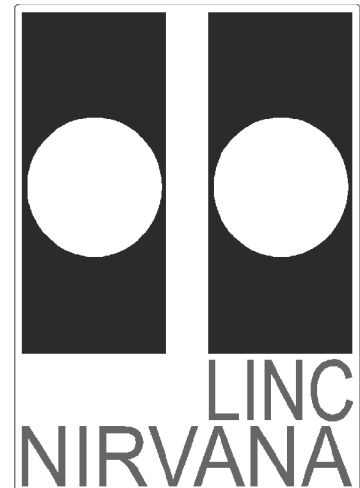


# LINC-NIRVANA

The **L**BT **I**Nterferometric **C**amera and  
**N**ear-**I**nfra**R**ed / **V**isible **A**daptive  
**i**Nterferometer for **A**stronomy

A collaborative project of the MPIA Heidelberg, INAF-Arcetri,  
Universität zu Köln, and MPIfR Bonn

<http://www.mpia.de/LINC>



## LINC-NIRVANA

-

### Observation Preparation Software

Doc. No. LN-MPIA-FDR-ASW-001  
Short Title FDR-ASW-OPS  
Issue 1.1  
Date 19 May 2004

Prepared	<u>W. Gaessler</u>	<u>19 May 2005</u>	
	Name	Date	Signature
Approved	<u>T. Driebe</u>	<u>19 May 2005</u>	
	Name	Date	Signature
Released	<u>M. Kuerster</u>	<u>19 May 2005</u>	
	Name	Date	Signature

## Document Change Record

Issue	Date	Section/ Affected	Paragraph	Reasons / Remarks	Name
0.1	20 October 2004	All		created	gaessler
1.0	21 December 2004	All		editing for release	gaessler
1.1	19 May 2005	All		PI comments	gaessler, pavlov

# TABLE OF CONTENTS

<b>1 Scope .....</b>	<b>5</b>
<b>2 Applicable documents .....</b>	<b>5</b>
<b>3 External Interfaces .....</b>	<b>5</b>
<b>4 Acronyms and abbreviations .....</b>	<b>5</b>
<b>5 Introduction .....</b>	<b>6</b>
<b>6 General Description .....</b>	<b>6</b>
6.1 Software Perspective.....	6
6.2 User Classes and Characteristics.....	6
6.2.1 Astronomer.....	6
6.2.2 Support Astronomer/Operator/Observer.....	7
6.3 Operating Environment.....	7
6.4 General Constraints, Assumptions, Dependencies, Guidelines.....	7
6.5 Design and Implementation Constraints.....	7
6.6 User Documentation.....	7
<b>7 Specific Requirements .....</b>	<b>8</b>
7.1 Constraint requirements.....	8
7.1.1 Browse images.....	8
7.1.2 Browse astronomical catalogs.....	8
7.1.3 Overlay instrument configuration.....	8
7.2 Capability Requirements.....	9
7.2.1 Write/display parameters.....	9
7.2.2 Estimate performance.....	9
7.2.3 Estimate exposure time.....	10
7.2.4 Build observation program .....	10
7.2.5 Evaluate scheduling.....	11
7.2.6 Submit observation program.....	12
<b>8 User Interface Requirements .....</b>	<b>14</b>
8.1 GUI.....	14
8.2 CLI.....	16

8.3 API.....16

8.4 Diagnostics.....16

8.5 Operational scenario for astronomer.....16

## List of Tables

## List of Figures

**Figure 1. Use case diagram for LINC-NIRVANA Observation Preparation Software. The requirements are derived from this use cases.....13**

**Figure 2. Skycat graphical user interface.....14**

**Figure 3. The Scientist's Expert Assistant (SEA). .....15**

**Figure 4. Activity Diagram shows the operational scenario for an astronomer.....17**

## 1 Scope

The document contains the requirements for the observation preparation software. An overview of a possible software design for this part is given. A deeper and more detailed software design will be given in future documentation. The main goal for the FDR is to constrain the Observation Preparation Software and define the constraint for the other software components.

## 2 Applicable documents

No.	Title	Number & Issue
1	Software style guide	LN-MPIA-MAN-ICS-001
2	Observation Support Software	LN-MPIA-FDR-ICS-002

## 3 External Interfaces

Item	Short description
OSS	Software interface to the Observation Support Software. Parameter files from the OPS have to be stored in a repository of the OSS and executed from a observation tool of the OSS.
User	Graphical user interfaces for astronomers are needed.

## 4 Acronyms and abbreviations

HW	Hardware
PDR	Preliminary Design Review
SW	Software
OPS	Observation Preparation Software
LBT	Large Binocular Telescope
DRS	Data Reduction Software
LN	LINC-NIRVANA
LINC	Large binocular telescope Interferometric Camera
NIRVANA	Near InfraRed and Visible Adaptive iNterferometric for Astronomy
GUI	Graphical User Interface
OP	Observation Program

OB	Observation Block
TBD	To be Defined

## 5 Introduction

Astronomical observation with interferometers is more complex than conventional imaging or spectroscopy. It not only needs advanced data reduction algorithms to extract the information from the data, but also needs a well-prepared observer at the time of observation. In this document, we will define the requirements for the observation preparation software for LINC-NIRVANA the Fizeau interferometer for the Large Binocular Telescope. The observation preparation software will be a tool to support the observer in advance to get an idea what kind of observations he has to do and what quality of data he can expect under given conditions. Later, this software can be extended to other interferometric instruments.

## 6 General Description

### 6.1 Software Perspective

The observation preparation software will be a stand alone package with interfaces to the observation block repository and observation block executor (see LN-MPIA-FDR-ICS-002 Observation Support Software).

### 6.2 User Classes and Characteristics

#### 6.2.1 Astronomer

The OPS is mainly used by the astronomer. He wants to have software which he can use **off and online** on his own computer. The package should be simple to install and not need any external pre-installed software. It should provide a simple GUI by mean of which he can **interactively** prepare his observation and already gets an idea what he can expect. **Feasibility, performance as well as efficiency checks of the final observation** are needed. A **short execution time** is required, checks should be performed within minutes. The observation should be **structured hierarchically** to support easy repetitions with small changes of some parameters. Single-target programs as well as survey-like programs should be supported.

The characteristics in short:

- Should work on and off-line
- GUI should support interactive preparation
- Feasibility, performance and efficiency checks of observation
- Short execution time
- Hierarchical structure of observations

### 6.2.2 Support Astronomer/Operator/Observer

The Support Astronomer/Operator/Observer will use the software at the telescope site. He will use the tool mainly to **browse the prepared observation blocks**. The OPS has to **work together with the observation block repository, scheduler and the observation block executor**, all tools which will monitor the execution and completion of the programs at the site and part of the OSS. **Online changes** of the blocks through the OPS at the site should be allowed. **Consistency checks and update with the other tools should be automatic.**

The characteristics in short:

- Observation block browser
- Interface to repository, scheduler and executor
- Online changes should be possible
- Consistency check and update with other tools automatically

## 6.3 Operating Environment

The software should be written as a **platform-independent** application.

## 6.4 General Constraints, Assumptions, Dependencies, Guidelines

As platform-independent development is quite difficult, we focus first on an implementation under LINUX and will extend it later to further platforms.

A **web interface** to the repository for observation block submission is required.

At the telescope site, the OPS has to have **interfaces to the repository, the scheduler and the executor**.

A possible approach could use Jsky as base interface and use XML as interface language or hierarchical FITS keywords.

## 6.5 Design and Implementation Constraints

UML design is suggested.

The implementation should not need extensive resources (<1GB disk space, <128MB RAM).

Algorithms should be fast (<1min if possible).

## 6.6 User Documentation

- User manual (PDF) with a tutorial guiding through all functionalities (examples)
- On-line help (optional)

## 7 Specific Requirements

### 7.1 Constraint requirements

#### 7.1.1 Browse images

**Description:**

- The GUI must be able to browse FITS images from a local disk as well from web-based image databases.
- The image browser could be JSky from ESO. (Jsky is freely available under the GNU public license.)
- The image browser should provide the same functionalities as JSky or Skycat.

**Priority:** High

**Stability:** Stable

**Verifiability:** Visual checks.

#### 7.1.2 Browse astronomical catalogs

**Description:**

- The GUI must be able to load and display astronomical catalogs through the web or from local files (ASCII).
- The GUI must have same catalog functionalities such as JSky or Skycat from ESO.

**Priority:** High

**Stability:** Stable

**Verifiability:** Test catalogue for test images.

### 7.1.3 Overlay instrument configuration

**Description:**

- Selected reference stars, the target and the FoV must be visualized.
- Must be able to select reference stars and target interactively.
- Must be able to move the FoV interactively

**Priority:** High**Stability:** Stable**Verifiability:** Visual check. Certain care for zoom function.

## 7.2 Capability Requirements

### 7.2.1 Write/display parameters

**Description:**

The astronomer must be able to define following

**primary parameters:**

- target (through coordinates [RA,DEC])
- filter/band
- signal to noise ratio (SNR)
- magnitude of target in selected band (MAG)
- (automatically retrieved from astronomical catalog if available)

**secondary parameters:**

- seeing -> constrain strehl and fringe contrast
- (position of max. strehl?)
- airmass -> constraint on scheduling time

**tertiary parameters:**

- all instrument parameters and commands

**Definition:**

- Primary parameters are mandatory.
- Secondary parameters are optional.
- Tertiary parameters are defined automatically but can be overwritten.

**Priority:** Medium**Stability:** Parameters can change during different development phases.

**Verifiability:** Visual checks and test scripts to check parameter files.

### 7.2.2 Estimate performance

**Description:**

- An estimated performance of the observation under given conditions is required.
- The performance should be shown with a
  - strehl map,
  - differential piston map and
  - fringe contrast map.
- Map display through a menu/button.
- Input parameters for such maps are:
  - strehl map:
    - ◆ seeing, atmospheric layers,
    - ◆ AO reference stars magnitude,
    - ◆ AO reference stars geometry
  - differential piston map:
    - ◆ strehl map
  - fringe contrast map:
    - ◆ differential piston map,
    - ◆ FFT reference star magnitude,
    - ◆ FFT reference star position,
    - ◆ target position

**Priority:** Medium

**Stability:** Parameter and algorithms will evolve during different development phases.

**Verifiability:** Compare to numerical simulations and real observations. Manual checks.

### 7.2.3 Estimate exposure time

**Description:**

- Depending on the estimated performance and resulting constraints, the exposure time must be estimated.
- The total integration time should be taken and automatically divided into incremental exposures (automatic multiplication of parameter files).
- The exposure time depends on following parameters:
  - SNR
  - magnitude of target in selected filter
  - fringe contrast at target position

- type of target (pointlike or extended)

<b>Priority:</b>	Low
<b>Stability:</b>	Will evolve within different development phases.
<b>Verifiability:</b>	Observational checks.

#### 7.2.4 Build observation program

##### Description:

- The Observation program consists of one or more observation blocks.
- Each observation block consists of one or more parameter files which can be combined flexibly.
- Only a limited number of different templates for parameter files exist.
  - Target parameter file:
    - ◆ RA
    - ◆ DEC
    - ◆ EPOCH
    - ◆ (siderial tracking)
    - ◆ parallactic angles
  - Setup parameter file:
    - ◆ filter
    - ◆ reference stars (RA, DEC, mag)
  - Exposure parameter file
    - ◆ exposure time
  - Offset parameter file:
    - ◆ offset RA, DEC
  - Constrain parameter file:
    - ◆ seeing
    - ◆ airmass
- Templates of observation programs should be provided depending on observation mode, nodding/dithering sequence, maximal and incremental exposure time, number of parallactic angles, etc.
- Note: This process will happen after a first assessment of the performance.

##### Definition:

- **Parameter file:** Consists out of a sum of parameters defining the setup of the instrument and telescope.
- **Observation block:** Consists of several parameter files which define and constrain a number of exposures. It is the smallest schedulable unit to be

executed.

- **Observation program:** Sum of all observation blocks which have to be executed until the full scientific goal can be reached.

**Priority:** Low

**Stability:** Will evolve over different development phases.

**Verifiability:** Test scripts to check parameter file, example of an observation program.

### 7.2.5 Evaluate scheduling

#### Description:

- Possible scheduling slots should be provided (earliest start time, latest start time for certain parts of the program depending on the day of observation).
- Selection of certain scheduling models may be provided (one night, within days, within month, same conditions scheduling).

#### Note:

- Assuming that each observation program will demand 3 parallactic angles within 60 degree distance, it might be difficult to finish several OPs within one night. (First - last observation has 12h distance, 1/2h - 1h per target -> in a long winter night ~3-6 maximum)
- An observation program will probably be queued over days maybe even months. Therefore it does not make sense, that a absolute and detailed schedule is selected by the astronomer at home.

**Priority:** Low

**Stability:** Will evolve over different development phases.

**Verifiability:** Test scripts and manual checks. Observational checks.

### 7.2.6 Submit observation program

#### Description:

- The observation program should be submitted as XML ASCII files.
- It is required to submit the observation program to a repository through the net (ftp could be a temporary solution)
- The observation program should be protected (not everybody should have access the repository)
- A submitted program can be modified later again (re-submission).

**Priority:** Low

**Stability:** Will evolve over development phases.

**Verifiability:** Test scripts for test submission.

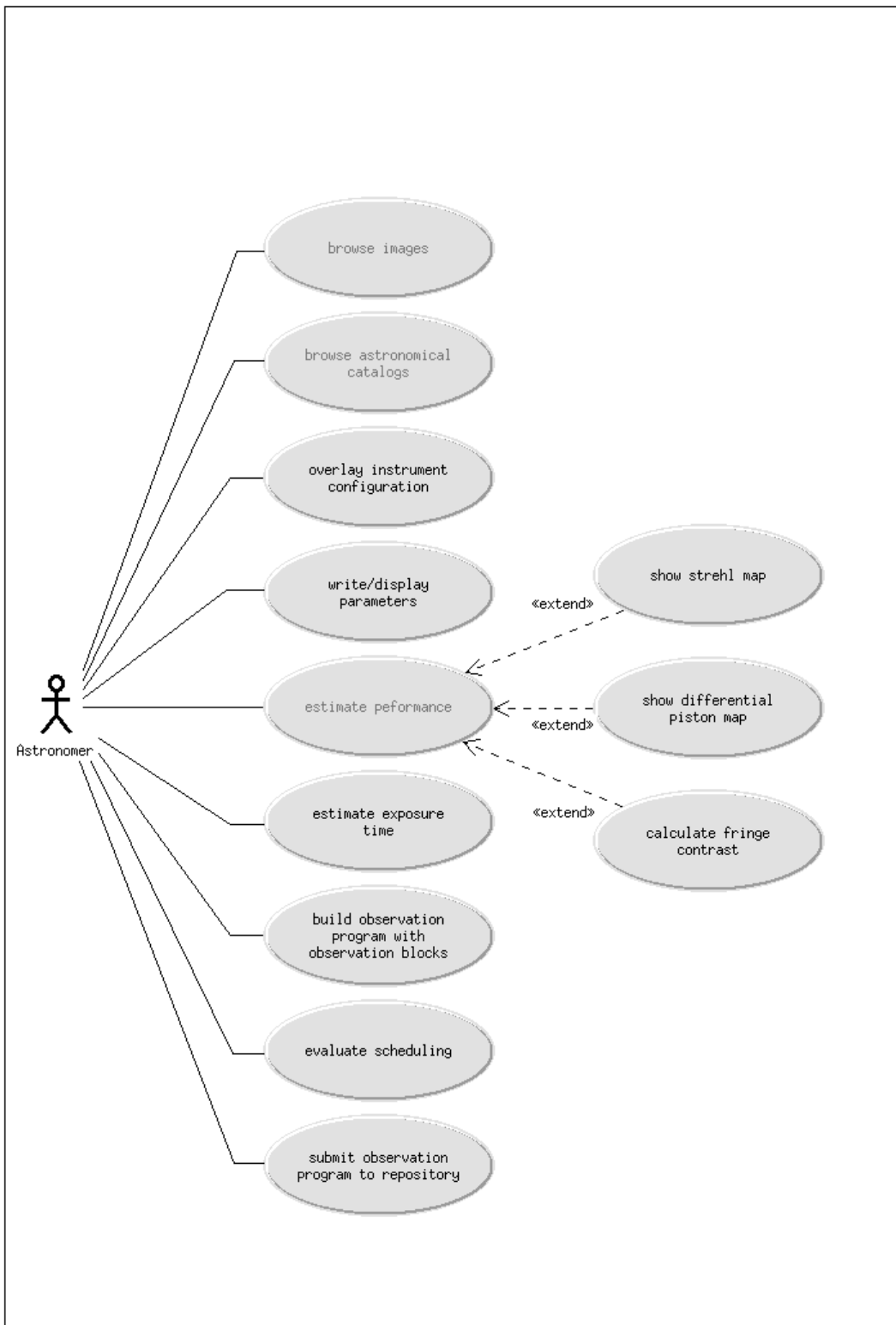


Figure 1. Use case diagram for LINC-NIRVANA Observation Preparation Software. The requirements are derived from this use cases.

## 8 User Interface Requirements

### 8.1 GUI

The image browser could be derived from Skycat or the Scientist's Expert Assistant (SEA). Skycat as well as SEA are systems that combine interactive visual and expert system approaches to assist astronomers in planning observations. SEA was developed by the Advanced Architectures and Automation Branch of the NBASA Goddard Space Flight Center in conjunction with the Space Telescope Science Institute. Skycat was developed by ESO.

The image browsers shown in Figure 2 have following menus:

- File
- View
- Graphics
- Go
- Data-Servers

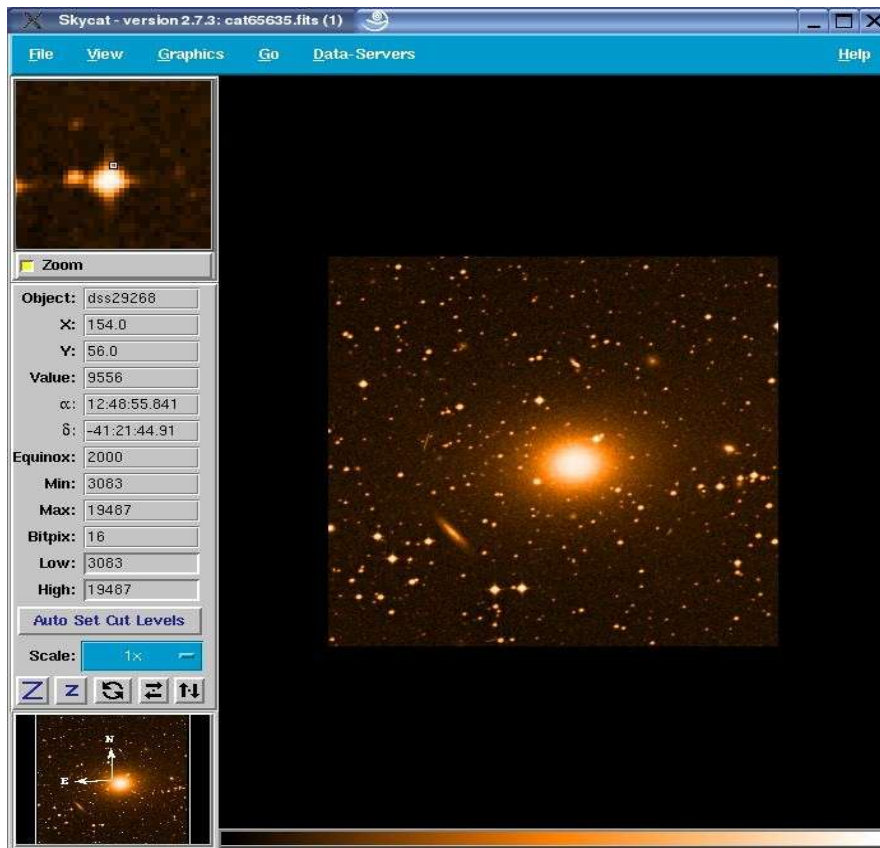


Figure 2. Skycat graphical user interface

The interface will be extended with an LN-Program menu containing following sub-menu:

- Target
  - Target list
    - ◆ Magnitude of targets for the selected filters
  - Field list
    - ◆ Offset
- Filter
  - All filters with radio buttons to select
- SNR
- Nodding scheme
  - Certain nodding schemes with offset parameters
- Observation angles
  - Scheme of angles, zero angle, angles free to define
- Seeing and airmass parameters
- Distance from moon
  - Bright time, dark time, minimal distance

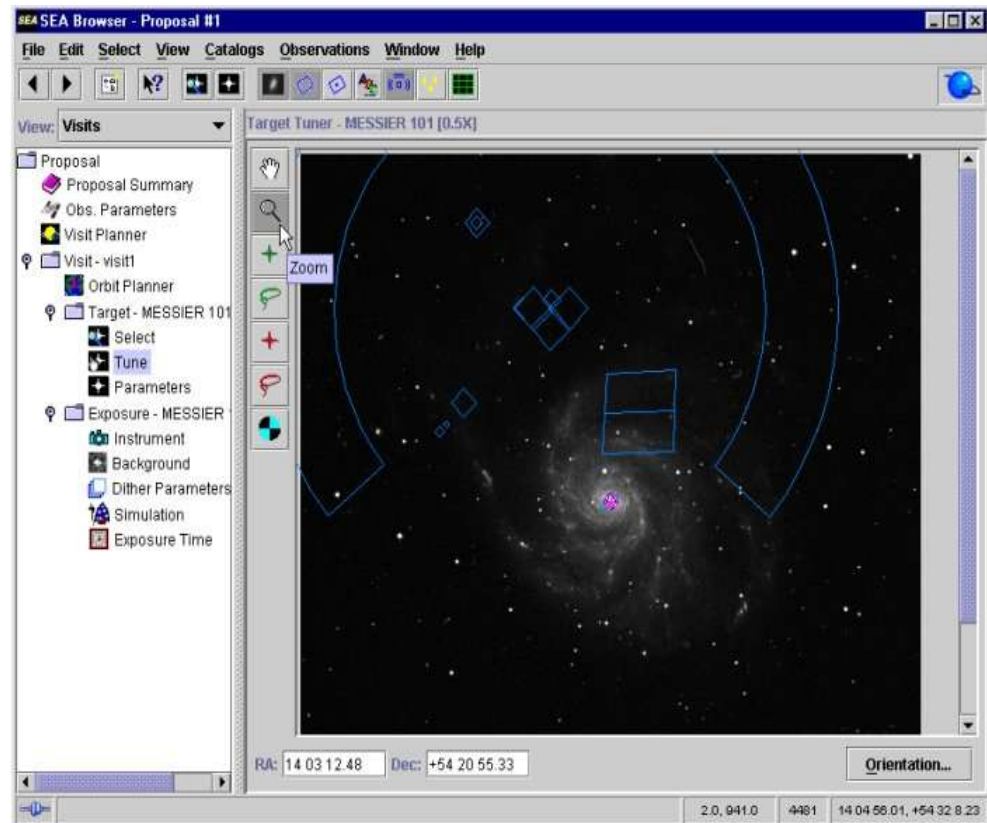


Figure 3. The Scientist's Expert Assistant (SEA).

## 8.2 CLI

No command-line interface is planned.

## 8.3 API

The application interface will depend on the basic image browser chosen.

## 8.4 Diagnostics

Debugging and diagnostic data will be in ASCII format in log files.

## 8.5 Operational scenario for astronomer

The astronomer has only interface for observation preparation through the main GUI:

- He starts the preparation process by starting the main GUI for OPS.
- He defines his targets and basic parameters for his observation program
  - Target list
  - Filters
  - Nodding scheme
  - Parallactic angles

- Seeing and airmass
  - Distance from moon
- Reference stars are selected automatically and a performance estimation is done. If the astronomer is not satisfied he can interactively change the reference stars or constrain parameters.
- The astronomer does an evaluation of the overall exposure time depending on the performance analyses, definition of SNR, atmospheric parameters and target magnitudes.
- The observer get an observation template which he can change for his needs. A re-evaluation of the performance might be necessary.
- Observation slots are proposed. The observer can constrain them but has to be aware that this will reduce the possibility of observation.
- The program can be submitted to the repository. Further changes can be done only on request or at the side.

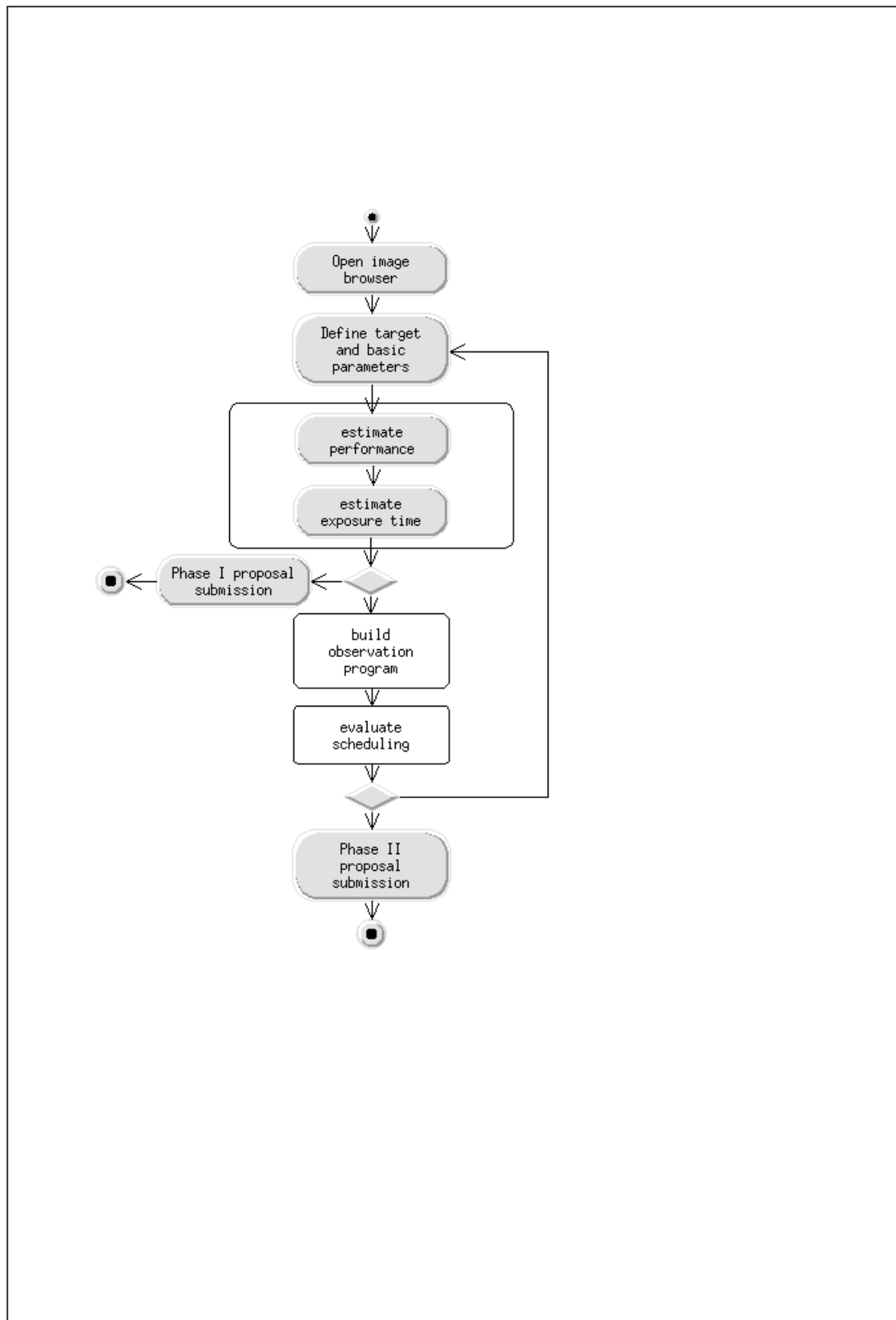


Figure 4. Activity Diagram showing the operational scenario for an astronomer.