WHT NAOMI project, proposed

Software Project Management Plan

wht-naomi-104

spmplanv3.doc

A. B. Gentles, R.M.Myers, A.J.Longmore

$Revision: 2.0 $

$Id: spmplan.rtf,v 1.2 1996/09/10 08:30:42 abg Exp $

1. <u>Introduction</u>

This Management Plan for the software development in the NAOMI Project recognises the nature of the collaboration between NAOMI and ELECTRA, the current status of ELECTRA as an already mature project and the need to optimise the transfer of technology and work product between ELECTRA and NAOMI.

The plan proposes a combination of ÒEvolutionary DeliveryÓ concepts (relevant and necessary because of the imminent availability of an ELECTRA prototype system) and tools from a ÒwaterfallÓ type of approach. This provides a process of software design and development which recognises the unusual starting point for implementation of a software product, i.e. the fairly imminent delivery of a separately developed prototype. The result is a draft plan to manage the software production process for the various phases of the NAOMI project.

At certain stages of the software production, releases of software are defined to be of ‘production quality’. In this document the term means meeting requirements for software standards as specified for the NAOMI project, including commenting, testing and documentation, appropriate for the type of release.

1.1 Product summary

The NAOMI project and the ELECTRA project have similar aims. Both intend to build adaptive optical control systems to operate on the same telescope and many of the critical hardware and software components will be functionally identical.

They also have some important differences. The ELECTRA system is an experimental, research oriented system *built by its end users* to investigate the potential for AO correction of the multi-segmented ELECTRA mirror and to further investigate new and challenging architectures and approaches to AO and AO control systems. The operational lifetime of any one version of the ELECTRA system will be a single telescope run; it can be expected that some components of the software will change between each visit to the telescope.

The NAOMI system, on the other hand is aimed at providing a common user facility for everyday operational use as a permanent integrated telescope facility. It will be built by adaptive optics experts but operated and maintained by people without specialist AO knowledge. The operational lifetime of each product of the NAOMI system will be bounded only by the availability of upgrade versions and final decommissioning or replacement which could be up to 10 years away. Thus the challenge of the NAOMI project is to deliver a system which can be a component part of the observing system and be robust and easy to operate whilst also providing a world class AO performance.

1.2 The ELECTRA re-use policy

Cost-effective production of NAOMI is most likely to result from a Òmaximal Ó or Òmost appropriateÓ re-use of ELECTRA software. The concept is based on the similarity of the ELECTRA and NAOMI systems, now that the ELECTRA deformable mirror has been chosen as the baseline active element in the AO system. The task of replicating the DM control software was thought to be wasted for NAOMI if a working prototype was available as a central result of the ELECTRA project. If components of the ELECTRA system can be proven to meet the requirements for the NAOMI system and their detailed structure and design are sufficiently well documented then they should be used as components of the production NAOMI system. Accordingly, a goal is to retain a high level of compatibility in the software architectures between the two systems, within the constraints of the system requirements at the various stages in the software development process.

1.3 The relationship of the software sub-project within the overall NAOMI project structure.

The NAOMI project is a multi disciplinary mix of electronics optics, mechanics and software. It has been split into workscopes by collecting functions, with the complete software development as a single workscope. The project control documentation structure is shown in Figure 1.

The operational requirements have been derived from a conceptual design, reviewed in April 1995. The work package descriptions for the optical chassis and WFS provide a high-level, although detailed, description of the opto-mechanical parts of the system. The software work package description (which appears in Fig. 1 as the Software URD for reasons given below) will have the following contents:

?? Brief summary of the overall purpose of the complete system

?? Description of the capabilities of the system, i.e. the required functionality of the delivered software

?? Description of the constraints on developing the system, e.g. hardware which must be interfaced to, restrictions on choice of implementations.

In the ESA PSS-05 model, this is extremely close to several of the functions of the User Requirements Document which is produced as the result of the software requirements definition phase. This proposal itself fulfils the function of the SPMP document in the PSS-05 method.

A separate OCD (operational concept description) document to cover the whole system will be referred to in the work package description. This can be to confirm the completeness and appropriateness of the system design solution embodied in the three work package descriptions. This will allow feedback to update the FPRD and work package descriptions as

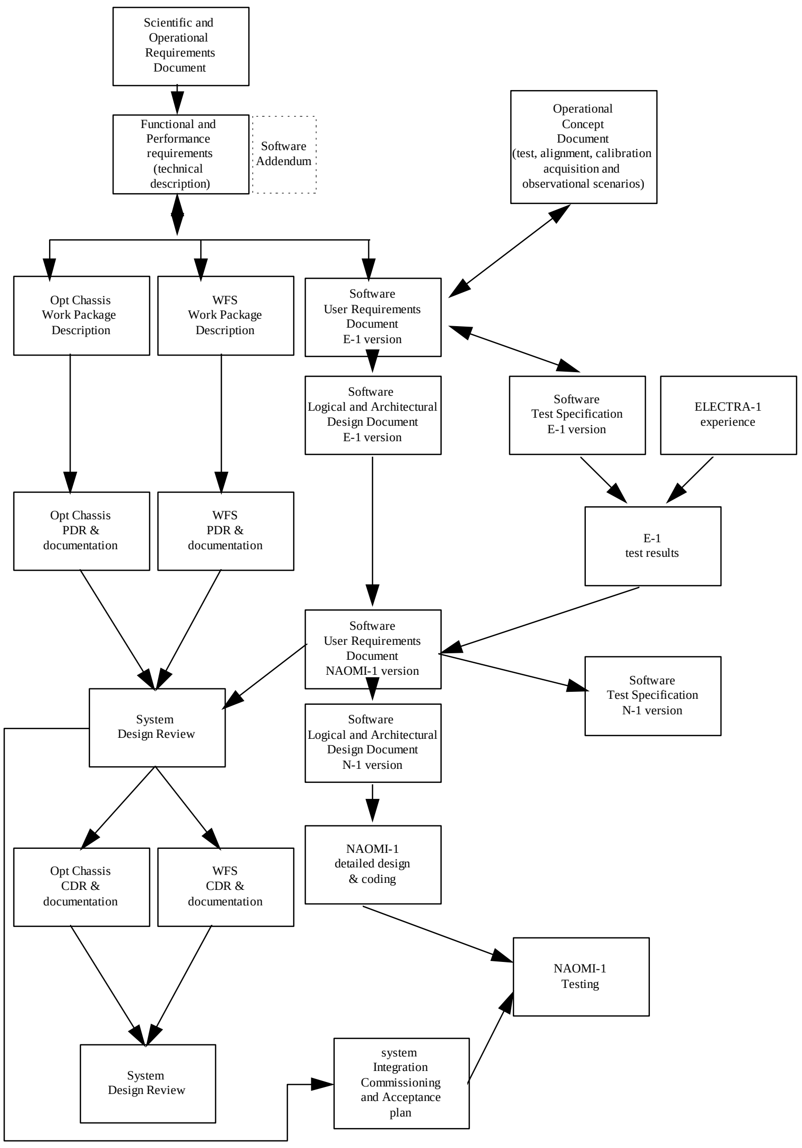

Scientific and Operational Requirements Document
Functional and Performance requirements (technical description)
Software Addendum
Operational Concept Document (test, alignment, calibration acquisition and observational scenarios)
Opt Chassis Work Package Description
WFS Work Package Description
Software User Requirements Document E-1 version
Software Logical and Architectural Design Document E-1 version
Software Test Specification E-1 version
ELECTRA-1 experience
Opt Chassis PDR & documentation
WFS PDR & documentation
E-1 test results
Software User Requirements Document NAOMI-1 version
Software Test Specification N-1 version
System Design Review
Software Logical and Architectural Design Document N-1 version
NAOMI-1 detailed design & coding
Opt Chassis CDR & documentation
WFS CDR & documentation
NAOMI-1 Testing
System Design Review
system Integration Commissioning and Acceptance plan

Figure 1 Project technical document structure

necessary.

The ESA model has several phases:

?? User Requirements	UR
?? Software Requirements	SR
?? Architectural design	AD
?? Detailed design and Coding	DD

?? Transfer TR

?? Operations and Maintenance. OM

The process is aimed at producing software with enough documentation such that it can be proven to be complete and maintainable in the OM phase.

This project is small enough that strict adherence to the above divisions is unnecessary Therefore it is propose that the Software Requirements and Architectural design phases be combined into a Software Design phase, which produces a Logical and Architectural design document, or SLADD. This should outline both the division of functionality into separate blocks and the assignment of those blocks to hardware and processors, at a high level.

1.4 Approaches to Software Management

This is a tutorial section which can be viewed by unhiding the Tutorial style.

2. Purpose and functions of this proposal

This document sets out a route for the production of the final software product which will be delivered to the ING. The overall task is divided into stages, each of which are designed to ensure that the final product is built in a maintainable fashion and does all that is needed without introducing unnecessary functions. It describes the stages to be followed for each product. It defines the relationships between the products and the feedback used to defined the functionality of each successive product. It should allow each of the stages to be sequenced properly, including identifying where parallel development is and is not appropriate, whilst ensuring that the output of each stage is well enough defined to be of use in later stages of the project.

It is expected that there will be updates to the document as prototype releases become better defined and specific quality control tools are chosen and used.

3. Overall Software Priorities and Risk Management

3.1 Priority List

Factors affecting decisions on software standards and allocation of resources include required performance, functionality, timescale, cost guidelines, risk and maintenance capabilities. A guideline for the relative importance of these is (most important first):

1. Meeting budget
2. Meeting Clause 1 in the SciOpReq document (reference 6) - the high Strehl requirement.
3. Reliability in operation - implies good architecture, good coding standards.
4. Ease of Maintenance
5. Meeting schedule
6. Minimizing risk.
7. Meeting all required functions

THIS IS A GUIDELINE ONLY. It is certain that the guidelines will need to be violated on many occasions. Common sense should apply at the same level as the priorities! The purpose they should serve is to ensure that when the guidelines are not followed this is done consciously. Therefore the working practice should be that if a step is proposed which does not follow the guideline, this should be brought to the attention of the local manager who should in turn notify the Project Manager if it could be an important issue.

3.2 Risk Management

The risk management strategy will be to allow some reasonable level of risk without further detailed analysis in areas where the project has relevant experience. A detailed analysis should be carried out to reduce risk or find alternative routes if the project has little or no relevant experience in the area of concern. The working practice should be that if a step is proposed which entail more than average risk, this should be brought to the attention of the local manager who should in turn notify the Project Manager if it could be an important issue.

4. Definitions of Documents

4.1 The OCD

The OCD is an operational concept description. It contains a list of mechanisms and software functions , such as Òthe WFS pick-off slideÓ and Òclose DM-WFS loopÓ . It gives descriptions of the procedures used to operate the instrument. It should be used during design and development to confirm that the hardware and software functionality are compatible, complete and necessary. After the design process these procedures should be used as the basis for writing the *how to do* sections of the Users Manual for the instrument.

It is proposed that a simple simulation of the system states be built which contains a record of the current internal system states for hardware and software and models the interactions between them. This can be exercised with the expected alignment and observational scenarios. This prototype system shall be able to record sequences such that the alignment and calibration scenarios can be generated and recorded and known to be both complete and possible. This should allow us to exercise the functionality defined in the URD.

4.2 The (software) URD

The URD should detail the required functions of a single software product which are described at higher level in the FPRD (or Technical Description document). The URD should also describe the restrictions on implementation of that product. It should not go into detail on describing how to use the system, but refer to the OCD for this function.

The software component in the current FPRD is not complete. An extension to the FPRD will be written after the E-1 commissioning has been evaluated.

4.3 The SLADD

The Software Logical and Architectural Description Document should break the monolithic system described in the URD into logical blocks, each with a single identifiable function. For this logical model, the requirements may be grouped as follows:

?? Functional

?? Performance

?? Interface

?? Operational

?? Resource

?? Quality

    ?? Verification

    ?? Acceptance testing

    ?? Documentation

?? Security / interlocks

?? Portability

?? Safety requirements

The SLADD also records the architectural design. It records which functions are run on which processor and how the processors and interface hardware are connected, and defines any extra software functions or modules needed to interface the component parts.

The choice of processor types and associated environment, or confirmation of an assumed choice, should be made at the completion of the first NAOMI SLADD.

A modified SLADD will be produced for each cycle round the evolutionary development loop. The amount of work involved will vary depending on how much of the existing system has to be reworked following the testing stages of the previous cycle, and how much extra functionality is being added.

?? The SLADD should be signed off by the Project Manager at a meeting attended by appropriate representatives from each group of the collaboration plus the Project Scientist and Systems Engineer.

5. <u>Delivery stages or products</u>

5.1 Proposed Software Stages

The following overall product sequence is proposed.

Proto-1	Synonymous with the first OCD. A simulation of the high-level system states, with which to test observational scenarios. This enables confirmation that the proposed FPRD and subsequent lower levels of implementation can in principle meet the top level operational requirements. It also confirms the need for each of the functions shown in the software URD.
Proto-2	Prototypes the proposed DRAMA based architecture, with GUI, sequencer core, and links to telescope and interlock processes and basic simulations of RTCS and MECH blocks running on the appropriate hardware platforms.
Proto-3	This adds an interpreted sequencer support to P-2 to expose any unexpected architecture constraints. This may done with ELECTRA.
ELECTRA	This is being developed in parallel. It will provide significant confirmation of the required functionality for NAOMI as well as a number of portable modules.
NAOMI-A	Like the Proto-1 and Proto-2 above this will be a highly stubbed system but will include production quality ports of core ELECTRA functions into the chosen NAOMI architecture. The aim is to prove the real-time and control capabilities and limitations of the architecture. Only the basic functions which are central to the system operation need be included. AO servo-loop functionality *may* be limited to ÒMartini-modeÓ only. NAOMI-A will have to pass acceptance tests before moving on to the NAOMI-1 product. Functions in the NAOMI-1 set may be included to aid debugging but will not be subject to acceptance tests. Control will be via the command line interface to provide an automation facility for testing. This will be developed with the NAOMI RTCS hardware and the agreed high level control processing platforms. The E-1 Optics and WFS will be used to verify the real-time performance.
NAOMI-B	This stage may be merged with the final integration stage. The working Real-time core developed in NAOMI-A will be integrated with the actual WFS hardware. The WFS EPICS mechanism control will be added. Further diagnostic and control functions will be added. These will include

3D visualisation, WFS calibration, WFS & DM offset & set-up functions. The visualisation GUI will be added at this point and possibly a stubbed opto-mechanical chassis GUI. Basic support for science target selection should be included here, although there is no requirement for a production level Telescope/observing system interface.

This system should be able to be operated on the telescope for sky tests if required.

NAOMI-1 This system aims to deliver the baseline NAOMI system for commissioning.

The major difference is support for the complete NAOMI hardware by adding the Opto-Mechanical Chassis mechanism control functions. It should add the capabilities of target source selection and full control of the telescope from the AO control task. Interlocks for various routine operations should be added to improve the robustness of the system. The DM-TCS focus loop should be implemented if possible. The precise functionality will depend on the lessons learnt from N-A and N-B.

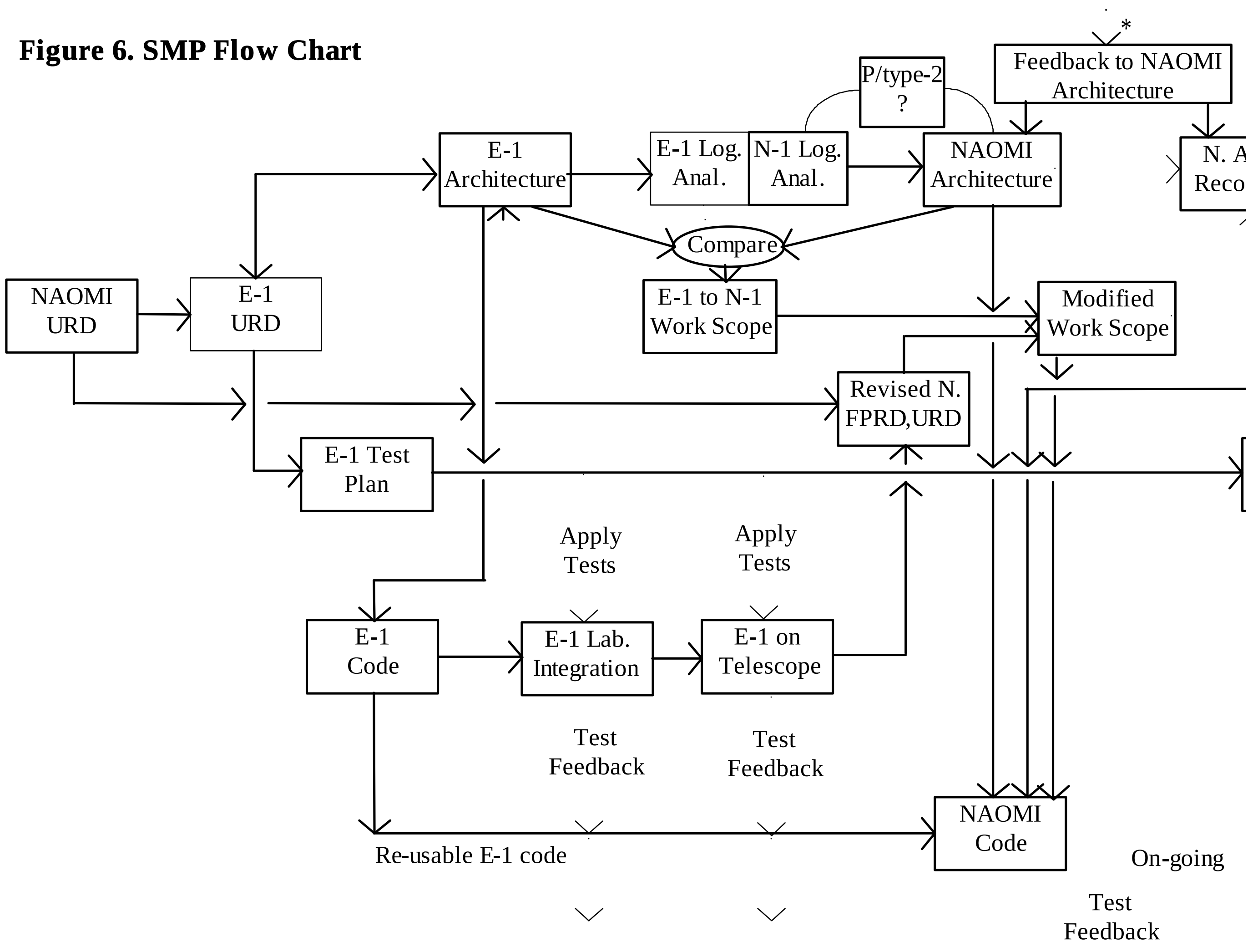
NAOMI-2 A 'NAOMI-2' concept introducing requirements listed under N-2 in the URD will only be realised if there is a system upgrade. Further development is not currently funded. Functions can be added on a case by case basis as funds allow.

5.2 Iteration between products

Each of the prototypes P-1, P-2, P-3 may be developed independently but will not be used in production code. The NAOMI-x systems are all production level systems. Pragmatic use of the waterfall model stages are intended for each evolution cycle. The goal is to maintain the integrity of each cycle, thereby controlling the integrity of the final product. The following list describes the process. An overview may be obtained by examination of Figure 6. Detailed explanations of the documentation loops are given in Appendix A.

1. Start with extant version of URD, OCD, SLADD and test plan.
2. Code and unit test modules, both with simulators and hardware.
3. Integrate and test sub-systems
4. Upgrade / refine the URD to include the appropriate function set.
5. Use this new version of the URD to upgrade/refine the acceptance test schedule.
6. Modify the high level design (SLADD) as needed
7. Identify those functions/modules of the previous product which are satisfactory, which need amendment, which are superfluous and which are missing.
8. Modify the Detailed design as needed.
9. Code and unit test new and modified modules, both with simulators and hardware.
10. Integrate and test sub-systems
11. Integrate complete system and run acceptance tests.
12. Monitor performance at telescope (if feasible) and relate to test performance.

Figure 6. SMP Flow Chart



$Id: spmplan.rtf,v 1.2 1996/09/10 08:30:42 abg Exp $

6. Software Quality Assurance

Factors and procedures affecting software quality which should be taken into account in the Quality Assurance plan include:

?? implications of requirement changes whilst the software is being coded must be followed through properly;

?? implications of architecture changes whilst the software is being coded must be followed through properly;

?? it is much more efficient to catch bugs during design rather than during or after coding;

?? what you cannot measure you cannot judge or improve;

?? unclear requirements definition causes more time to be wasted than almost any other cause;

?? solutions should be clearly separated from requirements;

?? faults in documentation, both for requirements and for design, are probably as influential in causing bugs and inefficiency as are direct coding errors.

?? quality covers design and documentation as well as code.

The QA plan for NAOMI is still being developed. It is proposed initially to adopt the following methodology and guidelines.

A recognised method will be used to assess general level of software quality of key releases and to monitor the product quality. It is proposed that the Fagan Inspection method be used. The basic measurement is the number of defects/bugs found in an one item such as a document / design /piece of test code /module of run-time code. The results of the monitoring will then be used to help determine resource allocations and to identify problem areas.

Self-documenting coding standards will be used for the actual code

Sufficient effort should be put into the documentation and detailed design that it is clear to the programmer what must be coded.

Outside of the real-time loops, in general speed can be sacrifice for cleanness in design. Code should of course still be kept reasonably efficient.

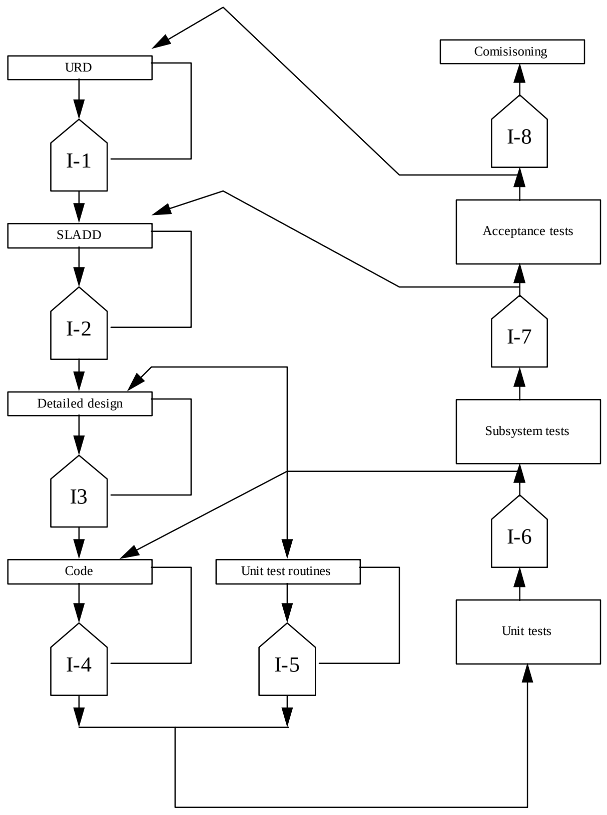
Modules successfully passing all tests should be listed as valid modules and put under strict change control.

Architectural changes made after coding has begun must be recorded and all modules which could possibly be affected must be re-tested and re-validated.

In general key functions will be coded before lower priority functions. No gratuitous functions shall be coded although of course functional requirements can be added as part of the iteration process.

An example inspection flow-chart is given in Figure 5. This defines inspections of the URD, the SLADD, the detailed design and the code, as well as all of the test code & methods. At each stage of first pass design or build, the product of that stage is inspected mostly on a stand-alone evaluation. A list of defects is drawn up which are actioned for correction. The stage is iterated until the product is of acceptable quality. After the first version has been produced in this way, subsequent inspections are done when the product can be evaluated against major new criteria (e.g. test performance, compatibility with a new document). The ideal conclusion of each stage would be 0 defects but it may be more efficient to allow a small number uncorrected. Such ‘threshold’ levels will be established for each product. Further information on the method is given in ‘tutorial’ mode.

Figure 5 Inspection flow chart.



7. Electra-1

7.1 Completing the ELECTRA URD.

This will be achieved by first completing the NAOMI URD. Then the sub-set of the NAOMI URD which will be met by ELECTRA will be agreed with Durham and this sub-set marked up on a separate version of the NAOMI URD.

7.2 E-1 Test Plan

The URD will be used to develop a test plan for E-1. First this will involve using the existing E-1 architecture and the URD together to produce a list of modules and their definitions. Tests to exercise these modules when coded should then be developed.

7.3 Logical Model of E-1

It is proposed to use Teamwork to produce a logical model of the E-1 architecture. This phase can make some use of ideas from the early SRD, but it should be noted that the early SRD was being prepared for an AO system with more functionality than NAOMI, but was never completed.

The proposed physical, or implementation, model of ELECTRA will then checked against the ELECTRA architecture. As the ELECTRA architecture is well developed the logical model will serve mostly as a learning process, a first stage in preparing a SLADD for NAOMI and one reference against which to identify mutual compatibility between ELECTRA and NAOMI software modules. It also provides the opportunity to check the model for completeness.

7.4 Coding and Testing of E-1

E-1 is being developed via E-0, for which some coding has already occurred. The general approach will be to apply the test plans to modules as they are developed, in the lab, more widely at the laboratory integration phases and then confirm them at the telescope.

7.5 ELECTRA-NAOMI Compatibility Evaluation

NAOMI-compatible components of the well-advanced ELECTRA system must be identified. Primarily ‘compatible’ means fully functional within the designated NAOMI architecture including meeting all interface requirements. Secondary levels of compatibility should include meeting a required function and fitting NAOMI coding and software documentation standards.

The compatibility evaluation will be done by progressing the NAOMI design after the logical model of the ELECTRA architecture until the NAOMI SLADD is well advanced. Limited iteration of the NAOMI architecture will be carried out until an *acceptable* design is reached. At this stage, the software modules which are compatible between the two architectures could be identified immediately using the logical models and further confirmed when results from E-1 tests when available. The evaluation period immediately after E-1 commissioning is likely to be the last major stage at which new input will be considered for NAOMI architecture improvements (see section 7.6). ??

Dummy modules will be written based on the working ELECTRA modules, with all their interfaces but with external functions stubbed where necessary. These dummy modules will be integrated with NAOMI supervisory software to form part of the sequence of prototypes as appropriate.

The workscope definition for conversion of incompatible modules, writing new modules and integration of compatible ones should be completed at this point.

7.6 Check E-1 URD vs. actual E-1 experience

While ELECTRA-1 is being commissioned on the WHT, the usage of the various functions in the ELECTRA system will be recorded. Subsequently their level of use and performance will be evaluated to set priorities for porting, re-works and new modules.

This information should also be fed back via changes to the URD to create suggestions for improving the NAOMI design and the design to date updated as necessary.

Video camera footage of E-1 operation should be considered as one tool which may be used in determining priorities on functions for NAOMI.

8. <u>NAOMI Prototype stages</u>

8.1 P-1 Update Prototype State machine for OCD

The initial, summary definition of this stage is given in section 1.1. Expanded details will be inserted here as appropriate.

8.2 P-2 Build Prototype for control architecture

The initial, summary definition of this stage is given in section 1.1. Expanded details will be inserted here as appropriate.

8.3 P-3 Add sequence interpreter to P-2 and investigate possible modes of use.

The initial, summary definition of this stage is given in section 1.1. Expanded details will be inserted here as appropriate.

8.4 Write/update functional tests based on E-1 tests and N-1 URD

These functional tests should be the basis of acceptance testing of the delivered product. It is important to write these directly after agreeing the URD in order to establish methods for testing and any software requirements needed to test the software. The tests should be written before the code is written and then revised directly before the tests to follow changes in design or structure and to allow the precise testing mechanism to be found.

It is important to devise a method of generating test input data and sequencing the software under test automatically without manual interaction. This can allow the tests to be rerun to spot regression failures and also not to waste staff effort watching tests happen.

Because of the advanced stage of ELECTRA, code has already been written, so in this case if practical, people not directly involved in the coding will devise the tests. These will be used to test the implemented software, both as thought experiments before coding via walkthroughs and to test the actual coded product.

8.5 Logical model of NAOMI

The primary aim is to identify a hardware architecture which conforms to the URD needs and software-specific requirements identified during the Logical Analysis - URD interactions. This interactive process combined with the E-1 analysis will produce the SLADD described in section 4.3.

The goal is to find a path which retains much of E-1 whilst also following the constraints of URD and the SLADD itself as it is developed. Ideally this would be an exercise in confirming that the data model of E-1 still applies and then extending it to cope with the N-1 functions. However there may be some differences in host architectures so they cannot be assumed to be compatible.

The second aim of this exercise is to identify all E-1 modules compatible with NAOMI and to determine reasons for non-compatibility of other E-1 modules. Common modules and methods of command interpretation etc. between E-1 and NAOMI have to be covered as these areas may contain incompatibilities.

Logical modules which are outside the scope of N -1 need not be analysed for the NAOMI product.

Any problems brought up in the ELECTRA telescope run should be factored in at an appropriate time to refine the models - see the evolutionary use of the SLADD described in section 4.3.

8.6 Draft and Confirm Durham and RGO workscopes

After the NAOMI Logical Model is complete there is enough information to define both the software work scopes in detail.

8.7 Define dummy module operation.

Each of the teams at Durham and RGO will have to be able to operate fairly independently but maintain good communication, for the development process to be efficient. Therefore a set of dummy modules will be produced, before the actual production code. These will have the same function set as the final module but allow integration in two stages.

Stage 1 is simply to write a function library with the functions having an agreed calling convention but the functions do nothing. This allows the modules above to be developed independently.

Stage 2 is to implement a remote dummy task using the communications layer intended for the real module.

If the final NAOMI architecture is simply an extension of E-1, further forms of stubbing (e.g. sequencer-based) would be readily available.

A client stub will allow each of the functional commands to be issued with the appropriate format and the responses monitored. A test facility should be able to use several stubbed modules to exercise the server program.
A server stub will provide a dummy set of null functions which can be called with the appropriate interface layer and can send an appropriate set of status returns, or generate test data for call-back routines or monitoring processes.

Both of these stubs should allow each end of the comms link to be exercised and the interface to this kind of comms link to be verified before too much coding has been done. During testing, this will also allow selective items to be stubbed off to test the comms layer separately from the application layer.

Dummy modules need have only a summary description of their function and design for review purposes.

8.8 Detailed design and Coding stages

The following phases are expected: they apply to both sites.

8.8.1 Code dummy modules.

The dummy modules as defined above in section 8.7 should be coded before the production system modules. This will allow knock-ons from the comms layer to surface early and allow separation of the comms layer from the application layer. The dummy modules will also prove....

Prototype 3 (onwards or only ??): Port and test E-1 compatible modules to NAOMI Architecture. This will provide production coded modules quickly from ELECTRA code. These will also identify any more incompatibilities. Modules found incompatible at this stage should join those which need detailed design and fresh coding.

8.8.2 Detailed design of N-1 modules (Durham + RGO)

A full description of the design, function and interfaces for each modules should be written.

The detailed design should be walked through and approved, before coding begins. This will allow the detailed design documentation to be confirmed to allow unit tests to be developed in isolation from the actual code. However, it is likely that there will be exceptions to this rule; these should be clearly identified and agreed with the software manager or equivalent.

8.8.3 Coding of N-1 modules (Durham + RGO)

Each developer is free to code in their own house style as long as the agreed requirements for software standards of NAOMI are adhered to (as described in references 2 and 3). The aim is to provide code which is effective but simple enough for someone else to maintain. In practice this means following chapter 18 rather than chapters 11-17 of Ranade and Nash (Reference 5). Code/test/diagnose/debug loops should be carried out in a way that meets the quality control and metrics requirements described in section 6.

8.8.4 Development of Opto-Mechanical control system

This is to be implemented via an EPICS database. This should be developed and coded in such a way that the following database configurations are available:

1. The real version: operated the hardware, controlled from channel access via defined interface.
2. Remote test version: hardware stubbed off. Implements defined control interface via channel access. Can be loaded into a VME rack with only the processor card & memory.

In addition, a separate module for engineering control is needed. The EPICS DM tools is acceptable.

A prototype system should be developed first, utilising the EPICS to DRAMA interface to test the operation of this part of the system.

8.9 Verification, integration and Commissioning of N-1

8.9.1 Unit testing of N-1 modules

A suite of tests should be devised for each module which can verify its operation. Ideally tests should be designed to ensure *condition coverage* of the source code (i.e. every condition/branch within the code should be tested. Realistically a lesser goal of (say) 80% statement coverage and 75% branch coverage may need to be adopted. Where practical the tests should be devised by someone other than the author of the code.

The Unit tests may be carried out by the code developers.

Where practical, tests should be automated. Logs kept of all test runs. The rate of improvement of test passes can be a good metric for how long the development of a particular module may take.

8.9.2 Integration and testing of N-1 modules

Each sub-system which is made up of a number of communicating modules should be able to be tested as a sub-system. Test scripts or programs should be created by ?? to exercise the sub-systems.

The ability to send all defined command types and the complete command set to a sub-system should be demonstrated. Correct operation of the status return mechanism should be demonstrated.

It is expected that the OCD will have a simulator which can generate system states. This should be used to confirm that correct sequencing of those states which correspond to the sub-system is possible. If the OCD model needs changing at this point, the changes should be approved and documented.

Any real-time requirements on speed or bandwidth should be tested and the performance verified.

It is important to carry out these tests without too much real hardware and with simulated data. This allows the data inputs to the processes to be known and thus the results can be compared with ‡ priori tests.

8.9.3 Sub-System tests with hardware

Once the various modules can be shown to be robust in isolation, they should be tested with as much real hardware as possible, starting from the bottom up. Reliable and robust operation of functional commands should be demonstrated. Robustness of the system to such events as bad pixels in a camera, noise spikes, missing or bad input data should be demonstrated.

At this point, the opto-mechanical control system should be partly integrated with the RTCS and overall control, in order to confirm robust data transfer between the appropriate elements (e.g. TCS slow pointing corrections, Autoguider outputs, feedback from WFS to WFS XY stage etc.) The Opto-mechanical hardware should not be present at this stage.

8.9.4 Complete system integration

The complete software system should be assembled in one place *without* live opto-mechanical control. The operation of RTCS, central control and mechanism control can thus be exercised and tested as a single unit. Once the operation is predictable and as expected, the hardware can be added and the inter-relationships explored.

Test scripts should be devised to exercise the operation for this stage. These should aim to establish that each of the mechanisms can be moved and that all of the RTCS and sequencer commands are available. Offsets and moving axis characterisation can be done here.

A detailed system integration plan should be created and the progress of the system integration monitored against this plan.

This phase is finished when the system is as completely functional as can be determined in the laboratory, and is ready for formal functional tests. It should be ready for shipping.

8.9.5 Functional tests of complete N-1 system

The aim of these functional tests are to prove that the system is ready for shipment to the telescope. They should demonstrate the capability to meet all the required functions and to be robust in operation. These tests should be run with enough of a range of environmental conditions to simulate operation on the telescope.

These tests will be formal in that the system should be baselined before testing and the tests carried out with that version. During the test run, all failures will be identified and appropriate remedial actions taken.

Reruns of the complete tests, or selected sections, should be done once bugs have been fixed. The testing, including repeated of sub-assembly tests, should be automated enough not to use up personnel resources unnecessarily.

8.9.6 N-1 commissioning

The commissioning for N-1 has several goals:

?? to ensure that the telescope environment effects on the NAOMI system. - noise, vibration temperature, pupil position, alignment, space etc. do not prevent it from meeting specification

?? to integrate the actual science instrument and its software

?? To operate the system with real astronomical aberrations and guide stars, demonstrating that the performance specifications are still met

?? To establish working ranges for guide star brightness and characterise performance in the various operating modes.

?? To confirm the suitability of the planned observing modes and procedures

?? To prove the default optimisation algorithms.

?? To confirm and document the actual operational procedures for mounting and dismounting the instrument and for operation in the test focal station.

?? To train the support Astronomer and AO specialist in operational use.

Very little time will be available during commissioning to do any software development so the aim of the software production process should be to provide a software product which is as robust as possible.

9. NAOMI-1 Software Design Reviews

Software Design Reviews will be closely tied to Prototypes and staged releases. These will not necessarily correspond one-to-one with PDR/CDR concepts. Potential Reviews (to be confirmed when definitions of the prototype stages have been agreed) are proposed at the following stages:

?? after completion of the N-1 URD and SLADD; passing this stage should result in approval to start purchasing NAOMI computing hardware and will define the remaining software work scopes for each of the collaborators.

?? after P-2

?? as part of the E-1 telescope commissioning evaluation

?? after P-3

?? after NAOMI-A

Except for the first of these, which will be paper-based only, the review will combine on-line examination of prototype or full release models with appropriate documentation.

Documentation for the first review should include a summary of the current status of ELECTRA, the N-1 URD, the OCD, the N-1 SLADD, an evaluation of these leading to a proposed work scope defining all expected E-1 to NAOMI software work, for each of the collaborating parties, ... ??

For all reviews after the first, the review documentation should include the appropriate architecture diagram, relevant data flow diagrams, a list of all modules incorporated, an indication of whether they are full or dummy modules, statistics on quality control results and formal tests carried out on the version on its release (presumably before the review!), a summary of quality statistics from previous reviews (if any), list of architecture modifications from previous review, ... ??,

For the NAOMI-A release review the documentation should also include counts of the fraction of documentation lines versus code lines within each module, a form against which further test results can be marked during the review, ...??

10. <u>Detailed design documents</u>

11. <u>Functional test procedures</u>

12. <u>Sub-System test procedures</u>

13. <u>Unit testing</u>

14. <u>Milestone list.</u>

1. Complete URD
2. Complete OCD
3. Update and Sign off OCD & URD
4. functional tests agreed.
5. analysis complete: URD, SLADD
6. (N-1) Agree on compatible module list with ELECTRA.
7. Complete extension to FPRD
8. SLADD signed off
9. Unit tests complete
10. Sub-system tests complete

11. Functional tests complete

15. References

Reference 1 (NAOMI management plan)

A J Longmore: ÒNAOMI Management Plan.Ó Internal document number AOW/MAN/AJL/7.1/07/96/NAOMI Management Plan, Version date: 10 June 1996

http://aops.dur.ac.uk/bscw/bscw.cgi/0/891?id=947

Reference 2 (FPRD)

Functional and Performance Requirements description of NAOMI system, or: ÒTechnical Description of NAOMIÓ .
http://aops.dur.ac.uk/bscw/bscw.cgi/0/891?id=2001

Reference 3 (SOF-STD-1)

Paul Rees ÒING Software Standards Overview Ò ING/RGO document SOF-STD-1

issue 1.1 24 June 1996 and later releases.

Reference 4 (SOF-STD-3)

Guy Rixon, Òconfiguration Management standards for ING Observing systemsÓ Issue 4.1 May 1996 RGO document number SOF-STD-3

Reference 5 (Ranade and Nash)

Jay Ranade and Alan Nash, ÒThe elements of C programming styleÓ ISBN 0-07-051278-7, McGraw Hill 1993

Reference 6 (SciOpReq document)

R.M.Myers: ÒTop Level Scientific and Operational Requirements for NAOMI.Ó Internal document number AOW/SYS/RMM/6.0/07/96/NAOMI S & O Requirements.