A PROPOSED OPEN INFRASTRUCTURE FOR CONSTRUCTION PROJECT DOCUMENT SHARING

SUBMITTED: July 1998
REVISED: November 1998
PUBLISHED: December 1998 at http://itcon.org/1998/2/
EDITOR: Z. Turk

Dr. Yacine Rezgui, Lecturer
Information Technology Institute, University of Salford
email:y.rezgui@iti.salford.ac.uk

Dr. Grahame Cooper, Senior Lecturer
Information Technology Institute, University of Salford
email: g.s.cooper@iti.salford.ac.uk

SUMMARY*: The paper gives a comprehensive overview of the European Esprit Condor project. The project provides a migration path from document-based to model-based approaches to information representation and structuring. The paper, first, presents the approach that is adopted to describe the business processes and document management practices of the project end-users, and also to capture the requirements of the Condor system. The chosen approach made use of the IDEF0 and UML (Unified Modelling Language) methodologies. A Strategy towards the integration of the Electronic Document Management (EDM) systems used within the project is then given. The paper also includes a comprehensive description of the overall modelling infrastructure that supports the proposed system. Finally, a description of the prototype V1, that implements the inter-working services between heterogeneous EDM solutions, is given. The project is ongoing and supported by a user interest group, which involves representatives from a variety of non-construction industry companies all over Europe.*

KEYWORDS: *Electronic Document Management, Object Modelling, Object Inter-working, Computer Integrated Construction.*

1. INTRODUCTION

Industrial processes are characterised by the intensive use and production of knowledge and information. Most of this knowledge and information, even when produced using computers, is still exchanged and conveyed on a paper-based medium. This can be viewed as a limitation to potential opportunities for process innovation as information is constrained by what can be achieved using paper. In fact, the format and layout of most documents was shaped before the proliferation of personal computers, and has not yet changed in response to more recent IT developments.

In the building process diverse and complex information flows between actors. This information is mainly conveyed using documents. The latter need to be highly consistent in order to provide a reliable basis for actors to perform their design, construction and maintenance activities. Document management has become a crucial issue within modern construction companies. The various solutions proposed by some software vendors have been revealed to be unsatisfactory, to a point where many leading construction organisations, with an advanced IT department, have undertaken the development of their own tools and solutions to support the production and maintenance of project documents. Even though such proprietary tools provide many helpful facilities, including support for document storage, retrieval, versioning and approval, they don't handle any semantics of the information being processed and therefore remain limited in their support of the end-user. In fact, construction project data and documentation (including full specification documents) constitute two fragmented information sectors where compatibility and interoperability are mostly needed. Moving these pseudo-sectors closer together to support construction project documentation as part of the life-cycle of the building product is becoming an actual and urgent topic for standard bodies and industry alike.

The background to electronic document management can be roughly summarised according to the two following approaches:

- The integrated document management approach (Bjork et al., 1993) in which documents are treated as black-boxes, and the aim of computer-support is to enable easy document storage, retrieval (using reference information), versioning and approval. This approach is becoming best practice due to the proliferation of EDM systems.
- The model-based approach (Rezgui and Debras, 1995) in which the information traditionally contained in drawings and text documents is described and represented through an object model, and is contained in integrated databases. These databases are then used as a basis for the production and authoring of project documents.

However, a literature review conducted in the area of electronic document management revealed that little attention was given to the issue of inter-working between dissimilar electronic document management solutions. Furthermore, it is not possible today to share documents, managed by dissimilar EDM systems, on a construction project. In order to pioneer a solution, a consortium driven by construction end-users (Kvaerner Construction from the UK, OTH from France, and JMBygg from Sweden) is currently collaborating within the frame of the European Condor project (ESPRIT 23 105). The consortium is aware of the fact that it is not possible to develop a unique document management solution for all the project partners, because of their investment in proprietary systems. The Consortium is promoting the use of the various existing systems, as described later in the paper, by the development of extensions which comply with the proposed approach. These extensions will enable all systems to communicate, and have access to advanced functionality, through a dedicated Application Programming Interface (API) developed within the project.

The paper presents a means of bridging the gap between the traditional document-centred and the proposed (model-based) approaches to project information structuring and representation. Robust models supporting this transition are proposed, together with a prototype implementation which demonstrates a pro-active use of document management and information management techniques in a collaborative multi-actor environment.

2. DOCUMENT MANAGEMENT PRACTICES IN THE CONSTRUCTION INDUSTRY

The business processes and the document management practices of the three end-users involved in the project have been analyzed (Rezgui and Karstila, 1998a). This analysis was then generalized to the whole industry. The latter made use of the Infomate model (Bjork, 1997a). The analysis revealed that present document management practices rely to a large extent on manual methods, although the production of documents is mainly done using computers. However, in recent years, many leading construction companies have enforced the use of Electronic Document Management (EDM) systems to support the effective and consistent management of the entire documentation produced and used in a project. These systems are now introduced in the construction industry much like Computer-Aided Design (CAD) was introduced about two decades ago. A brief description of the EDM systems used by the three project end-users is given below:

- SGT (Système de Gestion de données Techniques): this EDM solution is developed by Derbi (a subsidiary of OTH). SGT provides a set of functionality for storing, archiving and exchanging documents, including drawings and written documents, in a structured manner. This EDM system also offers a number of advanced services, including a function to co-ordinate the approval process of documents; a change request management service; an advanced construction specific financial tool; and a subsystem (GPP) that is dedicated to the production and management of drawings in a multi-actor environment.
- Eureka!Filebase: this EDM system is developed by Carasoft (a Swedish software house). It is presently widely used by JMBygg on most of their construction projects. Eureka!Filebase enables the storing and retrieval of documents. It supports document history, and administers the maintenance of file revisions. It bears similarities with the SGT EDM system.
- Kvaerner Hub: Kvaerner Construction have developed a document controller to set up an electronic hub for all parties involved in a construction project to manipulate, pass and distribute documentation. The

document hub provides a collection of services. For instance, it maintains some form of drawing control that keeps track of drawing versions, complemented with various functions, including, approval, receipt acknowledgement, and document distribution.

The overall analysis of the current document management practices within the three construction companies involved in the project revealed the following limitations, inherent to their current system and work methods:

- Every partner within the project must use the same EDM system on a project to be able to access and share documents.
- The document's semantics and internal structuring is not controlled by the EDM system. Documents are handled as black-boxes.
- The EDM system does not support document cross-referencing or semantic linking.
- Security is always an issue. It is not easy to implement as for printed documents. EDM systems require improved user authentication and document protection.
- The EDM system is not integrated with proprietary and commercial applications used within the company (e.g. CAD applications and word processors).
- Most end-users in the construction industry are not computer literate. EDM systems lacking user-friendliness, or used in a maladapted environment (e.g. network communication problems) discourage the user from using the EDM system.

Despite some minor differences, the analyzed EDM systems offer similarities across organizational boundaries. An example of this exist in document categorization, archiving, retrieval, versioning and approval. The services provided by these EDM systems are being used as a basis to define the Application Programming Interface that will support inter-working between these legacy applications, as described in section 4.

3. CAPTURING THE REQUIREMENTS OF THE PROPOSED SYSTEM

The choice of an appropriate methodology that is flexible enough to adapt to constant business environment change, and technology evolution, is essential for the success of a research project. We have chosen to adopt an incremental and iterative approach to develop the proposed solution (Figure 1). The software product is the result of a series of iterations. The important aspect is that by adopting this approach, a certain degree of incertainty in the specification of the users requirements is tolerated and supported. In fact, change is anticipated at any stage of the software product lifecycle. In addition, potential risks are identified and prioritized early in the lifecycle, and are at the core of each iteration leading to the final solution.

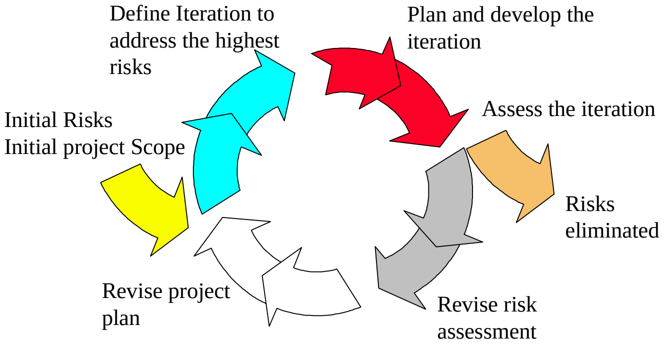

FIG. 1: Iterative and Incremental Development Approach (UML, 1997).

It was decided, given the timeframe of the project, that three iterations (the first being presented in the paper) will be conducted. Each iteration would be used to assess and validate the resulting prototype, and address the potential risks in relation to the implementation of the proposed solution. The iterative and incremental approach was provided by the Rational Objectory Process (Rational Software Corporation), and supported by the Rational Rose case tool (UML, 1997). The Unified Modelling Language (UML) methodology (UML, 1997) was chosen to specify, visualize, and document the underlying artifacts of the proposed system.

A hybrid approach making use of the IDEF0 methodology and the Use Case approach has been used to capture the requirements of the system. It was found that both approaches were complementary and necessary for the understanding and specification of the proposed system. IDEF0 is used to define a high level process activity model describing the document management practices taking place within the design and construction process. This is based on the analysis and abstraction of the business processes and document management practices of the project end-users. A Use Case diagram, at a lower level, is proposed and detailed to describe the system in use, as indicated in Figure 2.

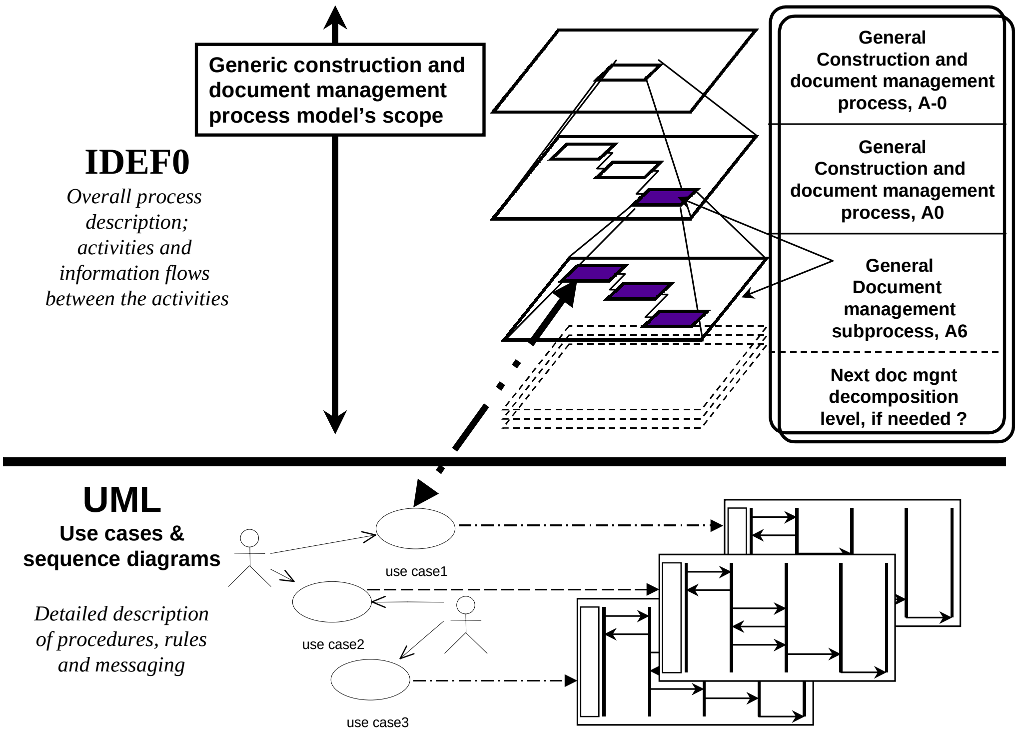

FIG. 2: The hybrid approach adopted to capture the requirements of the proposed system.

A Use Case describes a specific way of using a system. The identified Use Cases describe the functionality performed by the system as a result of a request from an Actor (physical person or other system interacting with the system). They provide a way to capture the requirements about the system, communicate with the end-users and domain experts, and test the system. The main Use Cases identified within the proposed system include:

- Add information to distributed system: this use case describes the process of adding a document or a piece of information to the registry. The latter is implemented via a centralized or distributed database holding references of all the documents shared in a project. Once referenced in the registry, the document is then accessible to all authorized users of the system.
- Browse information: this use case describes the process of browsing the information held by the various information providers registered with the proposed system. The shared information (e.g. documents) is referenced in the registry.
- Retrieve information: this use case describes the process of retrieving information held by an information provider via the registry.

We have then described in detail all the transactions involved in implementing every single identified use case. This was done via interaction / sequence diagrams. Figure 3 illustrates how IDEF0 is used in conjunction with use cases and sequence diagrams from UML. The IDEF0 document activity model presents the scope, context and construction environment in which the proposed system is used (Rezgui and Karstila, 1998a). It details the "Do building document management" process, and includes the following activities:

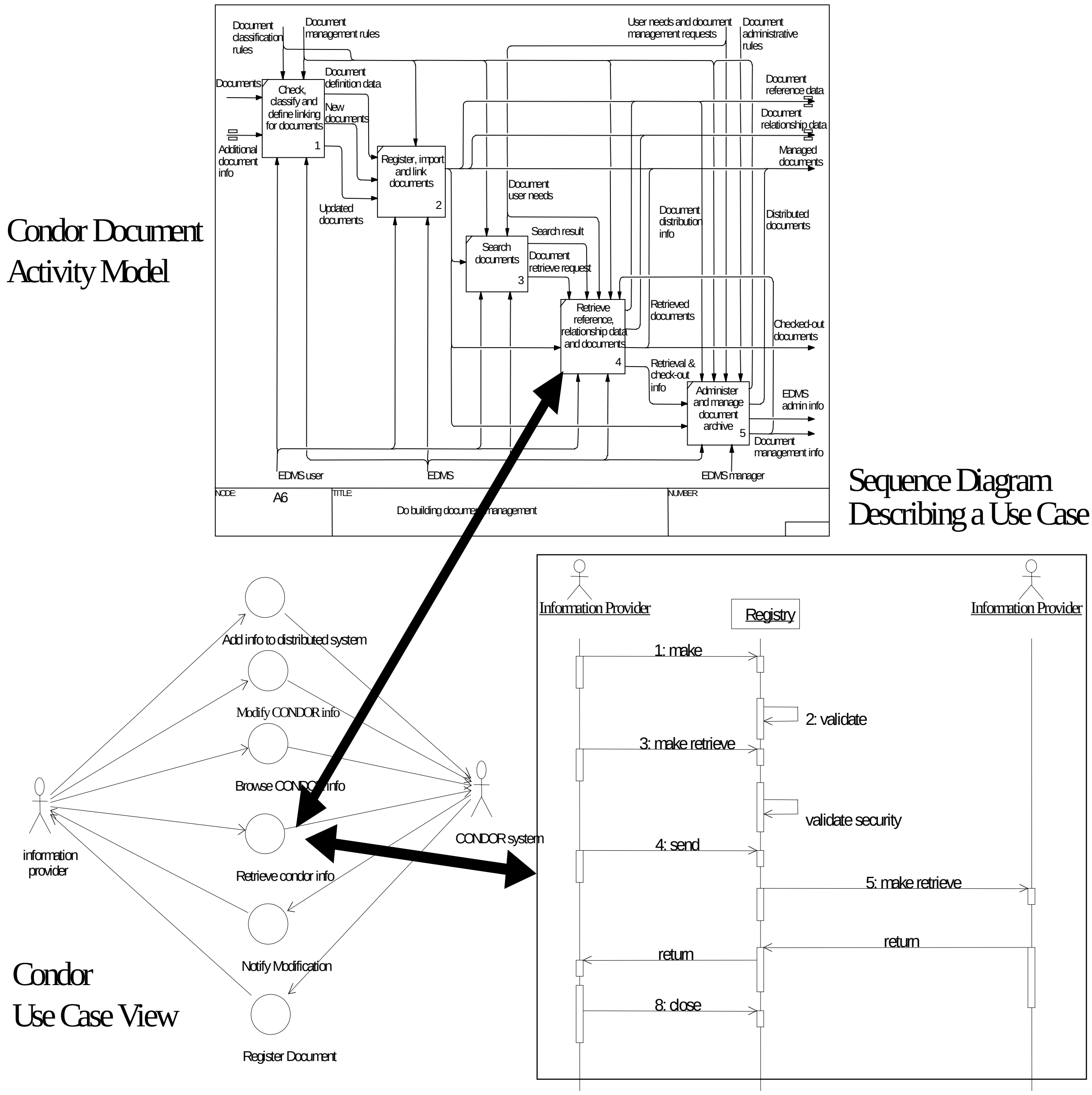

FIG. 3: Understanding and capturing the requirements of the proposed system.

- *Check, classify and define linking for documents:* this activity represents the authoring process of a document. It is supported by a document processor (e.g. CAD package for drawings, or word processor for written documents). The document being authored might not be visible, at this stage, by the EDM system.
- *Register, import and link document:* this activity represents the registration process of a document. Once registered, the document becomes accessible to authorized users.
- *Search document:* this activity describes the activity of searching for specific documents managed by the EDM system.
- *Retrieve reference, relationship data and document:* this activity describes the process of downloading a document using the EDM system. Figure 3 illustrates the fact that this activity is described by a UML use case and sequence diagram. The interaction diagram in figure 3 describes the process of retrieving information from an information provider. The user passes the ID of the document he/she wants to

retrieve. The registry requests this document from the information provider that holds the document. The registry passes back the requested document to the person who originated the request.

- *Administer and manage document archive:* this activity represents on the one hand, the pure administrative activities of the EDM System, and on the other, the management services that the document management system provides for the users. The administrative activities include defining the classification templates and the documentation structures, performing backups from the document archive, producing statistics from the system in use. The document management services may include the following activities: automatic notification of users for new imported documents in the system, automatic distribution of imported documents (according to document reference / distribution information), and copying, moving, deleting of documents in the system.

4. THE PROJECT VISION AND RESULTING SYSTEM ARCHITECTURE

We have identified, from the analysis of the end-users requirements, a strong need to enable dissimilar electronic document management solutions to co-exist within the same project. The need to access and cross-reference any document-based information regardless of its form, format, and location was also strongly expressed by the end-users. Therefore, the Consortium has chosen to promote the use of the various existing legacy systems as opposed to developing a new EDM solution. These EDM systems will co-exist and co-operate by the development of extensions which comply with the proposed approach. These extensions will enable all systems on the one hand, to communicate and inter-work, and on the other, to have access to advanced functionality, through a dedicated Application Programming Interface developed within the project.

Furthermore, the emphasis of the approach taken here is one of inter-working between dissimilar systems rather than sharing of common data structures. This is in line with current developments in software technology, which are based on object-oriented client-server approaches (Cooper, 1995). The approach also recognizes the need to allow the vendors of specific document management systems to make their own decisions on internal data structures and local functionality. One might argue that a pure component-based approach would rather tackle finer levels of granularity of information. The focus would then be on the document rather than the EDM system that manages it. Furthermore, the document would become a self-managing component with a set of public interfaces that provide similar services to the existing EDM systems. That is, in fact, supported by the proposed approach. It is part of its strategy to migrate to emerging component-based technology.

The project system architecture and functionality is defined in a manner that is independent of any particular object inter-working technology or standards. However, a prototype which makes use of the OMG's CORBA (OMG, 1996) standard is being developed within the second iteration of the project.

The core of the architecture is the API. It defines the services to allow on the one hand, inter-working between the project's legacy EDM systems, and on the other, semantic linking between different documents and between documents and other information objects. The precise services that are provided are determined from the results of the business process analysis conducted by the end-users.

Figure 4 shows the basic project architecture. The major components of the architecture are:

- the *Integration Services* (implemented as a class library in the demonstrator): these include the inter-working services and semantic linking services;
- the *API*, which defines the interface to the integration services;
- the *Adaptors*, which provide the mapping between the API and each of the document and object management systems to be integrated.

The architecture is designed to keep the size of the adaptors as small as possible, and this will be facilitated by means of a third set of API services: the Mapping Services. The latter will provide interfaces to a number of component objects that will aid in the creation of adaptors, thereby minimizing the code that will be required to implement adaptors.

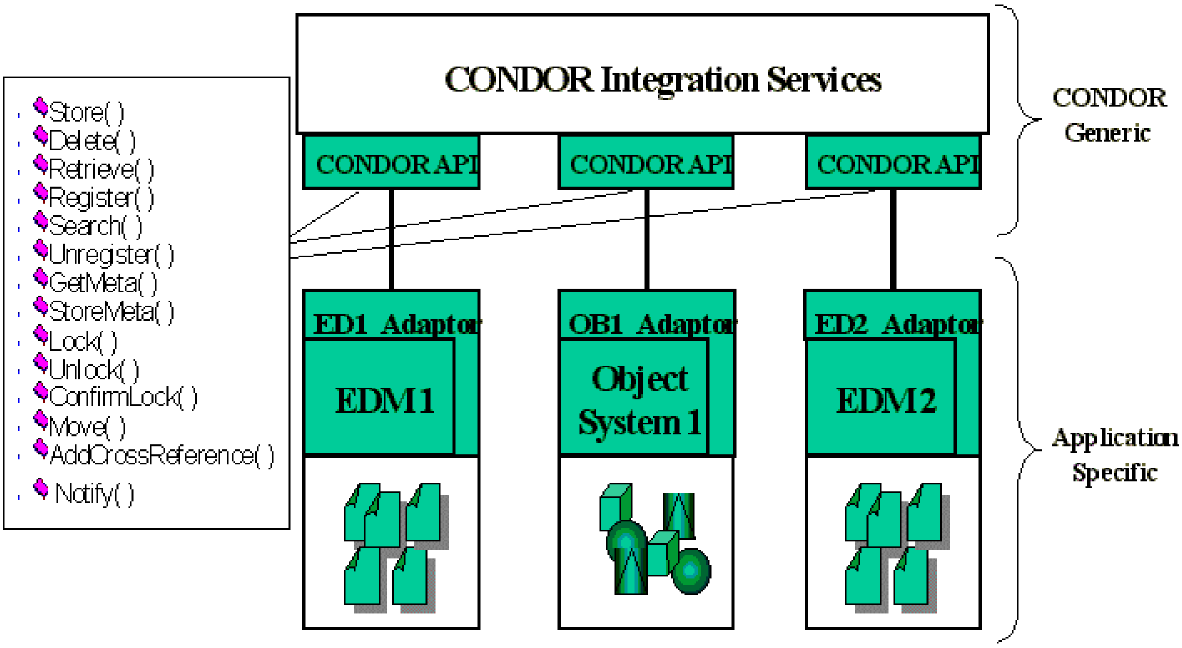

FIG. 4: The Proposed System Architecture.

The approach taken will ensure that the proposed generic functionality will be implemented only once in the system, whilst the functionality that is specific to a particular document or object management system will be implemented separately from the system. It is worth mentioning that much of the functionality of the proposed API can be provided directly by making use of CORBA horizontal services.

5. THE MODELING INFRASTRUCTURE

The Unified Modelling Language was chosen for the development of the models in the Rational Rose CASE tool. This language is rapidly being established as a de facto standard for object-oriented modelling, and is well supported by the Rational Rose CASE Tool. UML is developed primarily from two of the most popular modelling formalisms for object-oriented modelling, OMT (Rumbaugh et al., 1991) and Booch (Booch, 1994), and has been adopted as an international standard within the OMG (Object Management Group) that develops the Common Object Request Broker Architecture (OMG, 1996). The models shown in this paper are expressed in the UML.

A modelling infrastructure is proposed in order to address the information sharing, application inter-working, and semantic linking objectives of the project. This infrastructure is expressed by means of the UML package concept. The purpose of a Package is to group logically all the inter-related classes. Each identified class has a precise specification that includes attributes and operations that define the semantics of the class. The following packages have been identified and represented in the class diagram in Figure 5 :

- Information management package : it is at the core of the models. It contains all the classes that support information structuring, information versioning, and information cross-referencing. It also defines the inter-working functionality of the project EDM systems, along with the required extensions (adaptors).
- Document models package : it contains other sub-packages as it describes the structuring of the various forms of information used in a construction project. The latter include written document based information, CAD-based information, and muti-media based information.
- Building concepts package : it describes all the generic classes that are needed to describe a building product. These classes are then specialized in the KTH IFCs package to make use of the Industry Foundation Classes developed by the International Alliance for Interoperability.

The packages containing the classes describing the various forms of document-based information are given in Figure 6. The different interpretations (CAD, multimedia, and written documents) are specialized from the Generic Condor Document model.

It is worth mentioning that class diagrams are created to graphically depict the packages and classes needed to support the information sharing and semantic linking requirements expressed in the project objectives. The class diagrams identified in Figure 5 and Figure 6 contain only packages. The main class diagram for a package contains the public classes of the package (classes that communicate with classes in other packages). The relationships needed by the system are determined by examining the proposed use cases and sequence diagrams. The functionality specified in the use cases are carried out by defining the structure and behaviour of every single class contained in a package. The sequence diagrams are updated to show the replacement of messages with operations.

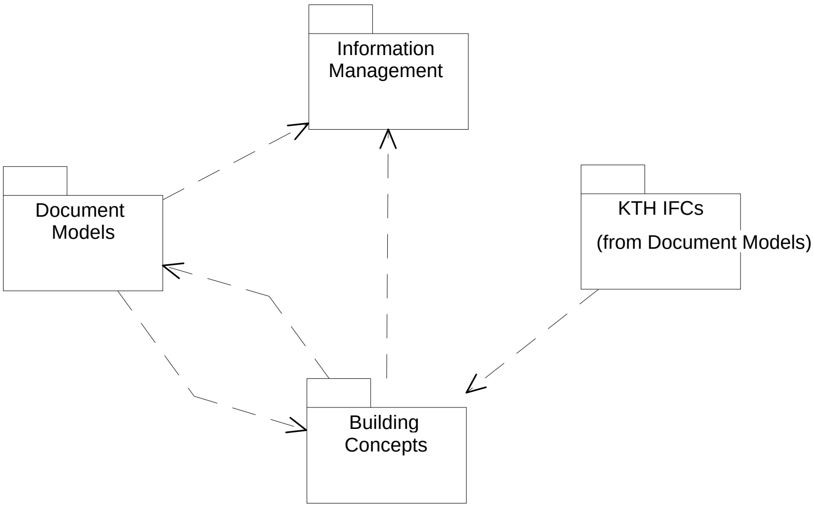

FIG. 5: Information management infrastructure class diagram.

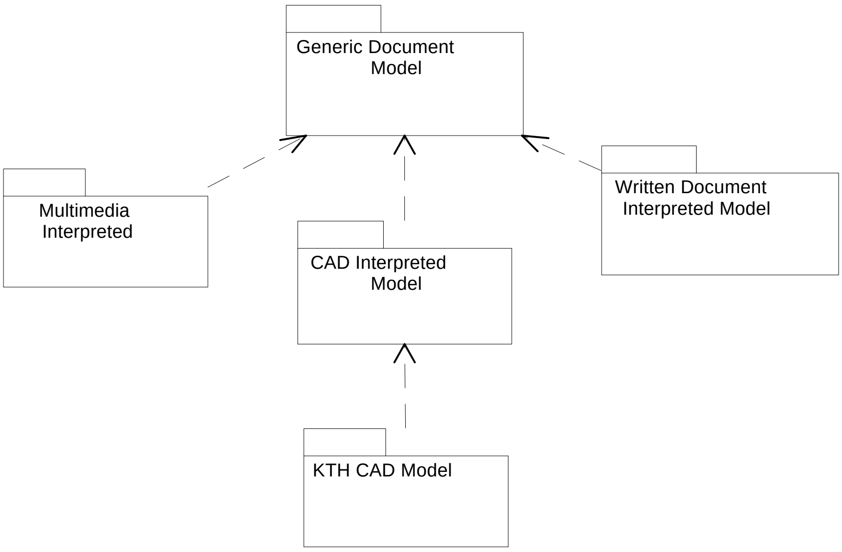

FIG. 6: Document model class diagram.

5.1 The Condor Information Management Model (CIMM)

This model describes how information is managed in the system. This is based on the analysis of the business processes and document management practices of the construction end-users involved in the project. . The model builds on results from the COMMIT project (Rezgui et al., 1998b). The CIMM (Condor Information Management Model) involves the information providers that represent the various EDM systems used in the project; the registry that holds details about all the registered documentation (Information Elements) used in a project; and the adapter that enables a given EDM system to communicate with the registry to invoke various services.

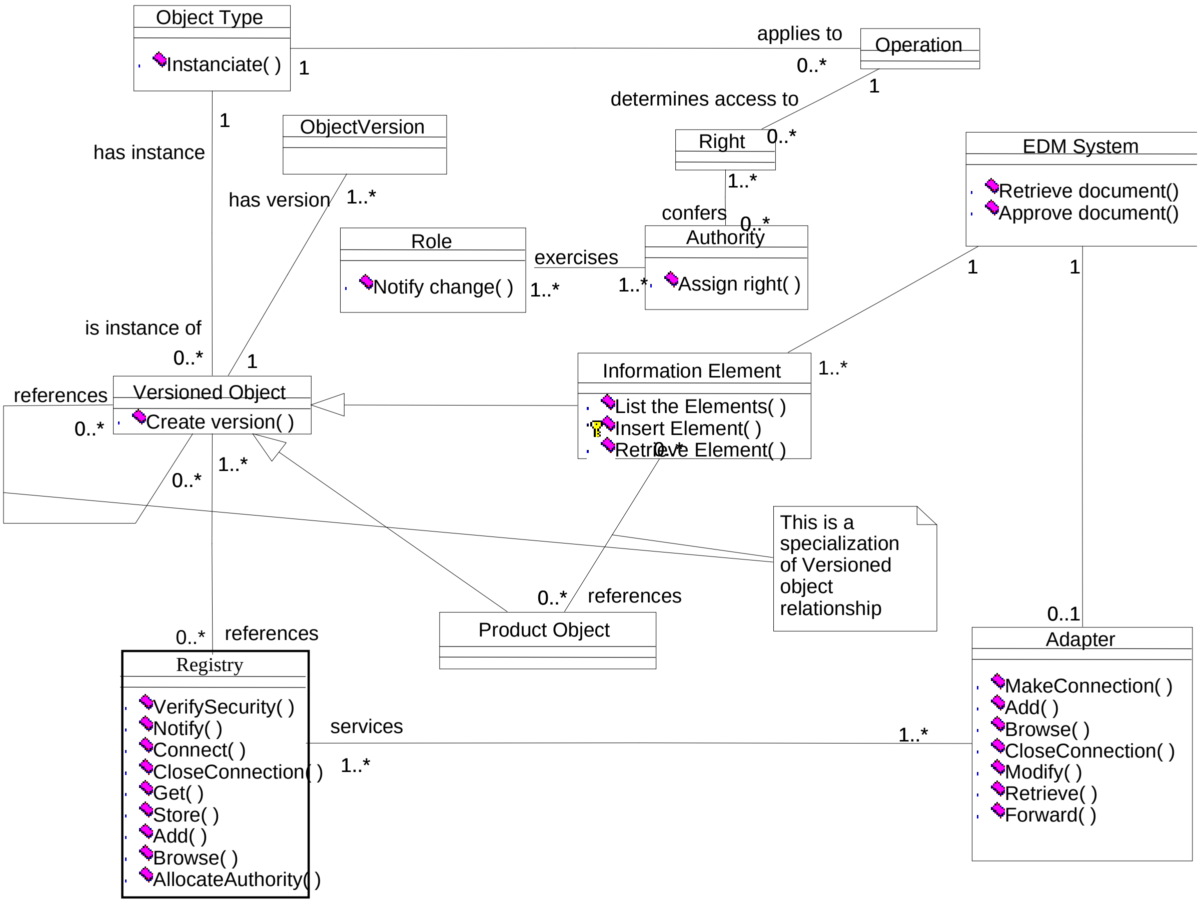

FIG. 7: Condor Information Management Model.

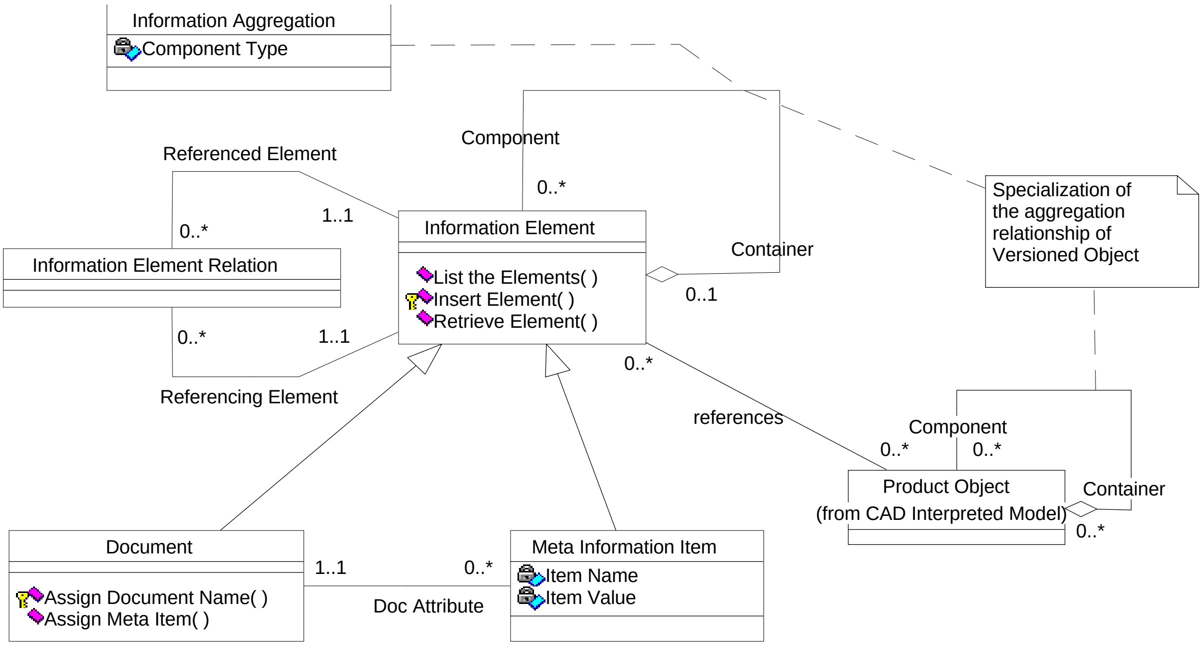

FIG. 8: Condor Generic Document Model.

Figure 7 shows the basis for the handling of information versioning (Rezgui et al., 1998c) and right over information in the CIMM. The concept of Role is introduced, in the sense that all actors participate in a project by means of one or more roles. Thus it is the concept of a role rather than an actor that falls within the scope of the CIMM. Through a role, an actor exercises authority over some parts of the project information, and each Authority is characterized by a number of responsibilities that relate to a particular object (Information Element). In order to discharge those responsibilities, the actor (through the role) needs to have certain rights to perform actions (or Operations) on the object in question. An important principle here is the use of operations on objects to define rights. This is in contrast to the conventional approach of assigning rights as: *Create*, *Read*, *Update* and *Delete*. In the proposed approach, the emphasis is on using real world concepts through the abstractions that may be represented in a true object-oriented model. For example, a project manager might have authority over a task in the project plan. In order to discharge the responsibilities associated with that authority, the project manager would need the rights to assign resources to the task, to move the start date of the task, etc. However, he or she might not have the right to delete the task, or change its duration. Naturally, this would depend on the specific situation, and the rights that the project manager would have over tasks might be different for different tasks. This suggests that rights must ultimately be defined at the instance level (e.g. over a particular task rather than over tasks in general).

In practice, however, there could be a huge number of combinations of rights to be assigned to roles and objects. For this reason, mechanisms are provided in the CIMM to allow default authorities to be defined over all objects of a given type. These authorities would apply in the absence of a specific authority for an instance. Similar mechanisms are being developed in relation to containment, whereby default authorities may be allocated in respect of an object by virtue of its being contained, or owned, by another object.

5.2 The Condor generic document model

The Condor generic document model describes the logical structuring and semantic linking between the various forms of information used in the construction industry. These include written documents, drawings and multi-media objects, and are all defined as documents. An information element might reference, or be referenced, by any number of other information elements. The semantics of any association between information elements is defined within the Information Element Relation class, as indicated in figure 8. An information element might also contain, or be part of, one or more information elements. For instance, a CAD layer might belong to more than one CAD drawings. The Information Aggregation class captures the semantic of this kind of aggregation between information elements. An information element is a versioned object (as indicated in figure 7) that is specialized into a Document, and a Meta Information Item (that constitutes a set of attributes to the document). An information element might reference one or more Product Object(s), this could represent an IAI / Industry Foundation Class, (IAI, 1997).

5.3 The Condor written document interpreted model

The Condor Written Document Interpreted Model describes the logical structuring of a document. It is derived and specialized from the Condor Generic Document Model. A written document may contain one or more sections, as indicated in figure 9. Every section might have a title and a section number. Sections can take several forms, including paragraphs, table of contents, and formulas. One section may reference, or contain, other sections. The mechanism of inserting a new section into an existing one is achieved by invoking the inherited Insert Element method from the Information Element class. It is worth mentioning that this model supports the SGML syntax.

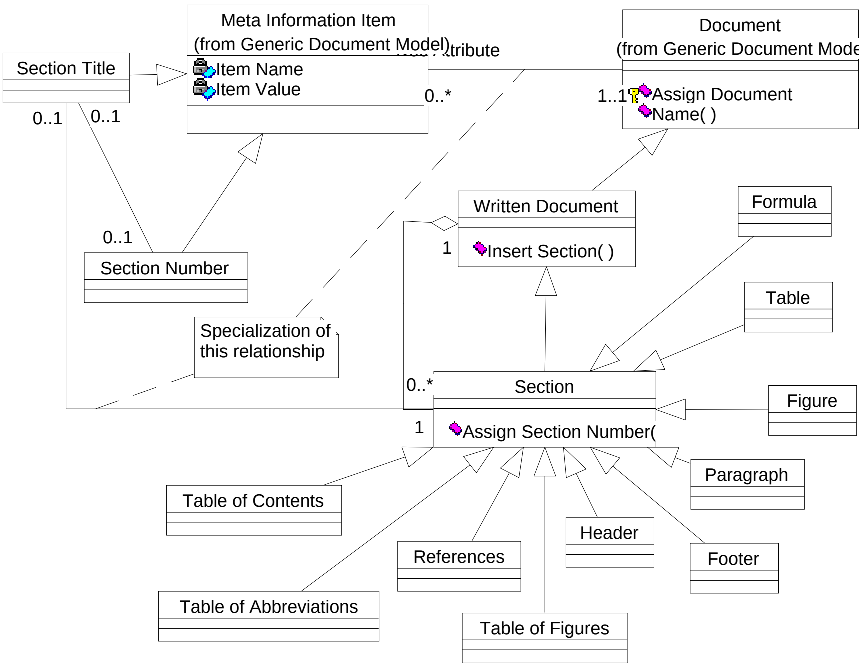

FIG. 9: Condor Written Document Model.

5.4 The Condor CAD interpreted model

The Condor CAD interpreted model provides an interpretation of the Condor generic model to describe the structuring of CAD drawings. The model has been developed from the ISO CAD Layering standard (Bjork et al., 1997b) and the ISO STEP AP 202 application protocol. A CAD drawing is primarily described by means of the layers that constitute it (figure 10). It is worth mentioning that the use of layers facilitates the creation of task-specific documents, e.g. for individual activities on the construction site. Such a method would allow more flexible work task instructions than the traditionally pre-produced documents. A layer template can be provided by the responsible agent, and applied on demand to produce a drawing of the actual part of the building with just the information content needed to perform the specific activity (Tarandi et al., 1997). This new approach to documentation also introduces new issues to be addressed, like the responsibility for the documents produced, and the procedures for revision. In addition, document search functions can use layers to identify documents or document sets concerning any of the properties covered by the layer structure. This may be useful in many situations, such as identifying all drawings that concern a certain building element or construction phase, or drawings that are produced by a certain actor (Tarandi et al., 1997).

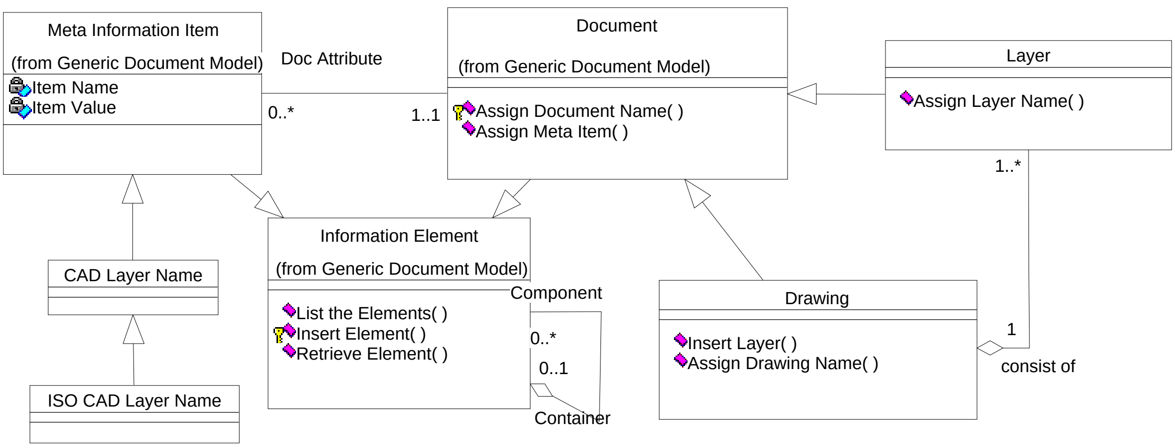

FIG. 10: Condor CAD Interpreted Model.

6. THE DEMONSTRATOR V1

The aim of the prototype V1 is to demonstrate the inter-working services of the proposed system. The demonstrator made use of the following EDM systems:

- NOVA: it is a general purpose, Web based, EDM system developed by the Cap Gemini Group. The users of the NOVA system can add, delete, modify, visualize or download documents stored in the server database. Full-text searches, supporting various commercial document formats (e.g. RTF from Microsoft), are available. The documents are stored in folders. The NOVA system is managed by a system administrator whose responsibility includes registering or deleting users, and also defining their access rights over the stored information.
- SGT: it is a conventional client-server application, running on both the Unix and NT platforms. The SGT NT version is used in the project. The connection between SGT clients and the server is done via dedicated ISDN connections.
- Eureka!Filebase (described in chapter 2).

Two strategies have been considered for implementing the inter-working services between the above mentioned proprietary EDM systems. In the first strategy, called c*entralized registration*, any event (e.g. add, delete, modify) occuring on a Condor compliant EDM server is notified to a central database (*Registry*). The meta-data (e.g. title, abstract, author) describing the shared documents are looked up in this central database, while the documents themselves remain stored on the servers. The second strategy is called *distributed registration.* The central database is replaced by a *Brokerage* mechanism (which is being implemented using CORBA). Shared documents are looked up by broadcasting requests to the connected servers.

The main advantage of centralised registration is that the centralised registry may be seen as a 'cache', speeding up the access to data : when a remote server is unreachable, the 'cache' can be used to display the latest available information obtained from that server. An obvious drawback of this method, however, is that it relies on the duplication of information : the same information may be found in a local server and in the centralized registry. In some cases, the information contained in the centralised registry may become inconsistent with the information contained in the local servers. For instance, a server may fall down just after a new document was stored on it, but before the centralised registry was notified. Conversely, the main advantage of distributed registration is that the retrieved data is always up to date. However, the access to data may be slower than in the case of centralised registration. Since there is no cache, any data request results in the broadcast of a message to all connected servers. Then, the broker must wait for the answers from all the remote servers before being able to send back a result to the server which originated the request.

For prototype V1, we have decided to make use of the two models of registration. The University of Salford developed a centralised registry, while Carasoft (that develops the Eureka!Filbase system) developed a *brokerage registry*.

The resulting architecture is shown in Figure 11. In the centre, the Salford registry represents the centralised registry (developed by, and physically located in, University of Salford), described in the above sections. The Salford Registry is a TCP/IP and socket-based server, written in Java, and using the ObjectStore object-oriented database to store documents' meta-data. To this centralised registry are connected the SGT server (Derbi), two Nova Web servers (Cap Gemini Group) and the *brokerage registry* (Carasoft). The latter implements an instance of the brokerage mechanism described above. The Carasoft brokerage registry acts as an information provider to the Salford centralized registry. It stores the references of documents managed by Eureka!Filebase (and, any other EDM system connected to the Carasoft brokerage registry). The Carasoft brokerage registry is also used to provide a proof of concept for the brokerage mechanism. It is worth mentioning that the second iteration of the project aims at migrating from a centralized to a brokerage registry mechanism.

Whenever a user adds, deletes, or modifies a document on his local database (using Nova1, Nova2, SGT or Eureka), the Salford registry's ObjectStore database is notified, and updated accordingly in order to reflect this modification. When a client user requests the list of all project shared documents, a request is sent by the local server to the centralised registry. The result of this request is sent back to the calling server, and displayed in the appropriate format. Whenever a document is added, deleted, or modified on the SGT server or on one of the

Nova servers, the centralised registry is notified. When documents are added on servers connected to the Carasoft Brokerage registry, the Salford registry is also notified.

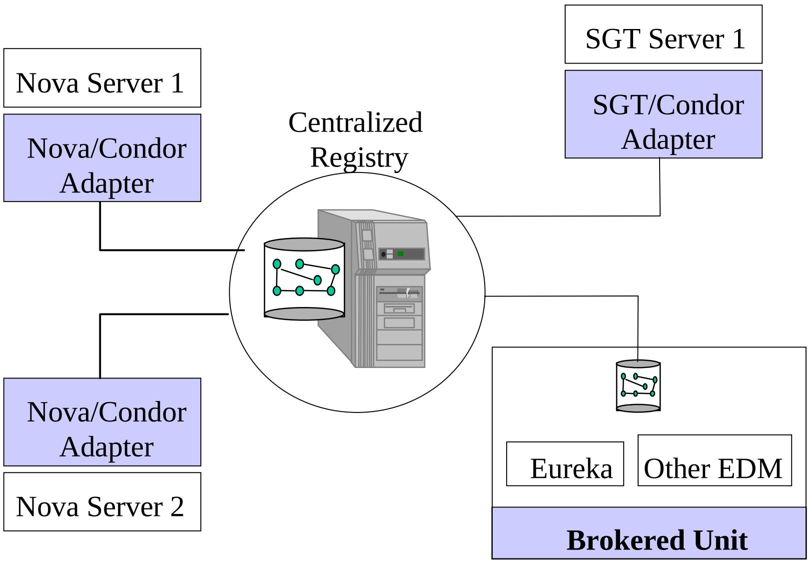

FIG. 11 : Condor Software Architecture V1.

The demonstrator V1 was tested on a real project. The three extended EDM systems were used in a scenario that involved the three construction end-users, acting from Sweden, France and UK. The shared project documents stored locally by the SGT, NOVA and EUREKA servers, were made accessible transparently via the registry physically located in Salford (UK). All interactions between the EDM servers and the registry were handled via TCP/IP and socket-based communications. However, the latter required the development of a specific protocol to support this type of communication. The prototype V2 is being developed using Corba 2.0. This will provide many advantages, including freeing the programmers from several difficult and error prone programming tasks.

7. CONCLUSION

This paper presented the European Esprit Condor project. An overview of the modeling infrastructure supporting the system is given. The presented models are being refined in the light of experience from implementation work. The prototype V1, supporting the inter-working services, is presented. The prototype V2 that demonstrates the semantic linking and cross-referencing services is currently being developed. This will be followed by the last prototype (V3) that will hopefully validate the overall approach taken in the project.

The main problem that the consortium is now trying to overcome is the change of culture. The organisation and processes of construction projects must change in order to deliver the expected benefits. Therefore, the construction project lifecycle, from concept development to demolition and recycling, need to be examined and re-engineered. In order to devise an effective strategy for the deployment of the proposed solution within the three end-users involved in the project, an approach based on surveys and data collection was conducted. The latter made use of in depth interviews and mail questionnaires. The interviews were conducted with key players within each company involved in the project. The initial findings from this survey suggest that reengineering implementation is a very complex process that involves a variety of human, cultural and organizational factors (Vakola et al., 1998). The Business Process Re-engineering research is still ongoing and aims at defining the appropriate process redesign strategy, within each company involved in the project, to implement effectively the proposed solution.

The project is ongoing and supported by a user interest group, which involves representatives from a variety of non-construction industry companies. It is hoped that the latter will help to ensure that the results of the project are sufficiently generic to be transferred to industries other than construction, and to assist in the wider dissemination of results. It is expected that some of these companies will be interested in exploiting the project results within their own organizations

8. ACKNOWLEDGEMENTS

The authors would like to thank Bo-Christer Bjork, Vaino Tarandi and Kurt Lownertz (from University of KTH), Kari Karstila (from EuroSTEP), Jean-Yves Magadur (from Cap Gemini Innovation), Marc Bourdeau and Jean-Christophe Escudié (from CSTB), and Duncan Rose and Maria Vakola (from University of Salford) for their contribution to the research. The authors would also like to thank all the other members of the Condor consortium (JM Bygg, Kvaerner Construction and Derbi) and wish to acknowledge the financial support of the European Community under the ESPRIT program (ESPRIT 23105).

9. REFERENCES

Bjork B-C., Huovila P. and Hult S. (1993). Integrated construction project document management (ICPDM*), In Advanced Technologies - architecture - planning - civil engineering, proceedings of the EuropIA'93 conference* (Behesti M. and Zreik K., editors), June 21-24, Elsevier, Delft, Holland, 135-146.

Bjork B-C. (1997a). INFOMATE: A Framework for Discussing Information Technology Applications in Construction, *In: proceedings of the CIB W78 conference on IT Support for Construction Process Re-engineering*, Cairns.

Bjork B-C., Lownertz K. and Kiviniemi A. (1997b). ISO DIS 13567 – The Proposed International Standard for Structuring Layers in Computer Aided Building Design, *Electronic Journal of Information Technology in Construction*, Vol.2, 33-56. ISSN 1400-6529.

Booch G. (1994). Object-Oriented Analysis and Design with Applications, 2/e. ISBN 0-8053-5340-2. Addison-Wesley, 1994.

Cooper G. (1995). Object Oriented Databases from an Information Technology Viewpoint: Knocking down some walls, *In proceedings of Object Technology and its Application in Engineering conference* (James Powell., editor), Glasgow. ISBN 0902 376 209.

IAI. (1997). Industry Foundation Classes, Version 1.5. http://www.interoperability.com/

OMG. (1996). Common Facilities RFP-4: Common Business Objects and Business Object Facility, OMG TC Document Number 96-01-04. http://www.omg.org/public-doclist.html.

Rezgui Y. and Debras P. (1995). An Integrated Approach for a Model-Based Document Production and Management , *Electronic Journal of Information Technology in Construction*, Vol. 1. ISSN 1400-6529.

Rezgui Y. and Karstila K. (1998a). D120: Building Process and Document Lifecycle Analysis, *Condor ESPRIT 23105 D120 Deliverable*, University of Salford.

Rezgui Y., Cooper G. and Brandon P. (1998b). Information Management in a Collaborative Multiactor Environment: The COMMIT approach, *ASCE Journal of Computing in Civil Engineering*, Vol.12, No.3, 136-144. ISSN 0887-3801.

Rezgui Y., Brown A., Cooper G., Brandon P. and Betts M. (1998c). Intelligent Models for Collaborative Construction Engineering, *Computer-Aided Civil and Infrastructure Engineering*, Vol.13, 151-161.

Rumbaugh J., Blaha M., Premerlani W., Eddy F., and Lorensen W. (1991). Object-Oriented Modelling and Design, Englewood Cliffs, New Jersey: Prentice-Hall, 1991.

Tarandi V., Karstila K., Noack R., Löwnertz K. and Bjork B-C. (1997). D213: Condor CAD Drawing Interpreted Model, *Condor ESPRIT 23105 Deliverable*, University of KTH.

UML. (1997). UML Document Set, *Rational Software Corporation*, Version 1.0. http://www.rational.com/uml/index.html.

Vakola M., Rezgui Y., Mitev N. and Thompson J. (1998). Condor Business Process Re-engineering Strategy, *Condor ESPRIT 23105 D310 Deliverable*, University of Salford.