A CORBA-based Facility for Information Access and Interchange between Cooperating Organizations

Arne-Jørgen Berre, Steinar Haukøy, Jan Øyvind Aagedal
SINTEF Informatics
P.O.Box 124, Blindern
0314 OSLO, NORWAY
E.mail: Jan.Oyvind.Aagedal@si.sintef.no
WWW: http://www.oslo.sintef.no
Phone: + 47 22 06 73 00
Fax: + 47 22 06 73 50

Abstract

In this paper we present a CORBA-based facility for information access and interchange, the Repository Services. This facility provides a flexible approach to information access and interchange by the means of a repository which contains information giving dynamically access to the detail information. The information may be available in different formats at different locations, and is grouped both within and between domains.

1. Introduction

The need for information access and interchange between and within decentralized organizations is increasing as corporate structures become more complex and cooperation increases and changes rapidly. In the times of Business Process Reenigineering (BPR) and virtual companies it is important to have a flexible means of information exchange across system boundaries, locations, representations, and companies. The information stored is often the backbone of a company, and it is a goal to enable the company to get good value of this information both internally and externally. When the content of the information is in constant change, one needs mechanisms beyond the traditional daily file transfers. In this paper we propose a CORBA-based facility for information access and interchange that can be used to achieve this goal. First an introduction is given into the area of systems integration and interoperability, and the ideas behind CORBA is presented. Next the Information Access and Interchange Facility is presented in detail by introducing the basic concepts and presenting the architecture of the solution. This facility has been implemented and used within the ESPRIT III project MARITIME focusing on information management in some of the phases in the life-cycle of ships. This project and its use of the facility is described before a comparison with some other approaches is presented. Finally there is a conclusion and description of future work.

2. Systems Integration and Interoperability

2.1 Goals of integration

The dictionary-definition of integration from [Web] is:

Integration *Complete (imperfect thing) by addition of parts; combine (parts) into a whole. (lat. Integrare: make whole) The focus is on making the whole by combining the parts.*

The challenge of a systems builder is to provide the users with systems supporting an integrated environment by seamlessly "gluing" different underlying system parts together. "Gluing" implies a need for interconnectivity between the systems. Two or more systems are interconnected if they can exchange messages. This, however, only guarantees communication. In order to achieve integration it is also necessary with interoperability between the systems. Interoperability means mutual accessibility and usability of information systems.

A definition of interoperability is given in [Brod92]:
Interoperability: *Two components X and Y can interoperate (are interoperable) if X can send requests R_i for services to Y, based on a mutual understanding of R_i by X and Y, and if Y can similarly return responses S_i to X.*

This means that two systems can interact jointly to execute tasks. What is required is a mechanism that facilitates interoperability between system components by providing support for high level interaction among components. An object-interaction mechanism such as that provided by OMG/CORBA , as described later, or COM/OLE is a suitable approach for this.

2.2 CORBA

The *Common Object Request Broker: Architecture and Specification* (CORBA) [OMG94], is a document from Object Management Group (OMG) specifying the technology which provides interoperability between applications on different computers in heterogeneous distributed environments, and seamlessly interconnects multiple systems. CORBA is the central part of the OMG architecture as shown on figure below.

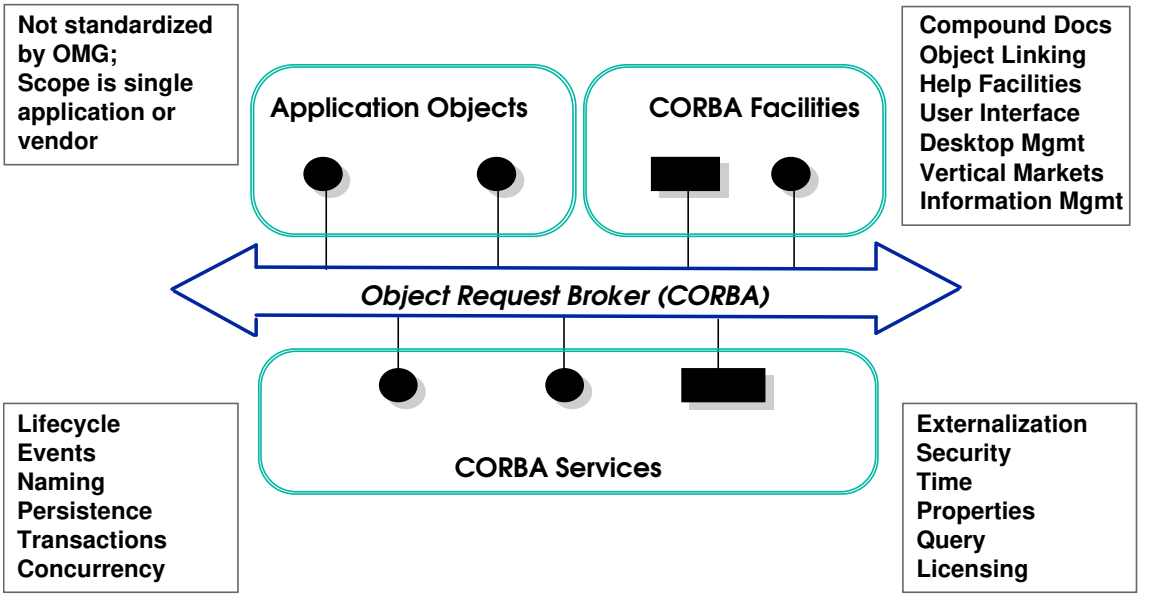

Figure 1 OMG Reference Architecture

Whereas the Object Request Broker CORBA provides a specification of a basic communication infrastructure and CORBA Services provide some additional fundamental services, do CORBA Facilities provide specifications for higher level services and vertical

market speciality areas. CORBA Facilities are “a collection of higher level services broadly applicable to many applications or high value capabilities for specific domains or vertical markets” [OMG95a]. There are two major categories of CORBA Facilities: Horizontal Common Facilities and Vertical Market Facilities. The horizontal set of CORBA Facilities represent technology shared by many or most systems regardless of application domain while vertical facilities includes functions that support various vertical market segments. Horizontal facilities may be facilities for user interfaces, information management, system management, task management, and others, while vertical facilities may support market segments such as CAD/CAM, CIME, Geoscience (Oil & Gas Exploration and Production), Geographical Information Systems, and others.

Common facilities for information management is one of the identified categories of horizontal facilities. The aim is to enable an enterprise to get good value from it’s investment in information by providing information to be modeled, described, stored, retrieved, moved, and interchanged within a system, that is between a set of application objects. Here we propose a facility that enables enterprises to retrieve and interchange information.

2.3 Problem

In work involving several systems (application objects) at several locations all having different representations of some rapidly changing information, the means of smooth information availability is crucial in order to make the different systems interoperate in an efficient manner. In cooperating work there is typically many information sources that should be available to many systems having incompatible internal representations. When this information is in constant change, there is considerable overhead involved if all information should be converted to all the needed representations whenever a change occur. The problems of different locations, representations, sources, and applications are all addressed next.

3. The Information Access and Interchange facility

3.1 Introduction

A typical situation in almost any area is that people are dependent on information made available by others. The goal of an Information Access and Interchange facility is to provide the means by which information may be made available (“released”) for use by others, in addition to providing the means by which the “released” information can be accessed via end-user applications in a secure and transparent way.

Some of the problems to overcome are:

- There are many information sources, that is, there may be a lot of people and applications involved, each of which may provide information to be used by some of the others.
- Information may be provided in different locations.
- Different applications will have incompatible internal representations, and may support only a limited number of export/import methods. The providers of information do not know (or necessarily care) what kind of applications that will be used to access the information. This means that they do not know the requirements the others have as to which formats they would like the information to be released in at the time of release.

- Several companies may be involved and information access and interchange solutions used within a company may be different from those to be used between companies.
- Security aspects both within and between companies must be handled. In companies organizing their work in projects, it is important that only the participants within a project have access to the information within that project. This is, in particular, very important when several companies are involved in the same project. (Each company may be involved in several projects simultaneously.)
- Persistence, the information should be available over time.
- Browsing-functionality is required to be able to find and access the requested information.

To solve some of these problems, a prototype (the Repository Services) of an Information Access and Interchange facility has been developed.

3.2 Concept

The Repository Services is a facility to support information exchange and sharing between and inside companies. The Repository Services is not concerned with functionality or resources used within local applications, but rather with making information accessible to those local applications/end-users that have the necessary access rights.

As can be seen in figure 2, the Repository Services can be used for transparent information access both within and between companies. An application will contact the Repository Service, using the Object Request Broker, which may give access to information stored both locally and in other companies.

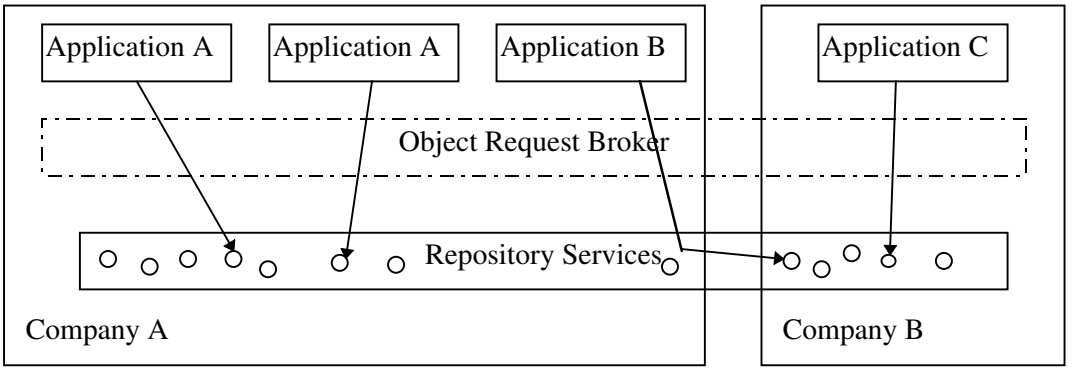

Figure 2: Information Access

Information "released" by an application is represented as an "Information Object", illustrated as circles in figure 2, in the Repository Services. Such "Information Objects" have two aspects:

- Detail information, which is the actual information other applications/end-users are interested in.
- Administrative information to be used by the Repository Services to provide access to the detail information represented by the object.

The "detail information" as such is of no concern to the Repository Services. It is assumed that it is kept under supervision by some other application, and that it can be made available upon request. In this way, the Repository Services are independent of what kind of detail

information is going to be interchanged. The semantics of the information to be delivered is then an aspect that must be agreed upon by the requesting and the providing application, i.e. the Repository Services only offers availability and delivers the information. As described below, there are, of course, some requirements as to how such requests for information must be resolved.

The Repository Services consists of an arbitrary number of Control Servers (see figure 3), to each of which an arbitrary number of applications may be connected. Each of the Control Servers has a database, which in fact can be looked upon as a metadatabase. It contains administrative information related to provided information, and to the Repository Services as such, but not the actual detail information.

It is the responsibility of each Control Server to keep track of all information "released" by some application connected to that Control Server, and to provide access to this information upon request. In addition, a Control Server may retrieve information from other Control Servers. All queries must, of course, be subject to access control.

There are two types of applications:

- An application which provides new information to the Repository Services.
- An application which makes use of the information offered by the Repository Services.

An application providing data to the Information Repository, must define how the information is to be made available to other applications, that is how the detail information should be retrieved, and also in which formats the information may be made available. The Repository Services requires that servers (Information Servers) are available to handle this. These Information Servers are the means by which access to provided information is handled. When the Repository Service receives a request for some information in one of the supported formats, it invokes the proper Information Server which performs whatever action needed to present the information in the requested format.

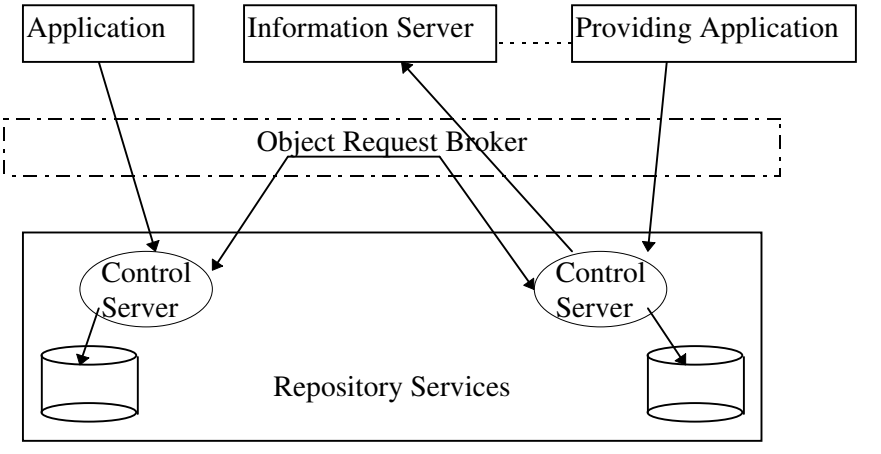

Figure 3: Information Retrieval

Since there is no predefined set of Information Servers, it must be possible to dynamically register Information Servers in a Control Server. Enough information must be registered to make it possible for the Control Server to use the Dynamic Invocation Interface of the Object Request Broker (ORB) to access the Information Server. In addition, it must be registered what kind of formats the Information Server offers to deliver detail information in. By formats it means some common representation of the detail information which both the requesting application can handle and the Information Server can convert the detail

information into, if needed. Typical formats for documents can be PostScript or SGML, for product models ISO/STEP standards.

When an Information Object is created in a Control Server, a reference to an Information Server must be added to that object. In this way, the Information Object "knows" from where to request detail information.

How detail information is made available to an Information Server and in which format it is stored is of no concern to the Repository Services. As shown in figure 4, the Information Server could even access the native database of the providing application. The information "released" into the Repository Services will be subsets of the total amount of information available in the native system, and there may be information in the native systems that are not available as Information Objects. This means that there is not necessarily a one-to-one mapping between Information Objects and constructs in the native system. As long as the Information Server knows how to retrieve the detail information, its representation in the native system is irrelevant to the Repository Services. Another alternative than a native database would be for the providing application to export the detail information and store it in a file, which the Information Server accesses.

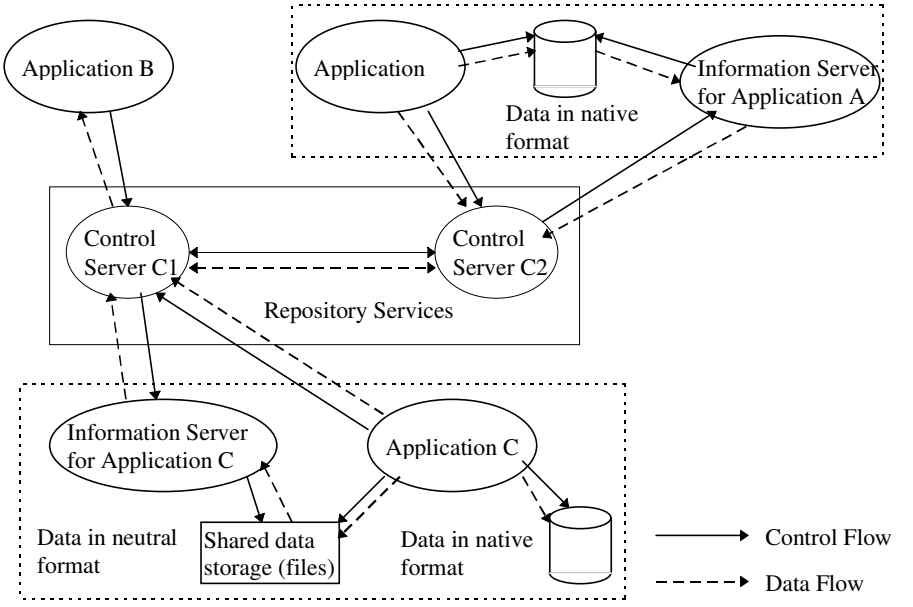

Figure 4: Technical Concept

The solution with Information Servers is a flexible one. Since the retrieval of the detail information is decoupled from the Repository Services, one could let the Information Objects represent "live objects", that is with a changing content, or static objects with a fixed content once defined. When having "live objects", if the information that is used to derive the detail information from the native system changes constantly, the Information Server needs to access the system for each request in order to get the latest version and then do the necessary processing to get the information in the requested format. However, if the information is static, the Information Server could store the detail information when it first has been requested, and on subsequent request only retrieve it locally. Using the Model-View-Controller [KP88] architecture or the event-service in CORBA [OMG95b], the Information Server could get told whenever the relevant information changes. How this is solved is however of *no concern* to the Control Server, it relies on the Information Servers to give the proper information.

Similarly as for Information Servers, there is no predefined set of Control Servers. Any number of Control Servers may be added at any time using the registration functionality of the Control Server. Since there is no automatic replication of information, new Control Servers must be registered in all Control Servers it needs/is allowed to communicate with.

The information in a Control Server is structured as follows:

- Registered Control Servers, and how to be able to communicate with them.
- Information Servers, and how to access them.
- Users (name, password, e-mail, ...)
- Contexts, which can be regarded as a work-environment involving several Control Servers. For each Context,
    - Information Objects, that is, all information made available within the domain of the Control Server. For each Information Object,
        - Which application (Information Server) to trigger to give access to detail information.
        - What kind of formats the detail information is available in
    - Which of the registered Control Servers that participates in current context. (These Control Servers have, in principle, been granted access to the information registered within current Context in current Control Server.)

Each Control Server represents a Domain, which usually is a company or a department. As depicted in figure 5, the Contexts, which usually represents projects, are orthogonal to Domains. A Context may span several Domains, as projects may span several companies. A Context may be further subdivided into Subcontexts having the same properties as the original Context. This is not shown in the figure. The Information Objects are defined within a Context in a Domain, but is made available across Domains within the Context it is in.

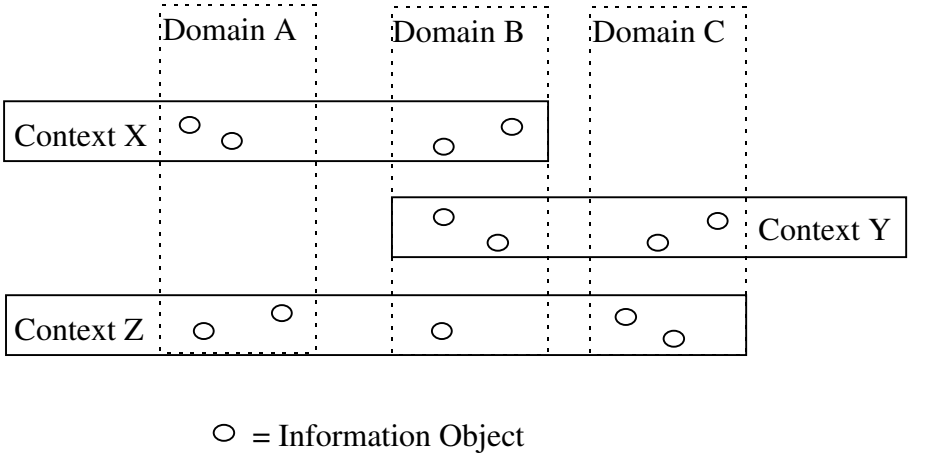

Figure 5: Information structure

The Repository Services as such provides an application programming interface, but no user-interface. An information browser has been developed to be able to browse in the Repository Services. This browser gives access to local information within the current Domain, and also transparent access to all information within a selected Context. It can be used both as an end-user tool and as an administrative tool.

3.3 Implementation

The Repository Services has been implemented using the CORBA implementation Orbix [Iona94] from Iona Technologies, Ireland. Orbix provides a C++ mapping from the Interface Definition Language IDL [Iona94], hence the Control Servers are written in C++. Currently we use TCP/IP as the transport mechanism, but it lies no restriction on using any mechanism as long as the CORBA implementation used supports it. Orbix do today run on top of TCP/IP, IPX/SPX, SNA, Tuxedo and ISIS. Currently all installations of the Control Server must be done on a site using Orbix, but when the inter-ORB communication standard in CORBA 2.0 is implemented, different sites may use different CORBA implementations.

The metadatabase is implemented using the object-oriented database OBST [FZI93] from Forschungszentrum Informatik in Karlsruhe, Germany. When the persistence CORBA service is available in the CORBA implementations, this could be an option to use.

4. The Maritime Project

4.1 Project Data

The Repository Services as described above, has been developed within the ESPRIT III project MARITIME, ESPRIT No 6041.

MARITIME (Modeling And Reuse of Information over TIME) aims at handling the Information Management in some of the phases in the life-cycle of ships. This includes the representation of product information, and also information access, in a distributed environment.

Project participants are BIBA (D), Bremer Vulkan AG (D), AT&T GIS Norway (N), SINTEF Informatics (N), TNO Building and Construction Research (NL), Kockums Computer Systems AB (S), RDM (NL), Det Norske Veritas (N), Technische Universität Berlin (D).

The project started in July 1992, and ends in October 1995.

The project is funded by the Commission of the European Communities, the Norwegian and Swedish national research councils, and the participating partners.

4.2 Problem Description

The life-cycle of a ship can be regarded to consist of the following phases:

- Bid preparation
- Pre-design and design
- Production planning
- Production
- Operation, maintenance and inspection
- Disposal

In each of these phases, there are many participants (companies) involved. For example, during design, a yard may use several subcontractors. Also, the participants have different responsibilities, e.g. a classification society will verify if the design done by the yard is according to the class-rules.

Since several participants are involved, the data will be stored in different locations. In addition, there are many different applications involved, depending on the tasks to be done. For example, a yard will use a specialized CAD/CAM system for the design, while the classification society will use a specialized application for class approval. Also, the participants involved may operate on a different level of detail, implying that there is not a one-to-one mapping between the data-requirements of the applications.

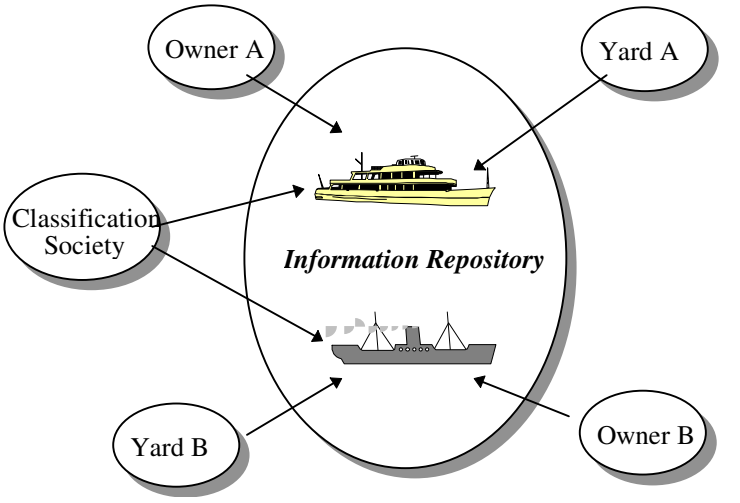

Figure 6: Company View

As shown in figure 6, each company may be involved in several projects (ships) simultaneously, and there are several participants involved in the same project. (The Information Repository is a term used to indicate an abstract view on the information to which the Repository Services may give access.)

It is important that information access is made as transparently as possible, while at the same time security aspects are taken care of.

4.3 Use of the Information Access and Interchange facility

In the project, a scenario for a demonstration to take place at the end of the project have been defined. According to this, it shall be possible to transfer data from Design Yard to Classification Society and back again. Figure 7 indicates two of the applications being used in the industry today, and can be regarded as an example on what this could look like. (Tribon is a CAD/CAM-system developed by Kockums Computer Systems, and is, in this example, used for the design of a ship. Pilot is a system used for class-approval, developed by Det Norske Veritas.)

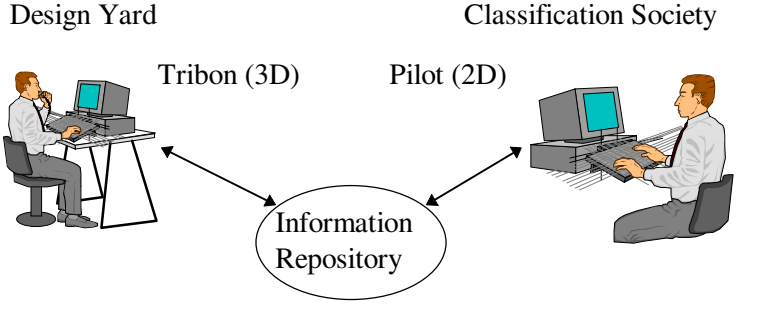

Figure 7: End-user View

Figure 8 shows how the applications involved has been integrated via the Repository Services.

Another aspect, which the Repository Services do not cover, is the mapping process between the native formats of the applications involved. A huge effort has been done within the project to try to establish a proposed standard to be used for the exchange of product model data within the ship industry. The standardization work has been done within the framework of ISO/STEP [ISO]. The Information Servers of the project can then offer to deliver product model data in a neutral format by supporting a subset of this new proposal.

The Information Servers can also be used to give access to documents/reports, e.g. in PostScript-format.

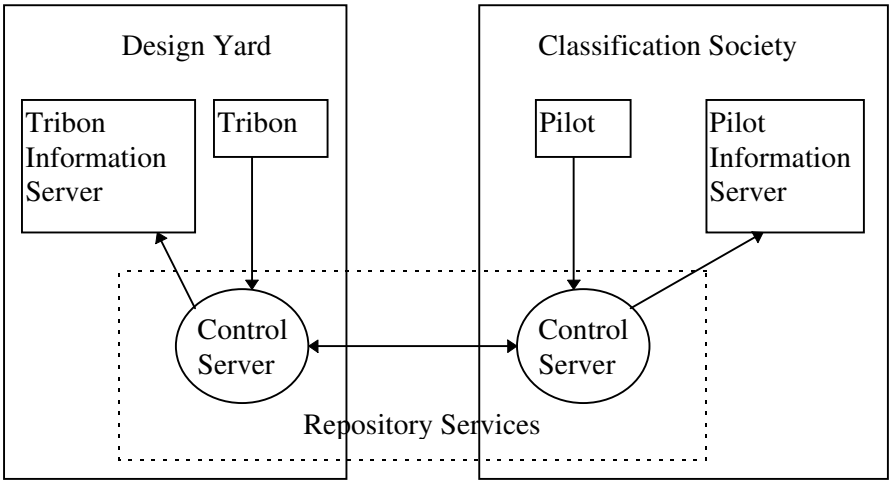

Figure 8: Technical View

Another issue is how to be able to identify information of interest in the Repository Services without having to do to much modifications to existing software. The approach used in MARITIME was to develop a neutral graphical information browser on top of the Repository Services. Using this browser, information in the Repository Services can be identified and selected. In addition, the browser is able to activate applications, e.g. Tribon or Pilot, and forward the selected information to the application. This browser is also used for the maintenance of all information in the Control Server it is connected to. Figure 9 shows how the browser looks like with some information in the Repository.

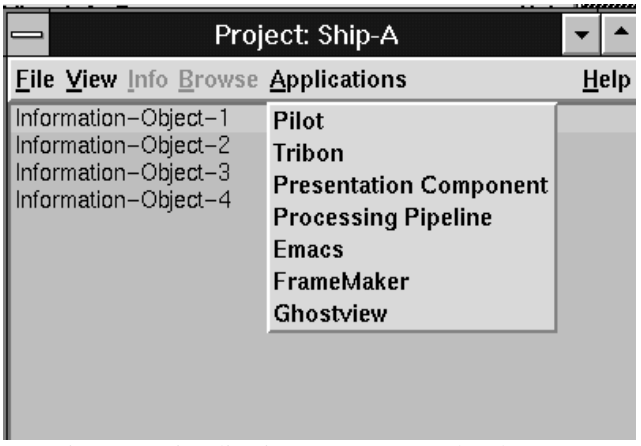

Figure 9: Visualization System (metadata browser)

5. Comparison to other approaches

When we decided to create the Repository Services, it was to our knowledge no technology available that addressed our problem. Apart from that, it was one of the goals of the project to gain experience in using CORBA-technology, hence other approaches were not given any serious consideration within MARITIME. Later, approaches have emerged that address much of the same problems.

In a distributed database system (DDBS) the data is managed by the DDBS and access is given to information through the DDBS. The Repository Services can be looked upon as a metadatabase with possibilities to access the actual data wherever they may reside. In the Repository Services the management of the data is done locally, maybe in a local DBS, and access can be given to only parts of the data. It is no problem to distinguish the internal information from what is provided to others using the Repository Services, this can also be achieved using DDBS using a federated approach, that is for each local DBS export schemas that are part of a federated schema. By not using DDBS we avoid the considering different schemas, the Repository Service is totally independent of native representation including schemas. Second, if the information is to be stored in only one native format, the approach with Information Servers is useful when other applications want access to the information in other formats. In a DDBS the information had to be stored in all possible formats if applications should be able to use them.

Information replication is another approach. A product with gaining popularity using this approach in Lotus Notes. Lotus Notes is a workflow facility supporting information access and interchange in a distributed environment. Lotus Notes replicates the information when making it available in order to reduce access time. It is a trade-off between replication and access time, a trade-off which changes with the amount of information, the liability of the information to change, and other factors. In some cases the best solution is to replicate information, in others not. In our case replication was no alternative due to the large amount of information and its rapidly changing nature. There may be other problems with using Lotus Notes regarding the possibility for native systems, as the CAD-system used in MARITIME, to access the information in Lotus Notes in a transparent fashion. Lotus Notes is however addressing a different market than the Repository Services is meant to address, being a higher level system with end-user tools. The Repository Services can in this context more be regarded as a component that can be used from Lotus Notes for the information access and interchange.

WWW is a rapidly growing, flexible facility for information access and interchange. Using WWW you can get access to information in different locations using HTML codes [Conn]. Location transparency is one of the major features of WWW. Usually people use a browser like Mosaic [NCSA94] as client to access different Web-servers on Internet. These servers may when accessed trigger scripts that may invoke programs including conversion facilities (like the Information Servers). When the data is returned to the clients, they may invoke programs to handle the data. Users of WWW will recognize the activation of GhostView (or another previewer) when receiving PostScript documents as an instantiation of this mechanism. To our knowledge, there is however problems with grouping information into Domains and Contexts and giving access to only a subset of users using WWW. That was one requirement when creating the Repository Services. With Java [Sun95], the recent development from Sun Microsystems, a greater flexibility is given, and it is possible that this approach can be used.

As a conclusion, of the three approaches mentioned, WWW is the approach most similar to the Repository Services and they are partly overlapping. A further comparison between the

Repository Service and WWW can be fruitful, identifying the weak and strong sides of the two technologies.

6. Conclusion and future work

The strength of the Repository Services is the handling of constantly changing information that is part of a larger native context. Using the Repository Services the integration is on the application level letting each application use its preferred format rather than on the data level representing the data in a common, neutral format and making that available.

Further development of the Repository Services includes implementation of both synchronous and asynchronous communication between Control Servers, using alternative transportation like ISDN, further development of the access control, and performance tuning. As more CORBAServices become available, the usage of these could be considered. The security service could take care of some of the access control, the persistence service could take care of much of what OBST is used for presently, and the event-handling service could be used implementing flexibility of what to do when the detail information changes. Threads could be used to handle multi-user problems, that is handling several requests simultaneously.

The Repository Services is a general approach to information access and interchange, and can as such be applicable in other domains than ship design, such as GIS (Geographical Information Systems), CAD/CAM, CIME, Geoscience (Oil & Gas Exploration and Production), and others. It would be useful to apply it within different domains to prove its generality.

There is also work carried out within the ANSA-project [ANSA95] that addresses the combination of CORBA objects and WWW. They are building tools to impose a distributed object model onto WWW, hence making WWW information accessible as objects in a CORBA environment. It could be interesting to use this with the Repository Service letting Information Objects be both objects created by the Information Servers as today, and objects accessed using WWW. As pointed out in the previous chapter, WWW is the approach discussed that is most similar to the Repository Service, and an integration of these two could prove very useful.

Bibliography

[ANSA95] ANSA
“Distributed Objects and the World Wide Web”
http://www.ansa.co.uk/phase3-activities/wwwCorba.html

[Berr95] Arne-Jørgen Berre, Bjørn Hjelle, David Skogan
"GISIP - An Object-Oriented Software Integration Platform for Geographical Information Systems" Proceedings of SCANGIS'95 - Scandinavian Conference on Geographical Information Systems, Trondheim, June 1995.

[Ber94a] Arne-Jørgen Berre and Magnus Rygh “GEOOSIP - An Object-Oriented Integration Framework for GeoScience Applications” New Trends in Geoscience computing, NPF, November 1994, Stavanger, Norway

[Ber94b] Arne J. Berre and Frode Høgberg, Object-Oriented Design and Implementation of an EXPRESS-based Product Model Database - On Object-Oriented Extensions to EXPRESS, EUG'94, EXPRESS User Group Conference, October 12-13 1994, Greenville, South Carolina, USA

[Berr93] Arne-Jorgen Berre "An Object-Oriented Framework for Systems Integration and Interoperability" PhD-thesis, University of Trondheim, Norwegian Institute of Technology, 1993, 370 pages

[Berr92] Arne-Jørgen Berre
"COOP - An Object-Oriented Framework for Systems Integration"
Proceedings of second international conference on Systems Integration, June 1992, Morristown, N.J. , USA

[Berr92a] Arne-Jorgen Berre "Experiences from Systems Integration through an Object-Oriented Software Bus" In Proc. of TOOLS-Europe'92, Tool2-7, Prentice Hall, pages 33-46, April 1992, Dortmund, Germany

[Brod92] Michael Brodie, Stefano Ceri "On Intelligent and Cooperative Information Systems" International Journal of Intelligent and Cooperative Information Systems 1(2):1-35, September 1992

[Conn] Daniel Connolly
"HyperText Markup Language (HTML)"
http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

[FZI93] Jochen Alt, Eduardo Casais, Bernhard Schiefer, Sameer Sirdehpande, Dietmar Theobald
"The OBST Tutorial", FZI.049.2, Forschungszentrum Informatik (FZI), Haid-und-Neu-Str. 10-14, D-76131 Karlsruhe, Germany

[Hauk93] Steinar Haukøy, Werner de Bruijn, Roland Oehlmann, Thomas Koch, Mart Käsper
D1401 - MARITIME Reference Architecture,
ESPRIT III 6041 MARITIME, February, 1993

[Hauk94] Steinar Haukøy, Jan Øyvind Aagedal, Knut Hasund, Roland Oehlmann, Thomas Koch, Mart Käsper, Are F. Tjønn
D1402 - Specification and Prototyping of MARITIME Repository Services,
ESPRIT III 6041 MARITIME, April, 1994

[Hauk95a] Steinar Haukøy
Information Access in a Distributed Environment (SINTEF in MARITIME),
SINTEF, April, 1995

[Hauk95b] Steinar Haukøy, Jan Øyvind Aagedal, Knut Hasund
D3501-A - Specification of the MARITIME Wide Area Network Services,
ESPRIT III 6041 MARITIME, May, 1995

[Iona94] Iona Technologies Ltd, Ireland
"Orbix Programmer's Guide", Version 1.3

[ISO] ISO 10303, Part 215, 216, 217, 218

[KP88] Glenn E. Krasner, Stephen T. Pope
“A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80”, Journal of Object-Oriented Programming, 1(3):26-49, August 1988

[NCSA94] National Center for Supercomputing Applications
“NCSA Mosaic for X Documentation”
http://www.ncsa.uiuc.edu/SDG/Software/XMosaic/mosaic-docs.html

[Nils90] E.G. Nilsson, E. Nordhagen, G. Oftedal
"Aspects of Systems Integration", Proceedings of first international conference on Systems Integration, April 1990, Morristown, N.J. , USA

[OMG94] Object Management Group
“CORBA: Common Object Request Broker Architecture and Specification”, OMG Document 94-1-1

[OMG95a] Object Management Group
“Common Facilities Architecture”, Revision 4.0, OMG Document 95-1-2

[OMG95b] Object Management Group
“CORBAservices: Common Object Services Specification”, Revised Edition, OMG Document 95-3-31

[Sun95] Sun Microsystems
“The Java Language - A White Paper”
http://java.sun.com/whitePaper/javawhitepaper_1.html

[Web] Virginia Thatcher and Alexander McQueen
"The New Webster Encyclopedic Dictionary"
Consolidated Book Publishers, 1980, 972 pages