

ESO

IT Talk #1

Use Cases

for Requirements Capture and Tracing
and UML as a standards modelling
language for SW projects

G.Chiozzi

European Southern Observatory



Garching 09/12/1999

ESO IT Talks

Introduction

Purpose of this tutorial is to:

- Introduce a **Use Case** driven development process
- Introduce the **Unified Modeling Language**

by means of a

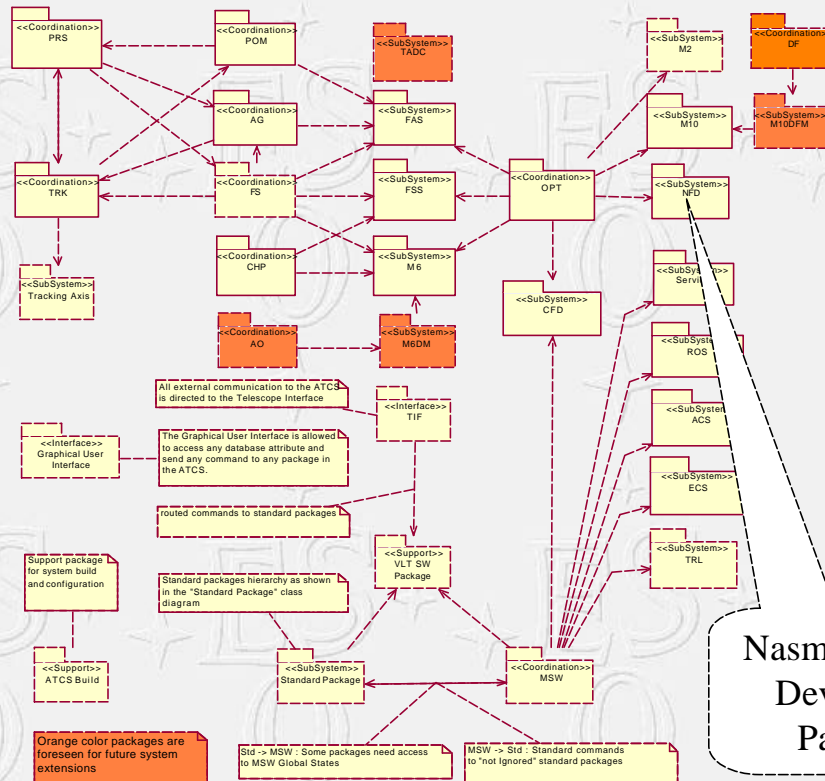
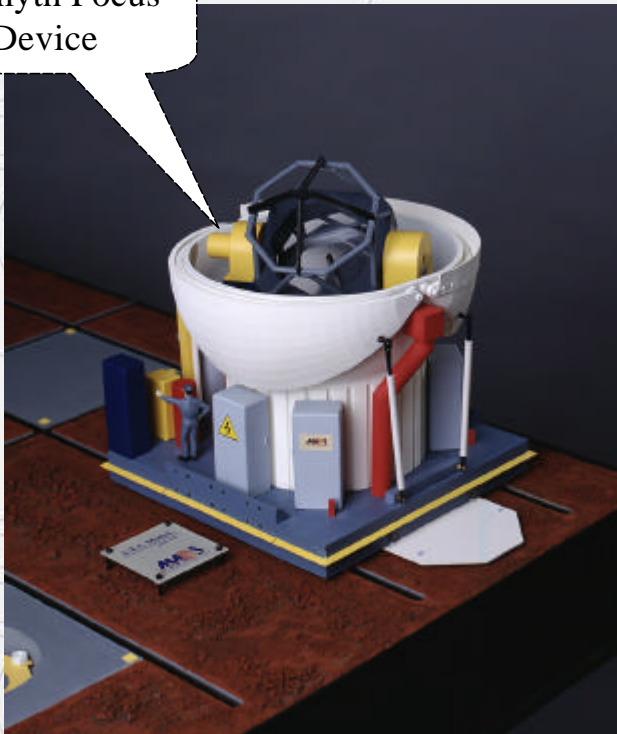
Real Life Example

The project: a simple control system

Model of VLTI Auxiliary Telescope (AT)

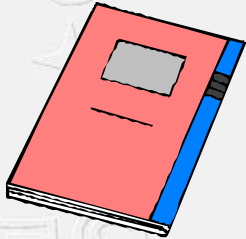
AT Software Packages

Nasmyth Focus Device



Nasmyth Focus Device SW Package

System Requirements



Statement of work



Other documentation



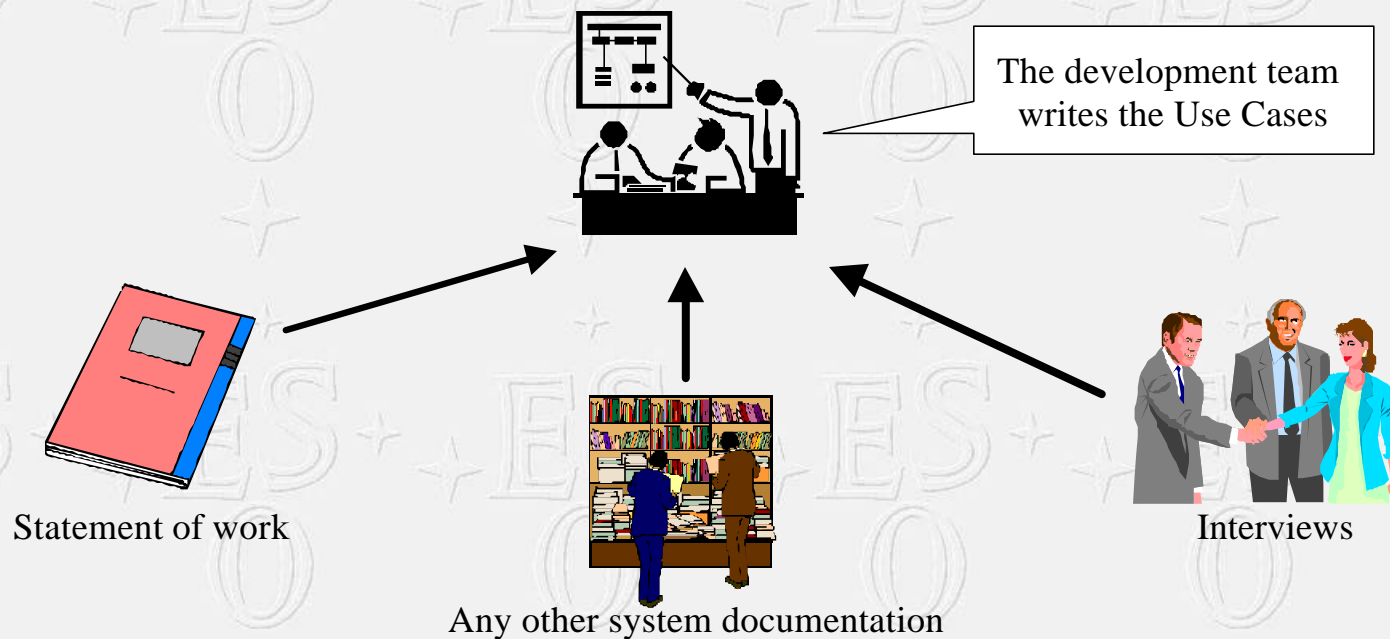
Interviews

- The rotating wheel to be installed at the Nasmyth focus shall have at least 4 selectable predefined positions:
 - 1: free hole. Diameter 30mm minimum
 - 2: Flat mirror
 - 3&4: Free for ESO alignment tools (Light Beacon)
- The rotation range of the wheel should be limited to 360 deg. The positioning repeatability in all directions shall be less than 0.5mm...
- In position 3&4 shall be possible to chop the wheel between two close positions...
- The Nasmyth wheel is an off-the-shelf rotary table from MICOS: DT-65...

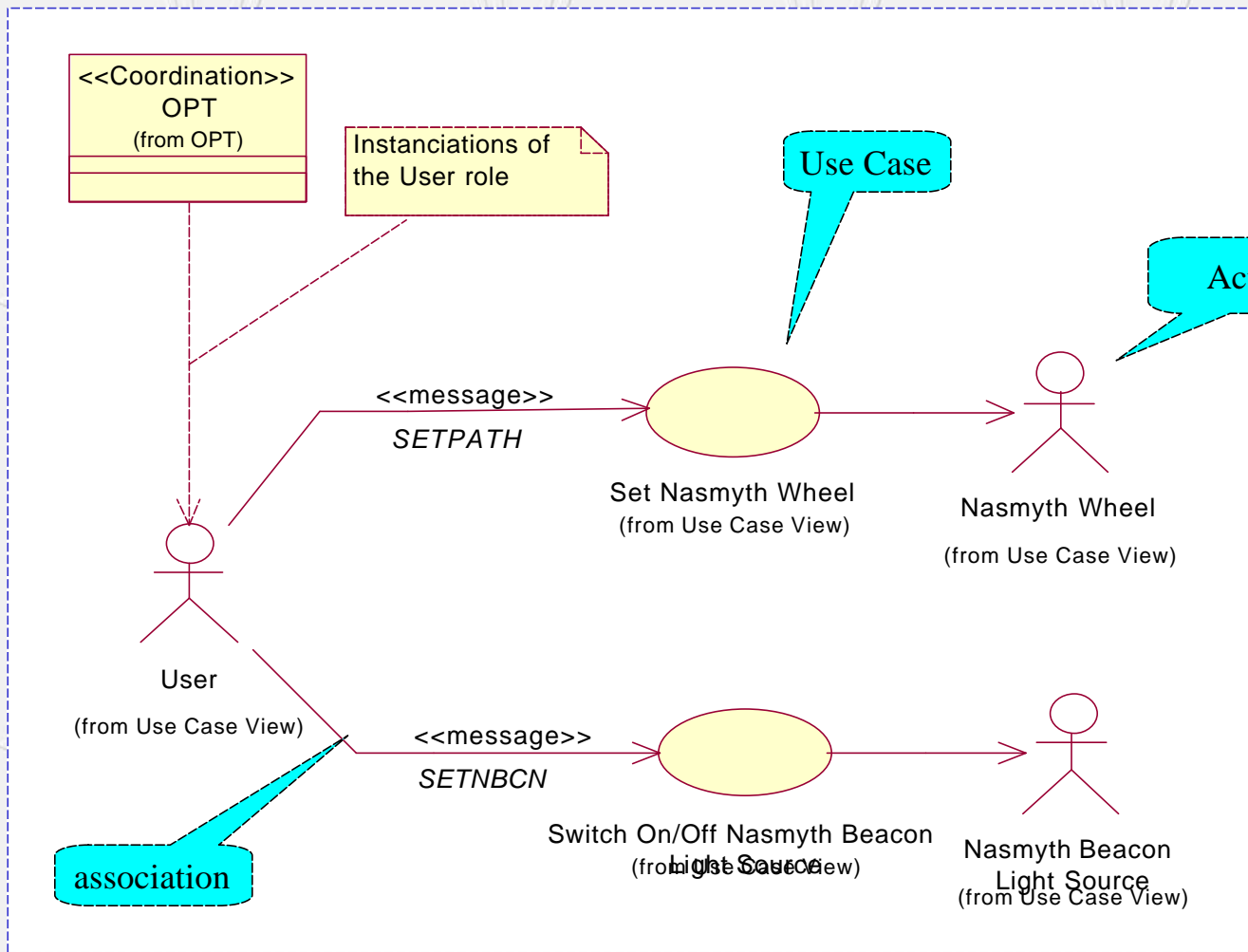
Identify Use Cases

- **Use Case:**

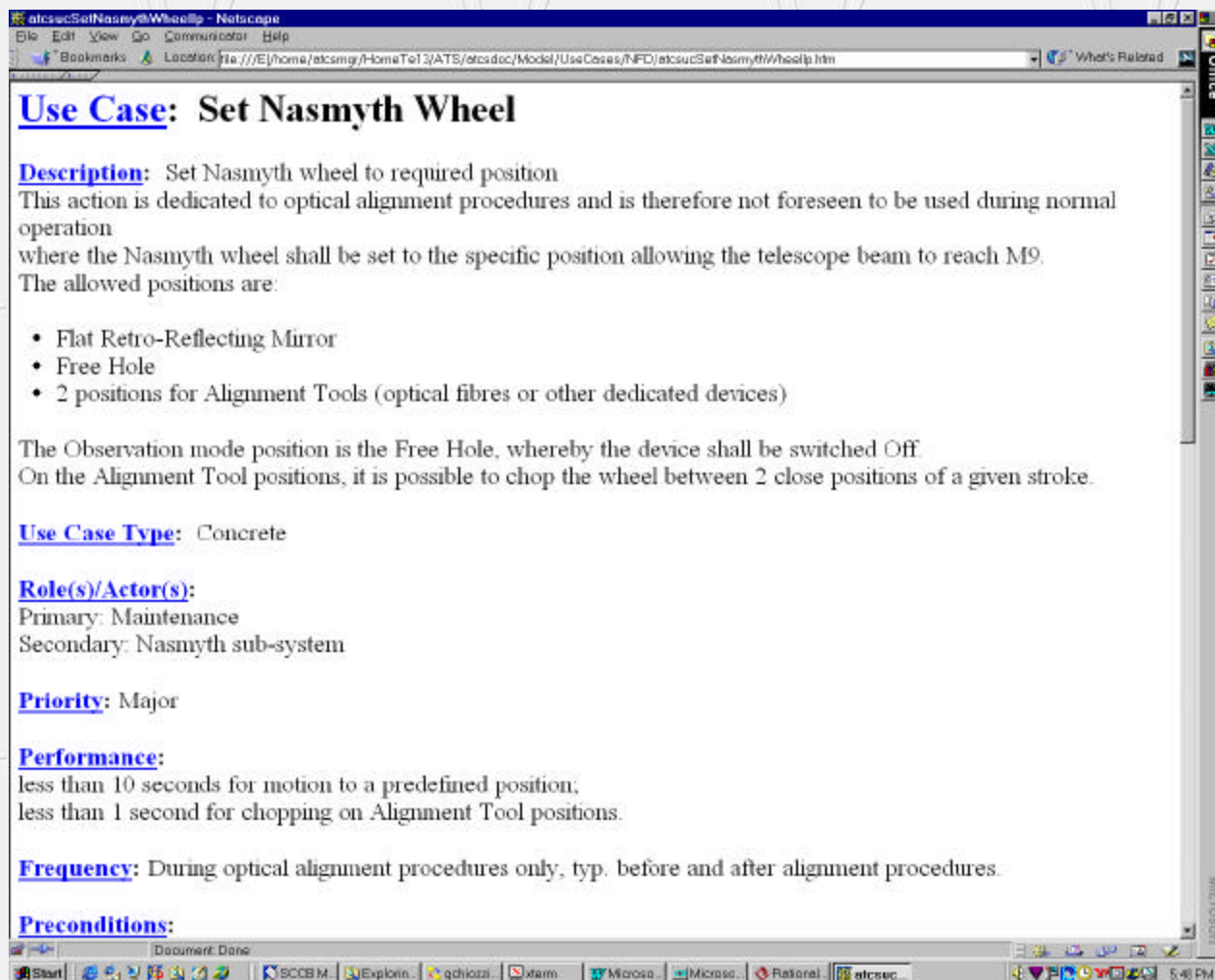
The specification of sequences of actions, including variant sequences and error sequences, that a system, subsystem or class performs that yields an observable result of value to a particular Actor.



Use Case Diagram



Use Case Description



Use Case: Set Nasmyth Wheel

Description: Set Nasmyth wheel to required position
This action is dedicated to optical alignment procedures and is therefore not foreseen to be used during normal operation where the Nasmyth wheel shall be set to the specific position allowing the telescope beam to reach M9. The allowed positions are:

- Flat Retro-Reflecting Mirror
- Free Hole
- 2 positions for Alignment Tools (optical fibres or other dedicated devices)

The Observation mode position is the Free Hole, whereby the device shall be switched Off.
On the Alignment Tool positions, it is possible to chop the wheel between 2 close positions of a given stroke.

Use Case Type: Concrete

Role(s)/Actor(s):
Primary: Maintenance
Secondary: Nasmyth sub-system

Priority: Major

Performance:
less than 10 seconds for motion to a predefined position,
less than 1 second for chopping on Alignment Tool positions.

Frequency: During optical alignment procedures only, typ. before and after alignment procedures.

Preconditions:

Use Case Description (cont)

Preconditions:

- Nasmyth Wheel is ONLINE

Basic Course: Motion to a predefined position

- Send command **SETPATH <FREE|RETRO|TOOL1|TOOL2>** to set the Nasmyth wheel to given predefined position
Exception Course: Command failed
- ATCS sends command to Nasmyth sub-system
Interface: [IfNasmythWheel Motor](#)
Exception Course: Command failed
- ATCS returns OK reply

Alternate Course: Chopping on Alignment Tool position

- Send command to chop the Nasmyth wheel of a given stroke
Exception Course: Command failed
- ATCS sends command to Nasmyth sub-system
Interface: [IfNasmythWheel Motor](#)
Exception Course: Chopping stroke >= 30mm
Exception Course: Command failed
- ATCS returns OK reply

Exception Course: Command failed

- ATCS returns ERROR reply
Postcondition: Nasmyth wheel not moved

Exception Course: Chopping stroke >= 30mm

Basic course

Single step

Error conditions

Design specific information

Requirements Specification

Understand basic requirements

- Use Case Model
- Actors
- Glossary and overall system description
- System Context Diagram
- General requirements
- Risk assessment

Inception - Preliminary Design

Identify system scope and basic architecture

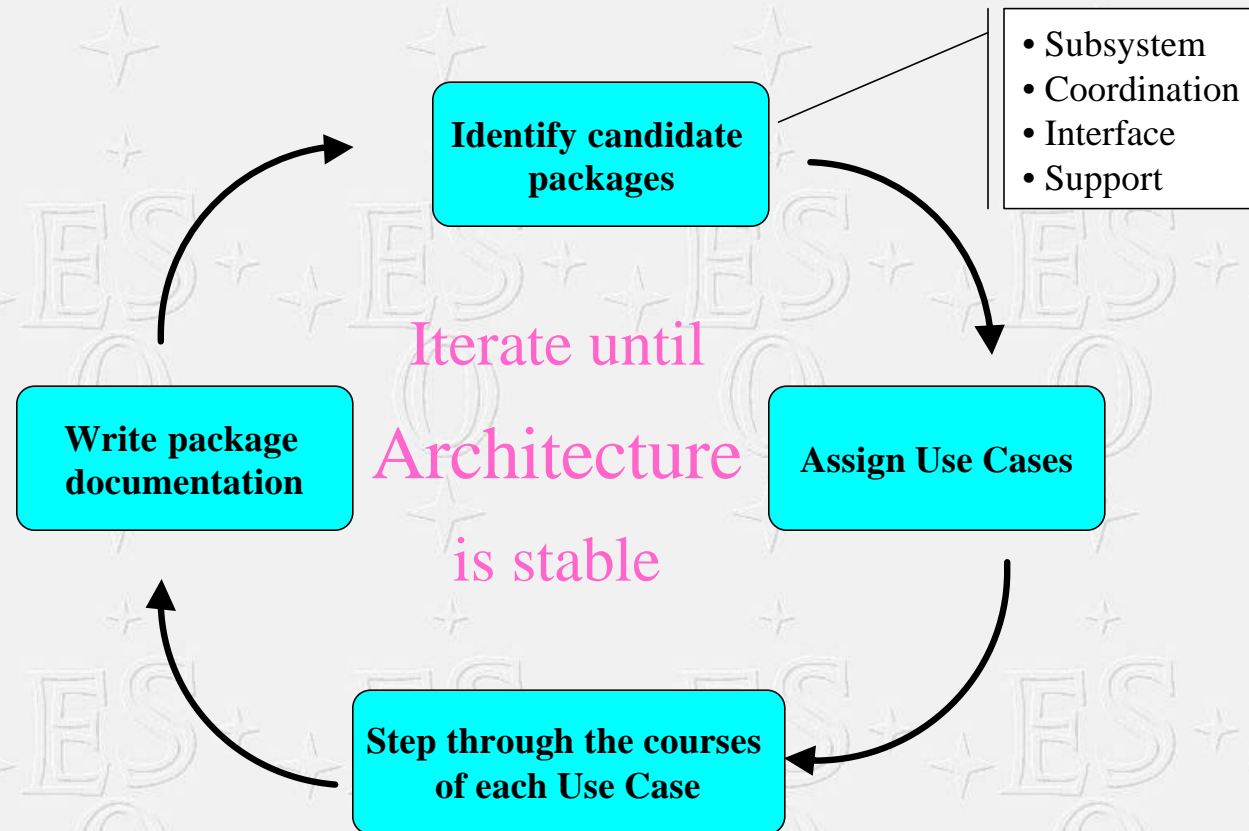
- Packages
- Interfaces with Actors
- Deployment Diagram and Process view
- Performance analysis
- Design of critical Use Cases
- Planning

Identify logical responsibility for Use Cases.

Every step in a Use Case with Actor interaction is an interface.

Assign priorities to Use Cases.
Assign Use Cases to iterations.

Identify Packages



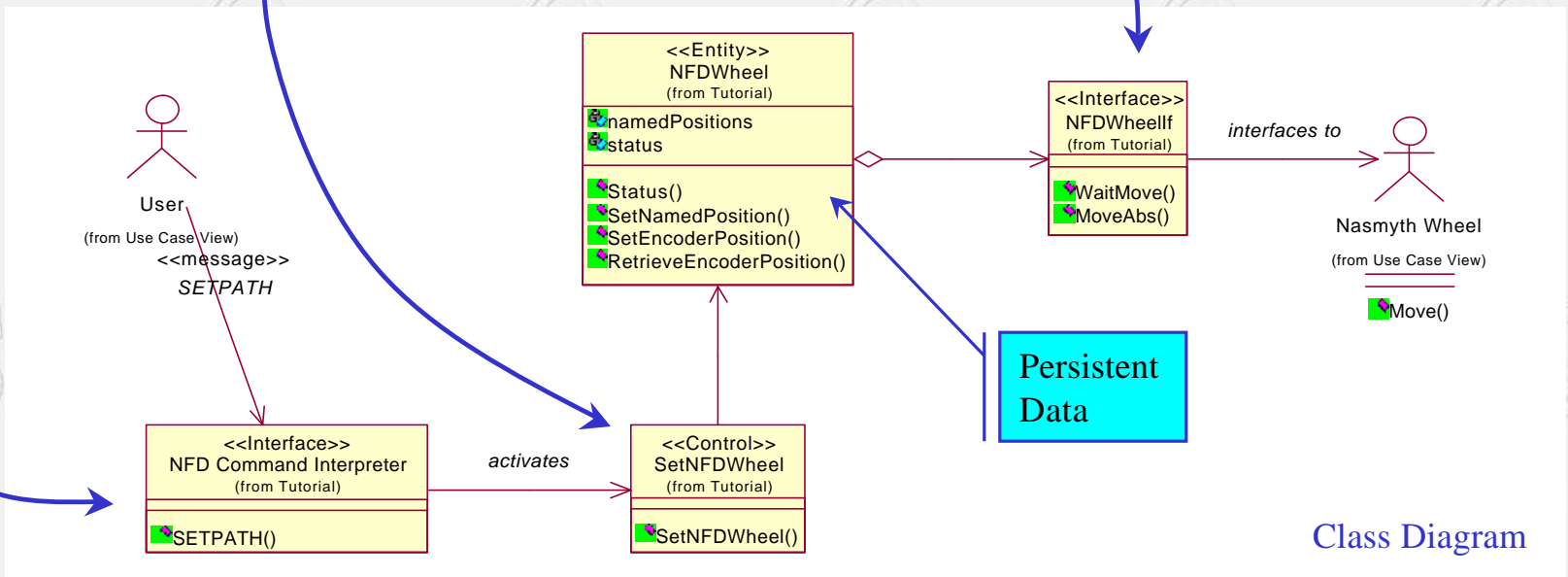
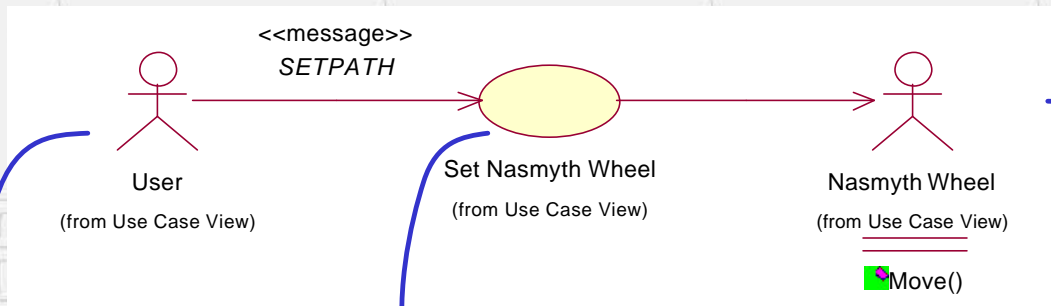
Elaboration - Detailed Design

Elaborate robust and resilient architecture

- Update and detail all deliverables of previous phases
- Project plan
- Executable architecture
- Prototypes for feasibility and risk analysis

Identify Package Architecture

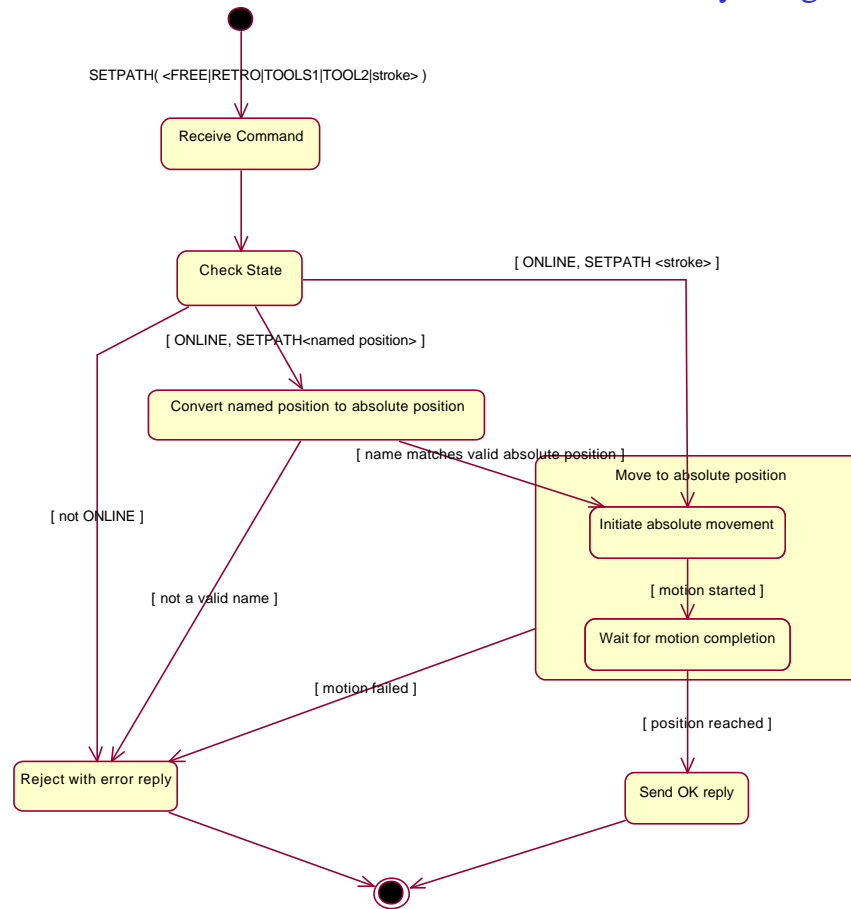
Use Case diagram



Class Diagram

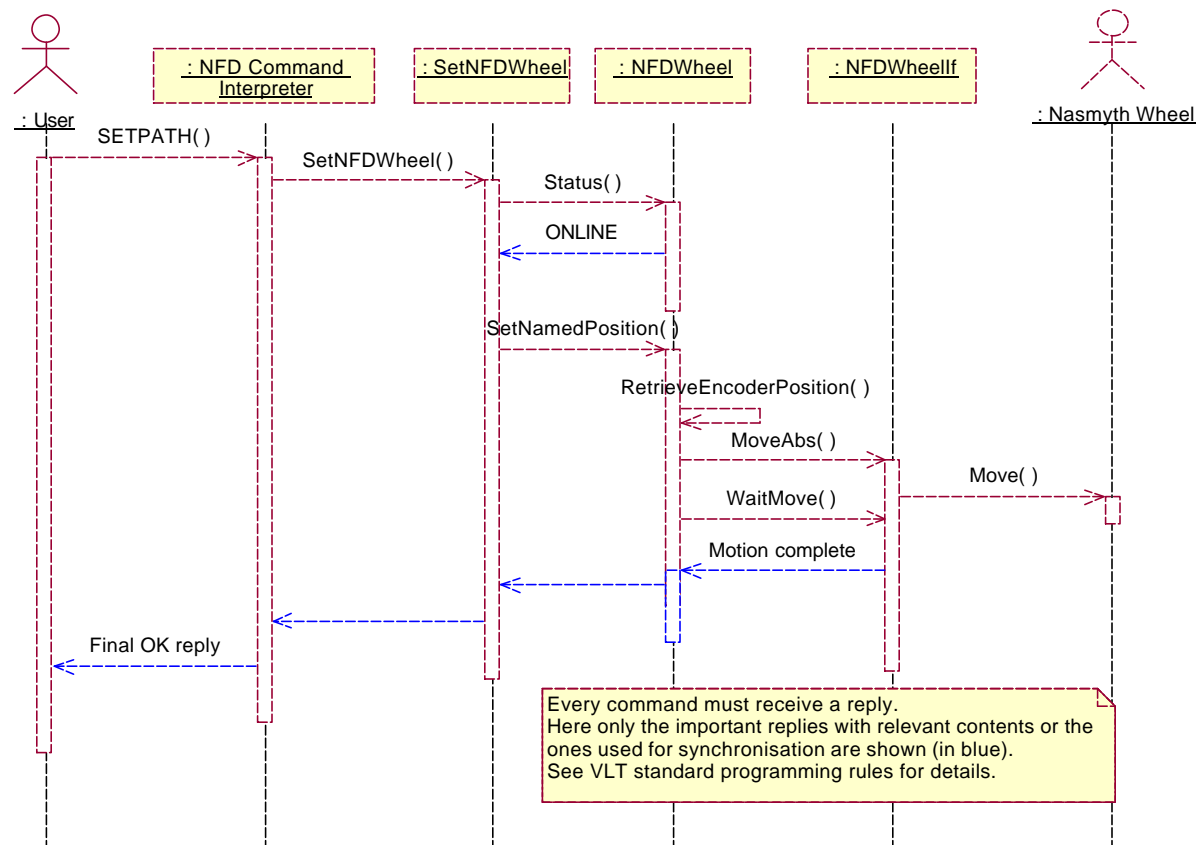
Detailed design of Use Cases

Activity Diagram



Detailed design of Use Cases (cont)

Sequence Diagram



Construction - Implementation

Attain initial operation capability

- Update and detail all deliverables at FDR
- Executable system:
 - Packages are assigned to SW developers
 - Packages are implemented and tested independently
 - Package implementation is implementation of the Use Cases assigned to them for the iteration
 - System integration and testing takes place regularly and marks iterations.
- Draft user and support documentation


Test Cases are derived directly from Use Cases

Transition - Deployment

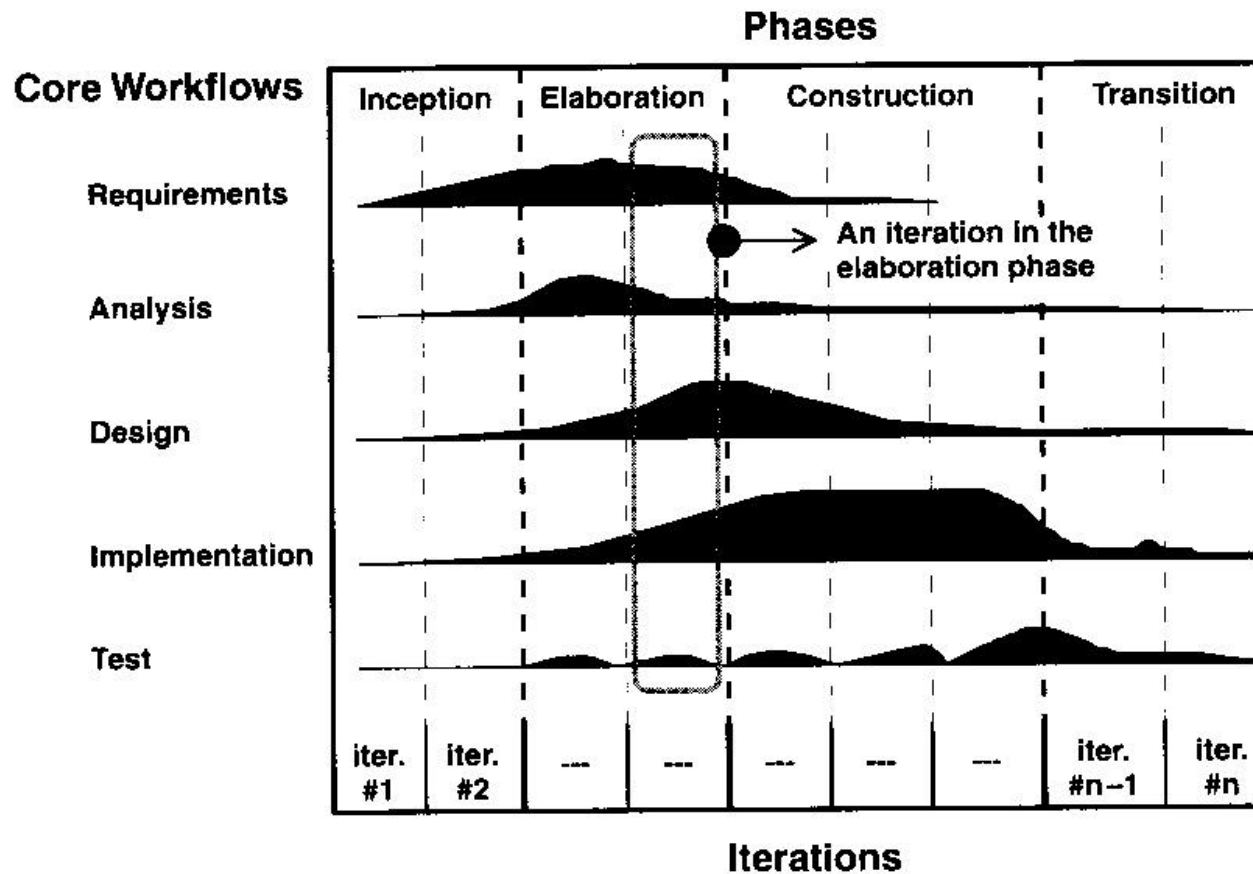
Make system ready for final users

- Update and detail all deliverables issued for the previous phase
- Final system
- Acceptance test procedure reports
- Final user and maintenance documentation

Acceptance Test Cases
are derived directly from
Use Cases



The Unified Software Development Process



Tools

- Requirements management:
 - DOORS (NRAO proposal):
<http://www.qss.co.uk/>
- UML
 - Rational Rose (ESO proposal, vs. GDPro)
<http://www.rational.com>
- WEB Editing/printable documents
 - MSWord (published as PDF)
- Code documentation
 - JavaDoc, DOC++

To know more

For a more detailed list:

<http://www.eso.org/projects/vlt/sw-dev/oowg-forum>

To contact me: email gchiozzi@eso.org

The UML User Guide

The UML Reference Manual

The Unified Software Development Process

G.Booch, J.Rumbaugh, I.Jacobson - Addison Wesley

Real-Time Uml,

Bruce Powel Douglass - Addison Wesley

Applying Use Cases

G.Schneider, J.P.Winters, I.Jacobson - Addison-Wesley

ESO ATCS Project

- ATCS Online Documentation:
 - <http://www.eso.org/~gchiozzi/ATS/atcsdoc>
(Uid: atcsdoc, Pwd: newgenuml)
- Technical Report on Analysis and Design with UML for the Auxiliary Telescope Control System (VLT-TRE-ESO-15151-1917)
 - <http://www.eso.org/~gchiozzi/ATS/atcsdoc/ArchiveDocuments/VLT-TRE-ESO-15151-1917>
- ICALEPCS'99 paper
 - http://www.eso.org/~gchiozzi/MyDocs/Icalepcs99/McI_p29PS.zip

How to learn?

Training on the job

- Small project (0.5 FTE)
- Small development team (3,4 people), possibly including one with good OO knowledge
- Book reading and weekly common discussion
- Distribute the trained people in the other projects or integrate new members in the team

Conclusions: Methodology Corner Stones

- The Unified SW Development Process
is meant to be adapted to project needs and can be kept light
- The Use Cases
drive the process and allow tracing Requirement in all project phases
- The UML Modeling Language
is THE standard visual language
- Standard milestones and deliverables
allow smooth integration with non-SW groups

Conclusions

???

Audience comments and feedback

???