FIRST/ESA/D/0023.01

Customer	: **ESA/ESTEC**	**Document Ref:**	**FDS.PWN.003**
Contract No	: **12628/97/NL/RE**	**Issue Date**	: **15 May 1998**
WP No	: **1130**	**Issue**	: **Draft B**

Title : **DEVELOPMENT OF A FINDAS PROTOTYPE: REQUIREMENTS, TOOLS AND SCENARIOS**

Abstract : This report describes plans for development of a FINDAS prototype. Test scenarios are derived from requirements drawn from the overall FINDAS User Requirements, or designed to explore potential risk areas. Recommendations for the development environment include a commercial database, other COTS tools, and a design methodology.

Author : ____________________
Phil Winder

Approval : ____________________

Accepted : ____________________

Distribution : P. Estaria ESTEC
J.-J. Mathieu ESTEC

FINDAS Project Documentation Library

Hard Copy File: FDS005
Network ID: P:\FDS\PWN\PWN003_B.DOC

2 Falcon Way, Shire Park, Welwyn Garden City, Herts, AL7 1TW, UK.
Tel : +44 (0)1707 391999
Fax : +44 (0)1707 393909

9. Object Oriented Analysis and Design with Applications, G. Booch, Benjamin/Cummings, 1994.

10. Object Oriented Modelling and Design, J. Rumbaugh et al., Prentice-Hall, 1991.

11. Object Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, 1994.

12. UML Distilled, M. Fowler, Addison Wesley Longman, 1997.

2. PROTOTYPE FUNCTIONALITY AND REQUIREMENTS SUBSET

2.1. Requirements

High-level requirements on the FINDAS prototype arise from several sources, each of which is examined in this section. There are more requirements than it will be possible to accommodate in a rapid prototype development, and so the background to each is stated briefly in order to establish priorities. Scenarios proposed for the prototype demonstration are presented in Section 3 on page 12, and treatment of requirements is summarised in Table 1 on page 17.

2.1.1. Key FINDAS Concepts, Cost Drivers and Risk Areas

The FINDAS ground segment introduces several new concepts which are described in our *FINDAS Concept and Architecture: Technical Report* (RD.4). It is important to establish during the initial prototype development phase that these concepts operate successfully in practical situations, and to identify any potential refinements, before commencing implementation of the main operational system. This list is a subset of requirements identified for the operational FINDAS system (RD.7, RD.8), reflecting our priorities for the prototype phase.

R1) **Seamless transition between phases.** It is a critical requirement that FINDAS supports seamless transition between mission phases, in order to introduce unnecessary delay and/or additional development or duplicated effort into a project with a tight schedule and restricted budget.

R2) **Distributed architecture.** Both FIRST and PLANCK project teams are geographically distributed, but FINDAS must integrate each site into a single system. This affects both the visible client interface aspects, and hidden networking and administration functions. It is important that data location should be hidden from the client interface, and that shadow databases operate efficiently.

R3) **Data size and system scalability.** Data requirements for both FIRST and PLANCK projects will probably be relatively modest in the initial stages, however the FINDAS design must anticipate a rise to several Terabytes after the spacecraft becomes operational.

R4) **Schema complexity.** In order to support efficient and easy access to the data, it is important that FINDAS captures transparently relationships between items in the database. For example, these links may be used to support efficient search and retrieval operations; or to present an enhanced interface to a science User by prompting with appropriate instances of associated information.

R5) **Adaptability, flexibility and durability.** It is predicted that FINDAS will remain operational for at least 20 years, spanning several mission phases. During this period it is inevitable that many elements of the ground segment will require significant revision, affecting the science, instrument, operations and/or networking components. FINDAS must demonstrate an ability to evolve functions and interfaces to meet this need.

TABLE OF CONTENTS

AMENDMENT POLICY

This document shall be amended by releasing a new edition of the document in its entirety. The Amendment Record Sheet below records the history and issue status of this document.

AMENDMENT RECORD SHEET

ISSUE	DATE	DCI No	REASON
A	13/5/1998	N/A	Draft
B	15/5/1998	N/A	Minor revisions

1. INTRODUCTION

1.1. Background: FINDAS Concept

Initial analysis of the FIRST ground segment has indicated that many high-level objectives, functional requirements, mission phases and data types are similar to those of other observatory missions. A number of practical issues distinguish the FIRST mission, however, and have important implications for FINDAS. In particular, the ground segment concept must support geographically distributed instrument group expertise and processing, and allow seamless transition between phases and evolution over an anticipated 20 year lifetime. In the context of this novel ground segment concept, a FINDAS prototype will be developed in order to reduce potential risk impacts for the science segment.

1.2. Purpose and Scope

The purpose of this document is to propose recommendations for development of a FINDAS prototype during Phase II of this contract. Each major issue is assessed, and VEGA's preferred approach identified and explained. Topics considered include a design methodology, software development environment, and COTS tools including database and related functionality. The principal goals for the prototype are:

- to demonstrate practically the feasibility and attractiveness of the FINDAS concept, and to encourage participation in its development;
- to demonstrate that FINDAS will offer significant benefits and added value relative to previous systems and approaches;
- to investigate key risk areas and potential cost drivers;
- to obtain additional understanding and information to produce consistent requirements and specifications for the operational FINDAS system;
- to implement an initial FINDAS environment which may be of immediate utility to the FIRST & PLANCK project teams, and which may be subsequently developed into a baseline operational system.

Demonstration scenarios are derived from a requirements subset drawn from User Requirements for the overall FINDAS system. FINDAS concepts are mapped onto specific software functions which may be tested and demonstrated. Additional functionality is proposed to support development of a bare-bones document management system. An architectural design for the prototype system is described, illustrating the components required to demonstrate each scenario.

1.3. Document Overview

Following this introduction, the document is divided into a number of Sections.

- Section 2 identifies relevant FINDAS requirements, cost drivers and software functions, which are candidates for inclusion in the prototype implementation.

- Section 3 gives an overview of major scenarios proposed for the prototype demonstration, relates them to the factors identified in Section 2, and outlines how the demonstration will be conducted at ESTEC.
- Section 4 summarises the rationale for our choice of the O_2 object-oriented database system.
- Section 5 presents our choice of the UML design methodology, and proposed software development environment.
- Section 6 summarises our conclusions and recommendations, and objectives for the prototype implementation during Phase II.
- Appendix A presents user requirements for the baseline FINDAS document management system.
- Appendix B lists details of COTS requirements for Phase II.

This document should be read in conjunction with its companion *FINDAS Concept and Architecture: Technical Report* (RD.4) which addresses the overall FINDAS concept.

1.4. Referenced Documents

The following is a list of documents with a direct bearing on the content of this report. Where referenced in the text, these are identified as RD.n, where "n" is the number in the list below.

1. Implementation of a Prototype Integrated Data Archive System for the FIRST Ground Segment (FINDAS) – TECHNICAL PROPOSAL, FUT.PRO.845 Issue 1, 20 October 1997.
2. Review of the use of CORBA in FINDAS, FDS.PWN.002 Issue 1, 17 April 1998.
3. O2 Object Database System Technical Overview, Unidata Inc., November 1997.
4. FINDAS Concept and Architecture: Technical Report, FDS.PWN.001 Draft B, 12 May 1998.
5. Minutes of the FINDAS Kick-Off Meeting, FDS.MIN.001, 28 January 1998.
6. Software Project Management Plan for the Implementation of a Prototype Integrated Data Archive System (FINDAS), FDS.PLN.001 Issue 1, 20 March 1998.
7. Critical Review of FINDAS User Requirements, FDS.PWN.004 Draft A, 13 May 1998.
8. FIRST Far-Infrared and Sub-Millimetre Space Telescope – Implementation of a Prototype Integrated Data Archive System for the FIRST Ground Segment – Statement of Work, PT-SW-04373 Issue 1, 4 June 1997.

R6) **Data access and security.** Security of proprietary and mission-critical data must be established in the context of a distributed system. All transactions will be logged.

R7) **Real-time performance.** It must be established that FINDAS will be capable of distributing data at a sufficient rate to meet requirements during both operational and test mission phases.

R8) **System availability**. FINDAS will support potentially mission-critical exchange of data between the MOC, FSC and ICCs. It is therefore important to establish that the system will achieve high availability, and that adequate recovery mechanisms exist to restore nominal operations following an unplanned incident.

R9) **CORBA client API**. A key feature of the FINDAS design is to adopt a CORBA client interface rather than the more established RPC mechanism (RD.2).

R10) **CORBA vs ODMG.** Because FINDAS has a three-tier client-server architecture (RD.1, RD.4), there are two interfaces to consider. Whereas CORBA has been adopted for the client interface, the middleware-to-database interface could be implemented using either CORBA or ODMG standards. This is an implementation detail hidden from client applications.

R11) **Robustness to transient network failure.** It is intended to distribute real-time data over FINDAS using relatively low-cost data links which will inevitably suffer occasional periods of interruption or unavailability. FINDAS must automatically resume a connection when the link is re-established whilst ensuring that no data is lost.

R12) **User categories**. FINDAS shall discriminate between classes of User to offer different access rights and interfaces. This principally affects internal vs external users, and novice vs. expert users.

Specific software functions arising from these requirements are derived below. Table 1 shows the subset of these requirements which will be implemented by the prototype.

2.1.2. Specific Requirements on the Prototype

In addition to the above requirements, further requirements have been placed on the FINDAS prototype in order to ensure that it immediately begins to support project needs, and can quickly evolve into a baseline operational system.

R13) **Configuration control.** It is essential that each version of an object is recorded within the database together with an audit trail which may be interrogated for change statistics.

R14) **Web User Interface.** The prototype will support a Web interface compatible with common Web browsers. This will support session management and navigation to related applications (a "FINDAS home-page").

R15) **Link to DMS.** FINDAS will support a link to the ESTEC document management system DMS. Owing to current limitations of DMS, this will be a URL to allow Web-based querying only.

R16) **FINDAS document management system.** In addition to DMS, the prototype will support an internal bare-bones document management capability (Appendix A). This will contribute towards FIRST/PLANCK project infrastructure, and serve as a vehicle to demonstrate FINDAS prototype capabilities.

R17) **Database administration.** It is essential to support an extensive administration interface in order to allow the prototype to be installed at different sites, to allow specific demonstrations to be set up, and to support routine operation.

R18) **Client API.** In addition to the Web-based interface, the prototype will offer a client API using the CORBA interface.

Where possible, these requirements will be merged with those listed above to form an integrated set of software functions scenarios.

2.2. Software Functions

This section expands on the requirements identified above by identifying specific software functions. These are employed to investigate risk areas or to demonstrate novel concepts in the scenarios described below on page 12. In order to demonstrate network aspects, it is assumed that the prototype will operate on at least two workstations.

F1) **Web access** (access using a NetScape client) – the prototype will offer pages via a suitably configured HTTP server. [R14]

F2) **Login via Web** to begin FINDAS session – password authentication. [R14]

F3) **Replication and distribution** – data supplied to FINDAS at one node may be retrieved and/or updated at another. [R2]

F4) **Notification** of events to a User via e-mail.

F5) **Configuration control** – version history and audit trail. [R13]

F6) **Access control** – restricted access and update rights. [R6]

F7) **RT telemetry flow** – real-time performance of database and recovery from transient network errors; integration with a simple low-level interface. [R2,R7,R10]

F8) **Store, query & retrieve** – interaction with the database. [R4]

F9) **Multiple views** on database – client interfaces to view data in different formats; update schema but maintain client interface. [R5,R18]

F10) **Links** between objects – "automatic binding". [R4]

F11) **Local & remote access** – data physically held at one node may be accessed at another. [R1,R2]

F12) **Database administration** – allocate data directories to servers, backup and restore, create users and privileges. [R17]

F13) **CORBA and ODMG** – assess merits of competing interfaces. [R7,R10]

F14) **FTP cache** – retrieval of large data items initiated from the GUI will be directed to an FTP cache area.

F15) **User categories** – offer different interfaces and functionality according to User category (internal/external, expert/novice). Note that these attributes are orthogonal from security-related issues such as access rights and group privileges (RD.4). [R12]

Certain additional generic software functions are necessary to provide a foundation to support these high-level functions and to ensure that activities can be co-ordinated and controlled. This infrastructure is included implicitly and may require significant development effort.

Treatment of these software functions by the prototype development is summarised in Table 1 on page 17.

3. DEVELOPMENT PROPOSALS AND DEMONSTRATION SCENARIOS

Section 2 has listed requirements for the FINDAS prototype by concentrating on risk areas, and the core subset of requirements for the operational system necessary to achieve a working system (RD.7, RD.8). Owing to constraints on the prototype development, it is not possible to implement all features at this stage: some will be included only in a restricted "concept proving" style; others must be placed on a "reserved" list or deferred. Other requirements are addressed by off-line means, such as automatic de facto implications of the FINDAS architectural design (distribution across a network, for example). Possible approaches are:

- demonstration of scenarios identified in this section by the FINDAS prototype;
- informal coding experiments which guide details of the software specification and implementation;
- off-line analysis of interactions, data flows and performance;
- surveys of available sources.

The choice of database package clearly has an impact on FINDAS, and we may draw on existing performance surveys in this area to broaden the scope of our analysis beyond what can be carried out directly within this project. Our proposed treatment of requirements is summarised by Table 1 on page 17.

This section concentrates on the FINDAS prototype, and identifies how requirements and software functions discussed in Section 2 will be treated during the development phase, outlining scenarios which will be employed during the demonstration at its conclusion. It is anticipated that some changes of emphasis will arise during Phase II: these will be discussed with ESTEC where appropriate to agree how these plans should be updated.

3.1. Restrictions on Prototype Development

It has been agreed with ESTEC that CORBA Security Services libraries will not be available during the prototype development, although they have been recommended for the operational system (RD.2). This level of security is not appropriate for a prototype system and as an established COTS product, it is not felt to be a risk area. Omission of these libraries means that many security and access control features will not be implemented by the prototype. This has some impact on several requirements but particularly affects R6 and R16.

It is possible that other COTS products may also be recommended to support particular areas of functionality as Phase II progresses, perhaps with similar implications for the prototype system.

3.2. Demonstration Infrastructure

The completed FINDAS prototype will be demonstrated in ESTEC at the Phase 2 Final Review Meeting, following installation and licence transfer. A Software User Manual, Software Transfer Document and Software Demonstration

Procedure (SVVP) will be written to aid these processes. This demonstration will make use of two networked SPARC workstations with local disc, in the configuration illustrated by Figure 1. Tools installed to support the demonstration must comply with requirements specified in Appendix B on page 27.

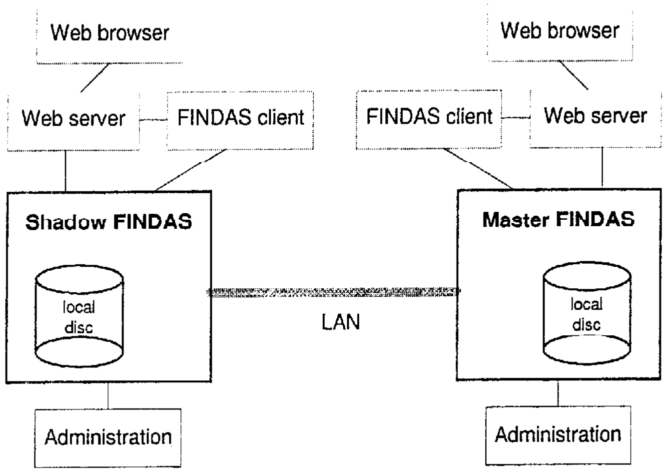

Figure 1 – Infrastructure for prototype demonstration

Although two workstations and attached clients and Web servers are shown, some scenarios will require only one workstation. It is not important whether the Web browser (such as NetScape) is hosted by the workstation executing FINDAS. The Administration module offers database administration functions which are required to configure each scenario, as described below.

Properties of the LAN used to support network communication during the demonstration differ significantly from the limited bandwidth leased lines presently envisaged for the operational system (RD.8). It is also likely that hardware specifications which will be proposed for the full FINDAS will not be met during the demonstration. This does not significantly affect the validity of the exercise, however, even for "real-time" aspects, because it will still be possible to assess the efficiency with which data is communicated over the network. These characteristics may then be extrapolated to operational data rates and network bandwidths to estimate future performance levels.

3.3. Deployment of Demonstration Scenarios

It will be necessary to install and configure the database and FINDAS software before any scenarios may be rehearsed. These operations will be the subject of a separate Software Transfer Document to be written during Phase II.

Most scenarios require the system to be placed into a known state before specified operations are performed by the operator. This will require uploading data schemas and test data into the database, configuring the Web server, and initialising appropriate client programs. Similarly, it may be necessary to remove data from the system when the scenarios have been completed. It is intended to use the database administration capabilities to support these operations, which will be made as straightforward as possible by batching together scenarios or by composing scripts to automate the process where appropriate.

3.4. Outline Proposals for Demonstration Scenarios

The scenarios listed in this section will be employed to demonstrate the FINDAS prototype following installation at ESTEC. Many features have been integrated into each scenario to avoid introducing artificial procedures where this is not necessary. The outline descriptions presented here will be elaborated into a Software Demonstration Procedure Document (SVVP) during Phase II.

Broadly, the scenarios proposed here fall into two groups. In the former group, it has been recognised that it is not possible to present a fully operational environment in which to perform the demonstration, and only the important features are preserved. Such scenarios are useful to examine the underlying functions or trade-offs, but are necessarily artificial in nature and have no on-going purpose. They will be implemented to a level necessary to demonstrate technical concepts only rather than providing significant support to domain-specific applications. Scenarios in the latter category still have a simplified domain, but it is less artificial so that they do offer a capability with immediate application. Both groups are built on the same foundation (utility classes and software infrastructure), so that the majority of the prototype development effort envisaged here contributes towards future implementation of an operational FINDAS system.

A Web User interface and CORBA client interface will be used for substantially all scenarios, and hence the Web User interface and CORBA API requirements are satisfied almost universally. Similarly, database administration functions will be exercised in establishing an environment in which each scenario will be demonstrated.

S1 – Schema evolution R1 R5

An object present in the database will be given modified attributes and methods (this will require FINDAS to be taken off-line for intervention by the database administrator). A client application may continue to interact with the object using the original interface; a new interface will also be available reflecting modifications made to the schema. Both eager and lazy update methods will be demonstrated to establish that the client application is fully insulated from schema changes. The ability to modify the data schema without disturbing client applications is vital to ensure seamless transition between mission phases and painless evolution of interfaces and detailed requirements.

S2 – Multiple views of an object R5 F9

A single object or data item will be accessed and manipulated in different ways using alternative interfaces supported by the object servers. Flexibility of access ensures that client applications are hidden from the organisation of data within the database. Coupled with the ability to modify the data schema, this ensures that client applications are insulated from revisions to detailed data formats and interface revisions.

S3 – Real-time transfer R2 R7 F3 F7

It is essential for FINDAS to distribute data in real time across a link of limited bandwidth. This requirement may fail if the transfer protocol is too inefficient, or heavy overheads cause the processor to become the bottleneck if line throughput is raised. In this scenario, a source of dummy "telemetry" data will feed one FINDAS node, whilst another client will act as a consumer at the other node, relying on FINDAS to distribute the data. Statistics will be gathered to estimate processing and communication requirements for the full FINDAS system. It is proposed to use the LAN for network communication in all cases. A simple low-level interface will be used, in keeping with that proposed for the full system. The receiving client *may* be extended to mimic FINDAS down-link processing (cross reference between telemetry and observations; update proposal database). This would allow major elements of the FINDAS data scheme to be illustrated, and may lead into replay of telemetry and more complex telemetry interfaces. Any such extension would be highly stylised.

S4 – Access DMS application via Web R14 R15

A URL link will be added to the FINDAS "home page" to allow navigation to the ESTEC DMS document management system.

S5 – Operate internal document management system R13 R14 R16 F1 F2 F5 F8

It is proposed to develop a bare-bones document management system within FINDAS (Appendix A). A series of related scenarios will exercise all features of this system, concentrating on four operations: browse and display, search, insertion, and revision. This will include situations where access should be barred, such as attempts to revise a document by a User who is not the author. In addition to purely document management system actions, this sequence will exercise a variety of core FINDAS functions such as configuration control.

S6 – Event notification F4

This scenario will trigger client processing based on an event within the database (such as an e-mail sent on receipt of a particular input stimulus).

S7 – Remote access and shadow database R2 F3 F11

A client application attached to a FINDAS object server on one workstation will access and manipulate data held on a remote workstation. It will be established that this operates smoothly, and that the location of the data may be completely hidden from client application.

In a revised version of this scenario, the master copy of the data will be held remotely but a (read-only) copy will be held in a local shadow database. Again, this will be hidden from the client application.

S8 – Robustness to network error R11

It is a critical FINDAS requirement that no data is lost. In particular, no telemetry packets may be mislaid during the real-time downlink period in the event of network errors. The ability of FINDAS to synchronise and resume the transfer once normal communication is restored will be demonstrated by temporarily interrupting the network link between two FINDAS nodes during a transfer.

It is also important to ensure that rapid recovery is possible following network unavailability over longer periods. This will be supported by demonstrating the ability to synchronise databases by transferring data to the remote FINDAS node off-line (e.g. on a floppy disc).

S9 – Logging of transactions R6 F6

A transaction log will be collected for data accessed during this scenario (both browsing and updating).

S10 – Investigation of CORBA vs. ODMG interfaces R10 F13

As explained in *FINDAS Concept and Architecture: Technical Report* (RD.4), there are two competing standards which may be used to implement the database to object-server interface. This scenario will ensure that this issue is fully explored by assessing the relative merits and overheads associated with each approach.

S11 – Configuration control R13 F5

Configuration control functions will demonstrated. It is possible that this may be achieved by the document management scenarios.

S12 – Third-party application client R18

Third-party development of a FINDAS application client according to a published interface specification has been discussed in principle. There is no commitment at present to integrate this scenario into the prototype demonstration.

3.5. Cross-Reference from Requirements to Scenarios

This section summarises how each requirement and software function will be treated.

Requirement/Function	Status	Comment
Seamless transition between phases R1	(✓)	S1
Distributed architecture R2 F3 F11	✓	S3, S7
Data size and system scalability R3	✗	Off-line
Schema complexity R4 F10	(✓)	Design
Adaptability, flexibility and durability R5 F9	✓	S1, S2

Requirement/Function	Status	Comment
Data access and security R6 F6	(✓)	S9
Real-time performance R7 F7	✓	S3, S10
System availability R8	✗	Not in scope
CORBA client API R9	✓	All
CORBA vs ODMG R10 F13	(✓)	S10
Robustness to transient network failure R11	(✓)	S8
User categories R12 F15	✗	Not in scope
Configuration control R13 F5	✓	S5, S11
Web User Interface R14 F1 F2	✓	All
Link to DMS R15	✓	S4
FINDAS document management system R16 F8	✓	S5; Appendix A
Database administration R17 F12	✓	To conduct demonstration
Client API R18	(✓)	S12 and others
Notification F4	(✓)	S6
FTP cache F14	✗	Not in scope

Table 1 – Treatment of requirements and risk areas

Notation: ✓– will be demonstrated; (✓) – will be partially demonstrated; ✗– will not be demonstrated.

Demonstration of requirements by the prototype will often use a restricted stylised environment in order to address technical concepts rather than domain issues.

4. CHOICE OF DATABASE SYSTEM: O_2 OODBMS

4.1. Database Recommendation

A commercial database package lies at the heart of the proposed FINDAS architecture (RD.4). COTS tools have been chosen where possible to ensure cost-effectiveness and to minimise development risk. VEGA are pleased to confirm our initial recommendation of the O_2 object-oriented database from Ardent Software Limited. O_2 is a versatile and efficient database, complies with relevant standards, and has an established User base. We feel that it is best placed to support the FINDAS system both now and throughout the project lifetime. Our rationale for proposing an object-oriented database and O_2 in particular, is summarised below.

4.2. Rationale for an Object Oriented Database

FINDAS is required to store both diverse data sets and their complex relationships, and to support efficient queries and retrievals including multiple logical views and alternative interfaces. Both short-term modification of data links and long-term evolution of the data schema are envisaged, within and between mission phases. These factors strongly suggest adoption of a truly object-oriented database management system (OODBMS), to take advantage of direct support for manipulation of sophisticated data objects and for extensibility of the data schema and database interfaces. In particular, an OODBMS is well placed to react to unpredictable changes to data storage requirements or access patterns. Object databases also fit well with GUI-based applications and Internet developments (including CORBA and Java).

By contrast, a traditional relational database management system (RDBMS) offers very limited support for complex data, typically manipulating only primitive data types (integers, character strings, etc.). An RDBMS excels when the database is populated with very many instances of simple static data records, which are accessed in a predictable and repetitive fashion.

Several vendors have recently adapted existing RDMBS to offer an object veneer. In these new hybrid systems, the core database engine is relatively unchanged, with an interface layer converting objects to a more primitive representation before insertion into the database. This necessarily imposes performance overheads that do not arise from a truly OODBMS, and may restrict flexibility to respond to changing data requirements. A potential advantage of hybrid systems is to provide a straightforward mechanism for data exchange with legacy RDBMS applications. Since RDBMS interface modules are offered by many OODBMS systems (in addition to universal interface standards such as CORBA), and legacy applications are not a critical area for FINDAS, adoption of core OO support with a true OODBMS is preferred.

4.3. Reasons for Selection of O_2

VEGA is independent of all database vendors, and we are therefore able to offer an unrestricted assessment of potential candidates. O_2 was the clear choice arising from this analysis of key FINDAS requirements: it satisfies all major criteria and offers additional support through add-on modules.

- **Scalable client/server architecture** – internally, O_2 supports a page server architecture. This devolves much of the database management system overhead to the client process, maximising the number of concurrent sessions which may be supported by the server. Clustering, page caching and adaptive locking raise performance and ensure that overheads vary linearly with database volume.
- **Large databases** – O_2 has been optimised for large databases. Existing very large (approaching Terabyte) database applications include a French military communications project, the Korean Patents Office, NatWest Securities information systems, and CNES "Baghera" space vehicle technical data management project.
- **Existing User base** – O_2 was first shipped commercially in 1991, and has been employed successfully in a wide variety of applications and Web integration projects, including: the French Ministry of Culture, Crédit Agricole, and Renault Formula One. Space applications include: Trasys (SPOT-4), Far Ultraviolet Spectroscopic Experiment (FUSE), and NASA Goddard's Mission Operations Control Architecture.
- **Replication** – O_2Replication offers COTS support for data replication, facilitating database distribution and high availability.
- **Web interface** – O_2Web supports dynamic HTML generation for any object stored in the database, simplifying construction of FINDAS Web interfaces.
- **Interoperability modules** – for connection to relational databases, Open Database Connectivity (Microsoft) databases, and CORBA.
- **Full ODMG standards compliance** – Object Query Language (OQL), C++ and Java bindings, and CORBA.
- **Flexibility and evolution** – O_2 supports flexible storage and schema management. For example, a single database may be physically dispersed across several discs. The schema manager accepts class modifications during development or at run-time with either eager or lazy updating. An API is available to browse class properties including attributes and inheritance, or to update classes dynamically.
- **Database utilities** – O_2 provides functions for database administration, including backup and restore, and database reorganisation.

O_2 also offer an on-going development program, with a major release approximately once per year, and support across Europe. At least some of these requirements were not met by competing OODBMS vendors. For example, ObjectStore does not offer support for database replication and has no track record of very large applications.

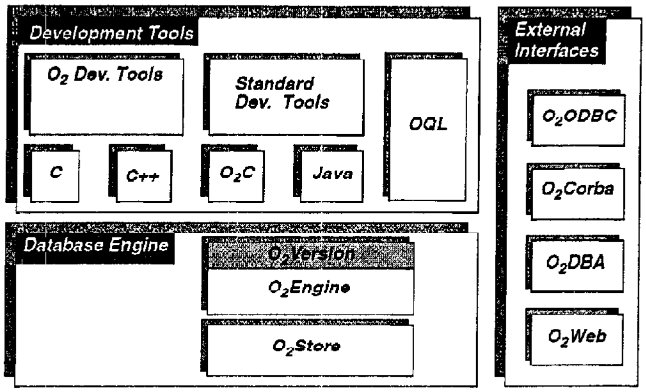

Figure 2 – O_2 system architecture

As has been indicated above, O_2 contains many supplementary modules in addition to the core database engine, many of which have relevance for FINDAS (not all will be used by the prototype). The architecture is shown in Figure 2. Extending prototype functionality by integrating COTS modules rather than with custom development offers cost-effective risk reduction. Three modules not mentioned explicitly above are: O_2Notification – asynchronous event-driven communication with other clients; O_2Version – support versions of object collections; and O_2Corba – to configure O_2 as a CORBA server. The modules required for the prototype development are identified in Appendix B on page 27.

4.4. Implications of Adopting an Alternative Database Later

We are confident of our recommendation to select the O_2 database system for the reasons summarised above. It must be acknowledged, however, that the prototype development will bring additional experience of O_2 which may lead to re-examination of this choice, and that other events during the project lifetime may dictate a change of database strategy.

A three-tier client-server architecture will be adopted for FINDAS (RD.4), with a CORBA interface presented to client applications. This means that clients are well-shielded from direct interaction with the database system by both CORBA and the object server middleware. For this reason, and the use of standard interfaces, any future switch to another database vendor would have little or no impact on client applications. Indeed, it would be possible to have a heterogeneous database architecture without this being apparent to client applications.

Substitution of an alternative database system has more impact on the object server because this interacts directly with the database engine. Here too, however, a standard interface is used (ODMG or CORBA), that will restrict recoding changes to truly database-specific functionality. Note that similar arguments imply that any evolution of interface standards during the project lifetime will also have only localised implications for FINDAS.

5. DESIGN AND SOFTWARE DEVELOPMENT ENVIRONMENT

5.1. Design Methodology

There are many prominent object oriented methodologies for which design tool support is currently available. However, there are probably only three clear leaders: the Booch (RD.9) method, developed by Grady Booch; OMT (RD.10), developed by Rumbaugh and co-workers; and Ivar Jacobson's Objectory method (RD.11). All three of these have their particular strengths and strong advocates, and also offer many common facilities, yet the notation used by the three methods is very different.

The Rational Software Corporation has long been associated with the design tools market and their Rose product has been a leading contender for many years. As a result there has been a migration of talent to Rational over the past few years and the situation has now been reached where all three of the above experts work for Rational. This has lead to an attempt to produce a standard design methodology by combining the best features of the three methods. The resulting modelling language is called the Unified Modelling Language (UML) and the resulting process is called Objectory (Jacobson is the person chiefly responsible). UML (RD.12) in particular has risen rapidly in prominence. We therefore recommend the adoption of UML as the modelling language of choice for FINDAS.

Rational Rose fully supports the diagram types prescribed by UML, and is also compatible with the use of Booch and OMT. It has a well thought out Windows interface, is easy to use, and has a number of useful features for navigating between diagrams. It includes the facility to generate C++ and Java code directly from the model, and the ability to reverse engineer diagrams from code. In common with other tools, however, reverse engineering tends to work best when applied to the code generated by the tool. An additional facility available is direct generation of CORBA IDL. Rational Rose has all the facilities needed for FINDAS, and given that its further development will be directly driven by the people behind UML, we recommend its adoption as the standard tool for use in FINDAS.

5.2. Software Development Environment

The FINDAS prototype will be developed on two networked Sun SPARC workstations under Solaris 2.6 as previously agreed (RD.5) in C++. Once an appropriate level of module testing has been completed, all code will be placed under configuration management and will be backed up regularly (RD.6). Sun's Visual Workshop environment will be used to support code development, including compilation, libraries and debugging, and GUI-building if appropriate. We plan to use the Sun Web Server to enable Web access to the prototype (it is available from Sun as part of Solaris 2.6). FINDAS will use the O2 database on both workstations, including additional modules listed in Appendix B, and Orbix ORB library. RogueWave class libraries will be used during development. Solaris 2.6 also offers support for Java – although there is no commitment to use Java for the prototype.

5.3. Overview of Prototype Architecture

Outline architecture (tie into scenarios). Make clear what will be delivered (object servers, clients).

6. CONCLUSIONS, RECOMMENDATIONS AND OBJECTIVES

Building on our review of proposed FINDAS User requirements (RD.7) and analysis of the distributed FINDAS concept and architecture (RD.4), we have identified topics which should be investigated in more detail with the development of a prototype system. These functions were derived from key FINDAS objectives and identified risk areas, and we have outlined scenarios for demonstration at the Phase II Final Presentation.

We have recommended a design methodology and software development environment for the prototype system, which will carry over into the full FINDAS implementation. Having assessed database requirements, we recommend the O_2 object oriented database system from Ardent Sofware. It meets all essential requirements and offers additional support for functions that will be used in both the prototype and operational FINDAS systems. COTS are preferred to fresh development where possible, and other suggested packages were also specified.

Building from the outline scenarios presented in Section 3, we have illustrated our architectural design for the prototype. This is consistent with the design proposed for the operational system. Features necessary to support each demonstration scenario were identified and described.

We are confident that the prototype will contribute significantly towards the development of an operational FINDAS system. It will establish the attractiveness of the FINDAS concept, and encourage early participation and interaction with potential FINDAS Users. This experience will be fed back into revised requirements, specification and design for the operational development, reducing risk and increasing understanding of technical issues, costs and co-ordination requirements.

APPENDIX A – BASELINE DOCUMENT MANAGEMENT SYSTEM

A.1. Background and Purpose

It has been agreed that FINDAS will support a link to ESTEC's internal document management system DMS, and it is anticipated that this will be used for official FIRST project documents. Additional documentation requirements have been identified, however, which DMS is not able to fulfil. It has therefore been decided that FINDAS will also support a document management capability, to be integrated with the distributed database functionality.

Development of a document management system is not a top priority for the prototype development because it is not a cost driver or a risk area, and a complete system with an attractive interface could absorb significant development effort. It is important, however, to bring documentation into FINDAS as soon as practicable, in order to enjoy a single and complete repository during subsequent mission phases. Therefore, it has been agreed that the prototype will support a "bare bones" document management capability: just enough to support the four critical functions of document deposit, interrogation, display and revision; but necessarily far short of a complete system. These limitations may be addressed during later phases of the project, as other priorities permit.

A.2. Functionality

The document management system will operate as a FINDAS client. This means that it enjoys the capabilities of the core system developed to support FINDAS prototype activities, and that it operates in tandem with a FINDAS database and object server, and a Web server. The document management system will be exploited during the prototype demonstration as a vehicle to illustrate many features and concepts relevant to the complete FINDAS system.

A.2.1. Document Data

The document data object held within FINDAS will have the following attributes:

- a display file in Adobe PDF format;
- a source file (in any format);
- author and FINDAS user_id;
- title (and version);
- keywords.

The binary document source file is not manipulated by FINDAS, and hence may be of any format (for example, a Word document, or an archive of TeX files).

A.2.2. Document Navigation, Search and Display

D1) The documentation client will be accessed via a Web browser from a link on the FINDAS home page, once a session has been commenced.

D2) An administrator will be able to create a multilevel document hierarchy, grouping together documents of similar scope (this will not be supported from the User interface). For example, a gross distinction may be drawn between FIRST and PLANCK documents.

D3) It will be possible to navigate the hierarchy directly by clicking on links presented in the Web browser.

D4) By default, only the most recent version of each document will be shown. However, it will be possible in the GUI to request that *all* versions are shown.

D5) Documents may be searched for by author, by title, or by keyword. A list of documents matching each query will be presented. Note: it is not possible to search the body of a document.

D6) Each document will be listed by its title. Clicking on this URL will download the document PDF display file to the Web browser (which may be configured to display it within the browser window using Adobe Acrobat).

Given a document, it is not possible to query it for the author or keyword list – the system is only capable of searching the catalogue for a list of documents which match a particular author or keyword. The document source file is opaque and cannot be interrogated directly.

A.2.3. Document Deposit and Revision

D7) Any FINDAS User may deposit a new document into the hierarchy (subject to permissions configured by the operating system) by clicking on a button presented in the Web browser.

D8) In order to deposit a new document, the User will be prompted for the fields listed in Section A.2.1, with the exception of the FINDAS user_id which is already known. The document display and source files will be uploaded from his local workstation (without format checking), but other fields must be entered manually into the GUI. When the User confirms these details, the new document is added to the database.

D9) Only the owner of a document will be able to download the source file (determined by FINDAS user_id).

D10) The owner of a document will be able to revise a document by uploading new information as described in D8.

D11) Only the administrator may delete documents or versions of documents.

FINDAS cannot verify the information provided, therefore it is important to take care that it is entered correctly.

A.3. Potential Future Enhancements

For the reasons set out in Section A.1, only a bare-bones system is proposed at this stage. Some future enhancements that may be considered are:

- more flexible access – update and read-only access to groups and to classes of Users rather than just individuals, allow wider access to source document;
- notification of new documents – register interest to receive e-mail when documents are added or modified;
- improve support for establishing links between related documents;
- augment document meta-information;
- refine data entry – prompt default values, improve error-checking.

Although desirable, these are not felt to be of immediate necessity. It is intended that future evolution of the system will not necessitate re-entry of existing documents.

APPENDIX B – DETAILS OF COTS REQUIREMENTS

Details of all COTS tools currently envisaged for the prototype development are presented in this section. These are subject to change (by agreement) during Phase 2. We distinguish between tools required to develop the system (at VEGA) and to operate it (at ESTEC). Several licences required during development have already been purchased by VEGA, and are not project deliverables. A warranty framework will be proposed in a separate document.

All licences will be covered by support and maintenance contracts during the project development phase, where available. Upgrades will be considered on a case-by-case basis as new versions are released.

B.1. Tools Required during Development Only

All licences listed in this section will be purchased by VEGA and are not project deliverables.

B.1.1. Sun Visual Workshop

Visual Workshop version 4.2 is available from Sun. It contains several relevant modules:

- C/C++ compilers – to compile FINDAS and client code;
- optimiser and link-editor – to build FINDAS and client code;
- debugger, source browser, other utilities – to support FINDAS development and investigation of anomalies;
- Tools.h++ – a module based on RogueWave Tools.h++ version 7.0.2 which will be used during development of FINDAS and clients;
- visu – a Motif GUI-builder based on XDesigner, which may be used to develop FINDAS or client GUIs.

Each of these modules is licensed, however no licences are required at run-time.

B.1.2. Rational Rose

Rational Rose Professional C++ Edition version 5 is available from Rational Software Corporation. It will be used at VEGA during the design and development process, but is not essential for subsequent further development of the FINDAS prototype following the conclusion of Phase II.

B.2. Tools Required during Development and Execution

Responsibilities for licensing and package installation are defined by the Project Plan (RD.6).

B.2.1. O_2 Database Modules

O_2 version 5.0.2P6 is currently available from Ardent Software. Two sets of SPARC server licences are required with identical sets of O_2 modules, one for each workstation.

- Main server: O_2 object database server 1–4 concurrent users.
- Shadow server: O_2 object database server 1 concurrent user.
- Modules for each server: OQL, O_2C++ binding, O_2Corba, O_2Notification, O_2Web, O_2Replication, O_2Version.

Other modules identified in Section 4 are included in the core O_2 server product: O_2Engine, O_2Store, O_2DBA.

VEGA will procure and deliver O_2 licences to ESTEC in time for the Phase 2 Review Meeting.

Note: the status of O_2Corba licences has yet to be confirmed.

B.2.2. CORBA Modules

Orbix version 2.2[1] available from Iona Technologies PLC will be used to implement CORBA interfaces. One copy of the development kit and runtime modules are required for SPARC Solaris Orbix-MT libraries.

Note: administration of Iona Orbix licences has yet to be confirmed.

B.2.3. Solaris Modules

Two modules available as Solaris 2.6 installation options may be used.

- Sun Web Server 1.0 – to act as the FINDAS Web server. The CGI will be configured to access the FINDAS prototype.
- CDE Motif 1.2.6 – may be used to support development of a Motif GUI for FINDAS or clients.

Solaris 2.6 also includes Java Virtual Machine 1.1 but it is not currently envisaged that this module will be required for development or demonstration of the FINDAS prototype.

ESTEC are responsible for preparation of the FINDAS prototype demonstration environment, including licensing and installation of Solaris 2.6 and any additional modules or mandatory patches.

---

[1] Orbix version 2.3 is preferred for use with O2 version 5.1 due for release during 1990.

B.2.4. Web Browser Client

A Web browser supporting HTML 3.2 and Java is required by a User wishing to interact with the FINDAS prototype. NetScape 3.0 or higher will be used at VEGA during development and will be specified for conducting the demonstration at ESTEC, but the browser software is not a project deliverable.