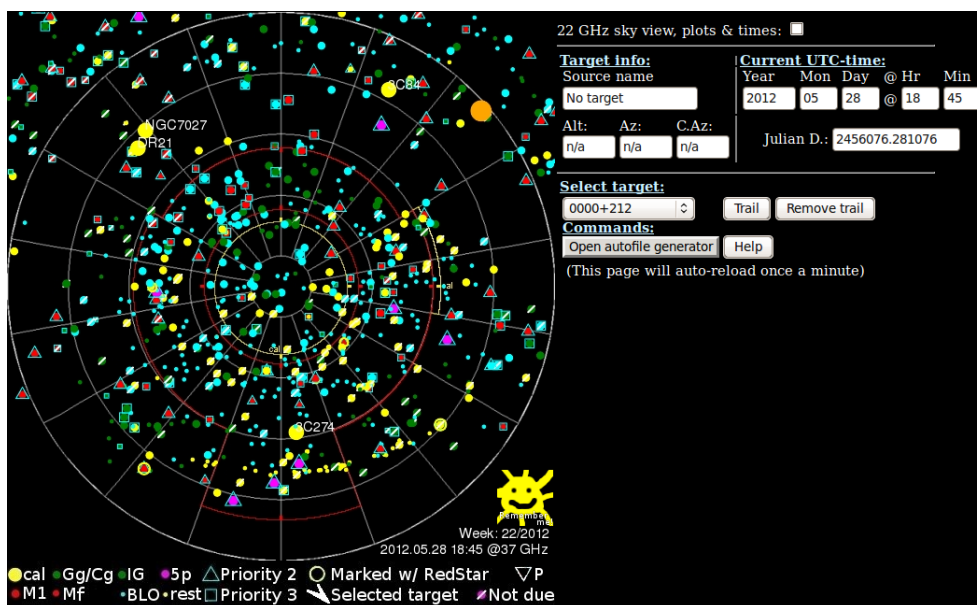


Figure 6. The real time sky view (28.5.2012). This tool displays the position of sources in the sky in real time and automatically updates once a minute.

2.7.4 Tau-tools

The tau tools are web based tools used in managing a separate database for atmospheric opacity, or tau values. The tools are used by observers to input and manage tau values for given time periods. The values are then read from the database to be integrated with other unreduced observation results in the observation summary tool (see 2.7.2). There are three separate tools for adding, finding and deleting tau values from the database.

The tool for adding new values to the database is depicted in Figure 7. The database query tool is shown in Figure 8 and the delete tool in Figure 9.

Tau database management

Start time (UTC): End time (UTC): Tau: Frequency:

Insert start and end time, tau and frequency. Date format YYYY-MM-DD-hhmm.

Status: [Insert values](#)

Input information:

Begin date:

End date:

Tau:

Frequency:

Status: Initializing... Input failed, check your dates and/or tau value

Time (UTC): 2012-07-03 12:05

Figure 7. The Tau-add tool (3.7.2012)

Tau database query

Start time (UTC): End time (UTC): Frequency:

Insert start and end time and frequency to search tau values in a time period. Leave end time empty to search for a tau value to a specific date.

Input information:

Previous begin date input:

Previous end date input:

Frequency:

Current time (UTC): 2012-07-03 12:15

Tau values in database

Use exact search date.

Figure 8. The tau query tool (3.7.2012)

Tau database ID search and delete

ID:

Insert ID to search for a tau value. If a value is found a delete button will appear.

Status: **ID does not exists**

Figure 9. The tau delete tool (3.7.2012)

2.7.5 MB data plotting tool

The MB data plotting tool is used for plotting graphs of weather and other data available in Metsähovi's network. MB's main user interface is presented in Figure 10 and an example of the graphs generated by MB is presented in Figure 11.



Metsähovi Local Pages

MB

**Following non-astronomical data can be viewed graphically.
Please choose first data types and a time period**

Data source(s)

- adam6017 - misc sensor
- antup-variables - Variables from antd process on antup
- celdam - H Maser Cellar ADAM/NuDAM module network
- clodi - Clock Difference Counter 10ch (uncalibrated raw counter values)
- clodi5t - (test data from kinukki) Clock Difference Counter (uncalibrated raw counter values)
- cloud - Cloud sensor
- contobs-daqqer - debug data generated when contobs is running
- dammer - control room ADAM/NuDAM module network
- daq-ch1-beam-diff - 1-second averages of beam difference using 37 GHz receiver
- daq-ch2-beam-diff - 1-second averages of beam difference using 22 GHz receiver
- hmaser - (not in use) GPS/H-maser clocks compared to H-maser Kvarz-69
- hmaser5t - (not in use) experimental, clodi5t data
- maturun test - Experimental tau-calculations
- datalogd - radome oil level sensor
- sundam - SunAnt ADAM/NuDAM module network
- sundaq - SunAnt data
- timediff - calibrated, averaged time difference between master maser and other PPS sources
- Vaisala WXT520 - Weather Station (High)
- Vaisala WXT520 - Weather Station (Low)
- GroWeather - Weather Station (Middle)

Time period

- Define a time period
- Last 24 hrs
- Last 3 days
- Last one week

To continue

2-define element parameters	reset values
3-define general plotting parameters	save settings to file
4-define picture parameters	load settings from file
5-make plot	

Please, send Your remarks and suggestions to: webmaster@kurp.hut.fi

Figure 10. The front page of the MB plotting tool.

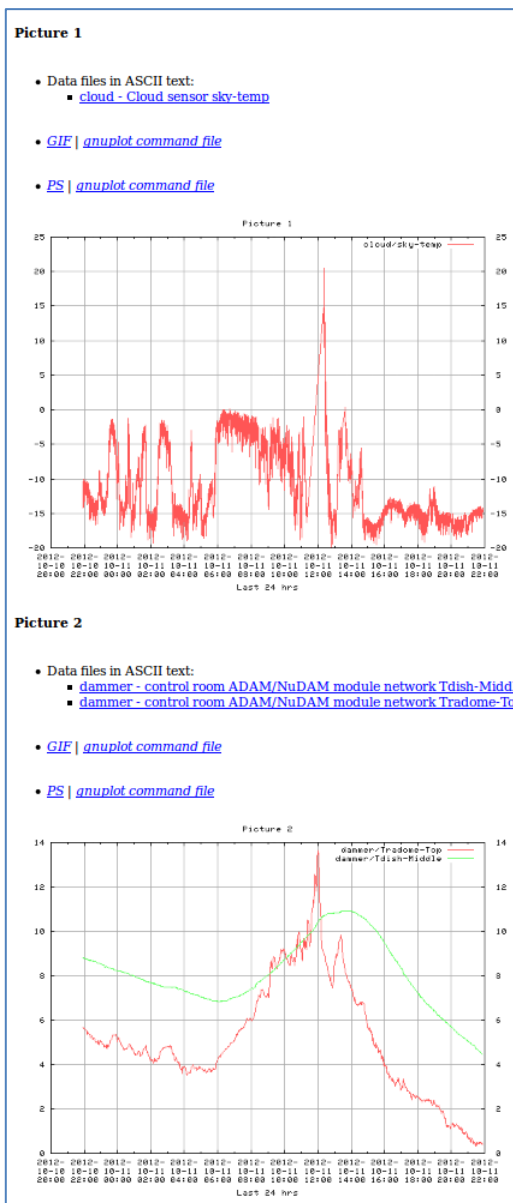


Figure 11. An example of the graphs generated by the MB plotting tool.

2.7.6 The pointing offsets tool

The Perl CGI based pointing offset tool is used for viewing and updating information on the pointing offset values of the telescope in various azimuth/elevation directions and with two different frequencies. The user interface of the tool is depicted in Figure 12.

Current offsets

Choose frequency: GHz

37 GHz:

Azimuth & elevation offset table last updated **UTC 2012-10-03 20:41:29**; offsets in degrees.
 A change done **>8 weeks >4 weeks >2 weeks <2 weeks** ago.
 Tick a check box if you want to Update the corresponding time stamp, but not change the offset values.

Elevation (°)	N			E			S			W		
	Az	EI	U	Az	EI	U	Az	EI	U	Az	EI	U
70	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>
65	+0.020	-0.017	<input type="checkbox"/>	+0.019	-0.014	<input type="checkbox"/>	+0.016	-0.015	<input type="checkbox"/>	+0.026	-0.015	<input type="checkbox"/>
60	+0.020	-0.016	<input type="checkbox"/>	+0.018	-0.014	<input type="checkbox"/>	+0.016	-0.015	<input type="checkbox"/>	+0.026	-0.015	<input type="checkbox"/>
55	+0.020	-0.016	<input type="checkbox"/>	+0.017	-0.015	<input type="checkbox"/>	+0.017	-0.014	<input type="checkbox"/>	+0.023	-0.013	<input type="checkbox"/>
50	+0.016	-0.016	<input type="checkbox"/>	+0.016	-0.016	<input type="checkbox"/>	+0.017	-0.014	<input type="checkbox"/>	+0.018	-0.015	<input type="checkbox"/>
45	+0.016	-0.016	<input type="checkbox"/>	+0.014	-0.017	<input type="checkbox"/>	+0.016	-0.014	<input type="checkbox"/>	+0.019	-0.019	<input type="checkbox"/>
40	NA	NA	<input type="checkbox"/>	+0.012	-0.018	<input type="checkbox"/>	+0.016	-0.014	<input type="checkbox"/>	+0.012	-0.012	<input type="checkbox"/>
35	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	+0.015	-0.015	<input type="checkbox"/>	+0.012	-0.010	<input type="checkbox"/>
30	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	+0.015	-0.015	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>
25	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	+0.010	-0.013	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>
20	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	+0.010	-0.013	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>
15	NA	NA	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>	+0.010	-0.013	<input type="checkbox"/>	NA	NA	<input type="checkbox"/>

Figure 12. The pointing offset tool (4.10.2012)

2.7.7 The surveillance cameras

There are six surveillance cameras in use at Metsähovi. The cameras are regularly used by observers to follow changes in observing conditions. An image of the web based surveillance cameras view is presented in Figure 13.

In May of 2012 Metsähovi acquired an all-sky camera for the monitoring of weather conditions. The camera points upwards to the sky and has a fisheye lens giving it a nearly full view of the sky above the observatory. Images from the camera are saved every 15 minutes to give observers the possibility to see how the weather conditions have changed during their observations. Quickly after the camera was taken into operational use, a Perl CGI based web gallery was set up to allow browsing the all-sky images. The user interface of the gallery is presented in Figure 14.

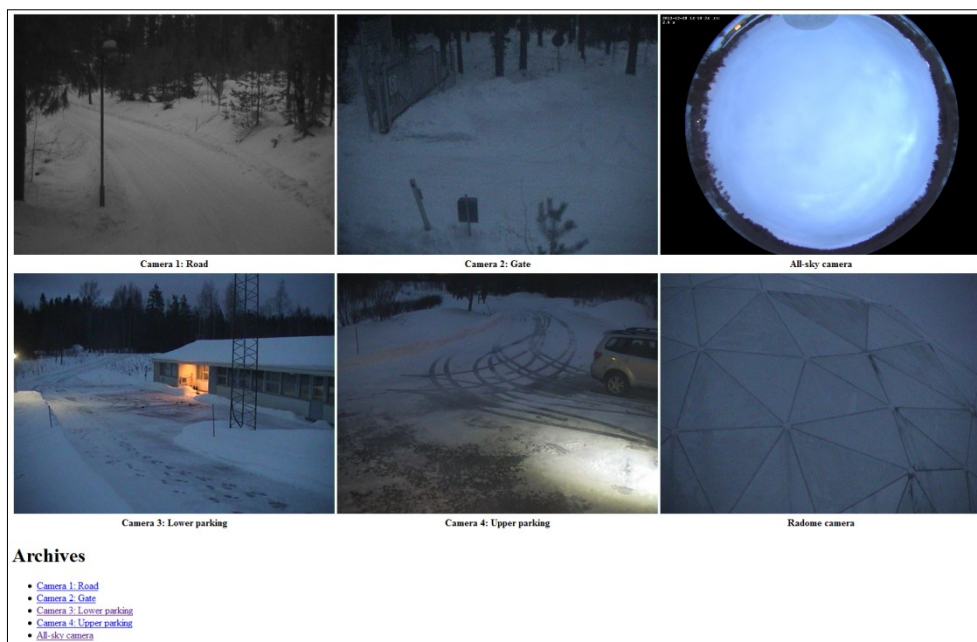


Figure 13. The web based surveillance cameras view (2012-12-28).

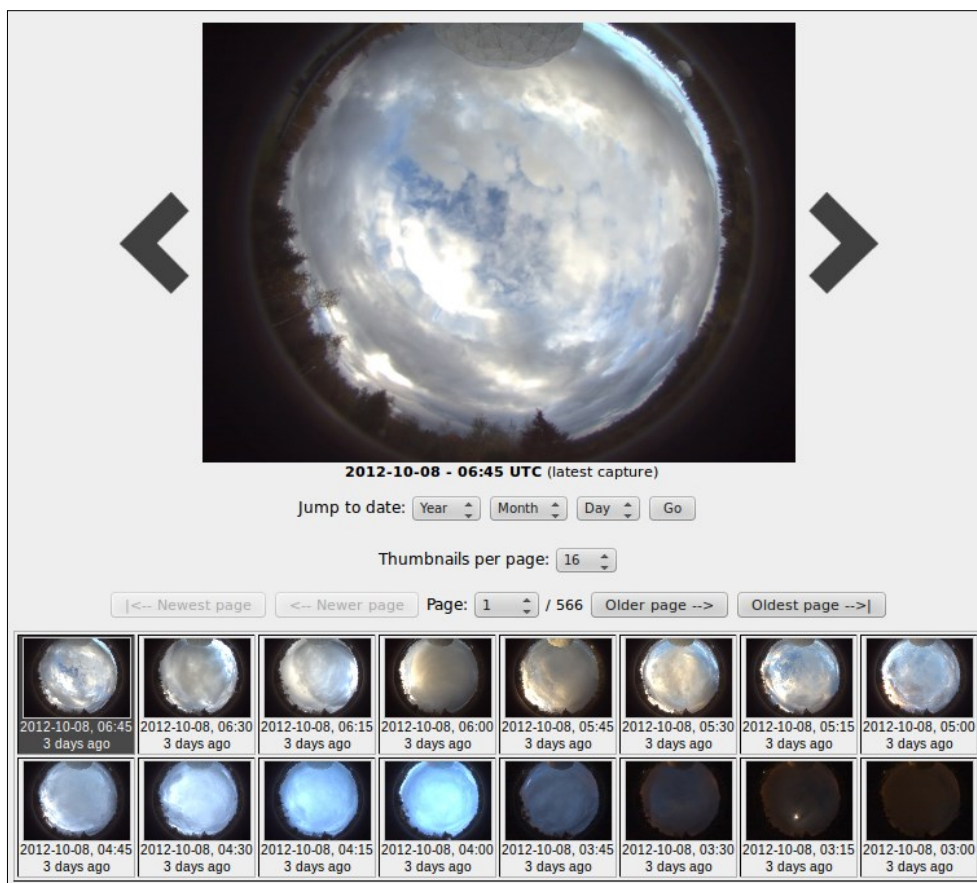


Figure 14. The web based all-sky image archive (2012-10-08).

2.8 Users

The users of the quasar observation system, or observers, as they are usually called, are mostly researchers and undergraduate or post graduate students of astronomy and engineering sciences. Metsähovi's staff also comprises of technical and support staff

that doesn't conduct quasar observations. In 2012 there were 10 active observers in Metsähovi.

Overall, the observers typically have above average computer skills and an academic background. Changes can stem from experience with astronomical observations (e.g. coordinate systems), familiarity with the science related to the sources and knowledge on radio receiver technology. Due to the international nature of research in Metsähovi, observers can represent many different countries and cultures. In recent years there have been observers from Finland, India, Spain and Russia in the observatory.

The observers can be divided into four different groups based on their education and experience with quasar observations: senior researchers, researcher observers, young observers and novice observers. Each group can have a slightly different view on the observations.

The most experienced observers in Metsähovi are the two senior scientists of the quasar research project. One of them acts as the director of Metsähovi and the other is the coordinator for the observatory's quasar observations and data handling. The senior observers started their observing career in the late 1980's or early 1990's. Both the senior observers have doctorates and have worked for over 20 years as researchers in the quasar monitoring project. They have used the data from the observations in their own research and one of them routinely reduces the observation data into its final form. They are intimately familiar with many of the sources in Metsähovi's source list and can often determine suitable observing times based only on the sources' coordinates and time of the year. They also remember the coordinate names and aliases for many sources.

The senior observers have seen the early days of the observation system when the observers had to conduct observations in situ at the observatory and manage almost every part of the observations manually. Their view on the usability of the system can be affected by these previous experiences. They may be less sensitive to minor usability problems and aspects of the observation work that require extra effort. From their point of view the current observation system is a massive improvement over what was in use 20 years ago. Their extensive observation experience and scientific knowledge gives them a comprehensively wide and deep view on the work flow and structure of making quasar observations. The senior observers are most likely to possess valid knowledge on how to conduct quasar observations.

A second group of observers can be formed of the younger researchers and graduate students in Metsähovi. There are currently three of these researchers who can be described as expert observers. Like the senior observers, the expert observers have been involved in at least some astronomical research and may have a different view on the observations than people with no background in research. The expert observers have over 5 years of observing experience. The expert observers can generally be expected to have extensive knowledge of the observations, e.g. the effects of weather and using the dome heater etc. However, some expert observers may carry with them some old misconceptions and habits that potentially cause sub-optimal results. Because of their experience, the expert observers may rely on their old habits and not update their knowledge often enough.

A third group of observers consists of observers with observing experience of about 1-5 years. These "young" observers are typically students of engineering and of astronomy. The defining factor between the expert and young user groups is the difference in research and observing experience. Compared to the expert observers the young

observers have learned the observation routine more recently and may thus have more up to date knowledge. However, the young observers may also have gaps in their knowledge due to their limited experience. They may not be aware of all the factors affecting the observations that the more experienced observers hold as self evident. Due to their relatively short exposure to the observation system, the young observers can generally be thought of being fairly sensitive to problems in the observation system and in the observing work. However, they may already have accepted some peculiarities as features of the system rather than as problems.

The fourth observer group is formed by the novice observers who have recently started making observations. The number of novice observers varies from year to year. Usually there are at most three novice observers in Metsähovi at any given time. A typical novice observer is a summer employee student hired mainly for observing duties. Last such student observers arrived in Metsähovi in 2009 (2 observers) and 2010 (2 observers). Graduate and post-doc students arriving to work in Metsähovi make observations and they too can be classified as novice observers in the beginning of their observation career. During the summer and fall of 2012 there was one such novice observer in Metsähovi. Novice observers can be considered to have made the transit to being young or researcher observers when they have gained a routine in making observations and have learned the most important factors affecting observations. Depending on the time spent observing and personal characteristics of the observer, this may take anything from a month to a year.

Even though the main goal for all the observers is the same, the production of good quality data, the different groups have different priorities and focus while conducting observations. The senior observers may have their minds set more on the use of the reduced data and e.g. the proper frequency of gathering data points for single sources based on their past behavior. They may adjust their use of the dome heater and of the integration times based on deeper understanding of the weather conditions than the younger observers. The younger observers may be more inclined to “get the job done”, that is, to observe sources based on more immediate and superficial goals. These goals may include observing sources based on the recent observation history and guidelines given by the senior observers.

Put simply, the seniors possess the knowledge on exactly why things are done the way they are done. The less experienced observers may do things more based on the way they have been told to do them. The challenge in teaching novice observers the basics is to get the silent knowledge from the senior observers passed on to the novices. From experience it can be said that this requires a lot of questions, interpretation and follow-up questions. This is, however, only the personal interpretation by the author of this thesis. To gain more accurate knowledge of the users, they have to be observed while they do their work. The contextual inquiries conducted in the usability study are a good source of such knowledge.

2.9 Use context

The observations are conducted using mostly the VNC desktop and the internal web pages of the observatory. The observations are almost always done using a desktop or laptop computer. It is also in principle possible to access the required observatory assets by using a mobile device such as a tablet computer or a smart phone. This possibility is as of yet very rarely exploited. Most mobile device use cases might include checking

the status of ongoing observations or giving single commands to the system to e.g. stop the observations through a terminal.

The environment in which observations are made varies due to the possibility to conduct remote observations. The most typical environments are the observatory, where most observers do their daily work, and the observers' homes. Some more exotic use environments where observations are known to have been conducted include a moving boat in the lake Saimaa, a summer cottage and a training field for search and rescue dogs.

When developing the observation system, the fact that the system is used remotely and sometimes through mobile internet connections, has to be taken into account. Some concrete limitations are especially evident; the bandwidth of mobile internet connections can be sometimes very low and the network latency very high. Additionally, the screen resolution of small laptop computers can be very low, e.g. 1024x600 pixels. These limitations set challenges for the observation system. The user interfaces have to be usable even with small resolution screens and with low performance network connections.

2.10 Development goals and known problems

The main goal for Metsähovi's quasar monitoring is to produce high quality data from the right sources at the right time. Because the observatory's staff runs observations in shifts parallel with other work, minimizing the staff's work load without loss of data quality is the main goal for the development of the observation system. This means reducing time wasted on completing routine tasks and making routine tasks more pleasant. In the long term, it is hoped that a more unified, flexible, usable and maintainable observation system can be developed. The adoption of modern Linux based low level observing software has been a major step toward such a system. The next major step is conducting the usability study in this thesis.

Metsähovi's management wished for the usability study to produce practical solutions for improving the quasar observation system. The goal was to not only to point out problems and give suggestions but to implement some of the solutions in practice.

The observatory's senior scientists don't wish to completely automate the observing routine. Human oversight of the observations is to continue in the future so that there is always someone personally responsible for the data produced at any given time. The avoidance of complete automation does not mean that many parts of the observing work shouldn't be automated and made easier. The observatory's director summed up her view by stating that everything that can easily be done by a human should be left for a human. Even with automated functions there should be a manual option available.

One of the more specific development goals originated from Metsähovi's senior researchers is the creation of an autonomous observing option. It would include adding automation to the observation system in such a way that it would be possible to tell the system to observe sources on its own for a given period of time. The system would automatically select sources for observing according to source priorities and coordinates. Behind this goal is the problem that the senior researchers are sometimes so busy with other duties that they can't concentrate on selecting sources for observing. This kind of a situation can arise e.g. after a rapid improvement in weather conditions.

This goal isn't meant to lead to the telescope becoming totally autonomous. Rather, the observers' work load would decrease when it comes to selecting sources for observing.

This gives an alternative to not observing at all and to making a very hasty job at selecting sources. The aim is to end up with a system that will generate data better in an autonomous mode than a completely novice human observer, but that will also allow totally manual control for expert observers.

Another goal is related to the assurance of high data quality, which is considered the most time consuming and highest workload step in the data production pipeline. According to the observatory's director, a long term goal in this sector is to generate final reduced data quickly from the preliminary results data without the current high workload final reduction. It is hoped that this can be achieved by better utilizing the sensor data available and with new hardware.

The single highest workload inducing thing is the usage of the high power dome heater which causes extreme jumps in temperature in the dome and in the receiver. This induces high workload in the final reduction as the reduction coefficients jump according to the jumps in temperature. To reduce the need to use the high power heater in the future, two lower power electric heaters have been acquired. They are to keep the dome's temperature steadily higher than the outside temperature. This would be enough to eliminate moisture on the dome and would not cause the dome's temperature to spike like with the high power heater. The high power heater could be left for melting snow and for warming the dome up for maintenance.

3 Theoretical background

This chapter presents the concept of usability, gives examples on methods of usability evaluation and gives an introduction on user centered design (UCD). This chapter also presents the principles of user centered design and gives examples of user centered design processes. An introduction is also given on different types of user interfaces and on the technologies used in web user interfaces, which form a central part of the quasar observation system.

3.1 Summary

In this thesis, Metsähovi's development goals were approached from the perspective of user centered design. In short, user centered design is an approach to designing interactive (computer) systems by concentrating on the needs of the end users (SFS-EN ISO 9241-210, 2010). Usability is a term that can be defined as the extent to which a product can be used by specific users to achieve specified goals in a specified context of use with effectiveness, efficiency and satisfaction (SFS-EN ISO 9241-11, 1998).

The SFS-EN ISO 9241-210 standard defines an overall framework for a user centered design process. It does not give suggestions on specific methods. They have to be selected based on the kind of data that is needed in the case at hand.

Methods of usability evaluation can be divided into expert evaluation methods and evaluations with users (SFS-EN ISO 9241-210, 2010). Examples of expert evaluation methods are heuristic evaluation and cognitive walkthrough. Usability testing is an often used and effective evaluation method involving users. Other methods involving users include questionnaires and interviews.

Computer user interfaces can be divided roughly into Command Line Interfaces (CLI) and Graphical User Interfaces (GUI). Currently the GUI is a widely used interface type and Windows, Icons, Menus and Pointing device (WIMP) is a widely accepted paradigm for desktop GUI's. The advances in *web user interface* technologies such as Dynamic HTML (DHTML) and Asynchronous JavaScript and XML (AJAX) has allowed introducing interactive features to web based tools that were previously only associated with desktop applications.

To design user interfaces, there are numerous usability *guidelines* available for different platforms. More general usability *principles* are also helpful when striving toward usable and consistent user interfaces.

In addition to the SFS-EN ISO 9241-210 standard, there are many user centered design processes available for evaluating and creating usable computer systems. Examples of such processes are Contextual Design by Beyer and Holtzblatt (1998) and the Usability Engineering Lifecycle (Mayhew, 1999). These processes offer possibilities for flexibly creating customized processes to suit diverse development goals.

3.2 Usability

Usability is a term that in the last ten years has been used increasingly in product reviews, in the media, in marketing and advertising and in everyday conversations between people. Typical examples of usability's visibility in the media are the problems of the electronic voting trial in Finland and the problems Nokia had with its Symbian

smart phones. In both these example cases at least part of the problem was the lack of adequate usability, resulting in lost votes and dissatisfied customers. A randomly selected person might define usability as “ease of use” or as “user-friendliness” of a device or a piece of software. This kind of a definition is somewhat descriptive of the general idea behind the term, but professionals of user-centered design (UCD) and of human computer interaction (HCI) need more precisely defined descriptions. As it turns out, usability consists of multiple components that can be accurately defined and measured. This is crucial in the systematic user centered development of a multitude of devices and software systems.

Usability has many different definitions created by standardization organizations and individuals, among others. Two often used definitions are the SFS-EN ISO 9241-11 standard (1998) and Jacob Nielsen's (1993) definition. The SFS-EN ISO 9241-11 standard defines usability as the extent to which a product can be used by specific users to achieve specified goals in a specified context of use with effectiveness, efficiency and satisfaction. The standard defines effectiveness as accuracy and completeness with which users achieve specified goals. Efficiency is specified as resources expended in relation to the accuracy and completeness with which users achieve goals. Satisfaction is defined as freedom from discomfort and positive attitudes towards the use of the product. Context of use comprises the physical and social environment in which a product is used: users, tasks, hardware, software and material equipment.

According to Jacob Nielsen (Nielsen, 1993), a well known “guru” in the field of human computer interaction, usability is defined by five quality components: learnability, efficiency, memorability, errors and satisfaction. Learnability defines how easy it is for a user to accomplish basic tasks when using a system for the first time. Efficiency describes how fast users can perform tasks with a system once they are familiar with it. Memorability defines how quickly users can regain their competence in using a system after not having used it in a while. Errors determines the number and severity of errors users make while using a system and how easily users can recover from them. Satisfaction measures the pleasantness of using a system. Nielsen classifies usability as a small part of a larger hierarchy of system acceptability which is depicted in Figure 15.

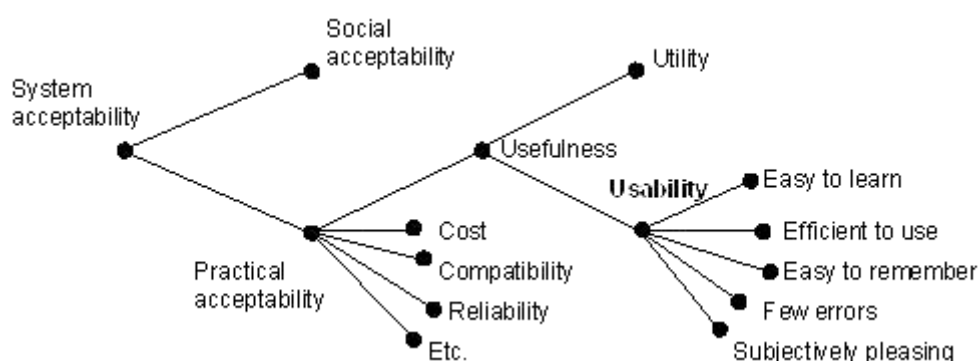


Figure 15. A model of the attributes of system acceptability (Nielsen, 1993)

3.3 User centered design

The term user centered design (UCD) is defined in the SFS-EN ISO 9241-210 (2010) standard. It states that user centered design, or human-centered design, is an approach to interactive systems development that aims to make systems usable by focusing on the users, their needs and requirements and by applying human factors/ergonomics, and

usability knowledge and techniques. The standard gives an overview of the user centered design process and the principles to follow. It does not tell which exact methods to use.

Below is a list of benefits of user centered design followed by the principles of user centered design as presented in SFS-EN ISO 9241-210.

Systems designed using human-centered methods improve quality, for example by:

1. Increasing the productivity of users and the operational efficiency of organizations.
2. Being easier to understand and use, thus reducing training and support costs.
3. Increasing usability for people with a wider range of capabilities and thus increasing accessibility
4. Improving user experience.
5. Reducing discomfort and stress.
6. Providing a competitive advantage, for example by improving brand image.
7. Contributing towards sustainability objectives.

Principles of user centered design (SFS-EN ISO 9241-210, 2010):

1. The design is based on an explicit understating of the users, tasks and environments.
2. Users are involved throughout design and development.
3. The design is driven and refined by user-centered evaluation.
4. The process is iterative.
5. The design addresses the whole user experience.
6. The design team includes multidisciplinary skills and perspectives.

Figure 16 presents the main parts of the user centered design process of the SFS-EN ISO 9241-210 standard.

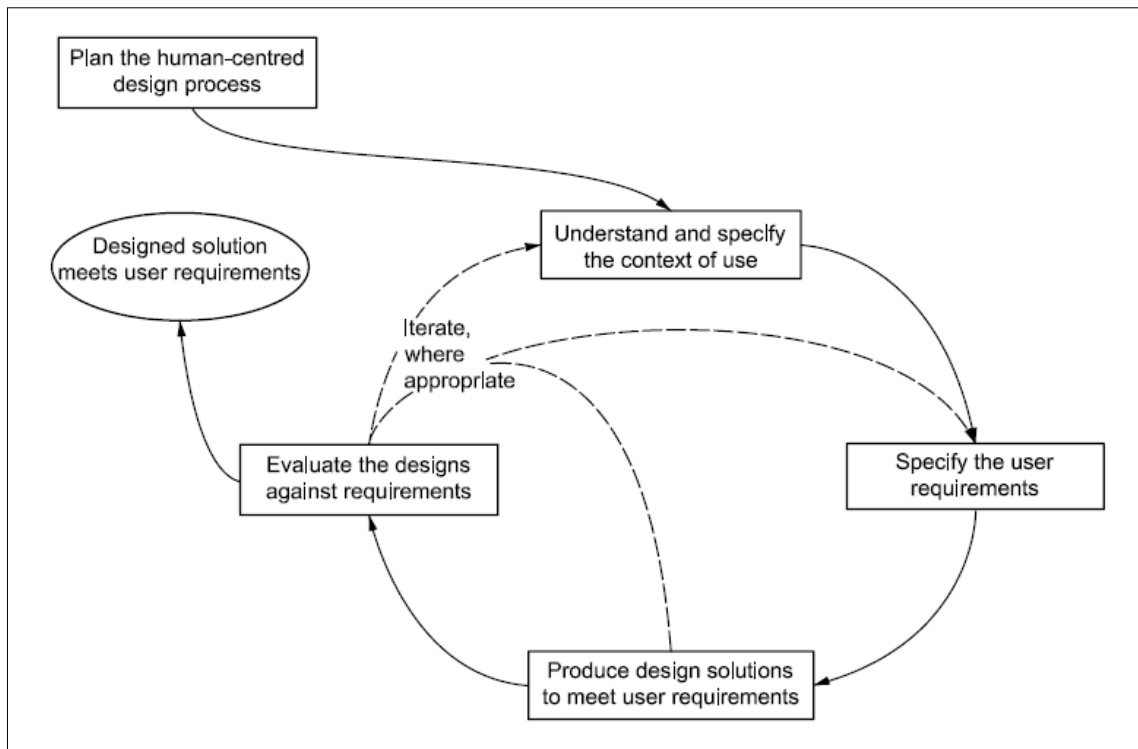


Figure 16. The process of user centered design (SFS-EN ISO-9241-210, 2010).

The process starts with understanding and specifying the context of use. The context of use description should include four components:

1. Identifying the users and other stakeholder groups and their key goals and constraints.
2. Identifying the characteristics of the users or groups of users, e.g. their knowledge, skill, experience, education, training, physical attributes, habits, preferences and capabilities.
3. Identifying the goals and tasks of the users.
4. Identifying the environment(s) of the system, including the physical (e.g. spatial layout, lighting), social and cultural (e.g. work practices, organizational structure, attitudes) environments.

The next phase is specifying the user requirements. The specification should include the following parts:

1. The intended context of use.
2. Requirements derived from user needs and the context of use.
3. Requirements arising from relevant ergonomics and user interface knowledge, standards and guidelines.
4. Usability requirements and objectives, including measurable usability performance and satisfaction criteria in specific contexts of use.
5. Requirements derived from organizational requirements that directly affect the user.

The third main phase is the production of design solutions that should include the following sub-activities:

1. Designing user tasks, user-system interaction and user interface to meet user requirements, taking into account the whole user experience.
2. Making the design solutions more concrete (e.g. making user of scenarios, simulations, prototypes or mock-ups).
3. Altering the design solutions in response to user-centered evaluation and feedback.
4. Communicating the design solutions to those responsible for their implementation.

The fourth phase in the process is evaluating the designs against user requirements. This phase should involve the following:

1. Allocating resources both to obtain early feedback in order to improve the product and, at a later stage, to determine whether the requirements have been satisfied.
2. Planning the user-centered evaluation so that it fits the project schedule.
3. Carrying out sufficiently comprehensive testing to provide meaningful results for the system as a whole.
4. Analyzing the results, prioritizing issues and proposing solutions.
5. Communicating the solutions appropriately so that they can be used effectively by the design team.

3.4 User interfaces

A user interface is a way to allow communication between a human and a machine. Mayhew (1999) defines a user interface as "...the languages through which the user and the product communicate with one another." Mayhew divides the communication between the user and the system into the application-to-user language and the user-to-application language. In a software application, the application-to-user language would consist of the design of the displays and feedback. The user-to-application language would consist of the interactions the users have with the display elements using input devices such as the mouse and the keyboard, indicating to the software what they want to accomplish.

In the broader sense of the term, there are user interfaces outside the computer world as well. Basically all control devices of machines can be considered as user interfaces. A door handle is the user interface of a door and a steering wheel is a part of the user interface of a car and so on. The user interfaces of mechanical systems, however, are typically classified in the fields of man machine interfaces (MMI) and human factors rather than in human computer interaction. In this thesis the focus is on user interfaces of computer systems.

Computer user interfaces can be divided into a number of main types. A high level division can be made between graphical user interfaces (GUI) and command line interfaces (CLI). In a GUI, users interact with a system using graphics and images, whereas a CLI is based solely on textual input and output. The GUI is often tied to the concept of direct manipulation that was first introduced by Ben Shneiderman in 1983 (Shneiderman, 1983). In direct manipulation interfaces, users typically interact with graphical representations of dialog objects presented on a screen (Nielsen, 1993).

Historically, user interfaces can be classified into three categories. Raymond and Landley (2004) point out three eras in the history of user interfaces. The era of batch computing spans the years 1945-1968, the era of command line interfaces (CLI) years 1969-1983 and the era of graphical user interfaces (GUI) spans years from 1984 onwards.

Raymond and Landley describe early batch computing as a non-interactive process where the user typically used a deck of punch cards to feed a program and a dataset to a batch machine. The commands were executed from a queue, and the total turnaround time could be hours or days. The first Command-line interfaces (CLI) were developed from batch monitor programs connected to a system console. Interaction was done using request-response transactions. The requests were textual commands given in a specific vocabulary. Today, there are many kinds of CLI's in use, from classic operating system terminals in Unix and Linux to the address bars of web browsers. According to Butow (2007), the need to learn and remember the commands to give the computer is the biggest drawback of CLI's.

3.4.1 WIMP

Also closely related to GUI, is the term WIMP, which stands for Windows, Icons, Menus and Pointer. WIMP is an interaction style that has been in use in desktop computers for the last 30 years. The first WIMP interface was introduced in the Xerox Alto computer in 1974 (Galitz, 2007). The Alto can also be regarded as the first personal computer equipped with a GUI (Butow, 2007).

Post-WIMP is a term used to describe interfaces that go beyond the WIMP paradigm (van Dam, 1997). Smart phone touch user interfaces and the user interfaces modern televisions are examples of post-WIMP interfaces.

3.4.2 Web user interfaces

Dynamic HTML or DHTML is a set of technologies based on HTML, JavaScript, Cascading Style Sheets (CSS) and the document object model (DOM). The principle of DHTML is that the elements of an HTML document are made available for modification with scripts through the DOM. This allows locally altering the appearance of a web page after it's been loaded on the web browser.

AJAX, which stands for Asynchronous JavaScript and XML, is a set of technologies for creating web applications. The term AJAX was first used in 2005 by Jesse James Garrett (2005). A central part of AJAX is the XMLHttpRequest (XHR) object, first requested to be developed into a web standard by the World Wide Web (W3C) consortium in 2006 (W3C, 2006). The traditional approach for web user interfaces has been based on reloading the HTML document on every user action. AJAX allows updating single parts of the HTML document, in a similar way as in a desktop user interface. Using the XMLHttpRequest allows retrieving data from the web server on the background without reloading the client's web page. This opens up possibilities for much more interactive web user interfaces.

3.4.3 User interface design

On what basis does one design and come up with user interfaces? How are real life problems translated to computer screens? These questions can be addressed with user interface principles, guidelines and models. Shneiderman and Plaisant (2005) present guidelines and principles that aid in designing user interfaces. Guidelines are generally

more specific in nature than principles. There are numerous different sets of guidelines available for different platforms. Examples of widely used guidelines are Microsoft's Windows User Experience Interaction Guidelines (Microsoft, 2013) and Apple's OS X Human Interface Guidelines (Apple, 2012). There are also sets of guidelines available for the Web and for various mobile platforms. Guidelines typically give specific instructions on how to create consistent user interfaces for a given platform. Shneiderman and Plaisant (2005) list four examples of topics which guideline documents often give advice on: navigation, organizing the interface, getting the user's attention and facilitating data entry.

User interface design principles are more general and more platform independent than guidelines. Shneiderman and Plaisant (2005) list the following principles:

1. Determine the users' skill levels.
2. Identify the tasks.
3. Choose an interaction style:
 - a. Direct manipulation.
 - b. Menu selection.
 - c. Form fill-in.
 - d. Command language.
 - e. Natural language.
4. Use the eight golden rules of interface design:
 1. Strive for consistency.
 2. Cater to universal usability.
 3. Offer informative feedback.
 4. Design dialogs to yield closure.
 5. Prevent errors.
 6. Permit easy reversal of errors.
 7. Support internal locus of control.
 8. Reduce short term memory load

Another important set of principles used in user interface design are the Gestalt laws of grouping (or the Gestalt principles of grouping). The Gestalt principles are a product of the Gestalt psychology movement that was launched in Germany in 1912 by Max Wertheimer. The Gestalt principles argue that the human mind perceives objects as organized patterns that in essence are more than the sum of their parts. There are four original principles by Wertheimer that have later been supplemented by Rock and Palmer. (Rock, Palmer, 1990) The original four principles are summarized below:

1. **Law of proximity:** A group of stimuli that are close together tend to be perceived as a part of a common object.
2. **Law of similarity:** A group of stimuli that physically resemble each other tend to be perceived as a part of a common object.
3. **Law of closure:** Incomplete figures tend to be perceived as wholes. The mind fills in the gaps to come up with familiar patterns when incomplete information is available.
4. **Law of good continuation:** Intersecting objects tend to be perceived as individual uninterrupted objects. Lines or curves following an established

direction tend to be grouped and organized more easily than those that have sudden corners or changes in direction.

There are many user interface models available. Foley et al. (1995) present a four level interface model:

1. Conceptual level.
2. Semantic level.
3. Syntactic level.
4. Lexical level.

The model's top down approach is convenient for designers due to good match with software architecture. This model was effective when command-line input was common. Currently it is starting to show its age due to the reduced need to implement syntactic and lexical levels (Shneiderman, Plaisant, 2005). Norman's (1988) seven stages of action model approaches human computer interaction through the stages of action that users go through when using interactive products:

1. Forming the goal.
2. Forming the intention.
3. Specifying the action.
4. Executing the action.
5. Preserving the system state.
6. Interpreting the system state.
7. Evaluating the outcome.

Norman's model places the stages in the context of cycles of action and evaluation. The first stage is for setting a goal, the three next ones for executing and the three last ones for evaluation. Norman also introduces the concepts of the gulf of execution and the gulf of evaluation. The gulf of execution describes the difference in the intentions of the user and the available actions in the system. The gulf of evaluation describes the effort needed for the user to interpret the physical state of the system to determine if the expectations and intentions have been met. In other words, the gulf of evaluation is about the difference between the feedback the system provides and how well the user is able to interpret it.

Based on the model, Norman (1988) presents four principles of good design:

1. **Visibility.** By looking, the user can tell the state of the device and the alternatives for action.
2. **A good conceptual model.** The designer provides a good conceptual model for the user, with consistency in the presentation of operations and results and a coherent, consistent system image.
3. **Good mappings.** It is possible to determine to relationships between actions and results, between the controls and their effects, and between the system state and what is visible.

4. **Feedback.** The user receives full and continuous feedback about the results of actions.

Shneiderman's (Shneiderman, Plaisant, 2005) Object Action Interface (OAI) model describes a user's work as objects and actions that have been mapped from the real world into the computer system. The user acts on (typically visual) representations of objects presented on the screen. Object-action design starts with understanding the real-world tasks and objects the users work on. After the real-world tasks and objects have been decomposed into atomic units, metaphoric representations of them are created into interface tasks and objects. Figure 17 presents the OAI model graphically.

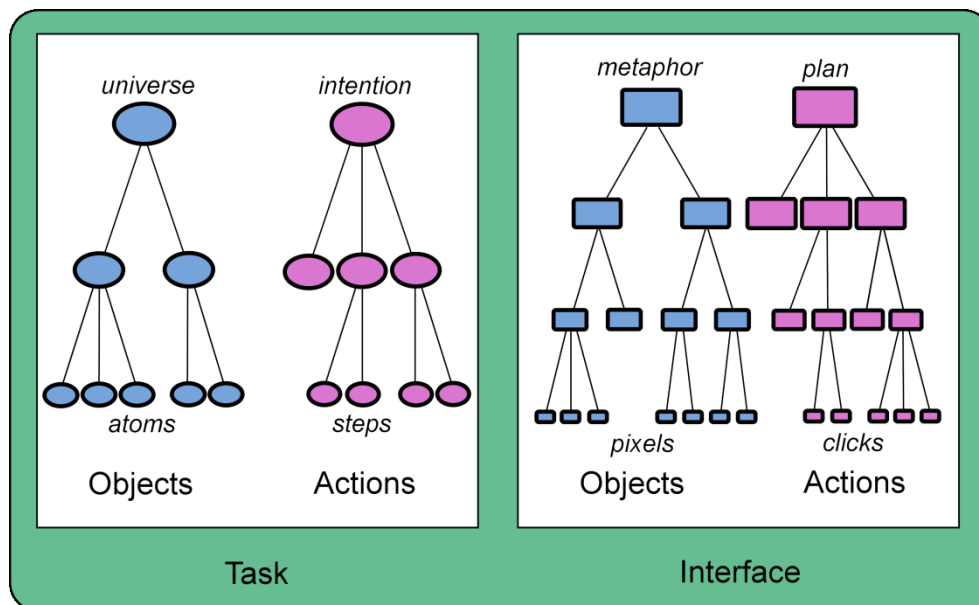


Figure 17. The Object Action Interface model (Shneiderman, Plaisant, 2005).

3.5 Usability evaluation

Usability of a machine or a piece of software can be evaluated in different ways. A system can be evaluated with or without real users and an evaluation can produce both quantitative and qualitative data. As presented in section 3.2, usability can be divided into elemental parts. In the SFS-EN ISO 9241-11 standard (1998) the elements are effectiveness, efficiency and satisfaction. Nielsen (1993) uses in part the same components in his definition that includes learnability, efficiency, memorability, errors and satisfaction. These components are all variables that can be measured in a quantitative usability evaluation. Qualitative usability methods on the other hand typically produce sets of usability problems that can be categorized by severity and type. Suggestions for new features to support successful user strategies are also a typical output of qualitative evaluation (Nielsen, 1993).

There are many different methods for evaluating usability. They can be divided into two main categories: usability testing that involves users and expert evaluations that are done without users (SFS-EN ISO 9241-210). In expert evaluation methods one or more usability experts evaluate a system to find out usability issues. In user testing, real users are involved. Examples of expert evaluation methods are heuristic evaluation (Nielsen, 1993), cognitive walkthrough (Wharton et al., 1993) and standard reviews. Methods involving users include user testing (Nielsen, 1993), contextual inquiry (Beyer, Holtzblatt, 1998), interviews and questionnaires.

Most usability evaluation methods produce a set of problems as their output. To help in deciding which problems should be fixed, the severity of the problems can be determined. The severity of a problem can be difficult to assess and e.g. Nielsen (1993) recommends using multiple independent experts in determining problem severities.

In the following sections an overview is given on the most widely used usability evaluation methods.

3.5.1 User testing

User testing is a widely used method of studying the usability of software systems. Nielsen (1993) describes user testing as the most fundamental usability evaluation method, because it allows gaining accurate knowledge on how people use specific systems and what problems they encounter. Typically, user testing involves having users who are representative of the actual users of the system, run test tasks with the system under evaluation.

The reliability of a user test is affected by the number of users in the test. Depending on the accuracy needed, the number of users can be raised. According to Nielsen (1993), having 5 test users would give a 70 % probability of getting within ± 15 % of the true mean result. Nielsen also states that the optimal pay-off ratio for the number of users compared to the resources expended is highest with 2-5 users with a slowly lowering ratio after 5 users.

The validity of a user test is affected by the right choice of tasks and users. According to Nielsen (1993), validity will be compromised if the right tasks are done by the wrong users or the wrong tasks by the right users.

When planning to run a user test, Nielsen (1993) recommends running a pilot test to refine the test procedures. A pilot test will help see if users understand the test tasks and to verify that the time estimates for the tasks are correct.

User testing is often run in specialized usability laboratories equipped with video cameras, two sided mirrors and screen capture software.

A central part of user testing is the thinking out loud method introduced by Lewis (1982). It includes having the test user explain what they are doing while they are running the test tasks. Thinking out loud makes the user verbalize their thoughts. This makes it possible for the designers to understand how the user views the computer system and makes it easy to identify the user's major misconceptions (Nielsen, 1993).

3.5.2 User feedback

According to Nielsen (1993), collecting user feedback can be a valuable source of usability information, especially on installed systems. Firstly, user feedback is initiated by the users and thus it reflects their most pressing concerns. Secondly, user feedback is an ongoing process and no special efforts have to be made to receive it. Third, user feedback quickly shows if the users' needs, circumstances or opinions have changed.

3.5.3 Questionnaires and interviews

According to Nielsen (1993), questionnaires and interviews are good for studying subjective satisfaction and anxieties and what features in a system are particularly liked or disliked.

One widely used questionnaire is the system usability scale (SUS). It was developed by John Brooke in 1986 (Brooke, 1986) to allow evaluating and comparing usability between systems that are very different in comparison. Usability can typically only be measured when the context of use is very well defined. In developing the SUS, Brooke attempted to create such a general questionnaire that the effect of context could be minimized. SUS measures the subjective usability experienced by the user with a ten-item questionnaire. The questionnaire is a Likert scale with options to answer on a scale from 1 to 5 on whether the interviewee agrees or disagrees with a statement.

3.5.4 Heuristic evaluation

Heuristic evaluation is a “discount” usability inspection method in which a relatively small number of expert evaluators go through a user interface and check if it complies with a set of usability guideline rules, or heuristics. It was introduced by Nielsen (1993). Table 1 presents Nielsen’s ten usability heuristics and gives short explanations on the meaning of each heuristic.

Table 1. Nielsen's usability heuristics (Nielsen, 1993).

Heuristic	Explanation
1 Visibility of system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2 Match between system and the real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3 User control and freedom	Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4 Consistency and standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5 Error prevention	Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6 Recognition rather than recall	Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7 Flexibility and efficiency of use	Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Heuristic	Explanation
8 Aesthetic and minimalist design	Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9 Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10 Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

3.5.5 Cognitive walkthrough

Cognitive walkthrough (Wharton et.al. 1993) is an expert usability inspection method that is used for simulating the thoughts and actions of a first time user of a user interface. The method focuses on the ease of learning by exploration of the user interface. A cognitive walkthrough is conducted on one specific task at a time and one step at a time. An expert evaluator first determines the most optimal course of action for a specific task, a so called “success story”. The evaluator then goes through the sequence step by step. On each step, the evaluator asks herself four questions related to the simulated user's actions:

1. Will the user attempt to achieve the right effect?
2. Will the user notice that the correct action is available?
3. Will the user associate the correct action with the effect they are trying to achieve?
4. If the correct action is performed, will the user see that progress is being made toward solution of their task?

If the evaluator gets through the task, the user interface supports the task. If there are problems in one or more of the questions, the sequence becomes a “failure story”. After encountering a failure, the problem is noted down and the sequence is continued as if no failure had occurred. Based on the failure stories, changes are implemented to the user interface.

To run a cognitive walkthrough, information needs to be available on who the users are, what tasks will be evaluated, what kind of action sequences the tasks will be achieved with and how the user interface will be implemented. Cognitive walkthrough is a good tool for recognizing differences between the way the designer and the users conceptualize tasks, for noticing poor wording in user interface elements and for revealing if inadequate feedback is received after action has been taken. The method can also reveal assumptions made by the designers about the users' knowledge of the task or about user interface conventions.

3.5.6 Problem severity ratings

A usability evaluation usually produces a list of usability problems as its output. To prioritize the problems and help choosing which ones to fix, it is beneficial to assign severity ratings to the problems. According to Nielsen (1993), a problem's severity is often evaluated based on its impact on user performance. One can take into account how many users are affected and how much time is wasted because of the problem.

If no data is available to assess severity, Nielsen recommends relying on the intuition of several independent usability specialists. He recommends not using the severity ratings of only one evaluator, because of the low reliability of purely subjective severity judgments. Nielsen (1994) writes that four independent usability specialists can get to within ± 0.5 points of the true severity of a problem on a 5-point severity scale with 95 % probability. One specialist can only achieve this result with a 55 % probability.

Nielsen (1993) presents two approaches to determining severity. Table 2 presents a single scale and Table 3 presents a combination of two orthogonal scales. The two-scale method can utilize measured values from a user test so that no subjective assessment is needed.

Table 2. Severity scale for usability problems found with heuristic evaluation (Nielsen, 1993).

Rating	Explanation
0	Not a usability problem.
1	Cosmetic usability problem. Need not fixed unless extra time is available.
2	Minor usability problem. Fixing should be given low priority.
3	Major usability problem. Important to fix, should be given high priority.
4	Catastrophic usability problem. Imperative to fix this before product can be released.

Table 3. A table for estimating the severity of usability problems. (Nielsen, 1993).

		Proportion of users experiencing the problem	
		<i>Few</i>	<i>Many</i>
Impact of problem on users who experience it	<i>Small</i>	Low severity	Medium severity
	<i>Large</i>	Medium severity	High severity

3.6 Task analysis

Task analysis is a process in which the current goals, approaches to tasks, information needs and handling of emergencies of the users of a system are analyzed. Task analysis is an important part in the early stages of designing a system. (Nielsen, 1993) Crystal and Ellington (2004) write that task analysis techniques enable rigorous, structured

characterizations of user activity and provide a framework for the investigation of existing practices to facilitate the design of complex systems. They also write that task analysis is vital in integrating complex real life task structures to often highly inflexible computer interfaces.

There are many methods for doing task analysis. Here, two examples of methods from different abstractions levels are given. Hierarchical Task Analysis (HTA), introduced by Annett and Duncan (1967), is a method for breaking tasks into subtasks and operations or actions. These task components are graphically presented using a structure chart. HTA includes identifying tasks, categorizing them, identifying the subtasks and checking the overall accuracy of the model. (Crystal, Ellington, 2004). Crystal and Ellington describe HTA as being system-centered and argue that this is both a strength and a weakness of the method. On the other hand, the system-centricity allows decomposing complex tasks but on the other hand has a narrow view on the task.

GOMS, which stands for Goals, Operators, Methods and Selection rules, is an analytical method originally developed by Card, et al. (1983). Goals are defined as something the user wants to accomplish, such as deleting a word in a text document. Operators are motor, perceptual or cognitive primitives the users have at their disposal. In other words, they are elemental basic actions of sorts, like clicking a mouse or memorizing a name. Methods are combinations of operators used in achieving goals. Selection rules help in choosing between goals, methods and operators if there are many options available. GOMS can be used to predict the usability of a system to some extent. It can for example be used in evaluating how long it might take for an expert user to accomplish a task by adding together the approximate run times of individual mouse clicks and decisions. Crystal and Ellington summarize the difference between HTA and GOMS so that HTA was originally meant for describing high level activities, while GOMS will produce a very detailed keystroke level model. They sum up the historical development of task analysis as a path starting from technically and ergonomically oriented methods to conceptual and information processing models and ending up to work-process oriented contextual models.

Task analysis is a part in many development processes such as Contextual Design (see 3.7.1) and Usability Engineering Lifecycle (3.7.2).

3.7 Development processes

In addition to single usability evaluation methods, a multitude of user centered design processes have been developed. The most typical processes have a lot in common with each other and they follow most principles of user centered design as presented in the SFS-EN ISO-9241-210 standard. The processes combine many of the methods introduced here, including various forms of user testing, task analysis and interviews. This section presents two processes as examples: Contextual Design by Beyer and Holtzblatt (1998) and the Usability Engineering Lifecycle by Mayhew (1999).

3.7.1 Contextual Design

Contextual Design (CD) is a popular user centered design process developed by Hugh Beyer and Karen Holtzblatt (1998). It is a process for developing hardware and software systems based on understanding the structure of people's work. In Contextual Design, data is gathered on the work practice of users and presented in a way that allows a team of multidisciplinary experts to understand the structure of the work. Based on the data, the system under development is designed so that it supports the users' work. Basically,

CD uses a bottom-up approach that starts with details observed in the users' work and builds upon that to create a large scale structure for a new system. Using the large scale structure, new design details can be tested with users. Iterative evaluation of an existing system alone typically can't introduce large changes to the structure of the work, but rather produces evolved versions of the original system.

CD mostly follows the principles of user centered design defined in the SFS-EN ISO 9241-210 standard. The process is influenced by participatory design in the paper prototyping methods. It has later been integrated with software development processes such as agile (Beyer, Holtzblatt, Baker, 2004) and developed as a field evaluation method (McDonald, Monahan, Cockton, 2006). Beyer and Holtzblatt emphasize that the process can be customized to suit the needs and restrictions of a given project and that running the whole process isn't always necessary.

The Contextual Design process includes six steps:

1. Contextual inquiry.
2. Work modeling.
3. Consolidation.
4. Work redesign.
5. User Environment Design.
6. Mock-up and test with users.

Contextual inquiry

Contextual inquiry is a field data gathering technique meant to reveal the structure of the users' work. It is based on conducting one-to-one interviews with actual users doing their every day work. The method is founded on four principles that guide the activities:

1. Context.
2. Partnership.
3. Interpretation.
4. Focus.

The interviews are run in the real *context* in which the users do their work. The interaction between the interviewer and the user is based on *partnership*. The goal is to form a master-apprentice relationship which allows a natural way for the interviewer to learn how the user works. The interviewer observes the user working and makes *interpretations* on what is the significance of what just happened. The interviewer shares the interpretations with the user to either confirm or refute and clarify them. To be able to effectively concentrate on the goals that the interviewer has set for the inquiries, a *focus* is determined. A clear focus helps the interviewer concentrate on the parts of the user's work that have significance from the design point of view.

The interview can be divided into four parts: the conventional interview, the transition, the contextual interview itself and the wrap-up. The interview starts with a conventional interview in which the interviewer introduces the user to the idea and focus of the interview. The user is promised that all recordings made will be kept confidential. The

user is asked to give an overview of the work to be done and asked about their opinions on the tools used.

After the traditional interview, a transition is made to the master-apprentice relationship. The user is told that from now on they are the master running the show and the interviewer is an apprentice wanting to learn the work. The user is asked to do his or her work as usual and told that from time to time the interviewer will interrupt the work to discuss what just happened. The interviewer makes notes and typically records the conversations. Videotaping is also a possibility.

After the work is completed, the interviewer wraps up the interview. The interviewer goes through her notes and summarizes her findings to the user. This gives the user another chance to adjust and correct the interpretations made by the interviewer.

When contextual design is run by a multidisciplinary team, the contextual inquiries can be run by different team members. To allow the team to create a shared understanding of all the data collected, instead of each interviewer only having knowledge of the interviews they themselves have run, interpretation sessions should be run. In an interpretation session, one interviewer goes through one of their interviews and explains the other team members what happened.

Work modeling

In work modeling, the data gathered in contextual inquiries is turned into graphical models to create a visual representation of the structure of the work. Contextual Design uses five different work model types:

1. Flow model.
2. Sequence model.
3. Artifact model.
4. Cultural model.
5. Physical model.

The *flow* model represents communication and coordination between people with different roles within an organization. The *sequence* model visualizes the concrete steps taken to achieve intents on different levels of abstraction. The *artifact* model presents tangible things, like calendars or spreadsheet documents that people use in their work. The *cultural* model visualizes the culture of an organization. The cultural model is created using influencers, i.e. people, organizations and groups, and influences, the effects of influencers on each other. The *physical* model is a caricature-like representation of the physical environment in which the work is done.

Consolidation

After data collected from users is turned into work models, the design team consolidates the data. In consolidation, the data is combined to create a coherent view of the different types of users in the customer population. Consolidation is done using work models and affinity diagrams. An affinity diagram is a hierarchical diagram into which findings from interpretation sessions are collected to form a picture of the whole work domain.

Consolidated work models bring together the individual work models from the previous step to reveal the underlying patterns in the work.

Work redesign

In work redesign, the work practice is restructured based on the consolidated data. The design team has discussions on how to improve the work and comes up with a vision of the new work practice. Based on the vision, storyboards are created, describing how people will work in the new system.

User Environment Design

A User Environment Design (UED) is a representation of the system work model that shows the functions and parts of the new system and their relations to each other. It is a floor plan of the new system that is not affected by user interface details. A UED is created by collecting together the different stories from storyboards made in the work redesign phase and assembling a structure that supports them all. The UED is divided in to focus areas that can be thought of as rooms in a floor plan of a house. Focus areas are places in the system where certain activities are supported. Different areas are connected to each other to allow the user to move in the system when doing their work. A focus area contains, organizes and presents objects the user needs when working.

Mock-up and test with users

In the mock-up and testing phase, design ideas are prototyped, tested with users and developed to better suit the users' work. Contextual Design emphasizes early testing to keep development costs down. The goal of testing with users is to communicate the design presented in a user environment design to the users in a concrete form and find where the design requires improvement. A concrete prototype is easier for the users to comment on than a User Environment Design diagram or a list of system requirements. Rough paper prototypes are created early on. They inherently invite the user to suggest changes to the design and keep the user concentrated on the structure of the work rather than on user interface details. The users are made to do real work with the prototypes, which gives them a possibility to make their unarticulated knowledge explicit. They can easily point out why a certain design fits their work and why another doesn't. Basically, the steps from contextual inquiry to User Environment Design are followed to create a good enough starting point for the user testing.

3.7.2 Usability engineering lifecycle

The Usability Engineering Lifecycle is a process conceived by Deborah J. Mayhew (1999). It is focused toward projects that have been defined, scoped and planned. It is also oriented toward creating new solutions rather than on reworking existing products. The process consists of three main phases. The first phase is the requirements analysis, followed by design, testing and development which are followed by installation. The process is presented graphically in Figure 18.

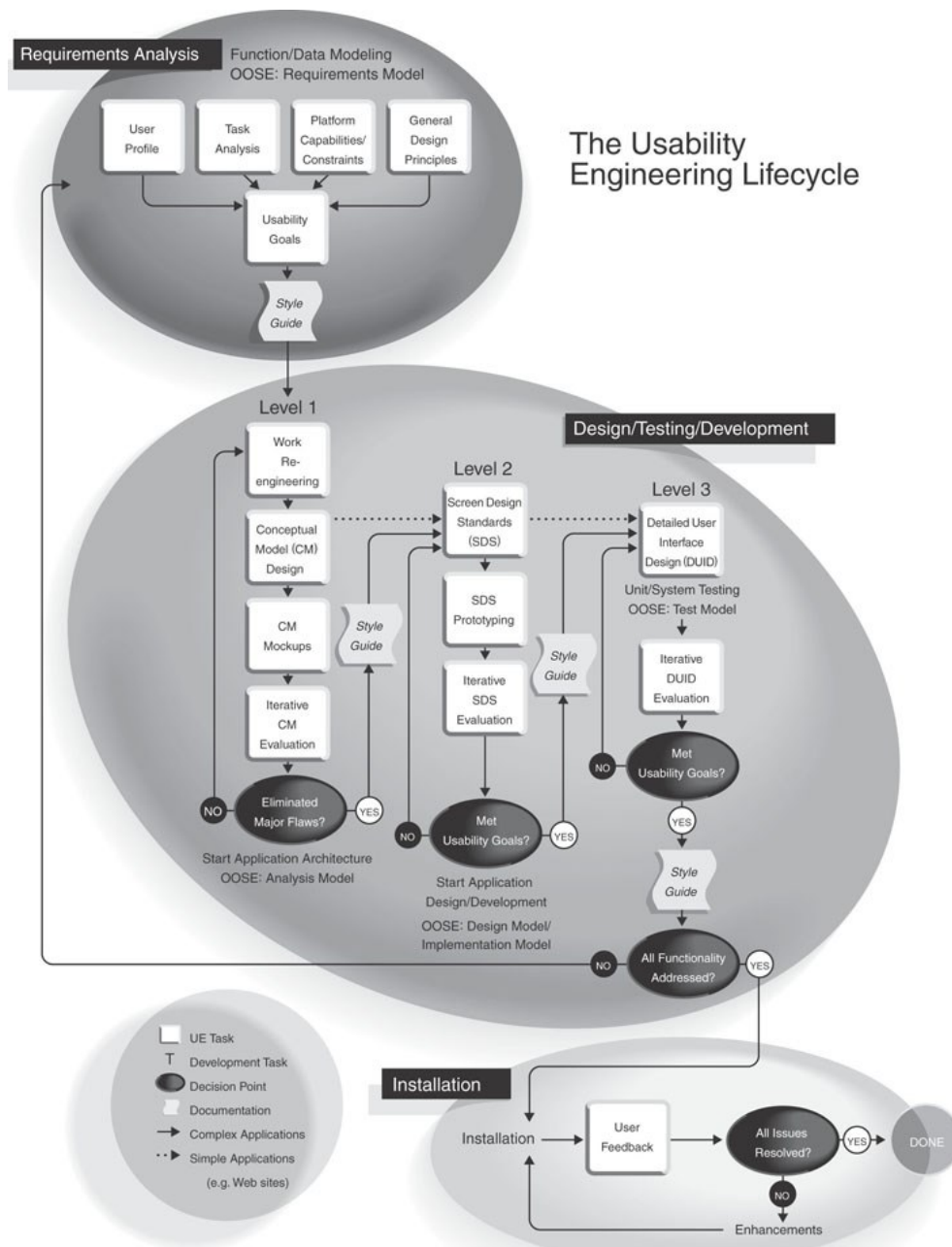


Figure 18. The Usability Engineering Lifecycle process. (Mayhew 1999).

The specifics of the process are left for the reader to examine, but overall, the usability engineering lifecycle has a top-down approach to design, like Contextual Design. Furthermore, just like CD, it is based on data gathered from users and the process can be customized based on specific needs of the project. Also, the involvement of a cross functional team is emphasized.

3.8 Evaluation and design process

The first research question defined for the study in chapter 1 was: what methods and processes could be used when studying the usability of the quasar observation system? In the preceding sections a literature review was made on the subjects of usability, user

interfaces, methods of usability evaluation and user centered development processes. In this section, an answer is formulated to the first research question based on the literature review.

Metsähovi's case as a development project can be classified as an evaluation and redesign of an existing system. The project was to yield concrete improvements in the observation system within the approximately 8-month long period reserved for it. Based on the literature review, the SFS-EN ISO 9241-210 standard (see section 3.3) is a good basis for the usability study. The four stage development process can act as a platform to which suitable development and evaluation methods can be attached. The principles of user centered design can be followed to ensure the development process will involve the users, be iterative in nature and be based on user centered evaluation.

What methods could be used to evaluate an existing system? Shneiderman and Plaisant (2005) list methods for evaluating a system that is in active use:

1. Interviews and focus-group discussions.
2. Continuous user-performance data logging.
3. Online or telephone consultant.
4. Online suggestion box or e-mail trouble reporting.
5. Discussion groups or newsgroups.

Beyer and Holtzblatt (1998) recommend using a partial version of contextual design (see section 3.7.1) when trying to find the top ten problems in a system. They recommend running contextual interviews to see how people use the current system. The focus should be on noticing how the system gets on the users' way or interferes with their work. Interpretation sessions should be run and notes should be captured. Low level sequences would be assembled to recognize intents and ways in which users interact with the system. In the end, the findings should be organized in an affinity diagram to help identify the main issues in the system.

As described in chapter 2, the starting point for the study is such that the current observation system allows successful completion of all observing and data pre processing tasks. The main problems are that the system is scattered over multiple tools and that accomplishing many tasks is inefficient, wasting observers' time.

As is explained in more detail in chapter 4, the literature review and the practical study were conducted in parallel. The exact methods to use in the study were decided "on the fly" as the author gathered more knowledge on the methods. Below, a brief justification is given on what methods were used and why.

User feedback and developers' ideas were collected to gather suggestions for new features and to gain information on the problems that the users have been struggling with. This also made it possible to collect and document development ideas that may have been floating around. Heuristic evaluation was used to quickly and easily collect problems that may be otherwise missed. These two methods fit into the evaluation part of the SFS-EN ISO 9241-210 process (see section 3.3). To understand and specify the context of use which is the first step in the SFS-EN ISO 9241-210 process, contextual inquiry was selected as the backbone method of the study. The purpose was to enable collecting data to be used in recognizing how the observers work and what problems they face. This allows task analysis and creation of user requirements. The data can also

be used in a future redesign effort. User testing was selected to be used on the second iteration to allow measuring the effect on usability by the changes made to the system.

3.9 Conclusions

In this chapter, literature review was conducted on the subject of user centered design, usability and user interfaces. The goal was to get acquainted with the principles and the methods applied in user centered design. The knowledge acquired was used to come up with an evaluation and development process to be followed in the usability study of the quasar observation system. The process was assembled so that the most high impact usability problems in the system could be recognized and some of them fixed.

The process was assembled to contain two iterations of evaluation separated by the creation of implementations between them. The process was based on the principles of user centered design, meaning that, among other things, the process was iterative, based on user centered evaluation and that users were involved in the development and design. Some principles, especially having a multidisciplinary team working on the design and addressing the whole user experience, had to be given less attention. This was done because of limitations in available time and manpower.

On the first iteration, the current status of the observation system would be evaluated using user feedback, heuristic evaluation and contextual inquiries. The data from the first iteration would be analyzed by utilizing affinity diagramming and the creation of sequence models. The data and results from the three evaluation methods would be combined to generate a set of usability problems. The severity of the problems would be evaluated based on Nielsen's definition (see section 3.5.6). Suggested improvements would be generated for fixing the detected problems. The improvements would be based on principles and theories regarding user interfaces, most importantly Nielsen's ten usability heuristics (see section 3.5.6), the Gestalt laws (see section 3.4.3) and the principle of direct manipulation (see section 3.4.3). When developing the solutions, the designs would be presented to the observers in small scale user tests to gather quick feedback.

The improvements practically implementable within the confines of the thesis work were to be implemented. Others would be implemented after the thesis project. The second iteration of the study would include measuring the changes in efficiency, effectiveness and satisfaction that the implemented improvements had brought to the table. The evaluation would be made using user testing as the main method.

The process is in many parts a compromise between strictly following processes that have been proven to produce viable results and the quick generation of practical results for Metsähovi. It can be argued that the process isn't as reliable and scientifically sound as it could have been if more resources had been available. Especially the analysis process of the contextual inquiry data was stripped down considerably from what was originally presented by Beyer and Holtzblatt (1998). The analysis would be more based on the personal intuition of the author rather than on thorough analysis by a multidisciplinary team. The risks of this might include missing some problems in the system and overemphasizing or misinterpreting others. The implementations of solutions might miss the points of the actual problems. The solutions could also be less systemic and more point solutions in nature than would ideally be attainable. The heuristic evaluation would be conducted in a sub optimal fashion, mostly due to the availability of only one expert evaluator instead of five recommended by Nielsen

(1993). However, as heuristic evaluation was utilized as a complementing method for contextual inquiry and user feedback, getting any results was thought of as a bonus.

Because of the deficiencies in the development process, it will be important to continue iterative development with users after the completion of the study in this thesis. This way the solutions can be honed to better match the users' needs. The data gathered in the contextual inquiries could be reused to make a more comprehensive system redesign effort. Compared to the problem-fixing approach followed in this study, the redesign effort should include a multidisciplinary team comprised of Metsähovi's staff and a more holistic approach to the data production and processing pipeline.

4 Execution of the study

This chapter presents how the study was executed. The first section starts by summarizing the whole process. The second section presents the activities of the first iteration round of usability evaluation. The third section presents how the system was changed based on the results of the first round. The fourth section presents the second evaluation round that was conducted to see how some of the changes to the system impacted the system's usability.

4.1 Summary

The study was executed in two main iteration rounds. On the first round, the observation system was evaluated by collecting feedback from observers and developers, by conducting a heuristic evaluation and running contextual inquiry interviews. The collected data was analyzed and arranged into a set of severity rated usability problems with attached suggested improvements. Based on the feasibility of implementing the improvements and the severity of the problems, changes were implemented to the observation system.

Some of the most obvious problems and needs appeared early on in the study, especially in the contextual inquiries and in the user feedback. Solutions to these early issues were developed rapidly in co-operation with the observers even before further analyzing the collected data. This included small scale usability tests with prototypes and informal talks involving the author, the users and other developers.

On the second iteration round, the effectiveness of the changes to the system was evaluated. The second evaluation round was conducted by running a heuristic evaluation and usability tests in addition to having informal discussions and running small scale usability tests with observers. The evaluation and the implementation of changes to the system were conducted over a period of many months. Some implemented changes were already in use while the first evaluation round was still ongoing and that's why contextual inquiries were used on the second round as well.

The practical work and the literature review were conducted in parallel. When the work started it was decided that the development process would be user centered in nature. The exact methods to use were yet to be determined. Collection of feedback and heuristic evaluation were selected early on, mainly because they were fairly easy to conduct in practice and because the author had previous experience of heuristic evaluation. These methods provided early data on what problems there may be in the system. The decision to use contextual inquiry and user testing was made later, after the author had come more acquainted with the usability literature. Contextual inquiry proved to be a suitable method for forming a picture of the structure of the observers' work. User testing was seen as useful in both acquiring quantitative measures of usability from the original and modified user interfaces and in unveiling new usability problems in the modified tools.

To satisfy the need to produce practical solutions within the thesis project, it was decided that a comprehensive high level redesign would be left outside the thesis work. The author feared that concentration on redesign might end up producing only suggestions due to the extensive analysis, and thus resources and time, required. The data collected especially in the contextual inquiries can be used in a future redesign effort.

It was expected that the full time contribution of one person would be available for running the usability evaluation and design. It was also expected that the users and other developers of the observation system would be available for feedback and interviews, for running meetings and for also doing some implementation of changes to the system. There was also a lot of subject matter know-how available on observing work, especially from the senior researchers.

The goal was to improve the usability of the quasar observation system as much as possible with the resources available. Focus would be on fixing high impact usability problems that are relatively easy to fix: the so called “low hanging fruit”. In spite of focusing on quickly fixing big problems, the goal was also to attempt producing systemic responses. That is, to try and see larger scale themes in the work and come up with solutions that are not only one shot solutions to point problems.

4.2 First iteration

4.2.1 User feedback and developers' ideas

Feedback and ideas from observers and developers were collected early on, starting in December of 2011. The early start was made to maximize the amount of feedback collected. Feedback was collected by having a kick-start meeting, many informal discussions and by setting up a table in Metsähovi's intranet wiki-page where feedback could be input and saved. The observers and developers were asked to input any problems or suggestions regarding the system into the wiki-page. A screenshot of the wiki page is presented in Figure 19.



Metsähovi Localweb Wiki Search [] Titles Text

Wiki
Recent Changes
Find Page
Help Contents
Observat.../Suggestions

Page
Edit
Info
Attachments
More Actions: []

Observations / GIX / Suggestions

Suggestions

Here you can report problems and issues and give suggestions and comments on the whole GIX observation system and its usability, e.g. automap, observation summary, MB, future development etc. All comments are welcome, from the the color of the target symbols in automap.pl to the complete architecture of the observation system. Input your comments to the table below.

If you want to comment a suggestion given by someone else, add your comment to the "Proposal for correction / response" -field along with your name/username. (Eg. "I think this could be corrected using duct tape. -nlavonen")

Comment / Problem / Suggestion	Proposal for correction / response	Author
Overall, the system is scattered over multiple web pages and terminal windows	Gather the most important tools for conducting observations in one place. To know what are the most important tools, one must know the most important goals and tasks of the observers. --> Apply contextual design. At least contextual inquiry + task analysis.	nlavonen
Tau values determined manually	Determination of the tau value should be automatized and/or be based on real measurements if possible. -nlavonen; I believe this is mandatory for 22 GHz observations. -maturun	nlavonen
Good quality observations have to be updated manually --> It's hard to keep track of sources that have recently been observed. This makes it harder to asses what sources need to be observed next. Having to memorize the latest good quality observations increases the observer's memory load. Alternatively, checking the observation results takes extra time and breaks the work flow.	Automatic initial classification of observations immediately after observation is finished --> confirmation from user. -nlavonen. A quick fix solution: Add a checkbox to each observation in obs_summary (or obs_delcom), so that an observation can be checked as a good quality observation. -nlavonen	nlavonen
It's not possible to add sources for	Creation of an "observation queue" type of a system, where	

Figure 19. A screenshot of the wiki page where developers and observers could input feedback regarding the observation system.

The table on the suggestions wiki page contained three columns. The first column was reserved for comments, problems or suggestions. The second column was reserved for

proposals for correcting a reported problem or for responses for comments and suggestions. The third column was for recording the author of each entry.

4.2.2 Contextual inquiries

In total, 13 contextual inquiry interviews were run with eight observers between July and October of 2012. The inquiries were organized and run by the author of this thesis. The goal was to collect usability data that could be used in recognizing usability problems and in developing features to better support the observers' work. The data could also serve as a basis for a more extensive redesign effort in the future (see chapter 6). The intention was to collect data from observers belonging to four different predetermined user groups and from six predetermined tasks of the observation work flow. The tasks that were selected as important were:

1. Observing manually.
2. Observing automatically
3. Processing observations after a shift and submitting the data for final reduction.

Additionally, three more specific tasks were listed as interesting:

4. Updating good quality observations in the source list.
5. Measuring and updating pointing offsets.
6. Using the dome heater.

The tasks were selected based on the personal observing experience of the author. It was expected that these tasks wouldn't necessarily represent the whole picture and that other tasks might also be observed. The inquiries can be roughly divided into two categories: observing and data processing. In the first category the observer was expected to conduct observations and work on related tasks. In the latter category the observer would go through their data from an already finished observing shift.

The times and tasks for the inquiries were negotiated beforehand with the observers via e-mail and by face-to-face discussions. A preliminary agreement was made on what kinds of tasks the observer would most likely be doing. The observers were to be interviewed and observed while doing their normal observation work. This would assure the observed work was as close to natural as possible and it would be easier to notice such problems in the work that the observers face in their daily routine. To enable favorable circumstances, the interviewer volunteered to conduct the inquiries where ever the observers would normally do their observations. One inquiry was run at an observer's home and the rest were run in the observatory's premises in Kirkkonummi.

It turned out to be impractical to attempt running inquiries on all the six selected tasks with all the eight observers. This was mostly due to scheduling reasons and also due to the fact that some tasks, such as updating good observations, take a relatively short time to finish. In most of the inquiries conducted, the observers told the interviewer that they had already completed some tasks of interest before the interview.

In hindsight, it might have been beneficial to ask more users to hold off doing some of the tasks to allow performing them during the inquiries instead. This view is supported by Beyer and Holtzblatt (1998), who write that asking the users to save the kind of work

the interviewer wants to observe, is worth while. The small effect on the work flow will be offset by the increase in relevant data. Beforehand, the author erroneously interpreted this kind of meddling in the observer's work as a potential results-skewing factor.

Table 4 shows the observers, tasks and dates that were included in the inquiries. As mentioned, the inquiries were run during a period of three months between July and October of 2012. Because of the long time period during which the inquiries were run, later inquiries were run with some user interface changes implemented. Most notable changes in the system were the adoption of the integrated tau value adding and introduction of the weather data widget in the observation summary tool. These changes are presented in section 5.4.

Table 4. The dates and preselected tasks of the conducted contextual inquiries. The tasks that were initially classified as being most important are indicated by bolded text. Entries in parentheses indicate incomplete or partial observations of the task.

Observer / task	Observing manually	Observing automatically	Updating good quality observations	Processing observations after shift	Measuring & updating offsets	Using dome heater
1	12.7	11.7	-	-	-	(11.7)
2	19.7	19.7	-	24.7	(24.7)	19.7
3	26.7	2.8	15.8	15.8	-	26.7
4	-	14.9	-	-	-	-
5	28.9	28.9	-	5.10	-	28.9
6	16.8	16.8	-	-	(16.8)	16.8
7	22.8	22.8	22.8	(tau: 22.8)	-	-
8	24.8	24.8	-	-	-	-

The contextual inquiries were started with a short preliminary interview. The point was to gather information on the observing experience, education background and research experience of the observers. This information could be useful later in the study in e.g. helping to understand reasons behind patterns in the observing work. The questions of the preliminary interview are presented in section appendix A. The interview was conducted once with every observer. Before the interview, the interviewer told the observer the purpose of the whole contextual inquiry, which was to learn how the observers do their work and to help in developing a better observation system.

After the preliminary interview, the interviewer told the observer that from now on the observer would be the “master” and the interviewer the “apprentice”. This was done in order to help create the master-apprentice relationship which is one of the key aspects of a contextual inquiry (see section 3.7.1). The observer was told she could do her work normally and that occasionally the interviewer would interrupt the work to ask clarification to observations he had made.

Audio recordings from the inquiries were made with a Nokia 6700 Classic mobile phone and bullet notes were made on a laptop computer. Confidential handling of the interview data was promised to the interviewee. No word-to-word transcripts were made due to the large amount, approximately 25 hours, of audio recordings accumulated in total.

The focus in the interviews was on the six tasks seen as key parts of the observation work flow. The interviewer focused on noticing the structure of the work and on recognizing problematic or exceptionally laborious tasks that could be made easier by

altering the system. Essentially, the interviewer was hunting for usability problems. Typically, when noticing the completion of a specific task or sub task, the interviewer asked the interviewee what she just did and what was the main purpose and goal. He would also share his own interpretation of the main goal with the interviewee. If the interviewer made an interpretation that something problematic had take place during the task, he would share this interpretation with the interviewee as well. This way it could be verified whether the interviewee agreed with the interviewer about there being a problem. Additionally, the interviewer shared ideas for improving the system and received instant feedback from the interviewees.

When the observer had finished her work, the interviewer went through his notes, summarizing his most important findings. In this phase the interviewer's goal was to further confirm his interpretations of the work with the observer and if necessary, correct them. During the discussion on the interpretations, there was often also discussion on how the system could be improved and the work flow made more streamlined. On many occasions, the observers gave suggestions on new features for the system and underlined their frustration toward problematic functionalities. Often the interviewer and the observer made drawings or sketches on paper on how a feature or a solution could look like in the future and how it could be used. This early user involvement was useful in steering the development of the design solutions to the right direction from the beginning.

The contextual inquiries lasted 1-5 hours, depending on the tasks the observer was running. The inquiries produced a set of audio recordings and notes. The notes contained observations and interpretations on the work flow of the observed tasks and ideas and suggestions on how to improve the observation system. The notes and recordings were used as a basis for further analysis. Early on, some of the observers and the author agreed on some general patterns in the work. One was the need to reduce jumping between tools and another was the need to integrate weather data for data pre-processing. Ideas for addressing these needs were developed already in the contextual inquiries and refined further later on.

4.2.3 Heuristic evaluation

Heuristic evaluation was conducted on most of the web and command line user interfaces that are used in quasar observations. The evaluated user interfaces include following user interfaces and tools:

1. The dome heater CLI (autoheat.pl).
2. The autofile generator.
3. The real time sky view.
4. The observation summary tool.
5. The observation delete status and comments tool.
6. The MB plotting tool.

The separate tau tools were not evaluated. It was evident based on the user feedback (see section 5.2.1) that these tools generated extra work and had usability problems. It was decided early on that the tau tools had to be completely replaced. A heuristic evaluation of the soon-to-be-replaced tools would not have benefited the development process.

In the evaluation, two sets of heuristics were used. The main heuristics were Nielsen's ten usability heuristics (see section 3.5.4) and they were supplemented by the gestalt principles (Rock, Palmer, 1990).

Heuristic evaluation was conducted after most of the user feedback had been received. Because of this, many problems that could have been noticed independently by the evaluator had already been reported by the observers or developers. Such problems were intentionally left out from the list of problems gathered by the heuristic evaluation.

There are several factors in the study that affect the performance of an expert inspection, most notably that of a heuristic evaluation. Most notably, there is only one expert evaluator available for this study. This affects the expected number of usability problems to be found. The expected number of problems to be found using a heuristic evaluation rises as a function of the number of evaluators (Nielsen, 1993). The number of problems found by a single evaluator is expected to be approximately 35 % of the total number of problems in the system (Nielsen, 1993).

Secondly, the single evaluator in this study is also a user of the system and has also participated in its development. This may affect the way the evaluator detects problems in the system, i.e. he might miss a problem because he has gotten used to using the system in a certain way. Also, being a user and developer of the system may affect the evaluator's objectivity. On the other hand, the evaluator also has used the system for long periods of time, which will have helped in noticing possible problems. Because of the reasonably short time it takes to conduct a heuristic evaluation and the lack of need to involve users, the disadvantages of the circumstances affecting the performance of the method are overcome by the advantages. All in all, despite a sub optimal number of evaluators, heuristic evaluation is a resource effective addition to the set of methods used in the study.

4.2.4 Analysis

Once data had been collected utilizing user feedback, heuristic evaluation and contextual inquiry, the data was analyzed. The main goal was to come up with a set of usability problems and to develop improvements on a high enough abstraction level. The purpose was to develop systemic responses and avoid addressing every isolated small scale problem individually. This principle was adopted from contextual design (Beyer, Holtzblatt, 1998), where it's emphasized that addressing every isolated problem individually can lead to short term solutions. Such solutions may not address the underlying problem and they may quickly become obsolete after a higher level solution is developed. Material from user feedback and heuristic evaluation required less analysis than data from contextual inquiries.

As presented in chapter 3, the full analysis and development process of contextual design includes seven stages. Going through the full contextual design process and pursuing a full redesign was deemed unfeasible considering the time and manpower available. Instead, based on the notes and audio recordings from the contextual inquiries, only sequence models were created. From these, a set of usability problems was assembled. During the creation of the sequence models, attention was paid to seeing the large scale structure of the work, similarly as had been done earlier during the contextual inquiries themselves. According to the contextual design process (see section 3.7.1), intents, triggers and breakdowns were noted in the work. The term breakdown was applied more loosely to label all findings that the author perceived as

problems in the work flow or in the user interfaces. An example of the sequence models created is presented in appendix C.

The problems from the sequence models were collected and categorized with the help of an affinity diagram. All the problems, marked in the models as “breakdowns”, were printed out on paper and placed on a table. The problems that were seen as related to each other were placed in clusters and the finished clusters were given a name based on their common denominator. This denominator was sometimes a task and sometimes a certain user interface.

The problems collected were given severity ratings based on Nielsen’s (1993) table for estimating severity (see section 3.5.6). As mentioned by Nielsen (1993), subjective severity ratings made by a single evaluator can be very unreliable. This is why the severity ratings presented in chapter 5 should be considered only as rough approximations.

The end result from the analysis was a collection of severity rated usability problems that were categorized by the method by which they were discovered, by task and/or by user interface. A typical problem type found in the analysis was an extra step in accomplishing a task. Such steps were seen as overly complicated considering the high level goal that the observer was trying to achieve.

4.3 Implementing modifications

Once a set of problems in the observation system were recognized, solutions were developed to overcome them. Focus was on developing solutions that were realistically implementable but that were also systemic rather than point solutions in nature. Solutions were developed by prototyping design solutions and showing them to users. These designs were often based on the talks and early ideation and prototyping with the users in the contextual inquiries.

Due to the limited resources available, it wasn’t seen as appropriate to change the platform of the user interfaces away from the mostly web based design. The changes made were modifications of existing user interfaces rather than completely fresh designs. Because of the fairly recent advances in web user interface technologies such as AJAX, DHTML and JavaScript libraries such as JQuery, it was possible to add interactive functions on top of the existing basic HTML page platform. This was especially the case with the observation summary tool in which high impact modifications could be made without fundamental changes to the underlying functionality.

The principles followed in developing the solutions included Nielsen’s ten heuristics (see section 3.5.4) and the gestalt principles (see section 3.4.3). The principle of direct manipulation (see section 3.4.3) was also widely applied.

4.4 Second iteration

Changes to the system were evaluated in three phases. The first solutions were developed and implemented while the contextual inquiries of the first iteration were still running. This provided an early possibility to check how the solutions fitted the observer’s work flow. Problems in the modified system were collected along the analysis of the contextual inquiries. Additionally, small scale user testing was done with some users to gather their opinions on prototype solutions. Small tweaks were made to

the user interfaces based on the feedback. Thirdly, five user tests were conducted, one of which was a pilot test. The tests were conducted to measure the efficiency, effectiveness and user satisfaction of the renewed observation summary tool and the old version together with the other tools related to it. This allowed comparing the usability attributes to see if the new version was actually better than the old one. The tests also served to reveal what kind of problems observers might have with the new user interfaces. They also provided a possibility to collect more feedback and suggestions. All this information could be used for developing the system further and in part for assessing how well the study had succeeded.

4.4.1 User testing

User testing was conducted in January of 2013 after main changes in the system had been implemented. The primary goal was to compare readings of efficiency, effectiveness and satisfaction between the original and modified versions of the observation summary tool. Secondary goals were the recognition of problems in the modified user interfaces and the collection of user feedback and suggestions.

It wasn't plausible to test all the changes to the system. Instead, the user interfaces to be tested were selected based on their suitability for measuring efficiency, effectiveness and satisfaction and based on the extent of the implemented changes. The observation summary tool fit best to these conditions. It had proven the most eligible part of the system for implementing high impact changes without extensive redesign. Many of the problems noticed in tasks involving the use of observation summary were related to poor efficiency, which was a core attribute for testing. An example of such a problem is having multiple tools for single tasks. Also, the solution for streamlining the single most problematic task of adding and modifying tau values had been implemented in observation summary. These factors made the tool a good candidate for usability testing.

The changes made in observation summary had made it possible to retire four specialized tools: the observation delete status and comments tool (`obs_delocm.pl`) and the three tools for adding, searching and deleting tau values (`tau_add.pl`, `tau_query.pl`, `tau_delete.pl`). As presented in more detail in chapter 5, the functions of the specialized interfaces were integrated into observation summary. Thus, the user interfaces involved in the tests were the modified and unmodified versions of the observation summary tool, in addition to the aforementioned specialized tools.

It was seen as important to properly test the replacing new functions in observation summary. It was already known that many users had tried out prototypes of them without major problems. Additionally, the integrated tau value manipulation functionality was in operative use for nearly four months before the beginning of the user testing. During this time the observers didn't report running into any critical problems in adding or modifying tau values. It was expected that the testing could potentially reveal problems that are less serious and that could be fixed to further develop the user interfaces.

Five usability testes were run, the first of which was a pilot test. The pilot test was conducted to see if the users would actually understand what was asked of them and to test the overall fluency of the test procedures. The users who took part in the testing included two senior researchers with over 20 years of observing experience, one Ph.D. level researcher with 4 years of observing experience and two Ph.D. students with 1,5 years and 4 years of observing experience, respectively.

The test procedure started with an introduction in which the idea of the test was explained to the user. After this, the user was introduced to the test scenario which was a hypothetical situation presented as a background story for the testing. Then a “practice run” of the test tasks was executed. The interviewer asked the user to do each test task once to get the user familiarized with the tasks. There was no hurry or pressure. The interviewer gave advice to the user if needed. This was done to reduce the effect of learning on the test results, as the point was to measure fluent expert user performance and not so much e.g. learnability. The practice run also gave the user an early experience of success that e.g. Nielsen (1993) recommends on including in the beginning of a usability test.

After the practice run, the user was asked to run the test tasks “officially”. The tasks were given to the user one at a time in written form on a piece of paper. The test tasks are presented in appendixes E, F and G. After each task, the user was given a paper note where the user was asked to evaluate the subjective pleasantness of the completed task.

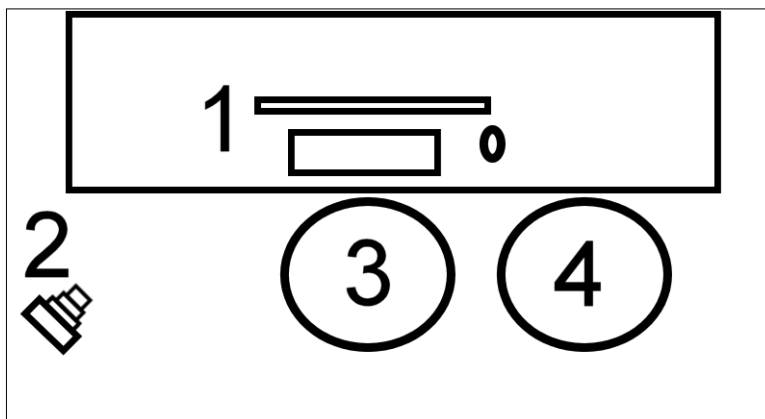


Figure 20. The arrangement in the user testing. 1. Screen, keyboard and mouse. 2. Video camera. 3. User's chair. 4. Experimenter's chair.

The events on the computer screen and the audio during the test were recorded in case events needed to be checked later. For example, when checking the number of erroneous choices made in the tests, the video was seen as possibly useful. On some of the tests the interviewer forgot to start the recording and on some tests the camera failed to record the whole task. In the end, the video recordings weren't needed much, so these problems didn't noticeably affect analyzing the results.

The measure of efficiency was obtained by timing run times of the test tasks. Effectiveness was measured through the number of errors and successful task completion. User satisfaction of individual test tasks was measured using a single likert scale from 1 to 5 for user perceived pleasantness. To measure the overall satisfaction of the two different software versions tested, the user answered a SUS questionnaire with ten questions, each on a 1 to 5 likert scale. The test procedures for the interviewer and the user are presented in appendixes D and E, respectively. The SUS-questionnaire template is presented in appendix I.

The tests were run in the observatory in two locations which were both quiet rooms. The tests were run on a Linux desktop computer. The computer had a 23 inch screen with a resolution of 1920x1080 pixels. The test arrangement is presented in Figure 20. To simplify the testing, the tests were done using the local machine's Firefox web browser instead of the browser on the VNC desktop. It was already known that the observation summary tool was too large for the low 1024x768 pixels resolution of the remote desktop. Had the test been run using the remote desktop, the results could have

been overshadowed by the problems stemming from the low resolution. Suggestions for lowering the resolution requirements had already been made on the first iteration round so it was seen as more worthwhile to concentrate on the other factors affecting the tool's usability.

Because the goal for the testing was mainly to measure performance of an expert user, the experimenter guided the users in finding the tools needed in accomplishing test tasks. This was done especially on the practice run that was done before the actual testing. Shortcuts were placed on the browser before the tests for quick access to the tools needed. Time spent searching for the various tools needed to complete tasks with the old version might have added significant time overhead to the task run times. The goal was to present both the versions a best case scenario, and including the times to search for the tools wasn't seen as useful.

The test plan and the wrap-up interview template are presented in appendixes E and H, respectively.

4.4.2 Heuristic evaluation

After changes had been implemented in the observation summary tool based on the first evaluation round, a heuristic evaluation was conducted on 2012-11-01, on version 1.38 of the tool. The version tested had been modified with integrated tau value adding, all-sky images and a weather widget displaying an all-sky image and a data graph plotter (see sections 5.3.4 and 5.3.8). The user interface of the tool was gone through piece by piece, considering whether the user interface agreed with Nielsen's ten heuristics (see 3.5.4) and the gestalt laws (see 3.4.3). As on the first iteration round, the found usability problems were collected in a table and were given an estimated severity rating in addition to a suggested improvement.

5 Results

This chapter presents the results of the study. The chapter is divided into six sections that present results from three main phases of the study: first evaluation round, modifications to the system and the second evaluation round. In the first section, a summary is given on the results. Section 2 presents the findings from the first iteration round, starting with user feedback and developers' ideas. It also introduces the problems and other findings aggregated from the contextual inquiry data and presents the problems found with heuristic evaluation. In section 3, the problems and findings from the first round are analyzed and the modifications made to the system are presented. Section 4 presents the evaluation results of the modified observation summary tool, including results of user testing and heuristic evaluation. Section 4 also presents changes made based on the second evaluation round.

5.1 Summary

The main goals of the study were to find the biggest usability problems in the quasar observation system, suggest improvements, implement some of the improvements and evaluate their effect on usability. The main problem types found in the observation system were having multiple tools for single tasks and having relevant information scattered over many web sites and tools.

Every day, observers spend extra time and effort in completing routine tasks, which wastes time and increases stress. Many separate tools and web sites have to be accessed to attain a clear picture of the weather conditions. The observers have to use multiple tools just to access data from Metsähovi's own weather sensors and cameras. In addition, the observers access many outside weather forecast services for complementing their picture of the current conditions and for seeing what will happen in the future. The lack of a centralized source of weather information means that the observers are constantly jumping between multiple different tools.

The biggest changes made to the observation system were implemented in the observation summary tool, which is used for viewing and modifying unreduced observation results. Multiple functions that were previously implemented in separate tools were integrated into the one single tool with the help of DHTML and AJAX technologies (see section 3.4.2). The changes include the integrated tau value, delete tag and comment modification functions. Another major modification to the tool was the integration of the all sky images to the results table and the introduction of the so called weather widget. The weather widget is a floating window on the web page that presents relevant information on the conditions related to a single observation. It displays an all-sky camera image and a data graph plotter. A small all-sky thumbnail image is attached to every observation result in the table. When a thumbnail or a table row is clicked, the weather widget displays data related to that observation.

User tests run with the observation summary tool show a decrease of 35% to 92 % in data pre-processing average task run times. The SUS questionnaire results show a 30 point increase in average perceived usability by the observers (on a non-linear scale from 0 to 100). Task specific questionnaires gave an increase of 0.8 to 3.5 points (on a scale from 0 to 5) in subjective single task pleasantness.

The autofile generator and real time sky view tools were slightly modified. Some unused controls were removed and sources of extra high priority were marked on the

sky view more prominently. Four observers had previously used manually created spreadsheets to gain information on the visibility of high priority multi-frequency campaign sources. They estimated that creating the spreadsheet took them approximately 30 minutes in the beginning of every shift. After multi-frequency campaign sources were marked on the sky view, the same observers reported that they no longer had to use the spreadsheets.

5.2 First iteration of studies

This section presents results of the first iteration of evaluation.

5.2.1 User feedback and developers' ideas

Table 5 presents the user feedback and developers' ideas collected from December 2011 onwards.

Table 5. Usability problems and other suggestions reported by the author and the users and developers of the observation system.

Problem or suggestion	User interface	Suggested improvement	Fixed?
Overall, the system is scattered over multiple web pages and terminal windows	All	Gather the most important tools for conducting observations in one place.	Yes (only partly)
Tau values determined manually	-	Determination of the tau value should be automatized and/or be based on real measurements if possible.	No
Good quality observations have to be updated manually --> It's hard to keep track of sources that have recently been observed. This makes it harder to assess what sources need to be observed next. Having to memorize the latest good quality observations increases the observer's memory load. Alternatively, checking the observation results takes extra time and breaks the work flow.	-	Automatic initial classification of observations immediately after observation is finished --> confirmation from user. A quick fix solution: Add a checkbox to each observation in obs_summary (or obs_delcom), so that an observation can be checked as a good quality observation.	No (Quality tag proto)
It's not possible to add sources for observation while the antenna is observing.	automap.pl	Creation of an "observation queue" type of a system, where the observer can freely add and remove sources to/from a queue while the antenna is observing.	No
No immediate feedback on the flux calibrated results of the current observation.	obs_summary.pl	Preliminary flux calibration of the current observation. Result presented to observer graphically as a part of a time series of the current source.	No
Multiple target dots that are near each other make it difficult to select the right target in automap.	automap.pl	Add a simple sky view zoom feature so that the distances between the dots become large enough	No

Problem or suggestion	User interface	Suggested improvement	Fixed?
The current ww/YY notation for good-quality observations is not accurate enough; due-markers change abruptly in automap come Monday	gixlist37.pl gixlist22.pl	As a part of automated good-quality observation suggestions, give an exact date that the observer can verify	No (quality tag proto)
Insertion of tau-values has to be integrated s.t. it's simpler to manage tau-values tied to observations	obs_summary.pl tau_add.pl tau_query.pl	Implement an Excel-like cell copying s.t. A tau-value can be copied to multiple measurements with a simple mouse gesture	Yes
Automap should contain filtering options to select only e.g. strong sources or targets of certain type	automap.pl	Add e.g. some check boxes for filtering sources and an approximate flux level slider	No
obs_summary should have an option for filtering measurements based on observer's name	obs_summary.pl	Add filtering based on observer's name	Yes
obs_summary should work faster when selecting long time periods	obs_summary.pl	Optimize obs_summary for larger data sets	No
It's annoying to write strings like "20111209122345" for obs_summary date fields	obs_summary.pl	Add some optional GUI feature for more user friendly input	Yes
There should be a simple GUI for autoheat. Repetitive use of autoheat is boring and slow.	autoheat.pl	Implement an optional GUI for auto-heat into the measurement program. The way heating is done now, when each heating session shall be input individually, is not convenient at all. The hypothetical GUI should implement more generalized approach, for example, when observer can set the start time (in the evening), stop time (in the morning), and the heating-on and -off cycle lengths, and than the heater will automatically be switched on and off within the set period, according to the cycle parameters. In a long run, it would be great to automatize the heating system completely, that it would learn to switch on and off in the proper moments automatically, based on the previous records of weather information and sensor measurements, combined with the records of the heating sessions. Of course the automatic heating system should have a manual option. (ER)	No

Problem or suggestion	User interface	Suggested improvement	Fixed?
Updating the source list (for example, adding new sources) requires modification of several files with exactly the same content (i.e. full and high priority files must be updated separately + gix_sources.list + textfield file). Very tiresome.	-	There should be one master file that is updated. Other files will be automatically generated based on that.	No
It would be nice to be able to update the offset-database directly from a 5-p observation	-	Create an interface through which observer can, after a good-quality 5 p obs., press a button "enter these values into the offset database" instead of manually editing it.	No
Collection of all offset data	-	From all good-quality 5 p observations the offset data (proposed offsets in a certain direction) will be automatically collected into a (nicely readable...) database. Then observer should be able to choose which offsets of these entries will be used in the de facto offset-database. Good quality here could be defined by rms and drift information, to leave out really anomalous data.	No
Single integrations directly through automap/skyview	automap.pl	Enable observer to click a source on the skyview plot, tick a box for 1/5 p obs and choose integration time (as in current automap for autolists) and then press "observe" for a single integration, instead of having to do the online command. Basically this should be a single interface: in the beginning of a session observer enters his/her initials and defines the default integration time, and then either creates an autolist or executes a single integration. Ditto for calibration.	No
Obs-delcom: update button also to the top	obs_delcom.pl	<p>Add another "update" button to the top, above the comment field. Makes it quicker to add a new comment, with no need to scroll down a lot.</p> <p>Comment from Anne: I asked Miika to add a button to the top, and it was there for a while (very handy) but it has now disappeared for some reason???</p> <p>Comment2: a test version of obs_summary was used in the vnc, it did not have the submit button on top. Restarted obs_summary in vnc should now be ok.</p>	Yes

Problem or suggestion	User interface	Suggested improvement	Fixed?
Comment fields to attach to solar observations	obs_summary.pl	Add a feature through which one can add comments about weather etc. also during a solar observation. If weather is relatively bad, there may not be any quasar obs. taken close by, so obs_summary does not show any information related to that. I don't know if there should be a separate solar obs log or what, this probably needs to be discussed with Juha K. Also, the observer information (initials) should somehow be attached there too, for possible further credit.	No
Tulen hulluksi tau-tietokannan kanssa!!!!	tau_add.pl	The current tau-add thingy is impossible to use. Fix at least: 1) Tau values can be added according to the UT times shown by obs_summary. 2) If error occurs, the error message should show a correct point of conflict (now it shows something arbitrary) and NOT reset any values in the fields; 3) Enable also an entry by ID for single desperate cases.	Yes

5.2.2 Contextual inquiries

The problems found based on analysis of the contextual inquiry data are presented in Tables 6-13. The results of the preliminary interviews conducted in the contextual inquiries are presented in appendix B.

Table 6. Usability problems found based on the contextual inquiries related to the task of adding or modifying tau values.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Some observations are left without a tau value even when a tau value in the database clearly overlaps the observation.	4	obs_summary.pl	-	Long term: Automate tau value determination and data integration so that observers only need to make tau observations.	Yes (Short term)
Upon erroneous input to tau-add.pl, the user interface is reset, forcing user to start the inputting from the beginning, instead of fixing the erroneous input.	3	tau_add.pl	Error recovery	Short term: Integrate tau value adding to observation summary. Integrate sensor data needed in determining tau values to observation summary.	

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
When an erroneous time span or tau value is input to the database, recovering requires searching the inputted value from the database and then deleting it, using two separate tools.	3	tau_add.pl tau_query.pl tau_delete.pl	Error recovery, Recognition rather than recall (memory load)		Yes
Adding tau values to the database from comments in the results table requires jumping between obs_summary.pl and tau_add.pl and requires memorizing time periods and tau values.	3	obs_summary.pl, tau_add.pl	Recognition rather than recall (memory load)		
Tau values have to be determined manually, which is very ambiguous.	3	-	-		
Data needed in determining tau is scattered over multiple tools and is not available in obs_summary.	3	obs_summary.pl	-		
The accepted tau value spans for summer and winter are not visible in obs_summary.pl or in tau_add.pl. If the user does not remember the values they have to be looked up from the localweb wiki.	3	obs_summary.pl, tau_add.pl	Recognition rather than recall (memory load)	Have the tau value spans visible in the tool where the value is determined.	

Table 7. Usability problems found based on the contextual inquiries related to adding or modifying comments and delete tags to observation results.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
It is not possible to add comments to ongoing observations. Observers make temporary comments or mental notes when they would like to add a comment to the currently running observation.	3	obs_summary.pl, obs_delete.pl	Recognition rather than recall (memory load)	Add support for commenting currently running observations. Also if an observing queue is implemented, add support for commenting upcoming observations.	No
Adding comments and delete tags to results requires opening a separate tool.	2	obs_summary.pl, obs_delete.pl	-	Integrate comment and delete tag adding to obs_summary.pl	Yes
obs_delete.pl lacks color coding for results which makes marking sources for deletion harder.	2	obs_delete.pl	Recognition rather than recall (memory load)		
Determining which results to delete is monotonic, repetitive and time consuming.	2	obs_summary.pl, obs_delete.pl	-	Integrate all the numerical and other criteria for deleting clearly to observation summary e.g. with table cell color coding and/or automatic comments in the comment fields.	Yes
Checking fulfillment of some of the deleting criteria requires the use of separate tools or live following of the results. (E.g. bad 5-point observation pointing, rain, unsettled receiver levels after heating) Also, some observers manually mark e.g. rain periods to the comment fields of observation summary.	2	obs_summary.pl, screen on daqqr, MB.cgi	-		
Determining which results to delete requires searching and remembering numerical rules.	2	obs_summary.pl, obs_delete.pl	Recognition rather than recall (memory load)		
No warning given to user about one character comments not being written to the database.	3	obs_delete.pl	Error prevention, Error recovery	Clearly warn user that one character comments won't reach the database. Remove limitation of comment length.	Yes (Limit removed)

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Some observers use a personal log of observing events such as heating cycles, weather changes etc. This might be an indication of a deficiency in the system.	2	-	-	-	No

Table 8. Usability problems found based on the contextual inquiries related to choosing sources for observing.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Observers aren't aware of the meaning of observations marked with a red star in the source list.	2	automap.pl	-	Replace term "Marked W/ RedStar" with "Urgent" or "Observe now!"	Yes
Observers don't notice observations with the symbol meaning a source marked with a red star in the sky view.	2	automap.pl	Match between system and real world (speak the users' language)	Update the symbol on the sky map to a more prominent one. Replace the term "Marked w/ RedStar" with "Urgent" or "Observe now!"	Yes
Multi-frequency sources are not marked on the sky view. Observers spend extra time to familiarize with mf-sources.	3	automap.pl	Recognition rather than recall (memory load)	Mark multi-frequency sources on the sky view. On the long term, allow automatic mf-status updates directly to sky view from online and other sources.	Yes
Good quality observations have to be manually updated to make the sky view "due" markings reliable.	3	automap.pl gixlist37.pl gixlist22.pl	-	Implement automatic initial good quality determination for finished observations.	No
In the case of uncertain good quality markings, the last good quality observation of a certain source needs to be manually checked in observation summary.	2	automap.pl	-	Make good quality flags visible and modifiable in observation summary. Automatic calculation of detection / non-detection.	

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Marking good quality observations requires the use of a separate tool.	2	obs_summary.pl gixlist37.pl gixlist22.pl	-		No
When marking a good quality observation, the observer needs to calculate if the observation is a detection ($S/N \geq 4$) or a non-detection ($S/N < 4$).	2	obs_summary.pl gixlist37.pl gixlist22.pl	Recognition rather than recall (memory load)		
Determining which results to mark as good quality requires searching and remembering of numerical rules.	2	obs_summary.pl gixlist37.pl gixlist22.pl	Recognition rather than recall (memory load)	Have relevant information on deciding if an observation result meets good quality criteria visible in the tools where the decision is made. In practice, cell color coding in observation summary could be utilized.	
The sources on the sky view are in places cluttered, which makes it hard to select individual sources.	3	automap.pl	-	Implement a sky view zoom feature Allow selecting which sources are displayed on the sky view	
Sources close to the Sun are not marked on the high priority “most due” source list.	2	gixlist37.pl gixlist22.pl	-	Mark sources close to the Sun on the list so that they don't produce “false positives” when looking for sources to observe. Integrate the information provided by the “most due” source list version to the sky view. This way high priority sources most in need of observing can be seen without leaving the sky view.	

Table 9. Miscellaneous usability problems found based on the contextual inquiries related to the observation summary tool.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Connecting a tau value to the time stamp of the observation on the same row is difficult because of the distance between the values.	2	obs_summary.pl	-	Implement mouse hover row highlighting to help connecting values on one row with each other.	Yes

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
It is not possible to open observation results based on observing shift or by observer name.	3	obs_summary.pl	-	Add possibility to filter results by observer name Add possibility to browse observation shifts and open observations based on shifts	Yes (Filtering) (Shift browse: UI proto)
It is not readily apparent whether or not observation summary is showing the latest available observations.	2	obs_summary.pl	System status	Consider adding a button for showing last N days or hours of observations. Consider adding a status display where it can be seen if observations shown are from the past or if they include the latest ones.	No
Returning pre-processed data files for final reduction requires remembering the file naming convention and where the files should be saved.	2	obs_summary.pl	Recognition rather than recall (memory load)	Remove the need for observers to return data files for final reduction. There could be a “Completed” or “Ready for reduction” flag that could be given to all the data from a certain observation shift after pre-processing. The lead scientist could export the data in one file from all the shifts that are ready.	No (UI proto)
Exported data files open in same browser tab as obs_summary.pl. Can cause unintended closing of obs_summary.pl	2	obs_summary.pl	Error prevention	Short term: Open the exported data files in a separate tab. Long term: Remove the need for observers to return data files for final reduction.	No
Scandinavian umlaut characters don't print correctly in the exported data files	2	obs_summary.pl	-	Add support for special characters or automatically convert problematic characters to ones that display correctly (ä → a, ö → o).	No
Long load times of observation summary may cause the user to prematurely press one of the buttons on the page.	2	obs_summary.pl	Visibility of system status	Consider deactivating the buttons until the page has fully loaded.	No
The observation summary table has some rarely used columns like, the tau ID (tID) and observation ID.	2	obs_summary.pl	Aesthetic and minimalist design	Remove or minimize the redundant columns in the table.	Yes

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
The connection between the weather widget and the gray highlighting in observation summary version 1.36 might not be clear to all observers.	2	obs_summary.pl	-	Change highlight color to a more bright and noticeable color. Explore ways to create a visual connection between the selected observation and the widget.	Yes (Color)

Table 10. Usability problems found based on the contextual inquiries related to giving observation commands and running observations.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Giving a single observation command requires memorizing offsets or azimuth-elevation coordinates AND/OR finding or searching for an observation command template OR creating and copying a single-source autolist file.	3	automap.pl screen on daqqr	Recognition rather than recall (memory load), error prevention	Allow adding sources to a queue directly from the sky map. From the queue the sources can be automatically selected for observing on their turn. Remove the need to remember any observation commands or variables in the process.	No
Default autolist file naming convention is different from the convention the observers actually use.	2	automap.pl	Recognition rather than recall (memory load)	Allow adding sources to a queue from which sources are automatically selected for observing on their own turn.	No
The system does not show the remaining sources and the expected end time for a running autolist	3	automap.pl watch_results	Recognition rather than recall (memory load)	Implement a source queue system where currently running and upcoming sources can be clearly seen along with estimates for time left and an end time.	No
When estimating the end time of an autolist, UTC-EET conversion has to be performed	2	automap.pl	-	In a future system, provide an end time estimate also in local time if user interface doesn't get too cluttered and/or users too confused.	No

Table 11. Usability problems found based on the contextual inquiries related to using the radome heater.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Inputting heater cycles is repetitive, prone to errors and requires concentration, especially with uneven cycle lengths.	3	autoheat.pl	-	Implement a user interface that allows programming both single and multiple heater cycles without the need to input all cycles individually.	No
Once a heating cycle has been input, the cycle can't be modified.	2	autoheat.pl	-		
Optimal use of the heater is stressful in some situations. Many guidelines apply: <ul style="list-style-type: none"> • Avoid heating in vain • Prevent moisture from forming on the dome • Avoid starting heating during an observation • Observe a calibration source before and after the first heating cycle Heater is too powerful for moisture removal.	3	-	-	Replace the heater with a less powerful one.	Yes (Two electric heaters acquired. Independent of this thesis)
Some observers mark down heater cycles in the result comments one cycle at a time. The process is repetitive and time consuming. Typically long heating cycles are marked near a time when the cycles start with the start time, cycle length and end time.	2	obs_delcom.pl	-	Implement a log where all actual heating cycles are written. Have observation summary access that log and automatically mark heating times into the data in a format most suitable for observers and the final reducer.	No

Table 12. Usability problems found based in the contextual inquiries related to the all-sky images and the all-sky image archive.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Unlike all the other tools in the observation system, the all-sky image archive uses EET time, which confuses the observers and causes problems.	3	All-sky archive	-	Modify the all-sky archive to use UTC time.	Yes
Some all-sky images have white balance problems and can be very dark, which makes detecting the conditions very hard.	2	-	-	Consider fixed white balance settings. Use longer exposure times, stack multiple images together or adjust the image brightness, contrast and/or gamma value.	Yes (WB, gamma adjust)
It can be hard to determine which way the images are browsed to see newer images, because newest images are on the top left and older images are browsed by clicking to the right. This can be counter intuitive.	2	All-sky archive	-	Consider changing the browsing direction and “newest at top left” logic to “newest at lower right”.	No

Table 13. Miscellaneous usability problems found based on the contextual inquiries.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Plotting long time periods takes a long time (MB)	2	MB.cgi	-	Make plotting faster. Create an alternative tool for MB that allows instantaneous access to relevant weather information.	Yes (weather widget)
For live monitoring of e.g. relative humidity, two tools are needed so that both recent development as a graph and the current situation as a number can be seen.	2	MB.cgi, antenna & weather	-	Create a weather monitoring tool that shows both recent development as a graph and latest value as a number.	No (Prototype)

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Setting up the MB tool is slow and all observers don't use the saving option.	2	MB.cgi	-	<p>Short term: Create a settings file in MB that contains values that most observers prefer using.</p> <p>Long term: Create an alternative tool for MB that allows instantaneous access to relevant weather information.</p> <p>Integrate weather and other information to observation summary.</p>	Yes (weather widget)
The fixed resolution of the VNC desktop causes problems. Too large for small laptops, too small for many user interfaces.	3	VNC session	-	<p>Help observers use VNC viewer clients with scaling, which allow full viewing of a VNC desktop on a screen with smaller resolution than the VNC desktop.</p> <p>Move to use web interfaces where different resolutions can be taken into account.</p> <p>Take low resolutions into account in user interface design.</p>	Yes (Scaling clients suggested)
When an observer has a local web browser and the VNC session open, she can get confused with what she was doing in which window.	2	VNC session	-	Having a VNC session with larger or dynamic resolution or moving the observation system completely to web-based user interfaces can help desktop clutter.	No

5.2.3 Heuristic evaluation

This section presents the results from the heuristic evaluation conducted on the first iteration of the study. The results are presented in Table 14.

Table 14. Usability problems found with heuristic evaluation.

Problem	Severity	User interface	Heuristic	Suggested improvement	Fixed?
<p>The controls for adding and removing sources in the autolist are deceptively far away and separated by horizontal lines from the list that their functionality affects.</p> <p>("Add" and "Del. last target" buttons and drop down menu of sources)</p>	2	Autofile generator (automap.pl)	Gestalt laws of proximity and closure.	<p>The controls should be closer to the list so that the user intuitively understands that the controls are connected to the list. Also, the controls should be inside the same enclosure as the list (horizontal lines).</p> <p>Due to limited space, can be hard to implement.</p>	No

Problem	Severity	User interface	Heuristic	Suggested improvement	Fixed?
There are some unused controls in Autofile Generator: The list type "Cntbs" radio button won't be used again for sure, with contobs being decommissioned. Also the "Gen. autofile" button is redundant as there are buttons for both 22 GHz and 37 GHz list generation.	2	Autofile generator (automap .pl)	Aesthetic and minimalist design	Remove the redundant controls (at least from view) to simplify the user interface.	Yes
The Add-button in Autofile generator does not stand out from the other buttons surrounding it (especially the Trail-button). This will make it harder to distinguish as an important button and make finding it harder.	2	Autofile generator (automap .pl)	-	Make the button stand out as an important button and locate it so that the user associates it with adding sources to the autolist. The current location does not suggest a strong connection with the list.	No
The update button in offsetapp.pl is hard to notice.	2	offsetapp.pl	-	Make the button larger and maybe also relocate it e.g. on the bottom right corner of the offset table. Add more separation between the button and the table.	No
The information presented in the "Autolist" listbox is hard to read. It is hard to distinguish one source from another and it's hard to find single parameter values for a source.	2	Autofile generator (automap .pl)	-	Arrange the information in a table with clearly distinguished rows and columns for each source and their parameters, respectively.	No
If the user inputs an invalid value to the "Dur. (s)" field, e.g. letters, the system defaults to the value 1800 without noticing the user of invalid input.	2	Autofile generator (automap .pl)	Error prevention, error recovery, system status	Give feedback on invalid input and let user re-input the value. Alternatively, switch the text input field into a drop down menu with the values 1200, 1400, 1600, 1800, because these are the only integration time values in use in the observations. (The drop down menu suggestion was also given by an observer in an observer's meeting)	No

Problem	Severity	User interface	Heuristic	Suggested improvement	Fixed?
It's not visible which offset values have been altered in offsetapp.pl. This may lead to erroneous and unwanted value modifications.	3	offsetapp.pl	Error prevention, system status	Mark altered offset values with a special pattern or symbol. This way user can clearly see, which values he/she has altered before pressing "Update".	No
"NA"-values are changed into "+0.000" values when updating with "NA"	3	offsetapp.pl	Error prevention	Make the program obey what the user tells it. "NA" is very different from "+0.000".	Yes
The term "Marked w/ RedStar" is technical and does not correspond well to a source in urgent need of observing.	2	gixlist37.pl, gixlist22.pl, automap.pl	Match between system and real world (speak the user's language)	Use a term that better tells observers that the source in question is in urgent need of observing. Suggestions: "Observe now", "Urgent"	Yes
The "Help" button does not provide any useful help.	2	automap.pl	Aesthetic and minimalist design	Until the button provides some actual help, hide the button from view.	Yes
It takes some extra effort to add a trail to a source in automap. The process of drawing a trail for a source can go like this: 1. Looking at a source on the sky map and wanting to draw a trail for that source 2. Moving the mouse cursor on the source and getting the name of that source 3. Finding and selecting the name of that source from the dropdown menu on the right 4. Pressing the "Add trail" button. This adds unneeded load on the memory of the user between looking up the name of the source and looking for it in the dropdown menu.	2	automap.pl		There could be e.g. a context menu selection available for drawing a trail for a source when right clicking on a source.	No

Problem	Severity	User interface	Heuristic	Suggested improvement	Fixed?
It can be tedious to figure out a proper offset value for a source because automap displays the offset value for only the current location of the mouse cursor.	2	automap.pl		Assuming the offset values automap uses are reliable, an optimal solution would be: When a source is clicked and added on a list/queue, automap calculates an optimal offset value based on the movement of the source during the whole observation. Even more optimal: measure would automatically change the offsets in real time during an observation. The biggest problem in this solution is assuming the offset values actually are valid, as they often are not. Another solution is that when a source is hovered with the mouse cursor, a track of the movement of the source with the current integration time is drawn on the map. This way the user could approximate an optimal offset value more easily than now.	No
It is hard to determine how long the currently running observation will last and at what time will it end.	2	watch_results	system status, memory load	This information is needed at least for determining a precise starting time for an autolist. Calculate the estimated end time in watch_results and display it to the user. E.g. "Est. completion time: 359 sec / 13:25 utc."	Yes
It's somewhat laborious to figure out which sources and when to observe during multifreq. campaigns.	2	automap.pl, multifrequency info page, gixlist37.pl, gixlist22.pl		In general, the system should tell the user what sources need to be observed. The system should tell the observer that a.) there are multifreq. campaigns going on right now and b.) the observer should observe certain sources belonging to the campaign at certain times (i.e. would tell to the user if mf. camp. observations are needed right now). This would transfer the knowledge of the situation to the system and minimize the observer's memory load. In practice the system could read a file of a certain format into which an mf. campaign expert would have fed the necessary information (e.g. via a web form).	Yes (MF-sources marked on map)
The time format (YYYYMMDDhhmmss) in the time fields is hard to decipher.	2	obs_summary.pl	Aesthetic and minimalist design	Change the time into a more readable format, e.g. YYYY-MM-DD hh:mm:ss	Yes

5.3 Analysis and recommendations

In this section, the most important findings of the first iteration of the usability evaluation are summarized. The findings are based on analysis of the usability data gathered using contextual inquiry, heuristic evaluation and user feedback. The findings are categorized by problem area, based on interpretations made on tasks and high level goals. Where possible, the problem areas are classified by task, by user interface and by other binding themes. These results don't yet attempt to systematically portray the tasks and goals in the work.

After the presentation of each problem area, recommendations for changes to the observation system and their technical implementations are presented. The recommendations are presented in Tables 15-17. Results from the evaluation of the changes that were implemented are presented in section 5.6.

The most pressing problems were related to having too many tools for accomplishing single tasks and relevant information being dispersed over multiple tools. The solutions recommended are mostly related to integrating functions and information to certain central tools.

Table 15. Recommendations for changes in the autofile generator tool based on the first iteration of evaluation.

Recommendation	Status
Remove unused controls from the autofile generator.	In operational use.
Change the sky map symbol of sources in need of special attention. Change symbol (marked with a red star in source list) into a white triangle and mark multi-frequency sources. Mark multi-frequency campaign sources as sources in need of special attention, making them on the sky map of the autofile generator and real time sky view tools.	In operational use
Change the term "Marked w/ RedStar" to "Urgent! (multifreq or due)" in automap.pl.	In operational use.
Implement a zoom functionality to the sky map.	Suggestion
Allow selecting which sources or source types are shown on the sky map in automap.pl.	Suggestion
Allow manual highlighting of sources on the sky map.	Suggestion

Table 16. Recommendations for changes in the observation summary tool based on the first iteration of evaluation.

Recommendation	Status
Integrate the tau value adding and modifying functionality to the observation summary tool.	In operational use. Five usability tests.
Make the recommended tau value spans easily available/visible for the observer.	Suggestion

Recommendation	Status
Integrate weather and other sensor data needed in determining observation quality, determining tau values and noticing changes in observing conditions into the observation summary tool.	In operational use. Five usability tests. (Weather widget)
Integrate the comment adding and modifying functionality to the observation summary tool.	In operational use. Five usability tests.
Remove the one character comment length limit from the system.	In operational use. Five usability tests.
Add support for adding comments to a running observation to the observation summary tool.	Two static HTML prototypes shown to users.
Introduce a three-level “quality” property for observation results. (Integrate the delete flag adding and marking of the last good quality observation into a tool for manipulating the quality property in the observation summary table.)	UI prototype implemented. Limited testing with users. Awaiting more extensive technical changes to the system.
Add automatic initial recognition and flagging of good quality observations. Include automatic recognition of detection and non-detection.	Suggestion
Make the rules for flagging an observation as good quality or for deletion clearly visible or at least quickly accessible.	Suggestion
Integrate delete-flag marking to observation summary tool. (Before “quality” property is implementable.)	In operational use. Five usability test.

Table 17. Miscellaneous recommendations for changes based on the first iteration of evaluation.

Recommendation	Status
<p>Implement a source queue system for running manual and automatic observations:</p> <ul style="list-style-type: none"> • Allow adding sources to the queue from the sky map and from a source list. • Allow modifying, moving and removing observations on/from the queue. • Remove the need to remember or copy commands or parameters. • Have running and upcoming observations clearly visible. • Have estimates for remaining time and for end time clearly visible for queue and for current observation. • If feasible, have end time estimate visible also in local time. 	Suggestion
Implement support for adding comments to queued observations.	Initial static prototype

Recommendation	Status
Migrate the source list into a database. Create a web user interface that allows modifying all entries in the source list.	Suggestion
Integrate the observation system into a logical collection of interconnected tools: <ul style="list-style-type: none"> • Organize the system into a relatively small number of “places” for accomplishing the most important tasks. • Add navigation to these places that allows quickly moving inside the system without the need to use browser bookmarks or links in the wiki pages. • Examples of possible places in the system: <ul style="list-style-type: none"> ○ Source selection, source queue and currently running observation ○ Observation results viewing and data processing ○ Weather data ○ Source list 	Suggestion

5.3.1 Choosing sources for observing

A central task in conducting quasar observations is choosing which sources to observe. This task is common to all observers in both automatic and manual observations. Several problems related to choosing sources were found in the evaluation. The most serious problem, based on the amount of extra work it causes to the observers, was getting information on multi-frequency campaign sources. Another major problem was that the sky map where sources are typically selected for observing is cluttered. This makes it difficult to distinguish sources from each other. Also, in general, the process of choosing sources inherently requires concentration from the observer. In the beginning of the study, the senior scientists set a goal for developing a feature to allow automatic selection and observation of sources. The feature would be used when the observer was so busy that no time was available for creating and running an automatic list of sources.

Source list

In general, properly addressing the problems associated with choosing sources will most likely require fairly extensive modifications to Metsähovi’s source list system. The current system is based on multiple text files that contain information on various attributes of the sources. These attributes include e.g. the latest good quality observation time and information on whether the source is to be observed in 1-point or 5-point observing technique. Adding new sources or new source properties requires modifying many or all of the individual text files. This causes extra work and increases the possibility of making erroneous changes. It would be reasonable to move the source list into a database where there would only be one occurrence of one source. A practically unlimited number of attributes could then be added to each source. A web based user interface could be created to support all necessary modifications.

Multi-frequency sources

Multi-frequency campaign sources (mf-sources) are sources that are observed in Metsähovi together with other observatories located around the world and in space. Multi-frequency sources are typically the highest priority sources to be observed, but in the evaluated system they weren't marked on the sky map where sources are most typically selected for observation. Instead, information on which sources are part of multi-frequency campaigns was listed on a separate web page. Observers either memorized which sources were mf-sources or created spreadsheets containing mf-source visibility times. Especially creating and following the spreadsheet was laborious for the observers and presented an extra step in the source selection process.

To remove the need for unnecessary memorizing and the creation of spreadsheets, the mf-campaign status of a source would have to be clearly visible to the observer. Because of the high priority of mf-sources, the symbol for marking the sources would have to stand out on the sky map. A quick solution was possible, because there was an unused white triangle symbol in automap.pl previously used for marking Planck satellite sources. After simultaneous observations with Planck were over, the symbol had been left unused.

The unused Planck symbol was adopted for marking multi-frequency sources and other sources in need extra attention on the sky map. The white triangle symbol was given a new annotation below the map: "Urgent! (multifreq or due)", "due" meaning sources that are due for observing. At the same time, the old yellow circle symbol for marking sources in need of extra attention was removed from use. The yellow symbol was hard to notice on the sky map and most observers also didn't know the meaning of the "marked w/ RedStar" annotation related to the symbol. In an interview, the lead researcher mentioned about observers not observing sources even when she marked them with a red star in the source list. This served as proof of observers not paying attention to the yellow symbol. Sources in need of extra attention are marked with a red star symbol in the web-based source list table, and were previously marked on the sky map with the yellow circle symbol. After the change, both mf-sources and other sources in need of special attention should be easier to distinguish from other sources on the sky map. The changes are depicted in Figure 21.

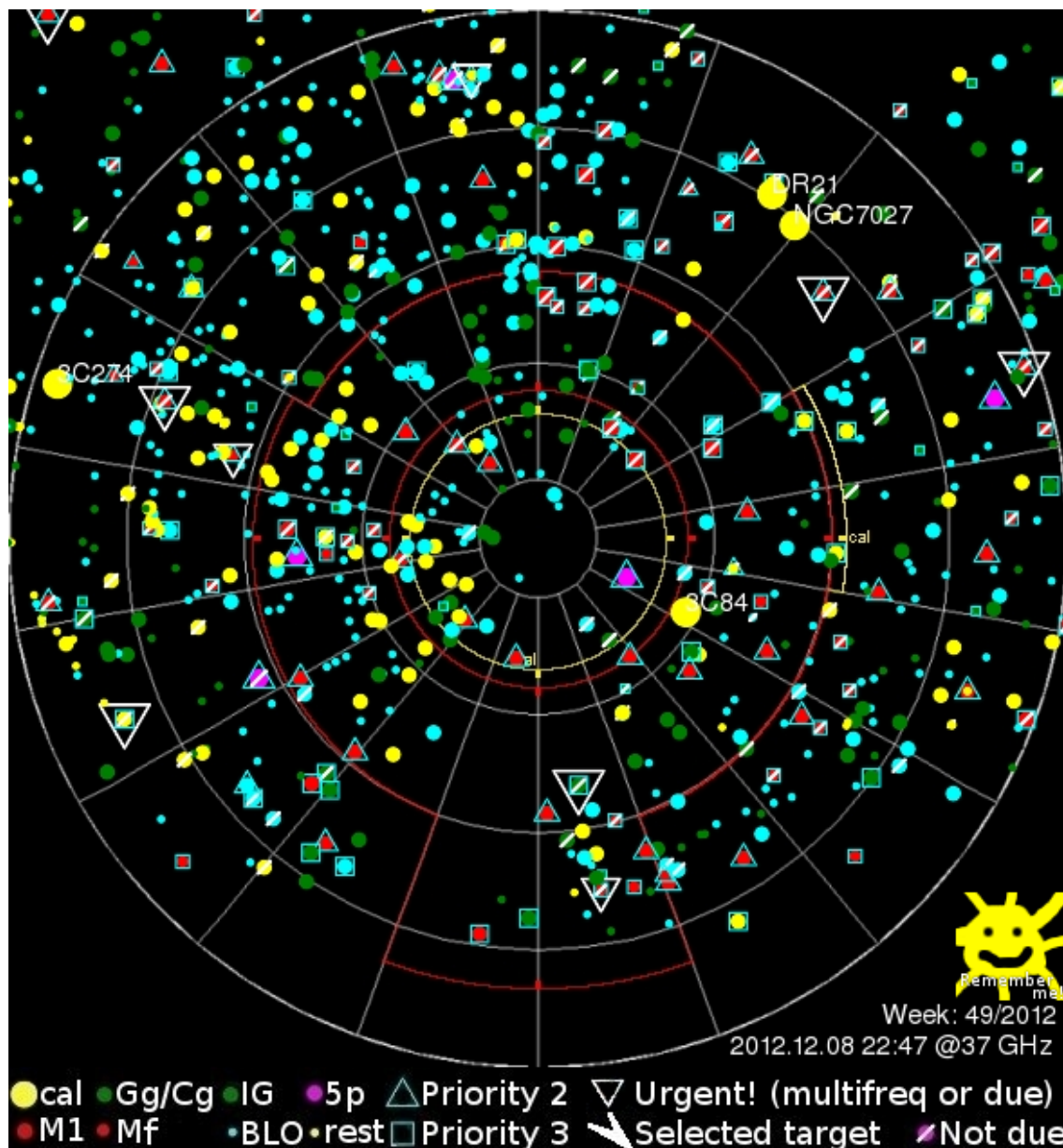


Figure 21. The revised sky map of the autofile generator and real time sky view tools. Based on the first iteration of evaluation, multi-frequency (multifreq) sources and other sources in need of special attention (due) were marked on the sky map with a white triangle symbol. The symbol is annotated by the text “Urgent! (multifreq or due)”.

Cluttered sky map

The other major problem in choosing sources for observing was the cluttered sky map. Sources are selected for observing by clicking them on the sky map. In many places on the map, the sources are so close together that clicking a specific source is very difficult. Observers also hover over the sources with the mouse cursor to see a flux curve graph the time of the last good quality observation of the source. This information is in a key role when deciding which sources to observe. Getting the system to show the flux curve of the desired source is very difficult for clustered sources.

To solve the clutter problem, two different approaches could be combined. The first solution is to implement a zoom feature to the sky map. By zooming closer to a cluster of sources the sources become easier to distinguish from each other, making it easier to

hover over or click a source. The second solution is to make it possible to select which sources and source types are shown on the map. Typically observers look for high priority sources to observe and only if there are very little of these sources left do they switch their attention to low priority sources. Low priority sources are usually also dim and observing them requires good weather conditions. Thus, displaying low priority sources on the sky map is usually not necessary and just adds to the clutter.

Due to the complexity of the technical implementation of the automap.pl Perl script that generates the sky map, it was not possible to create a functional prototype of the solutions within the confines of this thesis.

5.3.2 Running observations

When observing sources, the observer can either input each observation command manually to run one observation at a time or she can create a list of observing commands which is run automatically. Several areas for improvement were noticed in the tasks of running manual and automatic observations.

From choosing a source to giving an observation command

Extra effort is needed in transferring command information from autofile generator to the “screen on daqger” terminal. As a first step after choosing a source, some observers memorized or wrote down offset values or azimuth-elevation coordinates of the selected source. After this, they typically browsed the command history in the terminal window to find a previous command to use as a template for the new command. They then modified the old command with the new source name, integration time, observation type and offset parameters. Some observers checked the command once or many times during or after the modification the command. After this the command was executed. Another method of giving a single observation command was the generation of a single source “autolist” file. From the file, a complete observation command was copied directly to the terminal and executed. This method reduces the chance to make errors in the process and thus also reduces the need to check the command's validity before execution.

Creating autolists and observing automatically

When it comes to lists of sources to be observed automatically, it is not possible to add sources to a list that's running. The system does not support modifying an existing list. This means that the observer has to estimate the ending time of the currently running list and create a new list with the previous lists ending time as a start time.

A solution to these problems, which cause extra work and stress to the observers, should allow a quick start of observations after a source has been selected. The solution should also allow queuing sources instead of forcing observers to create separate fixed list files one after the other. These solutions were not addressed during the thesis project because of a lack of time. Detailed design and user testing will be done after the thesis project is finished.

5.3.3 Adding comments to the data

Adding comments to the observation data is in a key role when pointing out the most important changes in observing conditions. Comments are also needed when deciding which observations are not of sufficiently high quality when it comes to error values and the accuracy of the telescope pointing. Comments are needed both during observations and in the final reduction.

Problems

Two main problems related to adding comments were found. The first one is the use of a separate tool in adding comments and the second one is that there's no possibility to add comments to ongoing observations. A typical scenario where an observer adds a comment to the observation data is when he or she notices a change in the conditions or uses the dome heater. The observers usually desire to attach the comment to an observation that is currently running, right after noticing changes in the conditions. This was not possible in the system that was evaluated. Comments could only be attached to observations that have appeared in the observation summary (obs_summary) and observations delete status and comments (obs_delcom) tools. This forced the observers to memorize the comments, make notes on paper or to files on a computer. This is extra work and it would be best if comments could be added to running observations.

When adding comments to observations that already are on the results table, the observer typically notes a change in conditions at a certain time or a deviating value in the results themselves. In such a case, the observer has to open the separate obs_delcom tool which displays the data in a table in a somewhat different form than the obs_summary tool. The table doesn't have color coding and the table rows are higher than in obs_summary. This makes it hard to find the specific observation result which the observer already had chosen in obs_summary. This separation of adding comments slows down the observer's work and adds short term memory load. The observer needs to remember which observation she was looking at in obs_summary and then find that observation in obs_delcom.

Solutions

It was determined that the problems could be solved by integrating the comment adding functionality to obs_summary. This would eliminate the need to open a separate tool and locate the observation result anew. In some ways the simplest option would have been to add editable text fields directly to the comment column in obs_summary's table. The observer could simply move the cursor to the comment field of an observation result and write a comment. This, however, was deemed too difficult to implement. There might have been a risk of accidental inputs to the comment fields. It would also have introduced a technical problem on when and how to input the changes in the comments to the database. Would the insertion be executed on every keystroke, on the press of the enter key or on the moving away of the cursor? Sending AJAX calls to the database on every key press might be too heavy on the network connection. As some observers conduct observations on mobile internet connections and without the VNC desktop, too heavy a reliance on the reliability of the network might cause problems. It was deemed that there was not enough time to generate solutions to these challenges, and it was decided that a technically more simple solution would have to suffice.

The second simplest option was to implement a commenting dialog that opens when a comment field is clicked in the table. The comment could be manipulated in the box and either inserted in the table or canceled. This solution would follow the direct manipulation interaction style (see section 3.4.3) in the way that the comment fields themselves would be clickable. This solution was implemented and tested with observers. A screen capture of the commenting solution is presented in Figure 22. The solution would free the observers from the problems that arose from the use of a separate commenting tool, i.e. the lack of color coding and finding the target result in the obs_delcom table. Comments could be added and modified without losing sight of

the primary results table. The commenting dialog would require some extra mouse clicks compared to the direct-in-table commenting, but would be easier to implement.

0.28	2	152	419.88	-7.2	-11.3	81.8	0.047	2	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.	0.	0.	0.	0.	0.	0.	0.	0.	verama37
0.28	33	1833	420.53	-6.9	-9.9	83.3	0.047	2	skytemp going down to -20
0.25	26	1832	418.84	-6.8	-9.8	84.6	0.047	2	verama37
0.31	48	1831	418.84	-6.9	-9.8	84.9	0.047	2	verama37
0.38	2	150	418.84	-6.9	-9.8	85.1	0.047	2	verama37
0.24	14	1202	418.84	-6.9	-9.6	83.3	0.047	2	verama37
0.31	36	1831	419.80	-7.0	-9.4	82.7	0.047	2	verama37

Figure 22. Integrated commenting in the observation results table (*obs_summary.pl*). When the comment in the table is clicked, it gets selected and a commenting dialog box opens. The selected comment is highlighted in the table. The comment can be modified in the text edit field in the dialog box.

The evaluated observation summary tool displayed the ongoing observation but didn't allow attaching comments to it. To fix this problem, a comment field would have to be integrated to the ongoing observation's user interface element. A quick solution wasn't technically feasible due to the way the current observation is displayed separate from the finished observations table.

One option to consider in future development is to have the current observation displayed as one observation among others in the results table. This solution could also include presenting the queue of upcoming sources in the same table, using the same logic as e.g. play lists of music players. Because of the extensive changes to the observation system required, these redesigns were left for later design iterations. A static HTML demo page was prototyped to give an example on one way to present the upcoming, current and finalized observations in the same table. A screen shot of the prototype is presented in Figure 23.

ID	Start time	D	Source	az	el	aO	aC	eO	eC	T	Result	Err	drift	t(tot)	CAL	Tant	Tout	RH	tau	tID	Comments	Obsrv	f	ASC
	2012-10-14 17:11	.	J2049-1003	-3	40	4	-14	.	.	.														ej37
	2012-10-14 13:58	.	J333-459	93	54	10	-18	.	.	.														ej37
	2012-10-14 21:02	.	DR21	-89	52	6	-16	.	.	.														ej37
	2012-10-14 17:11	.	J2049-1003	-3	40	4	-14	.	.	.														ej37
	2012-10-14 13:58	.	J333-459	93	54	10	-18	.	.	.														ej37
	2012-10-14 13:58	.	J333-459	93	54	10	-18	.	.	.														ej37
	2012-10-14 21:02	.	DR21	-89	52	6	-16	.	.	.														ej37
	2012-10-14 17:11	.	J2049-1003	-3	40	4	-14	.	.	.														ej37
	2012-10-14 13:58	.	J333-459	93	54	10	-18	.	.	.														ej37
	2012-10-14 21:02	.	DR21	-89	52	6	-16	.	.	.														ej37
	2012-10-14 17:11	.	J2049-1003	-3	40	4	-14	.	.	.														ej37
	2012-10-14 13:58	.	J333-459	93	54	10	-18	.	.	.														ej37
	2012-10-14 21:02	.	DR21	-89	52	6	-16	.	.	.														ej37
1350246999	2012-10-14 20:39	.	CAL	90	45	C	432.94	6.26	130	85		10.0	4.4	85.8	-999	0.			ej37
1350246777	2012-10-14 20:33	.	CAL	90	45	C	432.94	6.26	130	152	404.97	10.0	4.4	85.8	-999	0.			ej37
1350243727	2012-10-14 19:42	.	CAL	90	45	C	404.97	2.1	14	151	404.97	9.9	3.8	88.2	-999	0.			ej37
1350241133	2012-10-14 19:00	.	DR21	54	64	14	2	-18	1	5	93.61	0.93	24	1205	404.97	10.0	3.9	88.5	-999	0.			ej37	
1350238285	2012-10-14 18:12	.	2337+264	-44	51	4	-18	.	.	.		0.94	0.40	74	1832	395.58	10.2	4.2	87.1	-999	0.			ej37
1350237817	2012-10-14 18:04	.	CAL	-90	45	C	395.58	0.34	3	152	395.58	10.2	4.3	86.8	-999	0.			ej37
1350237328	2012-10-14 17:55	.	CAL	-0	45	C	397.43	2.69	27	152	397.43	10.0	4.3	86.4	-999	0.			ej37
1350236930	2012-10-14 17:49	.	CAL	0	45	C	400.54	1.78	11	152	400.54	10.1	4.3	86.4	-999	0.			ej37
1350234650	2012-10-14 17:11	.	J2049-1003	-3	40	4	-14	.	.	.		2.58	0.28	54	1830	400.54	10.7	4.2	84.4	-999	0.			ej37
1350231729	2012-10-14 16:23	.	1824+271	34	54	7	-14	.	.	.		204.99	0.33	44	1830	396.20	11.5	4.9	82.4	-999	0.			ej37
1350229981	2012-10-14 15:53	.	NGC7027	-53	69	12	1	-14	-2	5	27.74	0.55	22	1203	396.20	12.4	5.3	81.2	-999	0.				ej37
1350229768	2012-10-14 15:50	.	CAL	-90	45	C	396.20	0.54	10	152	396.20	12.5	5.6	80.8	-999	0.			ej37
1350227558	2012-10-14 15:13	.	D112-283	-56	50	4	-18	.	.	.		2.54	0.34	63	1831	396.20	13.5	5.8	78.5	-999	0.			ej37
1350225348	2012-10-14 14:37	.	1732-094	10	39	4	-14	.	.	.		0.37	0.34	49	1826	388.44	14.4	7.1	72.0	-999	0.			ej37
1350222939	2012-10-14 13:58	.	J333-459	93	54	10	-18	.	.	.		1.49	0.42	53	1833	388.44	14.9	8.1	68.2	-999	0.			ej37
1350222745	2012-10-14 13:52	.	CAL	-90	45	C	388.44	0.75	4	151	388.44	14.9	8.3	68.6	-999	0.			ej37
1350221274	2012-10-14 13:28	.	DR21	-89	52	6	1	-16	-1	5	101.50	0.88	25	1203	387.75	14.9	8.2	67.9	-999	0.			ej37	
1350220123	2012-10-14 13:08	.	CAL	-90	45	C	387.75	0.37	2	151	387.75	14.9	8.4	68.0	-999	0.			ej37
1350219215	2012-10-14 12:54	.	CAL	-90	45	C	387.85	1.21	2	153	387.85	14.8	8.0	71.9	-999	0.			ej37

Figure 23. A static prototype of the observation summary tool where the upcoming, current and finalized observations are presented in the same table. The queued observations are highlighted blue, the currently running observation is highlighted green and the finalized observations have a white background. This approach would allow adding comments to all the observation types. It would also allow the observer to easily keep current with the observation queue.

5.3.4 Adding and modifying tau values

One of the most severe problems in the observation system was the determination of atmospheric opacity, or tau values and integrating them with the observation result data. There was no definitive guide or procedure available for the observers to determine the tau values. Instead, different observers use different methods to “raffle” (Finnish: “arpoa”) the values, as most of them call it. Most observers take into account the cloudiness conditions and relative humidity readings. There is variation on whether the values are temporarily written up every time the tau value is seen as changing, or whether the value is determined from scratch later on.

Based on user feedback and contextual inquiry results, integrating tau values with the rest of the observation result data was the most disliked component of the whole observing work. Many problems were found in the tools with which observers added tau values into a separate tau database. These tools worked by assigning a certain tau value to a certain time period. Each combination of tau value and time period was given a unique ID-number, which allowed searching and deleting the value. These tools were separate from the observation summary and observation delete / comment -tools in which most of the observation result data is displayed and modified.

Below, there are some quotes from the users related to adding tau values to the database and the observation results. The comments were picked from the user feedback, the contextual inquiries and from the observation result comments.

”Voi elämä, mikä tuska!” (“Oh, the agony!”)

- An observer describing how she would feel when making an input error with tau-add.pl. Captured in a contextual inquiry.

”Pieni osa sielusta kuoli taas” (“A small part of my soul died again”)

- An observer after not being able to get tau values inserted to certain observation results. Captured in a contextual inquiry.

”Tulen hulluksi tau-tietokannan kanssa!!!!” (“The tau database is driving me mad!!!!”)

- An observer in the user feedback.

*“MÄ EN SAA TAAS TÄTÄ YHTÄ ***** TAUTA TYRKÄTTYÄ TÄNNE, VAIKKA MITÄ TEKISIN! Yritin ainakin kymmeellä (sic) eri tavalla antaa noita aikavälejä, mutta kun ei sen vesipäinen virheilmoitus edes kerro, missä oikeasti vika piilee. ****. “(“I CAN'T GET THIS ONE *** ***** TAU VALUE SHOVED IN HERE, NO MATTER WHAT I DO! I tried at least ten different ways to input the time periods, but that idiotic error notification won't even tell where the actual problem lies. ****. ”)*

- An observer after having tried numerous times to get a tau value to be attached to a certain observation and failed. Captured in the observation results comments.

There were two realistic levels on which the tau value -related problems could be solved. The most optimal solution was to remove the tasks of determining the tau values and integrating them with other data altogether. This was deemed feasible in the long run, because it's currently possible to make separate tau observations with the telescope. There isn't any practical reason why results of these observations couldn't be automatically integrated with the other results, leaving only making of tau observations for the observers. However, there were still some problems in the validity of the results from tau observations. Before these problems are solved, the observers would have to continue “raffling” the tau values themselves. Thus, an interim solution was needed.

One such solution was to make the task of integrating tau values with the observation results as fast, efficient and pleasant as possible. This could be achieved by integrating the task into the observation summary or the observations delete / comment tools. The observers already used these tools in integrating comments on events and weather conditions and in adding delete tags in the data. It was expected that the observers would most likely have a natural feel for handling tau values with these tools.

The obs_delcom tool had problems of its own. The whole tool only existed because delete tags and comments could not be added to the results data in the observation summary tool. It was thus seen as more sensible to integrate the tau-adding functionality to the obs_summary tool.

Tau values are determined for time periods with constant weather conditions. A certain tau value represents a time period with certain conditions. When the conditions change, the tau value changes. It is typical that multiple observation results have the same tau value. The solution would have to support inputting tau values for multiple results at a time.

The solution was implemented using the jQuery JavaScript library that allows creating highly interactive web user interfaces. When adding or modifying tau values, the user either clicks a single tau value or uses a mouse drag “lasso” technique to select multiple values. The selection of tau values opens a dialog where the user can select an appropriate tau value from a drop down menu. The dialog widget contains buttons for Insert, Cancel and Close. After choosing a tau value, the user can insert the values to the selected cells in the table or exit from the dialog by pressing Cancel or Close. Cancel and Close both close the dialog. The double controls were originally there to allow the user a clear exit even in the case of accidental click on a tau value cell. The integrated tau-value adding and modification tool is presented in Figure 24.

4.85	15.3	11.6	63.3	-999	0.	e 37	
4.94	14.9	12.9	57.5	-999	0.	e 37	
4.94	14.4	12.1	61.2	-999	0.	e 37	
4.94	14.4	12.0	60.6	-999	0.	e 37	
4.94	13.8	12.7	60.5	-999	0.	e 37	
4.94	13.5	12.9	60.8	-999	0.	e 37	
4.94	13.3	12.0	64.1	-999	0.	e 37	
4.59	12.8	11.7	65.9	-999	0.	e 37	
4.59	12.6	11.1	65.9	-999	0.	e 37	
3.39	12.6	11.3	67.5	-999	0.	e 37	nopeasti liikkuvia vaaleita pilviä pitkin taivasta, tau 0.054
2.80	11.7	10.1	77.4	-999	0.	e 37	
2.80	11.6	9.2	82.8	-999	0.	e 37	
2.80	11.6	8.9	83.8	-999	0.	e 37	tasaisesti vaaleaa höttöstä pilveä koko taivaalla, tau 0.054

Figure 24. The integrated tau-value adding and modification tool. Selected tau values are highlighted in the table. Tau values are selected by painting with the mouse. The gray dialog box opens. The tau value is selected from a drop down menu after which the value can be inserted to the selected cells in the table.

5.3.5 Marking good and bad quality observations

Marking good and bad quality observation results are two separate tasks that the observers carry out in various phases of their observing shift. When an observation result of a source is of sufficiently high quality, the time of the observation is added to the source list. This marks the source on the sky map as “not due”, which means it doesn’t need to be observed right now. Bad quality observations are flagged with a delete tag to prevent them from reaching final reduction.

The week, year and observer name of a good quality observation are updated to the “last good quality observation” column of the specific source in the source list. This way the observer knows that the source in question does not necessarily need observing until a certain time period has passed. This period, which depends on the priority of the source, varies from a few days to many weeks. The autofile generator and real time sky view mark good quality sources on the map with a “not due” symbol. The symbol is a white dash over the symbol of the source.

The deletion of an observation result is affected by several factors. High error and drift readings and rain are the most typical cues to delete a result. Other factors are high offset error readings in 5p observation results and water and snow on the radome, which can cause attenuation of the signal and therefore low flux readings.

The first iteration of evaluation revealed problems in both determining if a result needs to be deleted and the technical task of marking a delete tag to a result. Accessing the

information for determining if a result needs to be deleted was in places inconvenient. Especially gaining reliable information on whether it had rained during an observation was laborious. Typically observers wrote a comment in the results data about rain if they were observing manually. This way they would later know to delete results flagged for rain. When going through past observations with lacking rain comments, the observers typically plotted rain duration plots of the two weather stations using the MB plotting tool. Using the tool meant leaving the observation results table and coming back again. The observer needed to memorize the times when the plot indicated rain and see what observations overlapped with the time periods. With a separate tool (obs_delcom.pl) used to mark the delete tags, the process was somewhat slow and included a lot of switching between tools.

To ease the deletion process, relevant data on conditions would have to be integrated with the observation results table. This way the observer would not have to switch between tools and memorize time periods. Whenever the observer needed information on the conditions, it would be instantly accessible in a form relevant to the task. The solution was implemented as a weather widget that presents an all-sky image and a data plotter with weather data. The widget is a larger scale systemic response to the problem of accessing relevant weather and other information and it is presented in more detail in sub-section 5.3.8.

To ease the technical task of adding delete tags to the observation data, the deleting functionality was integrated in the observation summary tool. This way, observers wouldn't have to open another tool and look for the deletable results anew. The solution was implemented by making the cells clickable in the delete column of the results table. When the observer clicks a delete cell of a non-deleted observation result, the letter "X" appears in the cell, the cell turns red and the row turns pink, indicating that the result is now marked for deletion. When the delete field of a result marked for deletion is clicked, the "X" is removed and the cell and row turn white to indicate an un-deletion. A screen capture of the delete functionality is presented in Figure 25.

2012-12-15 17:46	.	DR21	99	46	4	11	-14	-2	5	86.55	1.29	18			
2012-12-15 17:17	.	0235+164	-37	42	2		-15			7.36	1.19	44			
2012-12-15 16:48	.	0106+013	-15	31	10		-18			6.33	1.05	35			
2012-12-15 16:17	X	DR21	79	57	0	51	0	1	5	95.98	6.11	14			
2012-12-15 16:14		CAL	90	45					C	420.38	1.52	6			
2012-12-15 15:42					6		-16			8.86	1.16	20			
2012-12-15 15:14					4		-18			26.17	0.84	32			
Start time			D	Source		az	el	aO	aC	eO	eC	T	Result	Err	drift
2012-12-15 14:45	.	3C84	-104	42	0	6	-13	-5	5	127.25	1.06	34			
2012-12-15 14:39	.	CAL	-90	45					C	419.35	1.37	15			
2012-12-15 14:10	.	2243-123	-17	17	0		-13			4.35	1.21	59			
2012-12-15 13:42		2227-088	-20	20	4		-14			3.02	0.97	40			

Figure 25. The integrated deletion functionality in observation summary 1.4.

The deletion functionality was first prototyped in a different form. A static observation summary lookalike HTML page was created. It had the "delete" column replaced by a "quality" column. The prototype allowed marking good quality tags and delete tags in one place. When a quality cell was clicked, a dialog appeared where the user could select a "delete" or "good quality" tag for the observation. Once shown to observers, it became apparent that adding flags included too much clicking even though the basic idea was liked. This, together with the technical difficulties related to implementing the good quality marking led to only implementing the deletion functionality to observation summary. Good quality marking would have to wait.

The deletion functionality was implemented using JavaScript, AJAX and server side Perl technologies. The behavior of the user interface was implemented using JavaScript and especially the jQuery JavaScript library. The command to delete or undelete a certain observation is sent to a server side Perl CGI script via an XMLHttpRequest by utilizing the built-in AJAX functionality of the jQuery library. The Perl script then makes the necessary calls to the MySQL database that houses the observation results.

It is worth noting that no feedback from an unsuccessful AJAX call was implemented before the second iteration of evaluation and introduction of the functionality into operational use. This means that if for example the observer's internet connection was to break while giving delete commands, the user interface would not give any indication that the delete markings weren't reaching the server. The only way the observer would notice the missing markings would be if she reloaded the observation summary table. The necessity of proper feedback is obvious. The current system violates Nielsen's error prevention and proper feedback heuristics (see 3.5.4). A feedback functionality for problems in the delete tag insertion should be implemented as soon as possible.

5.3.6 Exporting data files for final reduction

After their observing shift is finished, an observer goes through the data from the shift and pre-processes it. Pre-processing is a high level task that includes many of the tasks presented in more detail in this section. Pre-processing tasks include deleting bad quality observations and adding tau values and comments to the data. Some observers might also postpone marking good quality observations to the pre-processing, especially if they had a short shift.

When an observer is done with all the data pre-processing tasks, she submits the data for final reduction. This is a redundant task, because the first step in the final reduction process is combining the separate data files submitted by individual observers. The first step in reducing time and effort spent on managing the data files is removing the need for observers to submit the files for reduction. The lead scientist could export the data from the observation summary tool herself. In this case a way is needed to let the reducer know that data from a certain shift has been pre-processed and is ready for final reduction. Currently the final reducer knows that data from a certain time period has been pre-processed when files have been uploaded to a certain directory.

The alternative solution for this could be that the observations would be handled in shifts in such a way that data from a shift could be tagged as ready for final reduction. This solution would at the same time fit to the need of the observers to quickly access the data from their own shifts. In the observation summary tool, a feature could be implemented which would allow users to search data on observation shift basis. Next to the current search filters, there could be a list of the names of observers who have observed with the telescope latest. After finishing with the pre processing, the observer could tag data from this particular shift as ready for final processing. When it was time for start final reduction for a time period, the lead scientist could relatively easily browse the shift list to see if any shifts still had unprocessed data.

This solution could give rise to some problems: How to avoid mistakes in accidentally clicking a shift processed or unprocessed? A verification dialog might come to question, because a processing flag should be given very seldom. Thus, there would no big harm in an extra dialog. Another possible problem is how to make observers remember to mark processed flag. A likely scenario is that the lead scientist has to send an email to observers if their shift is really unprocessed or just unmarked.

5.3.7 Use of the radome heater

Based on the evaluation conducted, the use of the radome heater can cause stress for the observers. There are many rules regarding the way the heater should be used in relation to the observations. These rules result in making hard decisions based on sometimes inadequate information. In many cases both using and not using the dome heater can degrade data quality or make the data unusable. If the heater is not used and moisture gathers on the dome, the observation results will be severely affected, and in many cases render the data useless. Then again, if the dome heater is used excessively, the temperature inside the dome can rise too high. This will affect the receiver, which will severely affect data quality and make final reduction difficult. It's also recommended to observe a calibration source and make a diode calibration before and after starting a heating cycle. This way it's easier to take into account the effect the heater has had in the final data reduction. For the same reason it's not recommended to start a heating cycle in the middle of an observation. These guidelines made many observers use a lot of time and effort in finding out an optimal time to start heating the dome.

When the heater has been used for a longer period, when e.g. melting snow from the dome, the observer has to wait for the temperatures to settle before starting observations again. In the contextual inquiries, there was discussion about a functionality that would follow the settling of the temperatures and receiver levels. The idea was that the system would alert the observer to the fact that the temperature has possibly reached a level that allows observing.

The underlying problem is that the current dome heater is overpowered for the purpose of preventing moisture from forming on the dome. When preventing moisture from forming on the dome, the goal is to keep the radome's temperature a few degrees Celsius above the outside temperature. To meet this goal with the current heater, the heater needs to be operated in cycles so that the temperature does not rise too high. This results in temperature spikes in the dome, the effects of which on the receiver are hard to take into account in the data reduction. For this reason, a lower power electric heater was acquired in November 2012. This heater is planned to be used more continuously and autonomously without constant controlling by the observer. When the new heater is in use, a large portion of the heater related problems to the observer will be eliminated. In the future, the high powered heater would be used only in melting snow and drying rain from the dome, tasks which are much rarer than moisture prevention.

5.3.8 Accessing relevant weather information

In the contextual inquiries it became apparent that observers had to collect information on changes in conditions from multiple locations. This was especially apparent when observers pre-processed the data from an observing shift. The sources of weather information include the MB plotting tool, various outside weather forecast web services, the antenna & weather page, the surveillance cameras and the cloud sensor 13-hour graph. The MB plotting tool allows plotting useful graphs of practically all sensor data available at the observatory, but can be laborious to set up and updates somewhat slowly. The cloud sensor 13-hour graph is popular among observers, but it is mostly used for viewing only the sky temperature graph. The antenna & weather page gives accurate real time information on multiple important variables, but lacks any graphical representation of the temporal development of the conditions.

Based on the data collected, the author interpreted that the observers need to access graphical representations of the temporal development of variables relevant to the

observations. To reduce the need for observers to jump between tools when going through observation data, a solution was developed which would integrate relevant information on conditions with the observation results data table. The observation results are displayed in the observation summary tool. When going through their observation data, the observers spend most of their time in that tool. The idea was to have relevant weather data available for each individual observation result to allow quickly assessing what the conditions were like during that observation. The integrated data was also meant to ease noticing moments when the conditions change.

The variables that the users most often accessed were determined mainly from the contextual inquiry data. The variables are sky temperature, rain duration and relative humidity. Based on discussions with observers, it was determined that the total power and the beam difference values of the 37 GHz receiver are good indicators of the general quality of the observing conditions and of the individual observations, respectively. A rule of thumb followed by many observers is that when the conditions are bad, the receiver's total power is high and vice versa. A graph of the beam difference of the receiver, on the other hand, resembles a rectangular signal when the conditions are good and the when source is strong. In bad conditions and with weak sources the graph starts resembling random noise.

The solution was implemented in the observation summary tool as a floating widget that can be freely moved and resized with the mouse. On the top portion of the weather widget, as it was named, an all-sky image from the closest 15 minutes of the currently selected observation is displayed. Above the all-sky image there's a header text displaying the image's capture time. Below the image there's a data graph plotter with the five aforementioned variables to choose from: sky temperature, rain duration, relative humidity, total power and beam difference of the 37 GHz receiver. The variables are displayed in small tabs between the all-sky image and the graph. Clicking a tab plots a graph of the variable in question.

For all variables excluding the beam difference, the default time span displayed is observation time \pm 6 hours. The observation time of the currently selected observation is highlighted in the graph with a semi transparent light blue color. The capture time of the all-sky image is illustrated as a black vertical line. The purpose of these markings is to help quickly notice the conditions during the observation, while also perceiving the development of the conditions before and after the observation on a long enough time scale. The graph can be zoomed in the vertical and horizontal directions by painting an area on the graph with the mouse. The default view is restored by double clicking the graph or by selecting the same observation result anew.

A result is selected by clicking the all-sky thumbnail image on the right side of the table. The thumbnails alone can be used to get a quick view of the conditions. To allow quick use, the observations can be also browsed with the keyboard up and down keys and the page up and down keys. The widget can be closed by clicking a close link at the top right corner of the widget. A screen capture of the weather widget next to the observation results table is displayed in Figure 26 and a close up showing only the widget is displayed in Figure 27.

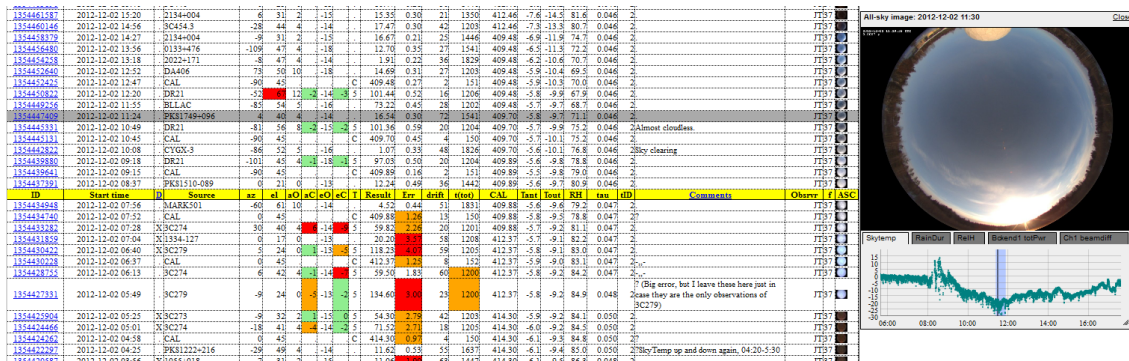


Figure 26. The weather widget for displaying relevant weather data integrated with the observation results table. The widget displays weather and telescope receiver data that is related to the selected observation, highlighted gray in the table. Also visible are the all-sky thumbnail images on the right side of the table.

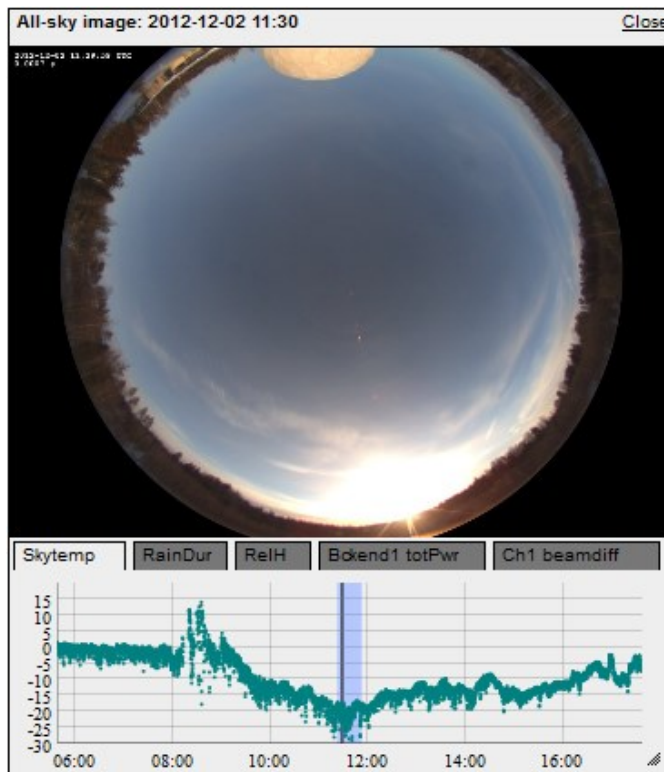


Figure 27. A close up of the weather widget. The running time of the currently selected observation can be seen highlighted in light blue in the graph. The capture time of the all-sky image is marked with a black vertical line. From the sky temperature graph and the all-sky image it can be seen that the current observation enjoyed a period of relatively clear skies. It was preceded by overcast skies, which cleared up. There was also a possible period of rain which is indicated by the peak in the graph. After the observation, cloudiness slowly increased.

The widget was first prototyped on a static observation summary lookalike HTML page that was shown to users. The initial prototype resembled the design presented in Figures 26 and 27. The users reacted positively toward the prototype and the design was adopted for further development.

The weather widget functionality was implemented by utilizing the jQuery JavaScript library (<http://jquery.com/>). The structure and styles were defined with HTML and CSS. The weather data plotter was implemented with the dygraphs JavaScript visualization library (<http://dygraphs.com/>). Data for the plotter is retrieved via AJAX calls to a Perl CGI script on Metsähovi's server. The script takes the name of the plotted variable

together with start and end times as input and returns available data from the time period in the Comma Separated Values (CSV) file format.

5.3.9 Overall dispersion of information in the system

One aspect affecting the observation work and especially setting up the observation environment is the fact that the system is scattered over multiple web pages and over the VNC desktop. In many tasks the observers have to shuffle between two or more web pages or other tools in order to have access to information needed for the completion of tasks.

It would make the observation work faster and more efficient if the core functions and information needed to conduct observations were as centralized as possible. The VNC desktop in a way already does this, but the limited resolution and bandwidth requirements present a challenge for e.g. use with mobile internet connections. Even with rising mobile network speeds, this continues to be a problem. Also, even with the VNC desktop, the observers still have to shuffle between several web tools in separate browser windows or tabs. Information has to be memorized and written down to allow transfer from one separate tool to another. A more coherent and interconnected system is needed.

5.3.10 Miscellaneous

The heuristic evaluation produced a set of fairly specific and technical findings from the evaluated user interfaces. The findings and the changes made to the system based on them are presented here.

The autofile generator web tool had a set of controls visible which hadn't been used for a long time. They cluttered the user interface and made noticing important controls more difficult. One of the first changes to the system was to hide those tools from view of the users. The changes made to the autofile generator are depicted in Figure 28.



Figure 28. The changes made to the autofile generator based on the heuristic evaluation of the first evaluation round. On the left is the user interface as it was in 2012-05-28 and on the right the version in use in 2012-12-14. The removed controls are highlighted with red ellipses and numbered from 1 to 3. 1. The list type selectors were hidden because currently only one list type is in use. 2. The Gen. Autofile button was hidden because the list type it creates is no longer in use. The Help button was hidden because no actual help was available by pressing it.

Two problems were noticed in the tool for viewing the pointing offset values (offsetapp.pl). It was not possible to overwrite an existing offset value with an empty one, which opened up a possibility for erroneous offset values being input in the table. This problem was fixed soon after it was noticed. Additionally, the offset table does not show the user which values have been changed during the current session. This can lead to the user's unintentional inputs being fed to the offset table without the user knowing about it. It is recommended that when the user makes a change in the table, all the changed offset values are clearly marked or highlighted. This problem is yet to be fixed.

5.4 Second iteration of studies

This section presents results of the evaluation of the modified observation summary tool. Results of user testing and heuristic evaluation are presented in addition to the changes made to the user interfaces based on these results.

5.4.1 User testing

User testing was conducted to evaluate the usability of the modified observation summary tool. Other modified tools were not included in the user tests. The results indicate that the modifications resulted in improved usability. On average, the results indicate a 75 % increase in efficiency and a clear improvement in satisfaction. Perceived usability measured with the System Usability Scale (SUS) rose from 30 to 87.5 points on a non-linear scale of 0 to 100 points. The average of single tasks' perceived pleasantness rose from 2.3 to 4.4 points on a scale of 0 to 5. Effectiveness remained on

a high level, as all the test tasks were successfully completed with the new system version.

There are some aspects to be taken into account when analyzing the results. The interviewer was the developer of the tested user interfaces, and the users knew that. According to Nielsen (1993), people tend to answer questionnaires and interview questions based on what they think the interviewer wants to hear. Also, the observers knew which tested version was supposed to be the newer and better one. They knew that the point of the testing was in part to verify that the new system was better than the old one. Remembering which pleasantness or SUS questionnaire grade they gave to the new system, they may have been reluctant to give a better grade to the old system, even if they had felt like doing so. Grading the new system as the worse one might have felt like giving the “wrong” answer.

Additionally, the observers had been personally involved in pointing out the problems that led to the development of the new version by giving user feedback and participating in the contextual inquiries. They had either given proposals on improvements themselves or at least commented on suggestions given by the interviewer. They had also mostly agreed on the existence of the problems and with the suggested improvements. That’s why it’s possible they may have been personally attached to the implemented improvements and thus inclined toward favoring the new system.

In addition, the observers had already used the new system for some time. Months in the case of the new tau functions. For some, it was the first time using the delete tag and commenting functions. Others had used the delete and commenting functions for one or two observing shifts. A fairly long time had elapsed since the observers had used the old tau tools. The long time from the last use may have skewed the results in favor of the new system.

To minimize the effect of the aforementioned factors, the test tasks had been formulated to be fairly mechanical. The focus was on determining how an expert user could perform with the system in an optimal situation. As presented in chapter 4, all the test tasks were first gone through once with the user before starting the actual test, to diminish the effect of learning. For some observers the “practice run” was actually the first time they successfully managed to use the old tau delete tool. The tool had had such a bad reputation among the observers that they had actively avoided using it.

The mechanical nature of the test tasks raised criticism among some test users, who argued that the tasks were unrealistic.

The results from the delete tag adding and comment adding tasks are presented in Table 18. The tau value adding and removal task results are presented in Table 19 and the rain duration task results in Table 20. The results from the SUS questionnaires are presented in Table 21 and averages of task pleasantness values in Table 22. Table 23 presents the problems and suggested improvements determined from the user tests.

Table 18. Results of the delete tag and comments adding tasks of the usability tests. Results are highlighted with green, orange and blue colors to indicate improvement, regression or unchanged values, respectively. Values in parentheses indicate results that did not affect successful completion of the task.

Observer		Add delete tags				Add comments			
		Success	Errors	Time	Pleasant	Success	Errors	Time	Pleasant
Pilot	Old	Y	0	1:11	2	Y	0	0:56	2
	New	Y	(1)	0:19	3	Y	0	0:24	4
	Delta	-	(+1)	-73 %	+1	-	-	-57 %	+2
1	Old	Y	0	0:35	3	Y	1	0:45	3
	New	Y	(1)	0:17	4	Y	0	0:23	4
	Delta	-	(+1)	-51 %	+1	-	-1	-49 %	+1
2	Old	Y	(3)	1:26	3	Y	0	1:15	4
	New	Y	0	0:23	5	Y	(1)	0:45	5
	Delta	-	(-3)	-73 %	+2	-	(+1)	-40 %	+1
3	Old	Y	0	0:38	4	Y	0	0:43	3
	New	Y	0	0:15	5	Y	(1)	0:52	3
	Delta	-	-	-61 %	+1	-	(+1)	+21 %	-
4	Old	Y	0	0:40	3	Y	0	1:06	2
	New	Y	0	0:10	4	Y	(1)	0:29	3
	Delta	-	-	-75 %	+1	-	(+1)	-56 %	+1
All	Old	Y	(3)	0:50	3,3	Y	1	0:57	3,0
	New	Y	(1)	0:16	4,5	Y	(3)	0:37	3,8
	Delta	-	(-2)	-68 %	+1,2	-	(+2)	-35 %	+0,8

Table 19. Results of the tau value adding and removal tasks of the usability tests. Results are highlighted with green, orange and blue colors to indicate improved, regressed or unchanged values, respectively.

Observer		Add tau values				Remove tau values			
		Success	Errors	Time	Pleasant	Success	Errors	Time	Pleasant
Pilot	Old	N*	0	3:33	1	Y	0	1:31	3
	New	Y	(1)	0:29	3	Y	0	0:13	4
	Delta		(+1)	-86 %	+2	-	-	-86 %	+1
1	Old	N*	1	2:10	2	Y	0	0:49	2
	New	Y	0	0:16	4	Y	0	0:06	4
	Delta	+1	-1	-88 %	+2	-	-	-88 %	+2
2	Old	N	3	8:47	1	Y	1	1:47	2
	New	Y	0	0:29	5	Y	0	0:08	5
	Delta	+1	-3	-94 %	+4	-	-1	-93 %	+3
3	Old	N*	1	4:15	1	Y	(1)	1:45	2
	New	Y	0	0:23	5	Y	0	0:07	5
	Delta	+1	-1	-91 %	+4	-	(-1)	-93 %	+3
4	Old	N*	1	1:41	2	Y	0	1:24	1
	New	Y	(1)	0:26	4	Y	0	0:06	3
	Delta	+1	-0	-74 %	+2	-	-	-93 %	+2
All	Old	N*	6	4:13	1,5	Y	2	1:26	1,8
	New	Y	(1)	0:24	4,5	Y	0	0:07	4,3
	Delta	+1	-5	-91 %	+3,0	-	-2	-92 %	+2,5

Table 20. Results of the rain duration checking task of the usability tests.

		Check rain duration			
Observer		Success	Errors	Time	Pleasant
Pilot	Old	-	-	-	-
	New	-	-	-	-
	Delta	-	-	-	-
1	Old	Y	0	1:14	2
	New	Y	(1)	0:30	4
	Delta	-	+1	-59 %	+2
2 (Missing training run, both)	Old	Y	(1)	2:44	3
	New	Y	(1)	0:33	5
	Delta	-	-	-80 %	+2
3	Old	Y	(1)	1:30	2
	New	Y	0	0:33	5
	Delta	-	(-1)	-63 %	+3
4 (Missing training run, old)	Old	Y	-	5:34	1
	New	Y	0	0:19	5
	Delta	-	-	-94 %	+4

Table 21. Results of the SUS questionnaires from the old and new observation summary tool versions. The scale is from 0-100. The scores can be compared with the global average result of 68 gotten from a set of 500 SUS studies (Sauro, 2011). I.e. scores over 68 can be considered above average and scores under 68 below average.

Observer / System version	Old	New
Pilot	52.5	85.0
1	25	87.5
2	42.5	90
3	35	95
4	17.5	77.5
All	30,0	87,5

Table 22. Comparison of average task pleasantness between the old and new observation summary versions.

Observer / System version	Old	New
Pilot	2	3.5
1	2.4	4
2	2.6	5
3	2.4	4.6
4	1.8	3.8
All	2,3	4,4

Table 23. Problems found in the observation summary tool based on the user testing.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Two observers accidentally moved the mouse while clicking a delete-cell, resulting in text being selected in the table instead of the delete cell being clicked.	2	obs_summary.pl	Error prevention	Consider adding a checkbox to the delete cells or prevent selecting text in the delete cells.	No
After canceling tau value modification, the values remain selected in the table.	2	obs_summary.pl	System status	Remove the selection when tau modification is canceled.	Yes
The Insert and Cancel buttons in the tau dialog are redundant resulting in unnecessary clicks when inserting tau values.	2	obs_summary.pl	Aesthetic and minimalist design	Remove the redundant buttons. Have the value inserted when a value is selected in the drop down menu. The Close link already works like Cancel.	Yes
An observation can't be selected to be viewed in the weather widget by clicking anywhere on the row.	2	obs_summary.pl	-	Make the whole row clickable.	Yes
Heater cycles have to be interpreted from radome temperature or manual comments.	2	obs_summary.pl	-	Make heater cycles automatically visible in the comments and/or in the weather widget's data plotter as highlighted time periods.	No
The weather widget does not display weather radar images for detecting rain.	2	obs_summary.pl	-	Add FMI's "Helsinki testbed" weather radar images to the weather widget. Make the images time browsable.	No
Delete tags can only be added one at a time	2	obs_summary.pl	-	Add an option to select multiple observation results for deletion at once in the same way as tau values are selected. However, maintain the option to mark single results by clicking them.	No

5.4.2 Heuristic evaluation

Table 24 presents the results from the heuristic evaluation conducted on the modified observation summary tool.

Table 24. Results of heuristic evaluation of the modified observation summary tool.

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
No <i>back/forward</i> functionality support in obs_summary's weather widget	2	obs_summary.pl	User freedom, Error prevention	Consider adding support for <i>back/forward</i> functionality if users seem to need it.	No
Weather widget can be moved out of view of the user	2	obs_summary.pl	Error prevention, Error recovery	Add automation to make sure the widget is never completely out of view.	Yes
By default, weather widget is out of view on small screens.	2	obs_summary.pl	-	Make the observation summary results table narrower. Add automation to make sure the widget is never completely out of view.	Yes (In part)
Weather widget can't be viewed on small screens without it covering a part of the observation results table.	2	obs_summary.pl	-	Make the observation summary results table narrower.	Yes (In part)
Weather widget resets to default settings when obs_summary.pl is updated or reloaded	2	obs_summary.pl	-	Add support for remembering the state of the widget across reloads.	Yes
Weather widget opens upon obs_summary.pl reload even when user has closed it before reload	2	obs_summary.pl	-		
Highlighting a selected observation in obs_summary with gray color may give an impression about the highlighted item being inactive rather than highlighted.	2	obs_summary.pl	-	Change highlight color to a more bright and noticeable color, e.g. dark blue.	Yes
At a glance, there's not a clear connection between the highlighted observation table row and the weather widget.	2	obs_summary.pl	-	Explore ways to create a visual connection between the selected observation and the widget.	No

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
Weather widget graph zoom feature does not work properly when the page is scrolled (Bug related to fixed position in dygraphs)	2	obs_summary.pl	-	Find fix or work around to dygraphs bug.	No
Weather widget graph doesn't print units on the Y-axis and it's not clear what is presented	2	obs_summary.pl	-	Consider printing units to the y-axis at the expense of space.	No
Weather widget graph doesn't tell the user what the light blue highlight and the black vertical line represent in the graphs	2	obs_summary.pl	-	Add a tool tip or link to an illustrative figure that explains the markings.	No
Ch-1 beamdiff tab of graph plotter is hidden from view when weather widget is at default size	2	obs_summary.pl	-	Add a navigation function that allows quick access to all the tabs.	Yes (arrow keys)
Ch-1 beamdiff graph displays invalid time period for either winter time or daylight savings time, depending which is selected in the code.	2	obs_summary.pl	-	Fix UTC time to Unix time conversion in the code.	No
Selecting a result in the results table with the mouse requires clicking a small thumbnail image.	2	obs_summary.pl	-	Consider making the whole row clickable.	Yes
There's no hint to the user that the observations can be browsed with the arrow and pgup/pgdown keys.	2	obs_summary.pl	-	Add a tool tip or info text on the page to guide the user on the available shortcuts.	No
There's no hint to the user that the weather widget is draggable.	2	obs_summary.pl	-	Add a tool tip or change the mouse cursor to one that suggests a window is draggable.	Yes (Tool tip)

Problem	Severity	User interface	Heuristic (if applicable)	Suggested improvement	Fixed?
No visible clue about weather widget being resizable other than mouse cursor over right and bottom edge and right bottom corner. (Missing symbol due to a technical problem with jQuery UI CSS linking)	2	obs_summary.pl	-	Fix CSS linking problem	Yes
When a new observation is selected, the graph disappears for a moment, making it harder to follow the progress of observations visually on the graph.	2	obs_summary.pl	-	Use two div elements on top of each other that allow updating the graph view only after the new data has been plotted.	No
Without reloading obs_summary.pl, there's no feedback to user if tau values do not reach the database	3	obs_summary.pl	Visibility of system status, Error prevention	Implement real time checking of unsuccessful AJAX calls and database insertions. Feedback to user about failures.	No
Not possible to input arbitrary tau values.	2	obs_summary.pl	-	Add possibility to insert arbitrary tau values.	No

5.4.3 Conclusions and recommendations

Table 25. Recommendations for changes in the quasar observation system based on the second iteration of evaluation.

Recommendation	Status
Move the column for delete tags next to the tau column in observation summary.	Implemented in observation summary 1.41.
Allow selecting an observation result to be viewed with the weather widget by clicking anywhere on the result row.	Implemented in observation summary 1.41.

Figure 29 presents the state of the observation summary tool after some of the recommendations given after the second iteration had been implemented. Figure 30 presents a jQuery calendar widget for selecting dates and times to observation summary's time fields.

Observation summary (version 1.43)

Current observation

(az 95, el 59) eni is observing 0804+499 using mode 1. The observation [136328860](#) started 1665 seconds ago.

Past observations

NOTE: Empty fields don't affect search results.
 Choose time range to display: and/or observer: and/or source: and/or frequency: and/or ID range:

ID	Start time	Source	az	el	aO	aC	eO	eC	T	RemH	Err	drift	(ttob)	CAL	Tant	Tout	RH	D	tan	Comments	Obsvr	IAS
136	2013-03-01 21:47	1058+018	-12	31	1	-16				12.83	0.41	20	1204	395	-3.0	-8.4	58.6		0.048	Tietokoneongelma yon mittaukset ovat tallennet	JTS97	
137	2013-03-01 21:15	4C29 445	-50	52	7	-18				4.98	0.40	19	1638	395	-2.4	-8.0	58.0		0.048		JTS97	
138	2013-03-01 20:36	MARK421	-51	62	11	-16				1.06	0.28	41	1800	395	-1.8	-7.4	61.2		0.048		JTS97	
139	2013-03-01 20:31	CAL	90	45						C 394.69	0.48	11	150	395	-1.8	-7.4	62.1		0.048		JTS97	
140	2013-03-01 20:07	3CB4	105	41	3	-19				5 126.33	0.88	6	1207	395	-1.0	-7.0	61.3		0.048		JTS97	
141	2013-03-01 19:42	3P287	-16	49	4	-14				28.46	0.35	30	1202	395	-0.4	-6.7	59.6		0.048		JTS97	
142	2013-03-01 19:06	3P489	-20	53	4	-14				10.35	0.33	60	1838	390	0.5	-6.2	59.0		0.048		JTS97	
143	2013-03-01 18:33	0607+179	39	43	4	-14				11.26	1.23	69	1638	390	1.4	-6.7	62.6		0.048		JTS97	
144	2013-03-01 18:07	3CB4	80	55	11	-19				5 137.88	0.54	4	1205	390	2.1	-5.4	59.0		0.048		JTS97	
145	2013-03-01 18:01	CAL	90	45						C 390.35	1.85	21	150	390	2.1	-5.3	59.1		0.048		JTS97	
146	2013-03-01 17:29	0137+476	102	52	9	-19				18.73	0.89	26	1637	390	3.0	-4.5	61.9		0.048		JTS97	
147	2013-03-01 17:03	0059+581	117	58	11	-20				14.22	0.85	31	1202	391	3.6	-3.8	62.0		0.048	Kirkasta	JTS97	
148	2013-03-01 16:36	3CB4	62	59	13	-19				4 144.68	1.81	19	1207	391	4.3	-3.0	63.3		0.048		JTS97	
149	2013-03-01 16:33	CAL	90	45						C 391.12	0.95	9	151	391	4.5	-2.8	64.3		0.047		JTS97	
150	2013-03-01 15:58	PKS0458-020	-9	28	1	-18				7.01	1.25	48	1734	391	5.2	-2.1	63.6		0.048	Kirkastuu	JTS97	
151	2013-03-01 15:25	0607+179	-28	46	4	-14				12.73	1.35	45	1637	391	5.8	-1.4	64.6		0.048		JTS97	
152	2013-03-01 14:57	TA026	-4	28	1	-18				8.65	2.68	87	1448	393	6.5	-1.8	67.5		0.048		JTS97	
153	2013-03-01 14:24	0235+164	6	47	4	-14				5.61	1.05	46	1637	393	7.1	-0.7	68.8		0.048		JTS97	
154	2013-03-01 13:54	0234+285	6	59	6	-16				18.70	0.73	29	1443	393	7.8	0.0	69.0		0.048	Perheleiva (kaan nousu edessä)	JTS97	
155	2013-03-01 13:29	BLLAC	90	51	11	-18				53.22	1.61	33	1204	393	8.4	0.8	68.4		0.048		JTS97	
156	2013-03-01 13:26	CAL	90	45						C 392.73	1.48	13	150	393	8.5	0.6	68.7		0.048	Tasaisempi piiripiste ja antenni	JTS97	
157	2013-03-01 13:02	DR21	101	44	6	-19				5 85.86	2.09	40	1202	393	8.9	1.2	64.8		0.048	Bad 5p fit. Ei?	JTS97	
158	2013-03-01 12:32	PKS0458-020	-9	28	1	-18				5 42.94	1.50	25	1208	392	9.6	1.6	63.1		0.048	Bad 5p fit. ?	JTS97	
159	2013-03-01 12:29	CAL	-90	45						C 391.96	0.75	21	150	392	9.6	1.6	63.6		0.048		JTS97	
160	2013-03-01 12:04	3CB4	-73	59	10	-17				5 139.61	1.90	24	1202	389	10.1	2.1	63.2		0.048		JTS97	

Right side widget: All-sky image: 2013-03-01 22:00. Skytemp, RainDur, Rel, Backend, totPwr, CH3, beamStd, TdshM.

Figure 29. A screen shot of the observation summary tool version 1.43 from 2013-03-15. The time format in the time fields on the upper left part of the image has been changed to the more readable “YYYY-MM-DD hh:mm:ss” format. The observations can be filtered by observer name. The whole table row is clickable when selecting an observation for viewing in the weather widget, excluding the delete, tau and comment fields. When hovering over a row with the mouse, the row is highlighted light gray. The delete column (“D”) has been moved next to the tau and comment fields. Many columns have been narrowed making the table 61 pixels narrower. The “ID” column is minimized by default and can be maximized by clicking the “ID” header. An automatic warning is written to the comment field of an observation with a bad 5-point pattern fit (“Bad 5p-fit”). The tabs of the weather widget graph can be browsed with arrow buttons. The widget remembers its size, position and visibility across page reloads.

Choose time range to display:

Feb 2013

Mo Tu We Th Fr Sa Su

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

Time: 13:11

Hour:

Minute:

ce	az	el	aO	aC
-12	31	1		
-50	52	7		
-51	62	11		
90	45			
105	41	3		
-16	49	4		
-20	53	4		
39	43	4		
80	55	11		
90	45			
102	52	9		
117	58	11		
52	66	13		
90	45			

Figure 30. A close up of the widget for selecting the date in the observation summary tool version 1.43 from 2013-03-15. The widget opens when a time field is clicked. The time widget is an add-on of the jQuery UI library.

Because the weather widget relies on the web server to access data, fairly high bandwidth is required for quick use of the widget. A typical 6-hour period of sky temperature data measures approximately 170 kilobytes and an all-sky image approximately 60 kilobytes. Other variables that are printed on the graph, such as rain

duration and relative humidity readings, take about 10 kilobytes of space per a six hour period. These figures aren't a problem when operating with a fairly quick network connection. They also don't present a problem when operating with the VNC desktop. In that case the network traffic takes place inside Metsähovi's local area network and the only thing generating traffic between the observer and the observatory is the updating of the VNC desktop view. However, with a sluggish mobile internet connection e.g. on a summer cottage and an SSH tunneled proxy connection to Metsähovi's local area network, problems might arise.

To give a rough example, the download time of 170 kilobytes worth of sky temperature data with an actual transfer rate of 64 kbit/s, is approximately 22 seconds. This kind of a load time does not allow quick browsing of observations with the weather widget. To reach a load time of 1 second or less, the transfer speed needs to be at least 1.4 Mbit/s, which can be an unrealistic figure to achieve in rural areas.

Considering the fact that the quasar observation system is relatively often accessed using a mobile connection from rural areas, solutions for lowering the network load of the system should be considered in the future. One solution could be based on loading the graphs as images from the server instead of as data. This would result in a more constant amount of data loaded from the server. This would, however, also limit the interactive features compared to the currently used dynamic graph.

Below are listed some aspects for further development:

1. The system should be addressed as a whole with systemic design responses. The goal should be in creating a consistent system that has a unified look and feel and that is divided into logical interconnected "places" for doing tasks. The system should provide easy navigation within the major tools.
2. Accelerators could be added for e.g. data processing in observation summary. E.g. key shortcuts ("d" for deleting, "c" for commenting, etc.).
3. Further reduction in the number of tools required in routine observations.
4. Focus on the final reduction process, which is currently very laborious.
5. The conducted study focused on improving usability for expert users. This was reasonable, because most users of the system are just that. However, when new employees come to Metsähovi, they have to learn to use the observation system.
 - a. There are two things to learn: use of the software and the art of quasar observations.
 - b. Learnability of the system can be tested with e.g. cognitive walkthroughs and usability tests with novice users

6 Conclusions

In this chapter conclusions are drawn from the results of the study.

In this thesis, a user centered evaluation and design process was conducted to improve the usability of the quasar observation software used in Aalto University Metsähovi Radio Observatory. The goals for the study were to determine what processes and methods should be used in improving the usability of the observation software. The primary goals of the study were to find out what are the biggest usability problems in the system, fix a selected group of the problems and evaluate if the changes led to improved usability.

In the study, a two staged evaluation and development process was followed. The process included collecting usability data from the observation system by three primary methods: user feedback, heuristic evaluation and contextual inquiry. This was the first evaluation round. The collected data was analyzed by applying methods used in the Contextual Design methodology. Based on the analysis, a set of usability problems was assembled. The problems were given severity ratings and they were classified by task and by user interface. Suggested improvements were developed, if possible, taking into account the larger scale context of the problems. Knowledge on the context was gained from the contextual inquiries and subsequent analysis. Based on the severity of the problems and the technical feasibility of implementing the suggested improvements, a group of suggestions were implemented. The most extensive changes were implemented in the observation summary web tool in which tabular form unreduced observation results are presented and processed. The changes were implemented with the help of JavaScript, DHTML and AJAX.

The second evaluation round included running usability tests, conducting heuristic evaluation and collecting user feedback on the changes made to the observation summary tool. The changes in efficiency, effectiveness and user satisfaction were measured between the old and new versions of the tool. The results point toward significantly increased efficiency and satisfaction. Effectiveness remained on a high level, although there was a small increase in non-critical user errors in one of the test tasks.

The changes made have reduced the need for the observers to jump between tools when processing data and finding relevant information on observing conditions. The most disliked tools in the system were retired and their functions were integrated into the existing observation summary tool. As expected, the number of catastrophic problems found was very small. Only one problem that regularly lead to an unsuccessful end result was found and it was in the tau value adding.

Based on the results, the study can be described as a success. The study revealed a wide variety of problems in the observation system and produced noticeable improvements in usability. In total, 98 suggestions for improvement were given. Out of these, 43 were implemented in some form and 55 were not. Out of the ones not implemented, 5 suggestions were addressed by developing prototypes for future development. The data collected can also be used as a basis for a higher level redesign effort.

Still, even after the changes implemented, the observation system is unnecessarily scattered. The task of giving observing commands still often requires for the observer to deal with a web tool, a manually saved autolist file and a command line interface. No concrete implementations related to the idea of having a queue of observing commands

were yet made. Design and implementation of automation for selecting sources for observing is also yet to be done. The observation system's user interfaces lack a common look and feel and are used as isolated tools rather than as one interconnected system. To address these remaining issues and others that are presented in the results, iterative user centered development of the system should be continued. The analysis conducted is not extensive enough to allow redesigning of the system on a high abstraction level.

The contextual inquiry data gathered could be reused in helping to redesign the system further. A partial contextual design process could be followed as described by Beyer and Holtzblatt (1998). For a redesign of an existing system, they recommend using contextual interview data to create sequence, artifact and physical models to understand how the system interacts with the users' world. Sequence models are already available. Based on the sequences, use scenarios could be created to tell the story of a typical user. An affinity diagram would be created and used together with scenarios and models to brainstorm issues and design responses. Artifacts in the physical model and natural clusters of work would be used in perceiving work structure. Based on key issues found, a visioning session would be run to come up with a design response. Storyboards would be created based on the scenarios, after which UI design and paper prototyping could be started. Organization of the user interface would be aided with the structural thinking behind the user environment design (UED). An actual UED would not be created. Beyer and Holtzblatt estimate that running this process might take two months with a cross-functional team of four to six people. Running such a process might be feasible to run in Metsähovi to continue on the path toward a more coherent and integrated quasar observation system.

References

Annett, J. & Duncan, K. 1967. Task Analysis and Training Design. *Occupational Psychology* 41. pp. 211-221.

Apple Inc. 2012. OS X Human Interface Guidelines. [Online]. [Referenced 2013-05-08]. Available: <https://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/AppleHIGuidelines/OSXHIGuidelines.pdf>

Beyer, H. & Holtzblatt, K. 1998. *Contextual design: Defining Customer-Centered Systems*. San Francisco, CA, USA. Morgan Kaufmann Publishers. ISBN 1-55860-411-1.

Beyer, H. & Holtzblatt, K. & Baker, L. 2004. An Agile User-Centered Method: Rapid Contextual Design. In: Zannier, C. & Erdogmus, H. & Lindstrom, L. (eds.) *Extreme Programming and Agile Methods - XP/Agile Universe 2004*. Calgary, Canada, August 15-18, 2004. Springer Berlin Heidelberg. pp. 50-59. *Lecture Notes on Computer Science*, Vol. 3134. ISBN 978-3-540-22839-4. (DOI: 10.1007/978-3-540-27777-4_6)

Bjerknes, G. & Ehn, P. & Kyng, M. 1987. *Computers and democracy – A Scandinavian challenge*. Awebury, England. ISBN 0566054760.

Brooke, J. 1996. SUS - A quick and dirty usability scale. In: Jordan, P. W. & Weerdmeester, B. & Thomas, A. & Mclelland, I. L. (eds.) *Usability evaluation in industry*. London. Taylor and Francis. pp. 189-194. Vol. 189.

Butow, E. 2007. *User interface design for mere mortals*. NJ 07458, USA. Pearson International. ISBN 0-321-44773-5.

Card, S. & Moran, T. & Newell, A. 1983. *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA. Lawrence Erlbaum Associates. ISBN 0898592437.

Crystal, A. & Ellington, B. 2004. Task analysis and human-computer interaction: approaches, techniques, and levels of analysis. In: *Proceedings of the Tenth Americas Conference on Information Systems (August 2004)*. Association for Information Systems. pp. 391.

Foley, J.D. 1995. *Computer Graphics: Principles and Practise in C*. Second Edition. Reading, MA, USA. Addison Wesley. ISBN 978-0201848403.

Galitz, W. O. 2007. *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques*. 3rd edition. Indiana, USA. John Wiley & Sons. ISBN 978-0-470146224.

Garrett, J. J. 2005. *Ajax: A New Approach to Web Applications*. [Online]. [Referenced 2013-01-29]. Available: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>

- Holsti, N. & Salonen & Urpo, S. 1983. A computer program for point-source microwave continuum observation. Espoo, Finland. Helsinki University of Technology Radio laboratory. Report pp. 141. ISBN 951-753-083-8.
- Horn, D. 2002. Alphonse Chapanis (1917-2002). Father of Ergonomics. [Online]. [Referenced 2013-05-14]. Available: <http://www-personal.umich.edu/~danhorn/chapanis.html>
- Hovatta, T. 2009. Radio variability of active galactic nuclei: Analysis of long-term multifrequency data. Thesis for the degree of Doctor of Science in Technology. Helsinki University of Technology, Faculty of Electronics, Communications and Automation, Department of Radio Science and Engineering. Espoo, Finland. ISBN 978-951-22-9812-9 (paper).
- Laurila, H (ed.) & Holsti, N. & Teräsraanta, H. & Urpo, S. & Jankko, T. 1989. A computer Program for Point-Source Microwave Continuum Observation (FIVE) Program Document. 02540 Kylmäla, Finland. Helsinki University of Technology Metsähovi Radio Research Station. Vol. II, System Description Manual. ISBN 951-22-0086-4.
- Lewis, C. H. 1982. Using the "thinking-aloud" method in cognitive interface design. In: IBM Research Report RC 9265. Yorktown Heights, NY. IBM T. J. Watson Research Center.
- Mayhew, D.J. 1999. The usability engineering lifecycle: a practitioner's handbook for user interface design. San Francisco, CA, USA. Morgan Kaufman Publishers, Inc. ISBN 1-55860-561-4.
- McDonald, S. & Monahan, K. & Cockton, G. Modified Contextual Design as a Field Evaluation Method. In: Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles. 14-18 October 2006. ACM New York, NY, USA. pp. 437-440. ISBN 1-59593-325-5. (DOI: 10.1145/1182475.1182531).
- Microsoft. 2013. Windows user experience interaction guidelines. [Online]. [Referenced 2013-05-08]. Available: <http://msdn.microsoft.com/en-us/library/windows/desktop/aa511258.aspx>
- Nielsen, J. 1993. Usability Engineering. San Francisco, CA, USA. Morgan Kaufmann. ISBN 0-12-518406-9.
- Nielsen, J. 1994. Heuristic evaluation. In: Nielsen, J. & Mack, R. L. (eds.). Usability inspection methods. New York, NY. John Wiley & Sons. pp. 25-62. ISBN 0-471-01877-5.
- Norman, D. A. 1988. "Psychology of Everyday Action". The Design of Everyday Things. New York. Basic Books. ISBN 978-0-465-06710-7.
- Raymond E. S. & Landley R. W. 2004. The Art of Unix Usability. [Online]. [Referenced 2013-02-13]. Available: <http://www.catb.org/esr/writings/taouu/html/index.html>.

- Rock I. & Palmer, S. 1990. The Legacy of Gestalt Psychology. *Scientific American*. Vol. 263. No. 6. pp. 84-90. ISSN 0036-8733. (DOI: 10.1038/scientificamerican1290-84).
- Sauro, J. 2011. Measuring Usability with the System Usability Scale (SUS). [Online]. [Referenced 2013-05-24]. Available: <http://www.measuringusability.com/sus.php>.
- SFS-EN ISO 9241-11:1998. 1998. Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. P.O.B 130, FI-00101 HELSINKI (MALMINKATU 34). Finnish Standards Association SFS. 43 p.
- SFS-EN ISO 9241-210. 2010. Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems. P.O.B 130, FI-00101 HELSINKI (MALMINKATU 34). Finnish Standards Association SFS. 32 p.
- Shneiderman, B. 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer*. Vol. 16. No. 8. pp. 57-69. ISSN 0018-9162. (DOI: 10.1109/MC.1983.1654471).
- Shneiderman B. & Plaisant C. 2005. *Designing the user interface*. 4th edition. MD, USA. Pearson Education. ISBN 0-321-26978-0.
- Taylor, F. W. 1911. *The Principles of Scientific Management*. New York, NY, US and London, UK. Harper & Brothers.
- Van Kesteren, A & Jackson, D. (eds.) 2006. The XMLHttpRequest Object W3C Working Draft 05 April 2006. [Online]. [Referenced: 2013-05-08]. Available: <http://www.w3.org/TR/2006/WD-XMLHttpRequest-20060405/>
- Tornikoski, M. 1990. Ohjelmisto auringon aktiivisuuden tutkimiseksi mikroaaltoalueella. Diploma thesis. Helsinki University of Technology. Department of Information Technology. Espoo. 68 p.
- Tornikoski M. & Mujunen A. (eds.) 2005. Metsähovi Radio Observatory annual report 2004. Espoo. Helsinki University of Technology Metsähovi Radio Observatory. ISSN 1455-9579.
- Tornikoski M. & Mujunen A. & Holmberg B. & Uunila M. (eds.) 2010. Metsähovi Radio Observatory Annual report 2009. Espoo. ISBN 978-951-22-9521-0.
- Tornikoski, M & Mujunen, A & Holmberg, B & Uunila, M. (eds.) 2012. Metsähovi Radio Observatory annual report 2010. Helsinki. Aalto University Metsähovi Radio Observatory. Unigrafia Oy. Helsinki. ISBN 978-952-60-4591-7.
- Tornikoski M. & Holmberg B. & Uunila M (eds.) 2013. Metsähovi Radio Observatory Annual report 2011. Unigrafia Oy. Helsinki. ISBN 978-952-60-5150-5 (printed).
- Urpo, S. 1982. Observing methods for the millimeter wave radio telescope at the Metsähovi Radio Research Station and observations of the Sun and extragalactic

sources. Report S 132. Helsinki University of Technology, Radio Laboratory. ISBN 951-752-696-2.

Van Dam, A. 1997. Post-WIMP User Interfaces. *Communications of the ACM*. Vol. 40. No. 2. pp. 63-67. ISSN 0001-0782. (DOI: 10.1145/253671.253708).

Wharton, C. & Rieman, J. & Lewis, C. & Polson, P. 1993. The Cognitive Walkthrough Method: A Practitioner's Guide. In: Nielsen, J. & Mack, R. (eds.) *Usability inspection methods*. New York, NY. pp. 105-140. ISBN 0-471-01877-5.

Appendix A: Questions of the contextual inquiry preliminary interviews

Below are questions that were asked from every observer who participated in the contextual inquiry interviews:

How long have you been doing quasar observations / when did you start quasar observations in Metsähovi?

What kind of education do you have and on what field?

Do you have any experience in astronomical observations?

Have you done research in the field relating to the quasar observations?

Have you been involved in developing the quasar observation system?

Appendix B: Results of the contextual inquiry preliminary interviews

Table 1. Results of the contextual inquiry preliminary interviews.

Observer	MH observing experience	Education	Research experience	Other observing experience	Obs. system development
1	1 year 2 months	Ph.D. astronomy	?	IR, Optical, single dish radio	Feedback
2	2 years	5 th year astronomy student	Pro gradu on AGN's. Familiar with AGN types.	-	Feedback
3	2 years (long gap in between)	B.Sc (tech.) geodesy, student	-	-	Tau database tools developer, feedback, discussions on development
4	22 years	D.Sc. (Tech), space technology, astronomy studies	AGN researcher, intimately familiar with MH's sources	SEST, optical	As a consultant and expert of the observing work, feedback
5	4 years 6 months (2011 no obs.)	M.Sc. Astronomy.	Working on doctoral thesis on x-ray binaries	-	Feedback, comments
6	3 months (2 shifts of 3 days)	M.Sc. astronomy	Working on doctoral thesis on quasars, data from single dish, vlbi radio + other freq. Follows a single source intently.	2 nights with a spectrograph	-
7	24 years	D.Sc. (Tech) Space technology, astronomy studies	AGN researcher, intimately familiar with MH's sources	Multiple places, software development at SEST	Currently more a user. Previously development of observing software.
8	1 + 3 years 7 months	Ph.D. astronomy	Spectrum and time series modeling with Metsähovi data included. ~20 sources familiar, some very closely	-	Instruction clarification, auxiliary functions for Planck observations.

Appendix C: A sequence model from a contextual inquiry

Manual and automatic observations 2012-07-19:

Other breakdowns and suggestions reported by observer:

- **Breakdown: No possibility to attach comments to ongoing observation** → has to either memorize or write comment down until the observation has finished
- Uses surveillance cameras to see changes in the wetness of asphalt in the parking lot
- Sometimes notices that she's made errors in inputting sources in an autolist
 - Recovery from such errors means deleting all sources from list up to the last good input.
 - **Breakdown: Not possible to modify autolist items**
- Sometimes unintentionally observes the same source twice in one autolist.
 - **Breakdown: Autolist generator doesn't show on map which sources are on list**

Sequence model:

- **INTENT: Continue remote quasar observations after solar observations**
 - Look for calibration sources on autofile generator (notice that there are non visible)
 - (Start diode calibration observation. Not in notes)
 - **INTENT: Check weather**
 - Look for weather in surveillance cameras page
 - Open yr.no weather forecast page
 - **INTENT: Observe calibration source**
 - Open autofile generator
 - Look for a calibration source in a suitable region of sky.
 - **INTENT: Observe a normal 5p-source**
 - **TRIGGER: No cal source at suitable region. 3C274 is moving towards sector, but not yet there.**
 - Choose normal southern 5p-source for observing.
 - Write source name and offset readings to paper
 - **Breakdown: Has to use paper and pen to help alleviate memory load while giving observation commands.**
 - Move to VNC desktop
 - Browse command history in screen on daqger
 - Write current source's name, offsets (and integration time) from the paper to the fields of a previous observation command
 - Execute observation command

- **INTENT: Make a note of the current conditions**
 - Check weather
 - Look at results of finished observations in obs_summary.pl
 - Open obs_delcom.pl
 - **Breakdown:** Has to open separate tool for inputting comments.
 - **Breakdown:** Has to locate observation result anew in the separate tool (in this case not a big problem, because the result happens to be the latest one)
 - Write weather description into the comment field of the diode calibration that just finished
- **INTENT: Manage and correct pointing offsets (Check status (error & pointing) of current observation)**
 - Look at result of current 5p-observation in watch-results terminal window: After two results 5p-fit is bad
 - Wait for third fit result
 - Quit current observation
 - Take offset correction from watch-results, half the suggestion and correct the offset values with new ones
 - Browse observation command history in screen on daqqr.
 - Update new offset values to previous command
 - Look at surveillance cameras page: Fast moving clouds
 - Execute observation command
- **INTENT: Follow changes in weather conditions**
 - Look at surveillance cameras (still) page. Update manually.
 - Look at current observation results in watch-results terminal: 5p-fit is on the center and the error isn't particularly large
- **INTENT: Prevent moisture from building up on the dome for the duration of the upcoming night**
 - **TRIGGER:** Planning to observe automatically the next night
 - **INTENT: Approximate starting time for moisture removal**
 - **TRIGGER:** Won't be present when the moisture removal is supposed to start.
 - Open MB plotting tool
 - **Breakdown:** Has to use external tool to access relh data
 - Look at relative humidity readings from last night.
 - See what time relH surpassed 70 %.
 - Convert time in graph from UTC to EET (calculate in head)
 - **Breakdown:** Has to manually convert UTC--> EET
 - Input heating cycles to autoheat.pl in VNC desktop

- **Breakdown:** *Has to use input multiple cycles with constant duration (inputting times is repetitive, unpleasant and is prone to errors)*
 - **INTENT: Decide on ending time for moisture removal**
 - Open MB plotting tool
 - Look at relative humidity readings from last morning.
 - See when humidity dropped below 70 %.
 - Decide on time to end heating cycles
 - Input last heating cycle to autoheat.pl in VNC desktop according to the end time
- **INTENT: Manage pointing offsets. (Check status (error & pointing) of current observation)**
 - Look at result of current 5p-observation in watch-results terminal window. Azimuth direction pointing isn't optimal.
 - Wait for one more fit to decide on whether to continue or to stop the observation.
- **INTENT: Keep aware of current weather conditions**
 -
 - Open browser tab with “Antenna & Weather” page
 - Check especially real time relative humidity reading from high weather station
 - **Breakdown:** *Real time humidity reading is not available as a number in MB plots. (Most likely a need for both latest real time value and history in a graph)*
- **INTENT: Manage pointing offsets. (Check status (error & pointing) of current observation)**
 - Look at result of current 5p-observation in watch-results terminal window.
 - Offset value that was corrected is fine, but the value of the direction that was kept constant has deteriorated
- **INTENT: Determine current tau value**
 - Look at weather conditions (very briefly)
- **INTENT: Manage pointing offsets. (Check status (error & pointing) of current observation)**
 - Look at result of current 5p-observation in watch-results terminal window.
- **INTENT: Observe automatically until the next morning**
 - **INTENT: Add multi-frequency campaign source to autolist**
 - Open spreadsheet file containing visibility times of multi-frequency campaign sources gotten from another observer
 - **Breakdown:** *Has to use external list to get needed information on multi-frequency sources*
 - Choose a suitable source from the list

- Open/navigate to autofile generator
- Locate the source in question from drop down menu
- Click “Add”
- **INTENT: Add regular source to autolist**
 - Look at source symbols on autofile generator's sky view
 - Choose a suitable source (e.g. without a “due” marking, high enough priority)
- **INTENT: Start automatic observations**
 - Export the list file from autofile generator
 - **INTENT: Determine name for autolist**
 - Look at the date
 - **Breakdown:** Has to manually check date to name autolist file
 - Look at rain duration plot in MB
 - Notice that it might be raining
- **INTENT: Determine if it's raining**
 - Look at obs_summary.pl, look at results (what she looked at in obs_summary, is not in notes)
 - Look at Antenna & weather page
 - Look at cameras-still page
 - Look at yr.no forecast: no rain forecast
- **INTENT: Add comment to observation results about rain shower**
 - Open obs_delcom.pl
 - **Breakdown:** Has to open separate tool for inputting comments.
 - Add comment about rain shower
- **INTENT: Dry the dome after rain**
 - Cancel the current autoheat.pl list
 - **Breakdown:** Dome heater list has to be completely redone to add one new heating cycle
 - Input new list to autoheat.pl
 - Run new autoheat.pl list

Appendix D: Interview notes from the user tests

Pilot test, 2013-01-03:

- “New is better, because I can see everything and because it doesn't require jumping between tools”.
- Problems in the new system:
 - She dragged the mouse while clicking a delete tag to an observation → text got selected in the table instead.
 - When clearing/deleting tau values and comments, she would prefer to have a button to clear the values instead of inserting empty values.
 - Wonders why empty tau values are marked with a “-999” instead of empty cells.
 - Associates the dark blue/gray color denoting a selected row with inactivity. The dark color also makes it harder to read the text on the row. She hopes for a lighter blue color.
 - When tau values are selected and cancel is pressed, why do the selected tau values remain highlighted?

1. user test, 2013-01-09:

- Accidental click-drag resulted in a single unsuccessful delete tag adding
- The area in the table for deleting/undeleting is small → makes clicking hard
- About the highlighting of deleted results: Suggests that the text of deleted rows could be gray instead of highlighted
- **Commenting:**
 - Pressing the Insert button in the end feels redundant
 - Having an extra box open in the first place feels redundant
 - Did not know about inserting with “enter”
 - Would like to have spreadsheet-like navigating with keyboard keys
 - Tab, arrows, enter, etc
- **Tau-values:**
 - Technically easy
 - Raffling the values and deciding if it's winter or summer is hard
 - After insertion and canceling the selected values were still selected in the table
 - Before the test, behavior with “Cancel” was altered so that selection is removed when “Cancel” is pressed.
 - Empty values should be marked with empty cells and not with -999
- There could be color coding for inserted tau values → would help getting overview of data (interpretation)

- There could be color coding for weather-related values
- **In test, observer clicked a row assuming the row would be selected and the widget be updated.**
 - The widget can only be update by browsing with arrow and page up /down keys.
 - Observer used arrow keys to get to desired observation's

2. user test, 2013-01-10:

- **NOTICE: No training run on rain duration task on either system!**
- **First time user of obs_summary 1.4**
- Criticized the test setup and tasks as unrealistic
- **Commenting:**
 - Accidental mouse scroll click on comment field
 - Reports no problems with the functionality
 - Cautiously prefers current version over direct in-table commenting
- **Rain duration:**
 - When clicking the widget to a certain observation, accidentally opens tau dialog
- **Deletion:**
 - Likes the pink highlight for deleted observations
 - Would like a multi select possibility for marking multiple results as deleted in a case where results from a long time period would have to be deleted.
 - Clicking multiple for deletion individually might lead to errors
- **Tau value adding:**
 - Custom values are not supported
 - *Cancel* and *Close* are duplicates of the same functionality
 - Prefers the simplified tau dialog with only the drop down menu and *Close* -link
 - Does not think of having the -999 as empty tau as a problem-fixing
- **Plotter:**
 - Would like to have heater events automatically in the comments
 - When suggested, would like heater cycles visible in the plotter as e.g. red vertical regions
 - Because RelH is visible in the table, it's not important to have it in the plotter (RelH takes up space from some more important variables.)

3. user test, 2013-01-16:

- Interviewee does the final reduction and has access and can modify the pre processed data files directly.
- Did not use the old tau tools regularly, because perceived them to be hard to use.
- Before the test had never used the tau query and delete tools.
- **Deletion:**
 - Thinks the new system is simple
 - Would prefer that the delete column be placed more to the right in the table
 - Closer to the values that typically effect deletion (drift, error)
 - Position between RH and tau would be good
 - Similar to what obs_delcom was like
 - Closer to the other modifiable columns (tau, comments)
- **Assumed that pink highlight is for critical error and drift values**
 - Should all rows with a critical error or drift reading be highlighted with e.g. pink?
 - Sometimes perceives it as hard to see on which row e.g. a certain red error reading is on
 - This can cause problems in attaching comments and delete tags
 - Maybe there should be a context menu for e.g. deleting the result on which the mouse is?
- **Commenting:**
 - Wonders why there needs to be a dialog window at all.
 - Prefers direct manipulation of the comments in the table
- **Tau values:**
 - Had wondered why there are Insert and Cancel buttons at all
 - Tests a version of the tau dialog without Insert and Cancel buttons without problems
 - Tells she prefers the simplified version
 - Does not mind “-999” as an indicator of empty values
 - Does not mind inserting -999 when “removing” tau values
- **Rain duration:**
 - Prefers the new system, thinks it is easier for accessing the data
 - Tells that she generally hasn't looked much at rain duration and sky temp for example.
 - Has relied more on radar images for checking rain
- **Might be somewhat distracted by the dot that tracks the mouse position in the data points in the graph.**
 - In an earlier test asked for the meaning of the dot

- Now user mentioned jokingly that when she's bored, she can play with the dot, looking at how it jumps around in various graphs.

4. user test, 2013-01-17:

- Has never even heard of rain duration and had no idea on its significance on detecting rain until after the introduction run
 - Tells that he normally uses FMI Helsinki Testbed radar images and relH and skytemp graphs for determining if it has rained e.g. last night.
- **Test director forgot to run training for checking rain duration on old system**
- **General:**
 - Possibility to be able to accomplish multiple tasks in one place
 - No jumping between tools and memorizing times and values
 - Color coding can now be utilized in the table
- **Deletion:**
 - Would like to be able to paint multiple values at a time if needed
 - However, wants to retain ability to single click individual delete tags
- **Commenting:**
 - Mentions that it might be good to be able to flag a time period with one comment or a “?” flag to indicate e.g. a period when there possibly has been snow on the dome.
- **Tau values:**
 - Does not consider “-999” as bad for inserting empty values, but for literally clearing/deleting tau values, would prefer a “remove” button.
 - Mentions about the large number of clicks required when inserting values
 - Is positive toward simplified tau dialog without seeing it
- **Plotter:**
 - Would like Tout and Tant added together in one plot
 - Behind this is the need to follow the changes in temperatures caused by use of the heater
 - These changes are also indicated well by delta(TcontFreeAir) and delta(Vpeltier)
 - The observer decides the effect of the possibly “compromised” temperature conditions on the observations
 - What to observe during changing temperatures?
 - Are results good quality, deletable during these conditions?
- **Obs_summary 1.41:**
 - Mentions general obs_summary notes:
 - Why are some values, e.g. CAL results, RelH, OutT, etc presented with so many decimals

- Possibility to save space by cutting down significant decimals
- New position of delete column is good
- Likes the idea of having heater running times visible in the graphs (especially the ones that have a non-obvious connection to the heater)
- **Related ideas on future heater UI:**
 - Start time, running time, time until auto shutdown visible when in manual operation
 - All heater activities in one UI
 - Prevent accidental use
- **Weather summary demo:**
 - (Page has dygraphs plots of outside temp, relH, skytemp and rain, together with dome camera and all-sky image in addition to Foreca Kylmäla forecast graphic)
 - Is positive toward it
 - Would like to have FMI testbed rain radar on the page as well

Appendix E: Test plan for the user tests

Test plan

Goal of the testing is to compare the usability of the old and new observation summary tool in the hands of expert users. The goal is to see if the changes to the system have improved its effectiveness, efficiency and user satisfaction. A secondary goal is to see what problems the users have with the new user interface and to gather feedback for further development. The testing is to take place in the Metsähovi radio observatory in January 2013.

Tasks to be included in the tests are:

- Adding comments to observation results data
- Marking observation results for deletion
- Adding tau values to the observation results.
- Removing tau values from the observation results.

The tasks are first run with the old system (obs_summary 1.33, obs_delcom, tau_add, tau_query, tau_delete) and then with the revised system (obs_summary 1.4). To reduce the effect of a possible learning curve, especially with the old system, the user is first given practice tasks. The more proficient the user is with both versions, the better.

For every task, three parameters are noted down:

- Is the task successfully completed? Yes, No
- How many errors did the user make?
- How long did it take to finish the task?

After running the test tasks with both system versions, a follow-up interview is run.

Appendix F: Interviewer's version of test procedures for the user tests

When setting up the test, tell the user:

- The point is to compare usability of old and new obs_summary
- The point is to test the system, not the user.
- In the beginning, the user is told that the test is recorded on video. Point is to record the actions on the screen, not the user.
- All the recordings are confidential.
- The user is told that he/she can stop the test at any time if she feels uncomfortable.
- The user is also told that the test director will be making notes during the test but the user doesn't have to worry about that.
- Thinking aloud isn't necessary, but the user is free to talk and make comments.
- The user can also ask questions. However, during the test, the experimenter can not answer all questions to avoid affecting the test results.
- The user can ask questions before starting the test if something is unclear.
- First, the user is asked to run practice tasks with the system.
- Then the user runs the test tasks.
- The goal is to attempt completing the tasks in a timely manner without unnecessary pauses. It's completely acceptable to have difficulties in running the tasks.
- After every tasks the user will be handed a short questionnaire.
- After completing the tasks the user will be handed a SUS questionnaire.
- After completing tasks with both versions, there will be a wrap-up interview.

Starting scenario:

You have observed automatically for a night and wake up in the morning to see what the observation results look like. You'd like to have the data from the night ready for returning for reduction in the evening. So you decide to go through the data and add tau values and comments to the data and mark bad quality observations for deletion.

I will ask you to run the tasks with two different software versions.

Test tasks**1. Marking results for deletion, OLD**

You would like to delete bad quality observations. You decide to only delete such results that have their error (Err) readings marked red in obs_summary.

The task is successful when all the observations that are eligible for deletion are marked for deletion. User is expected to click the delete field of a result.

Success	Y / N
Errors	
Time	

2. Adding comments to the data, OLD

You notice that some offset correction values in the results seem suspicious. These values are colored red in obs_summary. You want to notify Anne of the values by adding the comment "Notice!" to the suspicious results.

The task is successful when the user has added comments to all of the suspicious results.

Success	Y / N
Errors	
Time	

3. Marking tau values, OLD

- a) *You would like to add tau values to the results. You have already written the points when the tau value changes in the comments, so you only need to feed the values to the database.*

The task is successful when all the tau column cells in the results table have been filled according to the values in the comments

Success	Y / N
Errors	
Time	

- b) *You notice that you have made a mistake in adding the tau values. You now want to change the tau values for the observations X and Y*

The task is successfully completed when the desired tau values have been changed in the observation summary table.

Success	Y / N
Errors	
Time	

4. Checking rain duration, OLD

You notice two results near the beginning of the shift with high error readings. The error readings are marked with red color. You want to know if the the rain duration measured by the (upper) weather station has increased during these observations.

The task is successful when the user has opened the RainDur plot and determined if it has changed during either of the observations.

Success	Y / N
Errors	
Time	

RUN SUS!

5. Marking results for deletion, New

You would like to delete bad quality observations. You decide to only delete such results that have their error (Err) readings marked red in obs_summary.

The task is successful when all the observations that are eligible for deletion are marked for deletion. User is expected to click the delete field of a result.

Success	Y / N
Errors	
Time	

6. Adding comments to the data, New

You notice that some offset correction values in the results seem suspicious. These values are colored red in obs_summary. You want to notify Anne of the values by adding the comment "Notice!" to the suspicious results.

The task is successful when the user has added comments to all of the suspicious results.

Success	Y / N
Errors	
Time	

7. Marking tau values, New

- a) *You would like to add tau values to the results. You have already written the points when the tau value changes in the comments, so you only need to feed the values to the database.*

The task is successful when all the tau column cells in the results table have been filled according to the values in the comments

Success	Y / N
Errors	
Time	

- b) *You notice that you have made a mistake in adding the tau values. You now want to remove the tau readings whose value is 0.050*

The task is successfully completed when the desired tau values have been changed in the observation summary table.

Success	Y / N
Errors	
Time	

8. Checking rain duration, New

You notice two results near the beginning of the shift with high error readings. The error readings are marked with red color. You want to know if the the rain duration measured by the upper weather station has increased during these observations.

The task is successful when the user has opened the RainDur plot and determined if it has changed during either of the observations.

Success	Y / N
Errors	
Time	

RUN SUS!

Appendix G: User's version of test procedures for the user tests

Each task description and pleasantness question is handed to the user one at a time on a separate piece of paper. Task description is handed first. After the user completes the task, the pleasantness question is handed.

SCENARIO:

You have observed automatically for a night and wake up in the morning to see what the observation results look like. You'd like to have the data from the night ready for returning for reduction in the evening. So you decide to go through the data and add tau values and comments to the data and mark bad quality observations for deletion.

You would like to delete bad quality observations. You decide to only delete such results that have their error (Err) readings marked red in obs_summary.

Doing this task was pleasant:

Strongly disagree 1 2 3 4 5 Strongly agree

You notice that some offset correction values in the results seem suspicious. These values are colored red in obs_summary. You want to notify Anne of the values by adding the comment "Notice!" to the suspicious results.

Doing this task was pleasant:

Strongly disagree 1 2 3 4 5 Strongly agree

You would like to add tau values to the results. You have already written the points when the tau value changes in the comments, so you only need to feed the values to the database.

Doing this task was pleasant:

Strongly disagree 1 2 3 4 5 Strongly agree

You notice that you have made a mistake in adding the tau values. You now want to remove the tau readings whose value is 0.050

Doing this task was pleasant:

Strongly disagree 1 2 3 4 5 Strongly agree

You notice two results near the beginning of the shift with high error readings. The error readings are marked with red color. You want to know if the rain duration measured by the upper weather station has increased during these observations.

Doing this task was pleasant:

Strongly disagree 1 2 3 4 5 Strongly agree

Appendix H: Wrap-up interview template for the user tests

The following questions were asked from the user after the test tasks and the latter SUS-questionnaire were completed. The questions considered the new version of the system. The interviewer asked the questions verbally and noted down the answers. The interview was not recorded.

Deletion:

1. Was the task easy or hard?
2. What was hard/easy about the task?
3. If something proved difficult, how could it be developed in the future?

Adding comments:

2. Was the task easy or hard?
2. What was hard/easy about the task?
3. If something proved difficult, how could it be developed in the future?

Adding tau values:

3. Was the task easy or hard?
2. What was hard/easy about the task?
3. If something proved difficult, how could it be developed in the future?

Removing tau values:

4. Was the task easy or hard?
2. What was hard/easy about the task?
3. If something proved difficult, how could it be developed in the future?

Checking rain duration:

5. Was the task easy or hard?
2. What was hard/easy about the task?
3. If something proved difficult, how could it be developed in the future?

General question:

4. Which system version did you like better? Why?

Appendix I: SUS-questionnaire for the user tests

The SUS-questionnaire template below was used in a printed form in the usability tests of the second iteration round.

SUS questionnaire

*Pakollinen

1. I think that I would like to use this system frequently *

1 2 3 4 5

Strongly disagree Strongly agree

2. I found the system unnecessarily complex *

1 2 3 4 5

Strongly disagree Strongly agree

3. I thought the system was easy to use *

1 2 3 4 5

Strongly disagree Strongly agree

4. I think that I would need the support of a technical person to be able to use this system *

1 2 3 4 5

Strongly disagree Strongly agree

5. I found the various functions in the system were well integrated *

1 2 3 4 5

Strongly disagree Strongly agree

6. I thought there was too much inconsistency in the system *

1 2 3 4 5

Strongly disagree Strongly agree

7. I would imagine that most people would learn to use this system very quickly *

1 2 3 4 5

Strongly disagree Strongly agree

8. I found the system very cumbersome to use *

1 2 3 4 5

Strongly disagree Strongly agree

9. I felt very confident using the system *

1 2 3 4 5

Strongly disagree Strongly agree

10. I needed to learn a lot of things before I could get going with the system *

1 2 3 4 5

Strongly disagree Strongly agree