Columbus: A Solution Using Metadata for Integrating Document Management, Project Hosting and Document Control in the Construction Industry

Juan Jose Herrero

A thesis submitted in partial fulfilment of the requirements of the University of Greenwich for the Degree of Doctor of Philosophy

This research programme was carried out in collaboration with Ove Arup & Partners

January 2003

University Of Greenwich
London

Acknowledgements

I am indebted to many people who have helped me in various ways throughout the preparation of this thesis and over the lifetime of the Columbus product.

I owe special thanks to my supervisor, Dr. Chris Woollard, for his support and guidance over the duration of this research. Chris has been an excellent supervisor, providing insightful comments and constructive criticism throughout the project.

I would like to express my gratitude to Ove Arup for having the foresight to support this research and thank my colleagues in the Software Technology Group who have been involved with Columbus for their support and dedication.

A special word of appreciation goes to Alec Milton, who should receive joint credit for initiating the Columbus project and has done so much to promote the software and make it such a success. Without his efforts, Columbus would have just remained a research project rather than the practical solution that it has become.

I am extremely grateful to Adrian Conlon for his work in maintaining our C++ libraries and for his involvement in the development of the AutoCAD ARX module. Adrian has been a valuable resource for exchanging ideas over the development of the product. Also, Martin Cramp should be recognised for his work on the Columbus web site and Alan Ogden for all his patience in supporting users, writing documentation and providing training courses.

On a personal note, I would like to thank my mother, who from my earliest days pushed me so much in my learning and taught me my first words in English and Spanish.

To my father, who has sacrificed so much for me, I give thanks for dedicating the majority of his life to my well-being, providing financial and emotional support so that I could enjoy my student years without hardship.

And finally, to my wife Olga, who has enriched my life immeasurably. Words cannot express my gratitude and love towards her. Her moral and practical support has kept me going with this research. Her assistance in proof reading this report has also been invaluable. I am indebted to her for the patience that she has shown. In particular, I thank her for being so understanding whilst I worked into the late hours during the writing phase of the project, particularly as she was left on countless occasions to have dinner alone.

I dedicate this project to my wife and parents with all my love.

Abstract

This thesis presents a solution for integrating document handling technologies within the construction industry using metadata in a novel way and providing a working solution in the form of an application called Columbus.

The research analyses in detail the problem of project collaboration. It concentrates on the usage of document management, project hosting and document control systems as important enabling technologies. The creation, exchange and recording of information are addressed as key factors for having a unified document handling solution.

Metadata is exploited as a technology providing for effective open information exchange within and between project participants. The technical issues relating to the use of metadata are addressed at length.

The Columbus application is presented as a working solution to this problem. Columbus is currently used by over 20000 organisations in 165 countries and has become a standard for information exchange. The main benefit of Columbus has been in getting other project participants to send metadata with their electronic documents and in dealing with project archival. This has worked very well on numerous projects, saving countless man-hours of data input time, document cataloguing and searching. The application is presented in detail from both commercial and technical perspectives and is shown as an open solution, which can be extended by third parties.

The commercial success of Columbus is discussed by means of a number of reviews and case studies that cover its usage within the industry. In 2000, it was granted an Institution of Civil Engineers' Special Award in recognition of its contribution to the Latham and Egan initiatives for facilitating information exchange within the construction industry.

Contents

List Of Figures

Chapter 1

Introduction

1.1 Background and Aims

This research looks at the state of information handling in the construction industry and focuses on the use of Information Technology in document handling and specifically how effective information exchange using metadata can improve communication on a project.

During the lifetime of a project, there is a constant need to exchange construction documents amongst participants. Some information needs to be distributed to various team members for review or coordination; whilst other data has to be shared as part of a common live engineering model. The documents can be on paper or in a multitude of electronic formats and the process of managing the exchange, cataloguing and retrieval of them can be extremely costly. An example of this inefficiency is the need to manually re-enter drawing titles and numbers as information is moved between systems.

The research looks at how the construction industry is organised and how project participants can improve the way in which they exchange information. Though each team member can differ greatly in the level of investment available, closer cooperation and the usage of Information Technology to link document handling at all levels are key requirements as established in the landmark industry reports by Latham (1994) and Egan (1998).

The focus is on how document management, project hosting and document control systems are used to provide a unified solution to the problem and what inefficiencies exist when they are linked together. The work considers how information is shared within an organisation that uses different document handling applications and when information is exchanged amongst various project participants. The lack of interoperability between these systems is highlighted as directly responsible for many inefficient practices.

Special consideration has to be given also to project archival and how data can be accessed and restored many years after a project has been completed. This is particularly important if any litigation were to arise. Because of this, information needs to be stored, catalogued and accessed in a simple and reliable manner.

The way to improve these processes, as will be suggested in the thesis, is by producing a unified solution linking applications that uses metadata in a neutral format. Metadata is information about the documents that can be read and understood by all document handling applications. The thesis presents a detailed review of the different ways in which it is defined and used, describing the efficiency gains that can be made. To demonstrate the feasibility of this approach, an application called Columbus has been developed that is based on this solution. Columbus is a real product which has valuable document management and project hosting capabilities and links to external project hosting services and document control systems. The application is based on the concept of attaching metadata to documents externally, providing a practical solution to the problem. Columbus is the most tangible contribution that this research has made to the industry.

In 1995, Ove Arup was investigating how to adopt and implement a Document Management system across the firm. As a result of this, various options were considered and the Columbus product was chosen as the preferred solution.

The software is in widespread use within Arup and in other companies across the world, as it was decided to make the application freely available to the rest of the industry. Columbus is currently used by over 20000 organisations in 165 countries and is now an important corporate application within Arup. One of the key benefits of Columbus has been to get numerous project participants to exchange document metadata in a neutral format, therefore saving countless man-hours of data input time into different document handling applications. One other important saving has been to ensure that project information is archived with metadata in a simple and open format, which is guaranteed to be accessible many years later.

1.2 Research Questions

The specific questions that can be drawn from the aims of the research can be stated as:

- How well do team members collaborate on projects, what shortcomings exist with current approaches and what can be done to improve how they integrate together?
- As document handling is seen as a crucial aspect of the construction process, what type of applications are used and what requirements do project participants have of these systems?
- If document management, project hosting and document control systems are identified as the key document handling applications, what can be done to ensure that any information that is common amongst them is shared effectively?
- Is it feasible for a single monolithic closed document handling system to be imposed on all participants, or can an open and extensible solution be established?
- Can neutral metadata exchange be used as an enabler in an open solution, thereby allowing different document handling technologies to exchange information seamlessly?
- What are the different ways in which metadata can be represented and can a common format be suggested to exchange information?
- Can an application be created that demonstrates information exchange and suitable a long term project archival strategy that is based on the use of metadata?
- The thesis presents the Columbus application as a solution, with the specific goal of improving project collaboration. As such, what are the difficulties in encouraging organisations to use the software and how can the benefits of the software to the industry be quantified?

1.3 Thesis Structure

The thesis naturally falls into two parts, which are relatively independent. The first part, covering chapters 1 to 4, looks at current practices and existing technology. It also describes where problems exist and suggests how they can be overcome. The second part, starting at chapter 5 presents the Columbus application, describing how it works, its design and how it has solved many problems on real projects. Briefly, the scope of each chapter can be described as follows:

Chapter 1 is an introduction to the thesis, providing an overview of the work and describing the contribution to the project from various other individuals.

Chapter 2 looks at information handling in the construction industry. It describes how project participants are selected, the way in which they work together; their differences and how traditional contractual arrangement have put them in an adversarial position. There is a review of the Latham and Egan reports, which are key motivators for change and improving efficiency. The chapter then looks at Information Technology, specifically at document handling, describing how data sharing needs to be improved.

Chapter 3 builds upon the previous chapter, focusing on Document Management, Project Hosting and Document Control as the critical document handling systems that need to be investigated. After providing a detailed description of each one, it suggests that a unified solution based on metadata exchange and sharing data seamlessly between applications and participants can produce substantial gains.

Chapter 4 takes a detailed look at metadata, describing what it is, how it is used and different ways in which it can be implemented. This is then used in subsequent chapters as the basis of the Columbus application.

Chapter 5 presents Columbus, a practical solution to document handling. It gives a detailed description of the product from the user's perspective, separating the application into its key components: navigation, document viewing, publishing and collaboration, document acquisition and creation, document reporting and document activities.

Chapter 6 is a technical chapter from a computer science perspective. It presents the architecture and design of the application, which is mainly done using the Unified Modelling Language. If unfamiliar with software design, this chapter can be skipped without significantly affecting how the rest of the thesis is presented.

Chapter 7 describes how Columbus was released within Arup and subsequently to the rest of the industry, discussing the impact that it has had. By looking at case studies, reviews published in journals, user statements and awards received, it analyses how it provides a solution that has helped to improve information exchange within the industry. It also looks at the other benefits that Arup has received from the product, including the promotion of other services and increased sales of other products.

Chapter 8 presents the conclusions of the thesis. It begins with a brief summary of each chapter and is followed by a review of the research's outcome. This focuses on what its aims have been and describes the achievements and failings encountered. The chapter concludes by evaluating particular items that could be considered for further development.

The thesis report concludes with four appendices. These cover journal reviews, user statements, the Institution of Civil Engineer's award and procedural notes for the usage of Columbus on a particular project.

1.4 Contributions

This research was sponsored by Ove Arup & Partners, which is a global organisation of consulting engineers with more than 7000 staff and operating in 33 countries. The firm has worked on many of the landmark construction projects, from the Sydney Opera House and Pompidou Centre to major infrastructure projects such as the Channel Tunnel Rail Link.

Though all of the underlying research and technical content, such as the architecture, design and coding of the Columbus product have been carried out solely by myself, there has also been an input in the development, promotion and marketing of the software from various other people. It is important to recognise the contribution of these individuals so that they are acknowledged and credit is given for their involvement.

The original Columbus concept and its approach to document handling was the direct result of discussions between Alec Milton and myself following a visit to the design offices of Rail Link Engineering. Alec Milton, group leader of the Arup Software Technology Group, had the foresight to authorise the development of the project and should be credited as jointly responsible for the Columbus idea. In addition, Alec was the main force behind the promotion of Columbus within Arup and the whole industry and it is thanks to his persistence that the product is so widely in use.

After developing the first version of the AutoCAD metadata extraction module, I handed over the development of this module to Adrian Conlon. Adrian was responsible for porting the application to AutoCAD 2000 and adding a number of enhancements; he also added metadata creation facilities to the CADplot application. Adrian is also responsible for a number of low-level libraries that Columbus uses.

Columbus' support, training and documentation are the responsibility of Alan Ogden. Alan, who was the chairman of the Arup Columbus Client Committee before joining the team, wrote the "Teach Yourself Columbus" and "Advanced Columbus Configuration Guide" publications, in addition to producing the Columbus manuals and help file. Alan also provides on-site Columbus training courses for internal and external clients and first level support to users.

Web site development is Martin Cramp's responsibility. In addition to creating and maintaining the internal and external Oasys web sites, Martin has been responsible for the various incarnations of the Columbus site. Jointly with Adrian Conlon, he developed the software registration database and credit card processing software for Columbus CD-ROM sales. Recently, Zac Babawale, a new programmer, has been employed to help with the development of Columbus. Zac's major contribution, has been with the Email Filing System, which is described in Chapter 8 under the "Further Work" heading.

Columbus has also served as the basis for two MSc. projects, which have helped to improve the way the application interacts with other systems: the first project, "Design and Application of an XML Schema for the Interchange of Documents in the Construction Industry" by Olga Castillo (2000) looked at how a standard XML metadata schema for the construction industry could be defined and designed a Java based application that used Columbus format metadata. It was a prototype for a lightweight version of Columbus that could be accessed from any computer using a browser. It also acted as a sample client interface for project hosting sites. The second project, "A Reporting Application For the Columbus Document Manager" by Martin Cramp (2001), presented a way of gathering Columbus format metadata into a single database. The program that was developed showed how other document handling applications such as document control systems could read, process and import Columbus metadata.

Chapter 2

Information Handling In The Construction Industry

This chapter looks at the state of the construction industry in the context of project collaboration. First, the interaction between project participants is considered, looking at how teams are formed, their capabilities and how traditional contract conditions have put firms in an adversarial rather than collaborative position. Then, the chapter reviews the Latham and Egan report, which set milestones for change within the industry and discusses how team-working can be improved. Subsequently, it describes how improvements in the use of information technology and enhanced information sharing can result in big efficiency gains. This is done by reviewing the requirements for project participants to have their information well organised and gives consideration to how data can be exchanged and shared on projects. Emphasis is also made of the importance of maintaining records of all transactions and document archives.

2.1 Industry Characteristics

The Construction Industry can be defined as the sector of the economy concerned with building, repairing, maintaining and renovating the built environment. This covers things such as: buildings, bridges, tunnels, railways, docks, airports, pipelines, power plants, etc. It is characterised by the production of relatively large, single unit, custom designed and purpose built items. This is in clear contrast with other sectors such as the automotive or manufacturing industries where small, inexpensive and mass produced items are built. Small and large private businesses, government agencies, manufacturing establishments and public utility companies are all involved in the construction industry at one stage or another.

According to the Royal Academy of Engineering (1996), in the United Kingdom, the construction industry represents 9% of the gross domestic product and employs over a million people, which clearly emphasises the importance of the industry to the economy. In the same report, it is also highlighted that 90% of companies in the industry have less than 14 staff, which accounts for 50% of the employees in the industry. The British Department of the Environment, Transport and the Regions' statistical register lists 163,000 construction companies (Egan 1998). The result of this is an industry with a high degree of fragmentation and lack of strong leadership, which limits its responsiveness to change in what is a highly competitive market. This presents serious limitations to making any kind of radical changes, limiting organisational learning (Andreu and Ciborra 1996) and making IT solutions of any complexity difficult to implement.

2.2 The Project Team

2.2.1 Participants

When a new construction project starts, a great number of organisations are brought together to achieve the goal of completing it. The actual number of organisations participating in the project can vary from a dozen to several hundred, depending on its scale and degree of complexity. The size, experience, background and degree of involvement of each participant will vary greatly, which has resulted in the great diversity and fragmentation that is found within the industry. In general, on a construction project, we can find the following participant categories:

Client: This is typically the owner and promoter of the project, together with its representatives and may be a governmental body or a private institution. Client attitudes vary greatly, and whilst some will take a passive role, others for example will go to the extent of briefing trade contractors on site to ensure that they feel part of the team (Latham 1994). In certain cases, such as in Private Finance Initiative (PFI) projects (Bates 1999; Grout 1997), the distinction between consultant, contractor and client is blurred, as the client plays an active role in the construction process.

Consultants: This group encompasses the architect and consulting engineers. Examples of the later are the structural, mechanical and electrical engineers. They are generally active throughout the duration of the project and are responsible for the overall design, specification and quality of the construction. Consultants typically produce "high level" construction documents (e.g. structural framing plans) and review "low level" documentation (e.g. shop drawing) produced by contractors or specialist contractors. In the case of building projects, the important role of coordinating all the consultants' work has traditionally been with the architect, though there is a growing tendency nowadays to pass this role on to management professionals (Berman 1999). Often, consultants are criticised for seeking to transfer more of the design to specialist contractors and isolating themselves from the construction team (Latham 1994).

Contractors: Their responsibility is to materialise the consultants' designs. Depending on the contractual arrangement of a project, this role may lie with a single organisation, as seen in traditional contracts, or it might be divided amongst a number of "package contractors" with specific areas of responsibility. In this last scenario, the services of a construction management company is employed to coordinate all the contractors. Regardless of the approach adopted, contractors are unlikely to carry out all the physical work and commonly distribute the workload amongst specialist and trade contractors.

Specialist and Trade Contractors: These are companies that are highly specialised in a single area of construction, and between them all, carry out almost all of the construction work. Normally, they communicate only with the contractor and do not exchange information between themselves. Examples of areas where specialist contractors can be found include: piling, structural steelwork, lift and escalators, curtain walling, information technology, communications and networks, heating and air conditioning, lighting and power, public health engineering and security systems to just mention a few. As specialist contractors are highly focused on one task, one worry is that their commitment to the coordination of the whole project will be limited. A further area for concern is organisational size, which can vary greatly. Though there are some very large specialists, most of them are small companies with limited experience and investment in information technology and hence represent the biggest obstacle to implementing any of the proposed changes.

Others: In addition, there are many other participants which may not be directly involved in the construction process; for example: financiers, insurers and building regulation authorities. Their involvement, though crucial to the project, is of minor relevance within the scope of this research.

From this simplified description of project participants, it can be seen that there is a major coordination and management issue to be considered. Figure 2.1 shows the general interaction between the project team members.

The diversity and quantity of participants has caused the industry to be fragmented, resulting in poor communication and technological evolution. Improving communications and coordination between organisations is a complex issue, to the extent that project management professionals exist solely to facilitate the interaction between participants. After having personally worked in a construction management team over the period of a year on a prestigious building development in London's Canary Wharf, I was able to experience most of these issues first hand.

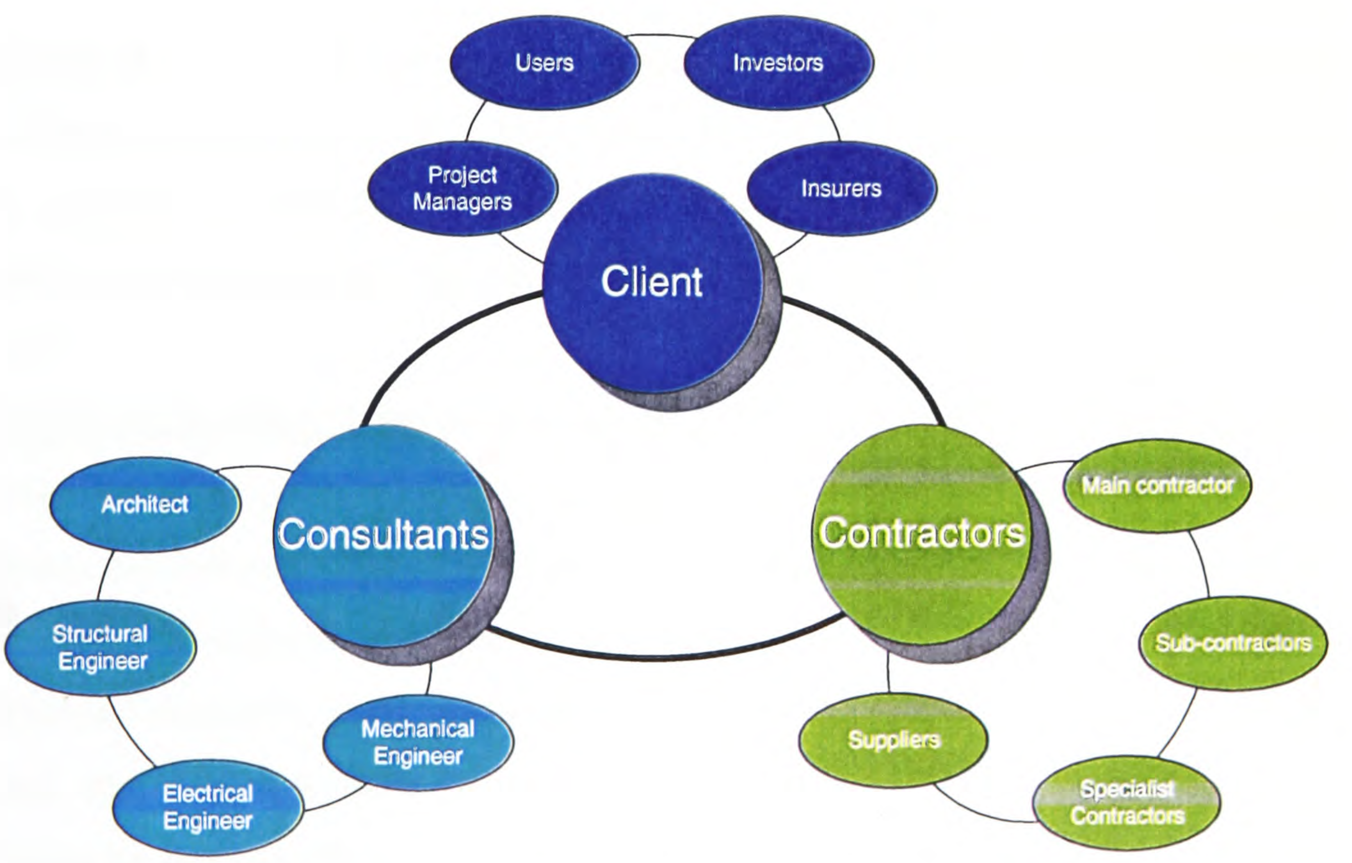

Figure 2.1 Project Participant Relationships

2.2.2 Team Selection

Before looking at how project participants interact and how they can work closer together, it is important to describe how the team is formed and how each member is related contractually. Though this is not intended as an exhaustive review, the following selection methods are described because of their relevance to this research: tendering, preferred supplier lists, partnering and strategic alliances.

Tendering: This is a selection process where organisations compete against each other for project work based on the submission of bids. Though a minimum standard is required from every bidder and other factors may be taken into consideration, it is generally the lowest cost submission that succeeds. The Construction Industry Board (1997) emphasised that for competitive tendering to be effective in providing best value for money, it must be seen to be fair and the processes by which decisions are reached must be as open as possible. Historically, the construction industry has worked on the basis of tendering and from survey data (Holt et al. 1996), it is confirmed that contractors achieve a contract award only once in approximately five tenders submitted. The result of this is that clients ultimately pay higher costs to offset the 80% of unsuccessful tenders. A further drawback of tendering is that it does not favour inter-participant collaboration, as each participant seeks to meet their own goals rather than playing as part of a team. This can greatly increase the cost of the project, as collaboration between participants is not imposed as a contractual requirement. A further issue to consider, is that many participants typically will not have worked together before, requiring project specific processes to be established, further adding to the cost.

Preferred Supplier List: In this type of selection process, the client has a pool of preferred participants which is typically based on past experience and recognition within the industry. Through regular repeat work, close involvement and specially negotiated contracts, they reach an understanding of each other's needs. As preferred suppliers, participants do not have the overhead of having to tender for work and as such are able to reduce their costs. However, reduced costs might not always be directly passed on to the client, as a monopolistic attitude is sometimes apparent. In terms of interaction amongst participants, there is still no implied need to improve the relationships between them. Though the diversity of organisations is reduced, there is still a tendency for each participant to be solely concerned with their own goals rather than with what benefits the project overall.

Partnering: This is a more recent concept, which involves two or more organisations working together as a single entity to improve performance for mutual and the client's gain. The hostile and adversarial relationship is replaced by good will and close cooperation. As Critchlow (1998) describes, the sharing of information is an essential element to the successful working of this framework arrangement. Teaming up to agree working procedures and establishing a track record before joining a project, demonstrates that partners are committed to collaborate. Barlow et al. (1996) describe a number of key objectives regarding communication within the scheme, which include:

- Provide open and flexible communications.
- Break down of formal communication hierarchies in an attempt to simplify information flow.
- Allow people working on the later assembly stages of a project to talk directly to those involved in the design and planning stages, bypassing the intermediate project managers.
- Avoid bureaucracy.

All of these encourage people to contribute and work for the good of the project. Nevertheless, whilst evidence shows that partnering provides considerably improved construction performance, it should not be seen as a universal panacea for the construction industry's problems (Barlow et al. 1997).

Strategic Alliances: Partnering can be applied at a project specific level or on a short-term basis to improve efficiency. However, as the Construction Industry Board (1998) emphasises, there can be major benefits when the relationships formed become long-standing across projects. These long-term relationships, known as strategic alliances are established with the aim of increasing efficiency, bringing consistency and engendering trust between participant organisations (Orange et al. 1998). Organisations then benefit from having access to skills or resources that they do not posses (Craverns and Shipp 1993) and there is likely to be a rapid diffusion of new technologies and mutual learning (Lorange and Roos 1991). This is particularly relevant when applied to information technology, as has been highlighted by many (Suomi 1992; Hørlück 1994). For example, computer system incompatibilities are likely to have been resolved before the project commences, and data exchange between the partners is based on a mature tested technology.

2.3 Improving Efficiency

There is a growing dissatisfaction amongst clients with the overall performance of the construction industry and a lot of research has been undertaken to investigate how it can be improved. Notable for helping to achieve this are landmark reports by Latham (1994) and Egan (1998), which established a framework for change.

2.3.1 The Latham Report

In the early 1990's, the UK Department of Trade and Industry (DTI) commissioned Sir Michael Latham to chair a committee to review the construction industry and investigate how it could be made more efficient. In 1994, the outcome of this work was published in a radical report entitled "Constructing the Team" (Latham 1994). Latham made 30 recommendations which, if implemented, he believed would lead to a 30 per cent reduction in real construction costs. Many of these issues have a direct bearing on the way information technology can act as a catalyst. Amongst other issues raised in the report, Latham emphasises that the client is the driving force and should encourage good team working, good design, and innovation with strong management. There is a call for an improvement in construction coordination by implementing effective pre-planning of the design process and better use of coordinated project information. Knowledge based engineering, which enables designers to see new ideas either through advance computer aided design or virtual reality systems and to quote Latham (1994): "the establishment of common standards for the exchange of electronic data would be highly desirable and further consideration should be given to this issue".

It is suggested that coordinating project information should be made a contractual requirement and that a quality register of consultants and contractors be maintained. Partnering is also suggested as a viable solution with the aim of improving collaboration contractually, this would bring together participants that already know each other and could work together with greater ease.

A strong emphasis is placed on research and development into technologies that would act as enablers for efficient project coordination. Latham highlights that trade and specialist contractors would benefit the most from being given access to shared project information, but also emphasises that they are the most difficult to integrate into new IT systems, due to their size and lack of investment, suggesting that they need to be better remunerated.

Though the Latham report looked at the UK construction industry, much of what is stated is equally applicable to other countries, as is highlighted in "Constructing the team: a U.S. Perspective" (King 1996). This is particularly important as the industry becomes more global and key players operate across national boundaries transparently.

The Latham report was well received throughout the industry and in particular, "Responding to Latham: The Views of the Construction Team" (Gruneberg 1995), emphasised that contractors are in agreement that a single point of responsibility creates an adversarial relationship and questionable performance. The client should seek greater and earlier involvement of specialist contractors, and that their contribution is rarely acknowledged.

Latham set the foundation for change within the industry, making many important recommendations for how it could be improved. Subsequently, others such as Cockshaw (1997) and Egan (1998) have followed with other constructive suggestions.

2.3.2 The Egan Report

"Rethinking Construction", written by Sir John Egan (1998), Chairman of BAA plc., presented the first major client-led review of the construction industry. Built upon the recommendation of the Latham report, it too is concerned with how to improve quality and efficiency in the construction industry. Egan identified the following key drivers for change in order to modernise the industry: committed leadership, focus on the customer, a quality driven agenda, commitment to people, integrated processes and an early involvement of contractors in the design stage. Though change is required in all of these areas, it is integrating processes and teams that this research centres on. One way of achieving this is by establishing long term relationships between organisations, as previously described, acting as a catalyst for standardising procedures and information systems.

Egan recommends that contractors are brought in at an early stage, and emphasises that smaller companies may be unable to cope with the new demands, and they may need to operate under the umbrella of a larger organisation. The supply chain is also looked at, and the adoption of partnering is recommended, with a particular emphasis on the use of standardised components.

Egan also suggests that eliminating confrontation, integrating processes, ending competitive tendering and forming long-term relationships would improve teamwork. Strong leadership is required particularly from the client and there should be a quality driven agenda. Major clients need to initiate change in the industry. They need to use their power to force the industry to work in a more efficient way. One way that this can be done, is by carefully scrutinising who will be on their preferred partner list. Clients need to change too, they need to stop thinking that price equates to cost, based solely on the tender price, which is seen as one of the major barriers to improvement.

As a direct result of the Egan report, the standard "Project Partnering Contract" (Mosey 2000), commonly known as PPC 2000, was introduced which amongst a number of important suggestions, emphasises:

- **Team-based multi-party approach:** All parties sign a single Partnering Contract, encouraging a team-based commitment to the project. There is a specific duty for all parties to deal fairly with each other and with their sub-contractors, specialists and suppliers, in an atmosphere of mutual co-operation.
- **Integrated Design/Supply/Construction Process:** Suggests the establishment of relationships between the design, supply and construction teams.
- **Core Group:** Involves having key team members who look out for problems and undertake regular progress and performance reviews.
- **Non-adversarial Problem Resolution:** Requires a problem-solving hierarchy of increasingly senior individuals from within each member of the team. It also suggests a facility for conciliation.

Essentially, Sir John Egan has had the vision of improving the construction industry by working with all parties to achieve savings. As chairman of BAA plc., he placed his company at the forefront of this challenge, by encouraging "all electronic" projects. An example of this is the £400 million Heathrow Express scheme (Cole 1997).

2.3.3 Learning from other industries

In addition to looking at new techniques to improve efficiency in the construction industry, it is worth investigating how other related industries have evolved over the years and tackled similar issues. The technologically closest, from which a lot can be learnt, include the aerospace, automotive and petrochemical industries. They all share a common engineering background, aiming to deliver products in the most efficient way to their clients.

However, it must be emphasised that the construction industry lags far behind the others technologically, as it does not benefit form the same level of investment that they enjoy. It is a low volume and low cost industry, which is made up of small fragmented organisations. One way forward is to try and use standard components and pre-assembly, which will help make it high volume and justify the level of investment required. This would also further integrate suppliers into the chain (Egan 1998). Other improvements include the use of more elaborate computing technology; for example, the automotive industry has been using full three-dimensional shared models for many years and implementing collaborative engineering quite successfully. In practice there are no major impediments to providing virtual engineering teams, as it is not a technical matter but a social and organisational one (Line 1997).

2.3.4 Appling Information Technology

Overall, this chapter highlights a number of inefficiencies that exist in the construction industry and describes changes that can be made to improve it. Though the areas and processes fingered for change are wide ranging, this research specifically focuses on the productivity gains that can be made by enhancing the use of Information Technology. As emphasised by Egan (1998) in Rethinking Construction, "good IT is an essential part of improving the efficiency of construction". A major survey report entitled "IT usage in the construction team" (Building Centre Trust 1999) also highlighted the need for changing, amongst other things: organisational attitudes to IT, project IT infrastructure and the usage of specialist software within the industry. The areas of IT where major improvements can be made are in document handling and electronic communications between participants (Amor et al. 1996).

For collaboration between participants to be a success, it is essential that every team member has a clear view of what has to be constructed, and this is achieved by providing a description of all the required processes and tasks through construction documents. These documents describe the extents of each team's work and the interactions between them. By improving the way that information flows between the teams, great savings can be made during the life of the project (MSM 2000). A number of key processes can be identified as crucial to improving how construction information is shared between participants and they include: production of data, information publishing, collaborative working, use of common models and recording transmitted information.

2.4 Information Sharing

Before investigating some of these key processes, it is important to consider the reasons why participants share information and their requirements. Though there are many reasons for exchanging information, the following are considered worthy of particular attention: instruction, review, linked information and coordination. All have their own requirements and need to be considered individually as a single solution is unlikely to meet the needs of all.

2.4.1 Instruction

In this case, information is published from one participant to another as an instruction to carry out an activity. A typical scenario where this occurs is when a contractor is asked to carry out a specific task such as building a wall. What is required in terms of construction information, is a printable or viewable representation of the task; i.e. a drawing showing the wall location. Traditionally, this has been achieved by exchanging documentation in printed or plotted form, without the need to give the contractor access to the underlying source electronic data files. Nevertheless, though the source files may not be required, clear benefits can still be identified if the drawings are handled electronically rather than on paper. For example, savings can be made in reducing the cost and improving speed of distribution. Beyond that, the data has no further value, and a publishable electronic format such as Adobe's PDF or Hewlett Packard's HPGL/2 language would suffice. The "reuse" value of the data is limited, but the ease with which it can be reprinted and the ability to guarantee that the correct information will always be reproduced far outweighs the need to have access to source data files. Currently most information exchanged within the industry falls into this category.

2.4.2 Review

In this situation, information is passed to another participant for review as part of a workflow process (Hollingsworth 1994). Typically, this occurs when documentation produced by a trade contractor requires validation from a consultant. An example is the requirement by steelwork contractors to submit their "shop drawings" to the structural consultant for approval. This requires feedback to be returned using a suitable format. Redlining and mark-up facilities within viewers have scope for doing this electronically, but are less than ideal. Unless the original files can be redlined, the information is only usable as an electronic "marker pen" and will need to be re-entered in the source files. Moreover, reviewing large drawings electronically is quite difficult as it becomes more difficult to spot errors on screen. Because this, many reviewers will choose to plot the information, review it on paper and re-enter their comments electronically, which is clearly inefficient.

2.4.3 Linked Information

This scenario occurs when information that is produced by one participant is required as part of another participant's work. An example is a floor layout that is drawn by an architect and passed to a mechanical engineer, who then places the ductwork in accordance with this floor plan. The most efficient way of doing this is to use the Architect's electronic Computer Aided Design (CAD) file as a background to the Engineer's layout. This is one of the most efficient ways of reusing electronic data, and is known as referencing. In the most popular CAD package, AutoCAD, this is called external referencing (XREF). When the original information changes, for example if a wall is moved, all that is required is that the

Architect's CAD file be reissued to the Mechanical Engineer and that he repositions his ductwork accordingly. In order for this to work, it is necessary to exchange information in a native or compatible file format, and that the data be in accordance with very strict guidelines for it to be compatible. The use of CAD standards for exchange of source information is of great importance within the industry and the use of the following CAD facilities should be standardised on a project: layering, text styles, dimension styles, title blocks, layouts, templates and design procedures (Green 2000).

2.4.4 Coordination Model

In this case, a model holding all project information is maintained to aid in coordinating and sequencing the whole project. The model is typically held on a central server and is directly accessible by all participants, who are given access to view and add their contribution to the design. This represents a major departure from the sequential data exchange processes traditionally used. All information is normally kept up-to-date either directly by each participant or by a coordinator.

The project is typically maintained as a three-dimensional graphical model, though facilities may also exist for non-graphical information. Coordination models are particularly valuable to construction and project managers, though other project participants can also benefit from seeing the "big picture". The main advantages of having such a model are that they give an accurate visualisation of the design prior to production, concurrent access to a single integrated model, availability of the most up to date project information and the use of error checking features such as collision and clash detection (Excitech 1999). However, because of the nature of the information required to build the model, source documents (e.g. CAD files) cannot normally be used directly, and information from the model cannot be reused for most other work.

There is also a need to "police" the information that is entered into the model to ensure that it is in a valid format, accurate and complies with appropriate standards. Because of this, a coordination model should be seen as complementary to the other project documentation, and not as a replacement.

2.4.5 Handover

Even though not strictly part of the construction process, at the end of the project all design information needs to be handed over to the client. The client may then choose to use some of this information, for example, to carry out future work or facilities management. Traditionally, little thought has been given to integrating the "handover" information, and this has resulted in a mismatch of information originating from diverse sources and of incompatible formats. The final outcome of this, on many occasions, is the need to re-survey the newly constructed project to ascertain the as-built information in the required format (Excitech 1999). The improvement of how construction documentation is handled will undoubtedly enhance the quality of the handover information given to the client.

Interestingly, Arup now sell copies of drawings produced as far back as the 1960's. These are made available, to amongst others, property developers who are refurbishing buildings that were originally designed by Arup. The information has been kept in the simple but reliable microfiche format and is now being scanned into the standard TIFF graphics format for reuse.

2.5 Information Production and Organisation

Before a project participant can be expected to share information with others efficiently, it is important that they organise themselves first. Some key points to consider are: management of documents, maintaining document metadata, using common applications, adhering to standards and having an appropriate archival strategy.

2.5.1 Managing Documents

Effective management of documents requires processes to create, find, edit and catalogue information. This can mean investing heavily in elaborate document management systems or simply that well considered procedures are adhered to in a disciplined manner. For example, in Arup there have been guidelines in place in the form of an internal publication known as the Arup Information Manual (AIM). The AIM describes amongst many other things the directories where project information should be filed on networked drives and a file-naming convention.

This contrasts with other organisations where data is kept at ad-hoc locations such as users' hard drives or in their personal directories. Wiggins (1999) highlights that professionals spend between 5 to 15 percent of their time reading documents, but an incredible 50 percent searching for them. One other important issue to consider if a document management solution is used internally, is that it must be compatible with the mechanism used to share data with other project participants, as it should complement and not hamper the way in which information will be exchanged. In many situations this has resulted in the adoption of hybrid systems (Excitech 2000) allowing document management systems to work seamlessly with project hosting systems. Though subsequent chapters will look at the subject of document management in detail, it is important to highlight now that regardless of whether it is considered essential to enforce the use of a document management system within an organisation or not, unless adequate procedures are established to carry out these processes, then the information will not be correctly organised.

2.5.2 Document Metadata

One important aspect to consider when sharing documents is to identify them clearly. Just using a filename is not sufficient; what is required is information about the information, i.e. metadata (Dornfest and Brickley 2001). Typically documents have many properties associated with them to help with classification and identification. Examples include title, author, revision history, contents, creation date, external dependencies, etc. This information helps to catalogue, index, track and share documents throughout the lifetime of the project and beyond. Though numerous standards exist for defining how metadata is created, maintained and exchanged, it is important that all project participants have procedures in place to handle this information both as part of their own document management processes and when sharing documents with others. Though the topic of metadata is discussed in chapter four, it is important to emphasise now that one of the most common problems encountered when trying to share information between participants is caused by internal document management solutions not handling metadata in a format compatible with other project participants.

2.5.3 Adhering to Standards

When project participants team up together, it is normally a requirement that they work in accordance with an agreed specification and follow set standards. This will cover, in a great level of detail the guidelines that should be adhered to in the documentation and construction of the project. It is essential that standards are adhered to when producing project documentation, as otherwise incompatibilities are introduced, which reduce the effectiveness of information exchange.

One important area for standardisation of information within the construction industry is in the production of drawings using Computer Aided Drafting (CAD). Examples where they are essential include file naming and layering, which are independent of the CAD application used. As Austin (1995) describes, there is a need to use standard line styles, character fonts and naming conventions across drawings, regardless of the software used. In 1994 Arup published the "CAD Good Practice Guide" (Arup 1994) detailing guidelines for how CAD information should be used on projects. Subsequently, this was published externally, and has been used by numerous organisations within the industry. Though a number of national standards exist covering drawing production such as British Standard 8888 (2000), British Standard 1192 (1998) and publications such as The Manual of Engineering Drawing (Simmons et al. 1995), standards are constantly changing and adapted as new technology is introduced or when national standards are replaced by international standards (Simmons 2002). As Fallon (1998) states, the key to information reuse in drawings is CAD standards, particularly layering standards that work for and are adhered to by all disciplines and outside consultants. Without such standards, project teams are unable to exchange meaningful information.

Arup have been active in setting industry standards. For example, we were involved in the development of British Standard 1152 part 5, defining a common naming convention for layers, blocks and files within the construction industry. In addition to standardising the usage of CAD, we are active in Technical Committee B/555 of the CICA (Construction Industry Computing Association), which is responsible for national standards in construction modelling, representation and information exchange in the construction industry and the UK working party for ISO 62045-1, entitled "Management Data (Metadata) for Technical Documents".

We were also actively involved in the Building Centre Trust report to the Department of Trade and Industry: "Effective Integration of IT in Construction" (Building Centre Trust 2001).

2.5.4 Application Compatibility

For two participants to share information successfully, it is important to ensure that their data is compatible, which means that the applications used should also be compatible. Though it would be ideal if all team members used the same software to produce their data, this is not always possible. For example, it is very common for an architect to use the CAD package MicroStation and other consultants to use AutoCAD. In this case, the best that can be done is to insist that the applications are capable of exchanging information in a neutral or common format. For example, in most CAD systems, lines, arc and circles have the same meaning and an exchange format such as DXF (AutoDesk 2002) can be used to share data. However, though numerous neutral formats have been defined, or application file formats have become de-facto standards, applications can only exchange data if the information is semantically compatible. A more doubtful scenario occurs when it is necessary to exchange information between software in dissimilar application areas. For example, if there is a design change in a floor layout which is maintained by an architect as a CAD model, it will be necessary for this change to be reflected in an engineer's structural analysis model. In this case, there is no direct translation for the data as the building is modelled for different purposes using dissimilar software. A more detailed description of the problems with data exchange across application areas is covered in section 2.6.3.

2.6 Exchanging Information

When considering how project information will be exchanged, it is important to take into account a number of factors. Amongst these, are the way in which organisations are inter-connected, the format that information will be in and the type of applications that will use the data. On many occasions, simple formats such as Adobe's PDF or Hewlett Packard HPGL plot files are suitable but quite often, there is now a need to exchange source data files between dissimilar applications.

2.6.1 Connectivity

Before the advent of computer technology, almost all information was exchanged on paper (with notable exceptions such as the facsimile and telex machines); letters were produced on typewriters, drawings on drawing boards and calculations were handwritten and they were all exchanged through the post. As computers became popular, most of these processes were computerised, yet they still relied to a large extent on producing paper output. Since then, the goal for technology has been to eliminate the hard copy and come closer to the concept of a paperless office (Bartholomew 1999); and to extend this to the paperless project, where all participants exchange information digitally. Data exchange in the construction industry started as the exchange of physical media, progressed to the use of private communication links and evolved into using the Internet to build virtual private networks (Morgan 1999; Compton 2002). The construction of restricted access networks over the Internet, known as extranets, combined with the vast improvements in communication speed, has made information sharing between project participants much easier. With connection costs now being insignificant, the only real expense is the expertise required to administer a system. Nowadays, the Internet has enabled everybody to be electronically linked, and many (Hannus 1996; Ouzounis 2001; Turk et al. 2000; Weisberg 2001) agree that the use of email or simple protocols such as HTTP or FTP have made considerable improvements to the way that distributed teams can work. Because these tools are mature and easy to learn, their fast proliferation can be expected to continue. The procedures for how those protocols are used in construction projects are well established and are supported by general practice in the industry. As the industry moves towards more elaborate project hosting technology, it remains to be seen if the same level of standardisation can be maintained.

2.6.2 Digital Paper

As described, in a move towards a paperless environment, electronic data files are replacing the use of printed and plotted information. Typically, these files are published in formats such as PDF or HPGL and though they may only be suitable for printing or viewing on screen, significant saving are made when compared to the cost of exchanging paper.

Despite this, though a number of organisations have embraced this technology, there is still an insistence by many others on receiving information in paper form. Though many are technologically capable of moving away from this, two issues are generally highlighted as concerns: firstly, sending or receiving paper appears to be more formal and secondly, it is not always guaranteed that the electronic document will be reproduced as intended.

The first point mainly focuses on the signatures that appear on drawing, which are required to validate document approval. This is gradually changing and techniques such as digital signatures, third party verification or widespread publication address this. The second issue is more difficult to overcome. Though, some organisations work digitally, they frequently insist that information is sent to them on paper. Their reasoning is understandable, as there is no guarantee that when printing or plotting an electronic file it will appear as originally intended. This is a common problem with HPGL (the format most commonly used for engineering plots) rather than PDF, where the result of plotting these files can on many occasions be quite variable. For example, if plotter configurations, such as pen mappings, are incorrectly set then the result can be inconsistent.

Interestingly, though many recipients insist on receiving paper, one of the things that they will do upon receipt, is to digitise the information by scanning, so that they can obtain the benefits of electronic document management.

Regardless of whether documents are scanned or exchanged in a format such as PDF, it is important to reiterate that they have no other use than just being printed or viewed. They are not suitable for editing by others or for use as part of a more complex assembly or common model. They are just "electronic photographs" of the printed information or digital paper.

2.6.3 Data Exchange

Though exchanging digital images of information is a great advantage over paper, its usefulness is limited, and it cannot be considered true data exchange. To allow full collaborative working with linked information, it is necessary that data files be shared. However, as described in section 2.5.4, compatibility between application software is essential to ensure that data is exchanged successfully. Some (Pollalis 1998; NASA 2002) have described this incompatibility as Islands of Automation. In Figure 2.2, Hannus (1998) shows an example where Computer Aided Design (CAD), visualisation, engineering analysis, cost estimation and project scheduling are handled by independent applications and how over the years various protocols have been defined to make information compatible and allow data exchange. Bridging the gap between the Islands of Automation is essential to the success of concurrent engineering (Krouse 2001).

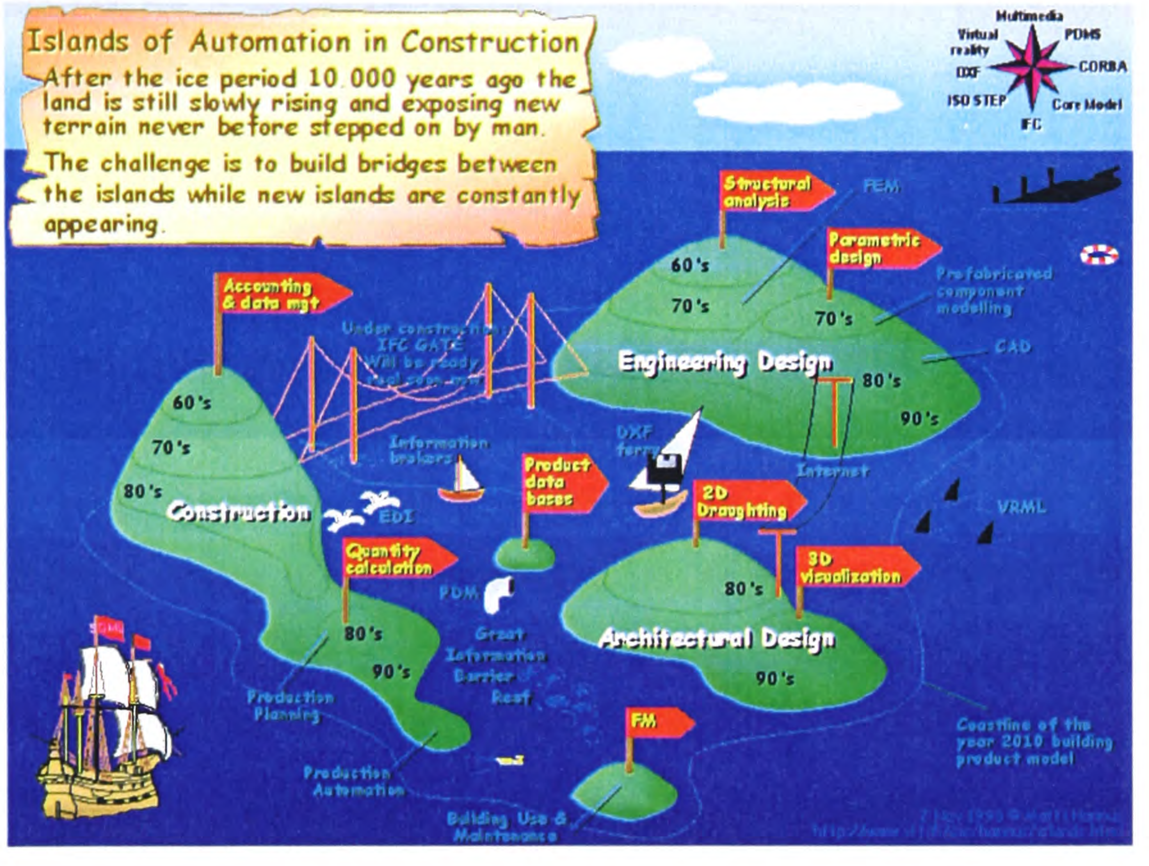

Figure 2.2 Islands of Automation

To take a specific example, when an architect changes the location of a column, the electrical engineer is notified so that he can change his electrical layout accordingly. However, depending on how the Islands of Automation have been bridged, these changes can be passed with a higher or lower degree of automation.

At it is simplest, data is not shared electronically. The changes are plotted on paper and the engineer needs to redraw or modify the electrical layout. This is still the most common situation today.

The other scenario is using simple CAD where, for example, the Architect issues files that can be imported or referenced directly into the Electrical Engineer's model. At that point, the electrical layout is repositioned according to the architectural layout background. This approach is now more common as participants standardise on compatible CAD packages and use common drafting standards. However, it will not work if the engineer does not use software capable of reading the CAD information.

One other way of linking applications is to use object modelling, which is considered to be the goal for information exchange. Though it is currently supported with various degrees of success (Day 1998), if and when fully developed, it could become the ultimate solution to data sharing. For it to work, each package maintains a model of the project as high-level objects, which hold the semantics of the information. When it becomes necessary to publish a change, the software passes on the data for any modified objects in a neutral format. The receiving application will then react to accommodate these changes. An example of object exchange technology are the Industry Foundation Classes (IFC), as defined by the Industry Alliance for Interoperability (IAI 1996). With IFCs, applications no longer exchange simple lines, arcs and circles (as was the case with AutoDesk's DXF), but higher-level objects such as floors, walls, door and stairs. IFCs evolved from the work on projects such as the General AEC Reference Model (Gielingh 1988) and the ESPRIT projects ATLAS (1994) and COMBI (1995). Other important milestones in information exchange within the industry were the definition of AP225 (ISO 10303-225 1997) and the CIMsteel steelwork exchange project (CIMsteel 1994). In addition, a number of vendor initiatives such as aecXML (AecXML 2001) and DesignXML (AutoDesk 2001) have since become more important within the industry. Though the definition of the object properties and behaviour are of major importance, object versioning, access permissions, parallel edit control and audit trailing also need to be considered (Woollard 1988).

The type of model generated by using objects is very high level and may not always be suitable for all types of engineering. For example, it is apparent when comparing the construction information produced in the UK with the US that the latter use a greater degree of performance specification rather than the detailed information shown in the UK (Flanegan et al. 1985).

Though objects may seem ideal, they are not yet ready for use with the industry, and Arup have expressed their concern over the current push to use objects before a number of major problems have been addressed. In an article entitled "Defeating the Object" (Day 1998) Arup, amongst others, highlighted potential problems with binary compatibility, archival, versioning and object exchange. Even if the idea is good, there are many hurdles which need to be overcome first.

2.7 Shared Project Information

The approach to information sharing described in the previous sections requires that each team member maintains the master data for each of their own processes and that copies of this information are transmitted to other participants when necessary. By following the traditional sequential linear and paper-based process, Khanzode and Fischer (2000) calculated that there is typically a three-week unavoidable delay in communicating design change information from an architect to a subcontractor, as the information is passed via the consultant and general contractor. Because of this, to improve efficiency, it is necessary to break the sequential flow of control, in which every participant is solely dependent on what is handed over to them by another participant and contractually only bound to comply with that small element of work. This is a reason why so many disputes and claims arise and therefore there is a need to have flexible links between all team workers. Change is required to make all participants have an overall input and feel responsible for the complete project. In particular, designers should work in close collaboration with all other project participants.

The use of shared information helps to utilise all the skills of the construction team and bring them closer together in a virtual community. Despite the fact that some of the information may be commercially sensitive, this can be addressed by having well defined teams with mature partnering agreements.

One way of improving team integration is to forward information electronically, use common project models and maintain document repositories.

2.7.1 Electronic Forwarding

As mentioned in various places throughout this chapter, substantial savings can be made by exchanging documents electronically rather that in paper form. These savings are typically in delivery time and reducing the costs of plotting, printing, postage and couriers. Using simple tools such as the File Transfer Protocol (Postel

1980) or email, physical document exchange can be replaced quite effectively. This approach, simply involves forwarding information electronically to a number of team members. Though this is a great advantage, it is still limited as data has to be sent to individual participants. This could still potentially be a problem, as for example, some may not be able to handle the type or quantity of data received. Nevertheless, this approach typically works well for ad-hoc data exchange when information does not have to be shared or published to a large number of users.

2.7.2 Common Project Models

An alternative approach that can be used is to maintain a common project model. The idea behind this is that everyone is connected to a globally shared virtual model of the project and work together live adding their contribution to the project. This is different to a coordination model, as that is just a snapshot of other information rather than the live data-set.

An example of the usage of a common project model, would be when the Architect designs the layout of a building's shell and core and after doing this the Structural Engineer is notified and he places the appropriate structural members to support that layout by entering his data onto the same model. In parallel with this, the Electrical Engineer would add his contribution to the model. Similarly, all other disciplines would work together in this way. Even if this might seem ideal, the reality is far from this, with some of the following reasons highlighting why:

- Each participant requires different software that would make the common model unreadable to others.
- Because of quality assurance, a participant should not publish his work to others until it has been approved and checked. Working live means that every time someone creates a building element, it is immediately visible to all others. Quality control procedures require that each design stage is reviewed and signed-off before being incorporated into the approved model.
- It is difficult to track the history of changes if everything is held in one large model.
- Most analytical processes are highly schematic and do not benefit from being modelled as correctly scaled three dimensional objects. For example,

an electrical engineer does not need to see a fully rendered view of a wall socket. Indeed a wiring diagram would be illegible if it were not produced schematically.

- Communication links are not at present suitable for working live across remote sites.

In practice, this does not provide a suitable solution on its own. Nevertheless, as previously suggested, it can be used as a coordination model. In a number of prototype projects that Arup were involved, such as the Genesis Project for British Airports Authority (Suchocki 1998), we found that services coordination and visualisation were ideal processes that benefited from being held in a central 3D model of the building. However, it did not provide a replacement for standard drawing production, review and approval cycle that most construction documents undergo.

It is important to note that the use of a central 3D model should be used as part of an established CAD strategy, where the project management team, working with contractors decide the best medium to describe the information that needs to be exchanged (Excitech 1999).

2.7.3 Document Repositories

One other approach to improving document exchange is to have a Central Document Repository. This is really some disk space at a single location where information is published amongst participants. This differs from the Common Project Model described in the previous section in that information is uploaded for a specific purpose, rather than maintaining a single source of information at a remote site.

In addition to the advantages of electronic forwarding described in section 2.7.1, the following are some notable benefits of this approach:

- Data needs to be published only once in order to be accessible by everybody.
- Data can be published in a neutral format, which is usable by all.
- An instant permanent record exists of all versions of the data.
- Records can be kept when data is accessed.

- Participants actively request to get information from the site, though email can be used to notify participants of changes.
- Each participant maintains the source files within their organisation in whatever format is most appropriate and issues information when ready.
- Other participants cannot modify source files, so quality assurance can be guaranteed.
- It provides the client direct access to up to date information as suggested by Latham (1994).

Whilst a document management system makes sharing information within an organisation easier, a shared project repository allows these advantages to be extended to all participants across the project. Project centred information access replaces organisational and departmental centred information management (Boddam-Whetham 1998).

Though Chapter 3 looks at how document repositories are implemented in more detail, it is worth mentioning that they can be realised in many different forms. At its simplest, FTP sites are used, which form the basis of rudimentary password protected extranets. More elaborately, project-hosting services are often used to manage project information. Interestingly, though many of these provide many additional facilities, it is the simple file sharing facilities that are commonly used. In fact, many of these sites have been described as solely being "glorified FTP sites" (Salimandro 2001).

Columbus, the working solution presented in this report, is mainly geared towards the use of Central Document Repositories as a way of exchanging information between participants.

2.8 Records and Archives

As seems obvious, it is very important that logs are kept of all communication and transmitted information between project participants. Despite a move towards partnering within the industry, the need for adequate record keeping still exists. Moreover, it is a requirement for obtaining quality assurance certification that strict procedures are in place to formalise the exchange of information. The ISO 9000 series of internal standards (Lamprecht 1993), their European equivalent EN 29000 and the British quality management standard BS5750 upon which the other two are based, highlight the importance of record keeping and document control to obtain accreditation.

Handling claims and legal action are also very compelling reasons for maintaining detailed records and archives. In fact, due to the statute of limitations, the risk of litigation makes it necessary to keep records for many years after a project has ended. When considering the type of records to maintain, there are two important aspects to take into consideration: firstly, it is necessary to have a description of what was transmitted, to whom, by whom, when and where. This activity is known as document control. Secondly, it is important to maintain and be able to access copies of the information itself. Without document control, an archive of a paper or electronic document has no value, as there is no record of the recipient or when it was sent. Moreover, without document archives, a document control system will for example show the title, number and description of a document, but the document itself will not be available. Therefore, though the activities are distinct, they are intrinsically linked. This typically results in the need to be able to link document management and document control systems.

2.8.1 Legal Status of Electronic Records

When it comes to litigation in the UK, the legal system has evolved to handle electronic record keeping and document archival. As described by the Building Centre Trust (2001), the Joint Contracts Tribunal (RIBA 1987) has published additional clauses for standard construction contracts where parties wish to communicate electronically. These clauses incorporate the UK Standard Interchange agreement. The main issues that they are concerned with are the validity of transaction logs and security issues such as authenticity and data integrity. It is emphasised that there is a legal requirement for each organisation to maintain records of their own transactions and not to rely on third parties. This is particularly critical when using project hosting services, as it implies that full records still need to be maintained by each participant. One approach that is considered acceptable, if it becomes necessary to validate the authenticity of a document, is to present an audit trail of the transitions as it was passed amongst multiple participants.

The Electronic Communications Act was published in the United Kingdom in the year 2000. This clarified that electronic information is acceptable if derived from a system whose integrity can be irrefutably proven, implying that it must be clearly fully integrated with an organisation's document management procedures.

2.8.2 Document Control

Because of all the legal and quality assurance reasons described in the previous section, it is vital for all participants to maintain appropriate records of all document transactions. Document Control Systems have been used for many years to keep records of all transmitted and received information. Regardless of whether information is sent on paper, as physical objects or electronic files, the principle of knowing what, when and where something was sent to or received from still applies. Before the advent of electronic document exchange, all that was expected from these systems was to provide a catalogue of these activities in a robust, reliable and unchallengeable format, whilst showing a document number or indicating a location where the information could be found. However, as document exchange becomes a digital activity, these systems are now expected to allow direct access to the information. One major question that this raises is: Should a copy of the information be duplicated within the Document Control System?

If the answer is yes, then the following questions also need to be considered:

- Will the amount of recorded information grow exponentially?
- Can the integrity of the database be maintained?
- Will it cope with all the intricacies of complex documents such as reference files or OLE linked objects?
- How will the documents be viewed?

This effectively means that all aspects of document management need to be duplicated in the document control system.

However, if the answer the question is no, then there are other considerations:

- Should there be links to external information on project-hosting sites?
- Can database records be linked to internal copies of the documents maintained by a document management system?
- Can the integrity of the information be guaranteed? For it to be admissible as evidence, both information sources will need to be unchallengeable.

- Will the information be accessible many years after the project has ended? This requires the long-term maintenance and an archival strategy for more than one application.

This highlights that there are potential pitfalls with either approach. An ideal solution would be to maintain these activities separately, whilst closely coupling them so that they handle the issues raised. Unifying document control, document management and project hosting services to achieve this can provide an answer and is discussed in detail in Chapter 3.

2.8.3 Electronic Document Archival

As described in 2.8.1, electronic archives of documents are acceptable if their integrity can be proven. Currently, as most information is transmitted as simple image files or paper, formats such as TIFF or PDF are best suited for long term archival. If necessary, even paper drawings can be scanned back in to show ink signatures. Though not ideal in terms of information sharing, this approach is currently seen as the most irrefutable.

As the industry moves away from paper drawings, it is uncertain as to whether drawing archival will be equally straightforward in future. As information is held in ever more complex document vaults, relational databases and project hosting services, many questions are raised:

- Will those systems still exist in a decade?
- Will they be compatible with future computer systems?
- Will anybody know how to extract and interpret the data?

Even keeping local copies of complex data may be unpredictable for similar reasons. The introduction of object technology can only further complicate these issues unless carefully consideration is given.

2.9 Efficiency Gains

As a general statement, to achieve gains in efficiency within the construction process, there is a requirement to change the way in which project participants collaborate. Some of the changes are organisational and contractual, whilst others are technical. The level of investment required from participants may present a challenge to smaller companies, but the overall result should justify it.

Improving how project information is shared between participants requires the appropriate implementation of document management systems or procedures, the adoption of project hosting extranet sites to exchange data and share models and the use of document control systems for recording document transmittals, which are all technologies that are being adopted within the industry. The next chapter looks at these technologies, emphasising that there is a particular need for them to work together. The Columbus application, which will be introduced in subsequent chapters, also highlights the way in which these technologies need to be used in order to achieve the required efficiency gains.

Chapter 3

Information Management

This chapter looks at the different document handling technologies that exist and how the goal of improving the integration of teams can be achieved by unifying these systems. Each of the technologies is reviewed in detail by describing the benefits and drawbacks that they have. Columbus, the working solution that is presented in this project, is also introduced. Though it is described in detail in subsequent chapters, some of the key issues that it addresses are highlighted initially, before formally introducing the software at the end of the chapter.

3.1 Technology Overview

In order to improve the efficiency of document handling in the construction industry, it is important to consider how documents are created, exchanged, archived and recorded. The technologies that deal with this are document management, project hosting and document control and can be briefly described as follows:

- **Document Management**: Covers the production, editing and manipulation of document files at source level. It is predominantly concerned with the creation and handling of information within an organisation. It may also deal with documents received from other project participants.
- **Project Hosting**: Is concerned with the exchange of information between participants via a shared repository or data vault. These are typically secure sites at remote locations. Project participants can interchange documents and usually communicate via proprietary methods. Typically, logs are maintained of all transactions on the server. Editing facilities are limited, as

the service is mainly concerned with the act of publishing and downloading un-editable information.

- **Document Control Systems**: These formally record and report on the details of document transmittals and receipts. They show when, how, where, by whom and to who documents and information are issued to or received from. These systems provide an essential link between manipulating a document internally and publishing it to other parties.

Each one of these application areas deals with different fundamental issues in electronic document production and exchange. However, it is a common mistake to assume that one piece of software will address all the requirements just because it has strong features in one of these areas (Grinfeld and Grinfeld 2000a).

Columbus is able to provide some of these facilities itself, whilst having links to other systems that provide some of the other capabilities.

The following sections look at each of these areas individually, focusing on the benefits and limitations when it comes to integrating teams.

3.2 Document Management

3.2.1 What is a Document?

Before looking at Document Management, it is important to define what a document actually is. Though this may seem quite straightforward, there are numerous definitions and interpretation of what it can be. According to Levien (1989), a document can be described as "a unit of recorded information structured for human consumption". Michalski (1991), rather elaborately describes a document as "a snapshot of some set of information that can incorporate many complex information types, exist in multiple places across a network, depend on other documents for information, change on the fly (as subordinate documents are updated), have an intricate structure and be accessed and modified by many people simultaneously". In either case, it is clear that a document can be something rather more complex than a simple computer file. For example, a situation where it is difficult to determine which is the document, is in the case of a financial report that has been created as a Microsoft Word file, with an OLE linked Microsoft Excel spreadsheet and which is distributed as an Adobe Acrobat PDF file. Which is the document in this case? One interpretation, is that the source document is a compound document made up of the Word and Excel files, which is then published in PDF format. The use of compound documents and the distinction between source and published documents is one of the most complex issues faced by document management systems (Bannan 1997). When it comes to the use of CAD in the construction industry, drawing files have all these complex issues associated with them. The use of nested reference and overlay files together with search path issues are a clear example of this problem. Moreover, the creation, publishing and distribution of plot, PDF, and DWG files from source documents add further complications.

In addition to documents that are created and published, it is important to consider a further category of documents, which encompasses those that are acquired from external sources. Typically, a large number of documents are received from other parties in paper form. Their receipt needs to be recorded and they have to be redistributed, either by being photocopied, or ideally digitally after being scanned.

3.2.2 What is Document Management?

In the same way that a document can exist in various formats (source, published, acquired, etc.), document management systems can vary greatly in complexity and role. To some (Botterill 1992; Head 1997; Avedon 1997), they deal with the process of storing multiple versions of document images. For others (Masinter 1995; Bartholomew 1999), the primary concern is with editing source documents at the composition stage and keeping track of any changes. And yet for others (Carr 2001; Frappaolo 1992), the concern is with managing documents as part of a workflow, notifying users of modifications and corrections and acting as a knowledge base of corporate information. As Frappaolo (1992) states, these systems can be used "to enhance and preserve the value of an organisation's information resources, and in doing so, optimise and streamline other business functions".

Regardless of the definition chosen, the use of document management systems in its various forms is a rapidly growing part of information systems in business today. It is generally accepted, as Parapadakis (1996) states, that a typical document management system should combine at least the following items:

- Document repository: This is the place where electronic documents are kept.
- Document creation: The ability to add documents to the storage area.
- Cataloguing: A method of indexing and finding documents.
- Document Editing: A way of accessing, modifying and saving documents.

And, in addition, any number of the following can be included:

- Check-in, check-out: This is a locking mechanism so that only one user can modify the document at a time.
- Version control and audit trail: Methods for monitoring the changes that the document has undergone.
- Security: Allows control over which users can access the documents and for what purpose.
- Organisational structure: Methods for organising documents in related groups, e.g. folders.
- Free-text searching: Provides facilities for locating documents based on the text they contain.
- Document attributes: Allow information such as the author, creation date and title to be associated with the document.
- Viewing and Red-lining: Have the ability to preview documents without using the application that created them and be able to "mark-up" documents electronically.
- Routing or workflow: Provide facilities for sending documents from one user to another in a controlled fashion.
- Imaging: Include methods for converting paper documents to an electronic format.
- Publishing: Allow ways of combining documents in coherent collections in order to distribute them to their target audiences.
- Archive Storage: Maintain large volume electronic storage media for permanent archiving of documents.

Based on this description of document management, it becomes clear why there are so many differences in the capabilities and expectations of document management systems. For example, some optional features such as imaging may be emphasised in certain systems, whist being dismissed in others.

In the Contract Journal (1998), an article entitled "From Paper to Electronic Data" was published highlighting various pitfalls that needed to be overcome before document management systems are adopted universally. It was stated in the article that far from lessening the amount of paperwork, document management systems create more, as hard copies are often produced and archived in the traditional way. That aside, other issues such as security, copyright, compatibility with other document management systems and applications such as CAD systems need to be considered. Moreover, the legal implications of whether unquestionable reliance can be placed on electronic information over paper need to be taken into account.

3.2.3 Document Repositories

Though it is not usually apparent to the end user, the way in which different document management systems store documents is an important issue to consider. In general, there are three ways in which a repository can be implemented within them.

- **Embeded documents**: These systems are typically implemented as large databases, which keep complete copies of each document within themselves. Each document is typically stored as a BLOB (Binary Large Object) (Oracle 1999). The main benefit of this approach is that it is easy to maintain referential integrity of the whole system. However, there are some negative aspects: the database will be extremely large, database corruption can easily lead to the loss of documents and it is difficult to handle compound documents. These systems are suitable for applications where small documents are created.
- **Associated documents**: In these systems, a database records all the details of the document, but copies of each version of the document are maintained on a file-system. When a document is required, a copy is "checked-out" to a workspace and "checked-in" when editing has been completed. Though documents are kept on a file-system, as updates occur, they need to be uniquely named to avoid clashes between different versions.

- **Native file system documents**: This is the simplest approach of all. Documents are kept and edited on the same file-system using their original names, rather than being accessed from separate vault. The database provides the complementary metadata for cataloguing, searching and event logging. A document management system can be built on top of the functionality embedded in the operating system, and many groupware products offered today are in fact filing systems with added functionality (Line 1997). The benefits of this approach are that the Document Management System does not interfere with the files, allowing applications to access them directly, which is important in the case of compound documents, backup, archival and document management system corruption problems are less problematic. On the negative side, documents are not secured by the system, though access can usually be controlled by modern operating systems. For example, user permissions can be set on a document-by-document basis on a Microsoft Windows NT file-system. One other potential problem, is that file versioning is not usually supported. However, this may not be an issue if versions normally need to be kept when documents are published rather than when they are edited.

Columbus makes extensive use of document repositories, being able to integrate with a number of dissimilar systems across various protocols. Nevertheless, the preferred repository solution, is to maintain documents on a native file system and link metadata with them.

3.2.4 Document Creation

One basic requirement of a document management system is to be able to create and add new documents. The way that this is done varies with each system, but typically they are based on one of the following:

- **Insert**: This is the simplest approach of all. The user is expected to create files outside the document management system and then add them via an insertion dialog box or command. Though this is flexible, the system does not guide the user in creating the correct document type and may also make it difficult to know where to create and place files prior to insertion in the system.

- **Full control**: This approach is very structured and imposes a series of rules on how a document can be created. These systems typically provide templates, which are copied and named. In addition, the user is then asked to add descriptive information that will allow the document to be catalogued. Nevertheless, this is not flexible enough for creating compound documents.
- **Externally assisted**: In this approach, the document management system works alongside external programs or application macros to create documents. They provide the greatest flexibility, but normally require additional software and only selected applications might be supported. An example of this can be an application that integrates with Microsoft Word and uses the Open Document Management API (ODMA 2002) to create and save files.

Columbus is able to work with all three document creation approaches. It allows documents that have been created externally to be added to the system, it provides templates as predefined files to copy and rename and allows any application to directly interface with itself and create documents directly.

3.2.5 Imaging

Even if a document management system is considered to solely handle the production and manipulation of electronic documents, it will no doubt be the case that on any project, paper documents will also exist. Though it is possible to just maintain a record their existence, it is ideal is to acquire them using imaging technology for the purpose of eliminating hard copies and ease distribution. This is done by scanning incoming paper documents (Koulopoulos and Frappaolo 1995), either as pure graphics in formats such as PDF or TIFF (Warburton 1998), or digitally converting them to text using optical character recognition (OCR) to produce an "index-able" and searchable document. After scanning, it then becomes an electronic file within the document management system. This document life-cycle is described by Mckie (1995) as "receipt, review and ready to file". The degree to which systems are able to control scanners varies, with most providing no support, expecting the user to acquire the document using external software and insert the new documents into the document management system.

Columbus includes a module for scanning paper documents, which can control any scanning device, save the document in PDF or TIFF format and name it in accordance with a predefined convention.

3.2.6 Identifying Documents

Once a document has been created or acquired, it is important to tag and index the information. This is typically achieved by providing metadata about the document. Though metadata is important for text-based documents; it is considered vital for graphical data such as CAD files, as it will be the only information which can be used for searching. Hewitt (1995), states that the following metadata should typically be included with a drawing when it is exchanged: drawing number, revision, date and time received, title, author, package number and date due.

The creation and maintenance of document metadata varies greatly amongst systems. Some allow for only a simple title and some predefined attributes, whilst others present detailed information about the document. Many systems force all document types to share a common set of attributes, which may be inflexible. For example, it makes no sense to show a "pen-mappings table" for anything other than a plot file, an "addressee" is only relevant for a letter, and having a "scanned by" field is only applicable for documents that have been acquired from paper.

Columbus uses metadata as the key technology to allow information to flow between systems. It supports a number of different standards and ways of storing this information. The subject of metadata is presented in detail in chapter 4.

3.2.7 Knowledge Management

In addition to providing a way of organising and accessing information, document management systems are required to find information easily. As systems have become more sophisticated, some have grown into knowledge management systems or have links to external applications with these facilities. This provides an enhanced way of accessing information in the document store, which becomes important as more data is captured electronically. Key requirements of knowledge management systems include:

- **Concept searching**: Allows information to be gathered based on loosely defined questions rather than using simple keyword searches. Knowledge can be extracted from documents that include the essence of the information.

- **Agents**: These are special rules that a user can create to monitor the knowledgebase for specific concepts. When they are found, the user is notified via an email message.
- **Tacit knowledge**: A lot of knowledge exists that cannot be extracted from documents and is only available as an individual's "know-how". Even though answers to specific questions may not be found, knowledge management systems can allow users with similar interests to find each other by suggesting a group of people who may have experience in a field.
- **Spider Information:** Systems can connect to other information bases to gather knowledge. Typically, other web sites or repositories can be accessed and indexed by the system.

Though these points highlight the benefits of knowledge management systems, it is important to consider the following too:

- **Document types**: Many documents are not suitable for indexing, making it difficult to differentiate between information and knowledge. For example, it is difficult to see how knowledge can be extracted from a spreadsheet or a drawing.
- **Indiscriminate indexing**: Sometimes it is difficult to decide when information should be indexed or not. Many systems take everything regardless of access permissions, which means that a lot of information that is confidential or restricted may be included. Indeed, most users are unaware that all the information that they produce is being used as part of the knowledge base.
- **Storage and processing requirements:** Consideration has to be given to the computer system needs when maintaining a knowledge management system. Typically, disk space and processing power requirements can grow rapidly.
- **Future proof:** As the amount of knowledge grows, it is important to consider how it will be stored. Most systems use proprietary closed formats that can tie an organisation to the same system. After a number of years, there are no guarantees that the knowledge will still accessible.

Though Columbus does not have any knowledge management capabilities, its ability to attach metadata and other related information to documents can be of great use when working alongside knowledge management systems.

3.3 Project Hosting

3.3.1 Description

Project hosting describes a service which is made available on a project for the purpose of exchanging information between participants. The terms project extranet and project collaboration site are also sometimes used to describe the same concept and can be use interchangeably.

The most important feature that a project hosting service offers is a document store or repository for sharing and holding files. Other important facilities to consider are notification and messaging facilities between participants, document viewing and commenting/redlining capabilities.

As identified by Doherty (1999), important reasons for using a project extranet, include:

- Fewer communication errors between project team members.
- Up to the minute decisions and collective information relating to a project.
- Less expense incurred with couriers, printing, copying, etc.
- Customised sites for each project and customised access for each user.
- Security.

3.3.2 Application Software

It is important to consider how the requirements of a project hosting system differ from those of a document management system. Document management is generally used within the enterprise to enhance document editing, though some can work across the Internet (Rein et al.1997). They are designed to work directly with source documents on a file system or using a multi-user direct access protocol. Project hosting, on the other hand, is geared towards sharing information across participant boundaries. The use of slower communication links and more limited protocols can make direct document editing more difficult and therefore are really only suitable for publishing information to other project participants.

The complexity of the software used to implement project hosting varies enormously. Some applications are quite ellaborate and provide powerful features, whilst others offer only basic facilities. At its simplest, many organisations just use email to exchange information. A recent survey (CAD Spaghetti Magazine 2001) found that 98% of users that exchange documents electronically use email as the medium to do so. Email, though adequate for exchanging information does not provide a suitable document store. The most common single solution for exchanging and storing shared information is to use an FTP server. The File Transfer Protocol (Postel 1980), which has been in use for over twenty years, allows users to exchange information using a graphical or command line client program. FTP is recognised as a mature, established and reliable technology and the procedures for using it in construction projects are well established (Hannus 1996), with many project participants maintaining an FTP site as a matter of course. However, as Maher (2000) states, though FTP sites are effective, setting them up and administering them can be a tedious task. Because of this, in the early 1990's, several software houses addressed the challenge of providing more sophisticated "collaborative software" (Hapgood 1998). These applications are typically accessible via HTTP (Berners-Lee 1993), using custom client applications or Java applets that run directly from the site. Custom client applications are more powerful, offering viewers, red-lining facilities, powerful upload and download tools, but need to be installed and licensed on each workstation. Java applets, on the other hand are more restricted but they typically can be used with a minimum overhead from any computer that has a browser. In addition to providing facilities for sharing documents, these tools may provide additional facilities such as project diaries, discussion forums, contract address books, project visualisation pages and an email system. However, in reality what often happens is that only a small subset of these features is used and simple, effective and reliable tools may often be preferred. In a recent report, the Construction Industry Computing Association stated: "The main beneficiaries of an extranet are either the client or the main contractor, who usually pay the direct costs and the view of designers was that the current offering is too 'heavy' and a simple FTP site is all that is needed, rather than the extra features that project-hosting sites are selling. For design organisations with established document management systems, an extranet imposes a double handling of the drawing environment. There is no reluctance to exchange project information electronically, it is the systems that surrounds such exchange that is the subject of contention and debate" (CICA 2002b).

Though Columbus can work alongside elaborate project hosting systems, it provides a number of basic capabilities for information sharing and document exchange using the FTP protocol. This satisfies the needs of many users, who consider this as one of Columbus' most useful features.

3.3.3 Accessibility

The nature of project collaboration software implies that users will be located at remote locations to the project-hosting site. Because of this, data communication links need to be considered carefully if the project is to be successful.

Before the massive update of the Internet, the general approach to information sharing was to provide leased lines or ISDN connections directly with a central server (Knutt 2000b). This provided fast and reliable dedicated links for each user, but proved expensive and feasible only for large organisations. Currently the trend is to use virtual private networks (Morgan 1999) or extranets (Franklin 1997; Wailgum 1998) to link participants. This allows private communications between users using Secure Socket Layers (Gay 2000) or password protection over the Internet with the minimum amount of effort (Grabowski 2000). Any organisation can connect for the cost a local phone call to any server and linking together participants electronically is no longer an issue, as the Internet can make them all accessible to each other (Wolton 2000). As higher bandwidth technologies such as Analogue Digital Subscriber Line (ADSL) develop, it becomes more practical for smaller organisations to also benefit from project hosting. Current developments in communications technology can provide a digital environment for collaboration, leading to new ways of conducting projects and acting as the key enabler for virtual teamwork (Line 1996).

3.3.4 Document Store

The most important element of a project hosting service is the document store, also referred to as the document vault. The collection of project documents is the knowledge base of design decisions made through the project's life (Amor et al. 1996). As described in chapter two, project participants need to be clear as to how the document store is to be used; which may be as a repository for sharing published information, maintaining a common project model or working on live data. Due to the limitations in connection speed and the protocols used, the possibilities for how different documents types are handled may be limited. Compound document assemblies, reference files and complex document dependencies may not be supported. Record locking, as opposed to file locking, may also not be possible, limiting multi-user access to documents such as Microsoft Access database files. Because of this, the main use of project hosting systems is to provide repositories of published documents. Laiserin (2001b), however, describes this "post-and-host" approach to collaboration as insufficient for architects and engineers collaborating during the design phase, where information needs to be exchanged constantly as the design is refined.

The way in which documents are organised in the document store will vary according to the standards adopted on the project. One possible layout proposed by Hannus (1997), is to create subdirectories for each project participant and allow them to upload their information there. Below these directories, there can be issue areas that are time-stamped and identifiable according to its purpose. These directories can be written to by the participant that owns them, but are read-only to all others. This structure has been widely used on project hosting sites that Arup have been involved in. This is the approach suggested in the Columbus documentation and is recommended as a filing system when used either with elaborate project hosting sites or simple FTP sites.

3.3.5 Viewing and Mark-up

One important requirement from a project-hosting site is document viewing. As the site gathers documents from diverse sources, it is important to able to view them without purchasing and installing additional application software. The viewer requirements within a project collaboration site are similar to those for document management and as such must handle a multitude of document types and formats. In the construction industry CAD is crucial and it is important to bear in mind that simple viewers may struggle with such complex information. Drawings, for example, may be made up of an assembly of reference files that might be held in different folders and locations. In addition, there are some specific CAD formats that viewers should handle. The most popular format for publishing information is the Hewlett-Packard Graphics Language (HPGL), which is the standard used with large format plotters. AutoDesk's Drawing Web Format (DWF) is also becoming increasingly popular with documents that are intended primarily for on-screen access (Knutt 2000a). Support for industry standard drawing formats such as AutoDesk's AutoCAD and Bentley's MicroStation formats are also particularly important when exchanging drawings that need to be edited. Additionally, most viewers need to be capable of viewing a number of common file formats such as Microsoft Word, Microsoft Excel, Adobe PDF and the more popular image formats. Columbus includes powerful viewing capabilities, supporting over 250 file formats.

As well as viewing documents, there is an expectation from project participants that facilities for mark-up or redlining exist. These allow reviewers to make comments on documents that can subsequently be seen by other project participants.

The approach to redlining varies between systems; however, most will generate a raster image file that is overlaid with the original document when viewing. The main problem with this, is that the comments can only be seen if the same viewer is used rather than the application that originally created the document or a different viewer. Once the document has been redlined, the software must be capable of uploading the comments, recording them with the original document and notifying other participants of these comments. Columbus' redline capabilities allow basic comments to be added to a file in graphical form which can be exchanged independently of the source document.

Redlining has been poorly received within the industry for a number of reasons. Though exchanging information electronically and minimising paper use is recognised by all as an ideal goal, correcting information and coordinating between different drawings becomes an impossible task on a small screen. Some users choose to plot the information, add their comments on paper and scan the document back in, therefore bypassing the redlining facilities. It is also important to consider the way in which comments are handled by project hosting systems, and redline information should not subsequently becoming detached or associated with the wrong version of a drawing.

3.3.6 Existing Project Hosting Systems

There are a number of project hosting systems available for use within the UK construction industry. Though not so long ago, the offering was much bigger, the market seems to have been consolidated into a few key players. In a review of the more popular collaboration tools, Kernon (2000) described the following facts about each of them:

- **ProjectNet** : Considered by many as the market leader, this system is offered by vendor Citadon. Extranet providers Cephren and Bidcom, both originally founded in 1997, merged in October 2000 to form Citadon. UK projects are hosted at a high security Internet data centre in England. The cost is based on storage, with 500Mb of data costing £833 per month.
- **Viecon** : Launched in 2000 by CAD vendor Bentley, it is hosted at a US site. Viecon has great potential for MicroStation users and for integrating other CAD systems.
- **ProjectPoint** : This system is offered by Buzzsaw and is more commonly known by that company's name. Buzzsaw was founded in October 1999 with financial backing from AutoDesk. The typical storage costs is $200 per month for 200Mb and servers are in the US. Because of Autodesk's influence, the product is very popular with AutoCAD users.
- **Project Information Warehouse** : Released in 1998 by Building Information Warehouse, it is hosted at a secure Internet data centre in London. The estimated cost for running a £5M project over five months is £12,000.

These are just some of the project collaboration tools available, showing the type of services offered and the cost of the systems. Reviews of project hosting services are often published in journals and across the Internet and one of the main resources for project hosting information is Dr. Joel Orr's Extranet World web site (Orr 2002).

Though Columbus provides basic project hosting capabilities in its own right, it also able to work with some of these and other systems. This is done either using specific application programming interfaces or by implementing a module within each system to import Columbus document property information.

3.3.7 Limitations and Concerns

There are a number of concerns from users as to whether project-hosting services will offer what the Architecture, Engineering and Construction (AEC) industry expects (Augenbroe 2000). Though Orr (2000) emphasises that it is possible to use generic tools to produce something that looks like an extranet, commercial products have many subtleties that are not easily emulated and what works well in one context may be clumsy in another. It is up to buyers to carefully define their needs and match them to the appropriate project-hosting system. As none of the systems available were able to satisfy all of Arup's requirements, the Columbus application was extended to provided project hosting capabilities.

It is also very common for project participants to have systems imposed on them by a client and as project participants move from project to project they find themselves using different systems. This is not a good situation and ideally long-term relationships should be established between vendors and participants so that they standardise on a system to cover a whole portfolio of projects.

There are also major concerns about the maintenance and ownership of project hosting systems. This, as already covered in previous chapters, is because of the possibility of litigation and lack of control over the document store. If an external organisation or third party manages the document vault, access to information can be seriously compromised. Because of this, project participants should download and archive any information that they use, whilst maintaining their own copies of uploaded documents. This imposes further document management requirements for project participants and is a reason why some organisations have been discouraged from using project-hosting systems. Columbus' capabilities in this area mean that information from project hosting sites can be integrated seamless with its document management features and information can be archived reliably for future reference.

When using project-hosting systems, close-working relationships can be established between project participants by forming "virtual teams". In a case study by the Building Centre Trust (2000), it was found that though many participants appreciated the rapid exchange of up-to-date information, it was important that professional responsibilities were not compromised if, for example, too many revisions were forwarded too quickly. This highlights that a formal recording and issuing procedure should not be bypassed.

The concerns about the existence of a multitude of project-hosting systems are also further accentuated because they normally store their information in proprietary closed formats, which may make it difficult to access information in the future.

A number of these issues were highlighted in a brainstorming debate transcribed in "Construction IT in practice" (Bunn 1998). Though it is clear that clients want to see better efficiency from interoperable project collaboration and better processes of the kind that Sir John Egan recommended (Egan 1998), many technical issues need to be addressed before it becomes possible. Ideally, project hosting systems should store information in an open format that is accessible from other systems and can use a universal data exchange format to interoperate with each other. This has been one of the key requirements of Columbus, which is apparent in the way the software was designed and works.

3.4 Document Control

Document Control can be defined as the process of recording, tracking and reporting on the existence, movement and history of documents that are sent to or received from other parties. Keeping formal transaction records can prove invaluable in the event of a claim or litigation. It is important to distinguish clearly when documents can be transferred in an uncontrolled manner between project participants and when they need to be included in a formal issue, which requires detailed records to be kept.

Tracking incoming and outgoing documents involves recording the movement of information between participants. This does not necessarily mean that copies of the actual information are maintained within the document control system. On most occasions, just a record of the document's basic metadata is made. This can include the document's title, number, status, purpose of issue, originator and recipients. If there is a need to access copies of the documents, then links can be made to the document management system, where archived information may be retrieved and viewed.

The purpose of recording all incoming and outgoing document information is to make it possible to query it at some later date if required. This is typically achieved by running database reports or queries using pre-defined forms.

Columbus is able to interact with document control systems by sharing document information in a common metadata format.

3.4.1 Document Control and Document Management

There is a clear distinction between the roles of a document management system and a document control system. The former deals with the creation and manipulation of source documents, whilst the latter, as described, is responsible for recording information exchange. Nevertheless, there are a number of areas where the two need to work together. Typically, documents are published in formats that are very different to their native application files. For example, an AutoCAD drawing may be published on paper and a Microsoft Word report may be shared in Adobe PDF format. Though the information may be in different formats, they are directly related and should have the same common information; this means that there should be a mechanism for using the same metadata between them.

Another area where both systems need to work together is in transferring published documents to other parties. This is typically the responsibility of a document management system, but a record of the transaction needs to be made in a document control system.

Historically, both of these activities have been kept separate by maintaining completely independent systems. For example, though drawing information such as titles and numbers have been accessible within a document management system, a large number of technical clerks have still been employed by organisations to re-enter the information manually into document control systems.

Columbus is able to transfer metadata to document control systems so that common information is shared. It is also capable of publishing documents to other project participants and notifying the Document Control System for record-keeping purposes.

3.4.2 Transmittals

When transferring documents to other parties, it is important to accompany the information with a report that lists what is being sent. This is called a transmittal sheet and has been in use well before the introduction of computers. This sheet includes basic metadata for each document that is being sent, states the purpose of the issue and lists the recipients of the information. Though they have traditionally been used to list physical items such as paper drawings, they are now regularly used to index electronic information. For example, electronic documents may be transferred via FTP or on optical media and listed in the transmittal. When transferring documents electronically, additional information such as the computer file name and file size is typically included. The use of compound documents such as AutoCAD assemblies of reference files can further complicate the transmittal.

When incoming documents are received by an organisation, it will typically be the job of a document controller or technical clerk to process them. This involves recording their receipt by entering the information that is printed on the document or transmittal sheet and notifying any interested parties. Very often, due to the nature of the systems used, this is a very inefficient manual process that requires information to be typed in. Even if the transmittals themselves may sometimes be delivered electronically, if they are in a format such as Adobe's PDF it will still be necessary to re-enter the information. An ideal solution would be to provide transmittal information in a standard electronic format that can be imported directly into the document control system, which is one of the goals of this project. If documents have been received in paper form, the ideal would be for them to be scanned and saved in a standard file format such as multi-page TIFF or Adobe PDF. This can then lead to a workflow process being initiated to dispatch the documents to all the necessary recipients. Columbus can generate transmittals of any activity in a format that is suitable for human use, or more appropriately to be used by other document handling applications.

3.4.3 Integrity and Maintenance

Document control systems are typically implemented using a database and maintain complex relational links between tables of information. Tables typically exist for document information, revisions, recipients, and originators. As this information grows and becomes critical to the project, it is important that it is maintained using a robust system. This has typically meant that document control systems are based on one of the major proprietary databases. Even when this is a common solution, it is important to consider that unless the database is properly maintained and is easily accessible, all the data may be left unusable in the future. It is possible for information to be required up to ten years after a project has been completed; in this scenario, unless personnel are available to reinstall, restore and run the document control system, it may be impossible to retrieve information at a later date. Because of this, it is important that alternate snapshot records are maintained of transmittals, either in paper form, or ideally in a neutral format such as text, XML or PDF files.

To preserve information for future use, Columbus exports information and documents to suitable archive media such as tape or CD-ROM. In order to do this, it uses metadata to identify any document dependencies and write out properties to assist with future identification. The usage of neutral non-proprietary data formats is considered essential for this purpose.

3.5 Unified Systems

As was presented in detail, document management, project hosting and document control are considered key information management technologies in projects. It is clear that a lot of the data that they deal with is common and improving the way that information is shared between them can lead to big efficiency gains.

3.5.1 Identifying the Requirements

Ideally, it should be possible to have a unified method of handling documents from the point of creation until the final project archival stage, to meet the needs of all project participants. Achieving this, however, can be difficult as each system has specific design requirements and is typically used by different people within a project team or organisation. Though it could be done within a single organisation, it is a complex task to unify systems across organisational boundaries. One obvious solution is for everybody to use a single monolithic system that carries out all the tasks. This is not normally possible as few systems exist that handle all the activities that are required and it is also impractical for organisations to constantly change applications between projects. Another solution, which is much more feasible, is to have standard interfaces between all the document handling applications and let them exchange data according to agreed standards. For this to work, the documents themselves and the information about them must comply with agreed standards. CAD drawings, for example should following guidelines for document naming and drafting standards such a layer naming, font usage and text heights. It is also important that documents' metadata, such as the title or revision, should be shared in a format that is suitable for electronic use by the recipient.

Linking information management systems remains a key issue as stated in a Construction Industry Computing Association report: "The need to develop a common interface to the various systems requires attention" (CICA 2002a). Ultimately, a solution that brings together the capabilities of all information handling technologies, either by providing them in a single system or combining different applications, should provide the following key facilities:

- Structure documents within the organisation, allowing easy editing, document creation, cataloguing, searching and viewing.
- Maintain documents in a non-proprietary format.
- Maintain descriptive metadata for documents that is detached from the physical data store.
- Issue information to other project participants in a seamless manner, providing full descriptive information about all transmitted information.
- Implement a simple project hosting system based on standard protocols such as FTP or HTTP.
- Work with other proprietary project hosting systems.
- Work with document control applications, tracking information sent to and received from other project participants.
- Exchange transmittal information in a neutral format.
- Keep record copies of documents and archive information reliably for long-term retrieval.

The aim of Columbus is to meet these needs by providing a unified solution linking all document handling technologies in a seamless way.

3.5.2 Finding a Solution

Grienfeld and Grienfeld (2000b), emphasise that companies should not be dazzled by expensive technology, but should analyse their own requirements carefully and seek tools that their employees can use with minimum training. For most organisations, the idea behind using a document management tool is not to impose revolutionary changes, but to implement some sensible rationalisation in the handling of information. With this in mind, in 1995 Arup looked at their requirements for implementing a solution within the firm, taking into account many of the issues raised. It was considered essential that the system would work alongside the current document control system and it should allow data to be shared easily with other project participants.

Arup evaluated a number of products that could provide an answer to their needs. Most were problematic: they were cumbersome to work with, were not really suitable for CAD use, stored information in proprietary vaults and had minimal document exchange facilities. Of all the solutions that were considered, the best document management capabilities were found in Documentum (Documentum 2002); however, with an initial cost of £2,000,000 for a pilot test, £6,000,000 for a full roll-out and £500,000 per annum for maintenance, it was considered an expensive option. This, coupled with the fact that only some document management requirements were addressed by the application, meant that working alongside document control and project-hosting systems would still be a problem.

The outcome of this review was that none of the commercial solutions satisfied the requirements of the firm and the Arup board rejected them all. Unless something better and cheaper appeared, the firm would hold back on the use of these systems.

3.5.3 Introducing Columbus

Whilst Arup were evaluating their options for implementing a document management system, my team leader and I visited the design offices of Rail Link Engineering (RLE) in 1995 to see how they were working. RLE were responsible for the design of the Channel Tunnel Rail Link and were using Documentum as a document management system. What we found was that they were mainly interested in indexing drawings according to their metadata, which was automatically extracted from the drawing title blocks. There was no use of the document management check-in/check-out and versioning facilities due to problems in handling reference files and storage requirements.

After careful consideration, we both reached the conclusion that a simple file system browser with facilities for showing metadata would have satisfied the requirements of the RLE project team and be ideal for Arup. Following this approach, and trying to meet the needs and requirements of the firm to solve the problem, I began developing the application that is now known as Columbus.

Having produced a prototype version of Columbus, it was released for comment to a selected group of users. On the basis of their feedback, the firm provided the required funding to progress its development. Though the company's policy was, and still is, not to develop software if a commercial alternative exists, when faced with the inability to find a viable commercial document management solution and considering the costs involved, it was decided that Columbus was a notable exception.

Columbus is now used throughout Arup and in over 16000 organisations in 165 countries across the world, making it one of the most popular document management solutions. Grienfeld and Grienfeld (2000a) describe Columbus as an example of a system that has emerged due to the lack of clearly defined market standards. Though introduced as a document management solution, Columbus has evolved to work with and provide project-hosting facilities (Knutt 1998).

The Royal Academy of Engineering (1996) stated that the industry should support initiatives that promote and develop the concepts of team working and identified a need for low price CAD and simple IT solutions. In accordance with this, the success of Columbus is based on its simplicity, non-proprietary way of storing information, links to other information management systems and its handling of metadata.

Following a review of metadata technology in chapter four, the remainder of this report will concentrate on the Columbus development, presenting a user perspective of the product, its architecture and design and a review of how the product has been accepted within the industry since its release.

Chapter 4

Using Metadata

As suggested in previous chapters, the use of metadata is proposed as the way of unifying information exchange between document handling technologies. This chapter takes an in-depth look at document metadata, describing what it is and how it is used, defined and formatted. Particular consideration is also given to the way in which it is handled by applications, looking at how it is stored, created and exchanged. The application of metadata is then discussed as a solution, which leads to the presentation of various XML schemas and a description of how the Columbus program uses this technology to improve information exchange between systems.

4.1 Metadata Basics

4.1.1 Metadata Description

Metadata can be defined as information about information. For many years it has predominantly been the domain of librarians, who have used metadata to catalogue publications. Indeed, the usage of metadata may be called "cataloguing by another name" (Milstead and Feldman 1999). Nowadays, the distinction between document management and digital libraries is becoming blurred. Documents created within an organisation may be catalogued, indexed and exchanged just like any publication. A key function of any document handling system is to classify and find a document easily. This is done by associating additional properties, attributes or custom variables with the document. This information is the metadata that is associated with the document.

The importance of metadata can not be over-emphasised. Even when a lot of documents are textual and information can be extracted from their content, many others are not. This is particularly true in the construction industry, where examples of this include graphical, CAD and engineering file formats. The only way of indexing and searching for these documents is by using metadata.

Metadata consists of a series of named elements which describe an object and its properties. Examples of elements used for cataloguing purposes include title, author, date, document identifier and subject. Each of these will have a value that can be combined to uniquely define the essence of the object.

4.1.2 Metadata Application

Metadata can be used for a variety of different purposes. It may be used to catalogue documents, to track their usage and changes over time or as a mechanism for transferring information between applications.

- **Cataloguing:** By using metadata, document handling systems can identify and categorise information easily. It assists in the description, organisation, discovery and access of information resources (Masinter 1995). To achieve this, documents need to be uniquely identified regardless of whether they are used within workgroups, across the enterprise or shared between project participants.
- **Finding Information:** Metadata is also used to find information based on loosely defined properties. An example of this is seen with search engines and crawlers on the Internet. However, it can also be used as a way in which proprietary applications can extract information from remote systems. Data mining programs can map database fields across systems using metadata. These techniques are also applicable when various document handling solutions need to be combined. The intellectual property in the information is often locked up in documents which are often hidden deeply in poorly named files and spread over many servers. This is particularly relevant when information is left for many years after projects have been completed. The usage of document metadata can greatly help with identifying and preserving the information.
- **History Tracking**: Tracking can be used by document control systems to maintain a history of documents as they are sent between project participants. This provides an important audit trail for future reference.

Most of this information is stored with each system but may need to be exported in a neutral format as a permanent record. A history of local edits may also be kept as associated metadata whilst documents are being edited.

- **Linking Systems**: Metadata can be used to link together separate applications. Though they may each have their own ways of storing data, metadata generated in temporary files or streams can be used to share information. In order to be able to exchange metadata, it is essential that it is transferred in a format that both systems can understand. The ideal solution is for all systems to use a single standard generic definition. This, may not always be possible, as each format may target different application areas, making them incompatible. The use of metadata as a temporary medium for information exchange can be seen when documents are uploaded to project hosting systems by a document manager. In this case, metadata is typically imported into the system's database and subsequently discarded.

4.2 Standardised Metadata

Metadata is made up of tagged information that other applications can interpret. In order for the data to make sense, it must be written in accordance with some predefined rules than govern its usage. Just like any other language, standards exist defining the semantics, syntax and structure for metadata usage (Masinter 1995). In addition, there are standards to allow different metadata mechanisms to be combined as a compound metadata resource.

4.2.1 Semantics

The semantics of a metadata item is its meaning. Regardless of how the information is stored, encoded or tagged, if two properties have the same meaning, then they are semantically equivalent. For example, a property may be labelled as *creator* in one format or *originator* in another, yet they may have the same meaning. On the other hand, two tags could semantically be different even if they are labelled in the same way. An example of this is the *date* property, which could be interpreted as a creation date in one format or as the last modified date in another. This clearly highlights that the precise meaning of the attribute has to be defined so that information cannot be interpreted out of context. When metadata is semantically equivalent, it is possible to exchange information between systems that use different formatting.

It is important to highlight that though there are automated ways of checking the structure and syntax of metadata using software, semantics can not generally be validated, which means that care should be taken with the meaning of metadata elements. When considering the use of metadata, it is necessary to establish the vocabulary for the data set that will be used to achieve the required exchange. Though it is easy to define a protocol for each situation, this does not help in allowing other applications to make use of the data; therefore standardisation is required.

A number of initiatives exist which have tried to standardise the semantics of metadata. As they are not specifically concerned with the format, they are suitable for use with any representation. Examples of semantic standards include: MARC, Dublin Core, IAFA, TEI, URC (Heery 1996). The basis for the semantic standard used in this research is the Dublin Core.

The Dublin Core (Weibel et al. 1998) is a standard definition of semantic elements developed in 1995 to facilitate the discovery of resources. It can be applied across a broad range of fields, from simple card based cataloguing to web based document exchange.

The Dublin Core Element Set, as defined by version 1.1 of the standard (DCMI 1999), establishes fifteen metadata elements that can be used as a common attributes during information exchange. The meaning of each of these elements is defined very precisely using ISO/IEC 11179, the standard for the description of data elements. The fifteen elements, together with brief description are as follows:

- Title: A name given to the resource.
- Creator: The entity responsible for making the content of the resource.
- Subject: The topic of the content of the resource.
- Description: An account of the content of the resource.
- Publisher: An entity responsible for making the resource available.
- Contributor: An entity responsible for contributing to the resource.
- Date: Date associated with the creation or availability of the resource.
- Type: The nature of the content of the resource.

- Format: The physical or digital manifestation of the resource.
- Identifier: An unambiguous reference to the resource within a given context.
- Source: A reference to a resource from which the present resource is derived.
- Language: The language of the intellectual content of the resource.
- Relation: A reference to a related resource.
- Coverage: The extent or scope of the content of the resource.
- Rights: Information about rights held in and over the resource.

Opinion on the usage of the Dublin Core is divided into two separate camps, the Minimalists and the Structuralists (Weibel et at. 1997). Minimalists believe that the Dublin Core should be characterised by a simple single set of metadata. This would mean that creation and usage of metadata could be widely supported by all applications and that this is the only way of maintaining semantic interoperability across domains. Structuralists, on the other hand, accept that semantics may drift and state that greater flexibility to extend or qualify elements is required to make metadata useful. With only fifteen elements, there is often a need to supply multiple values for each element and these need to be unambiguously differentiated from one another. This differentiation is usually done using qualifiers, which greatly extends the usage of metadata.

4.2.2 Syntax

Having looked at how the semantics of metadata deals with the meaning of information, it is also important to consider how that information can be used.

The syntax of metadata deals with a number of issues including: data types, allowable values, the ability to express optional or repeatable sequences and multi-language support.

The data type defines the kind of information that can be stored. This can either be in terms of simple primitive types, such as strings, integers, floats or more complex structures such as lists or sets.

The values that can be used for metadata will either be from a fixed or extensible list or as uncontrolled free text. This is particularly important when data integrity needs to be maintained between various systems and metadata values are used as database keys. Because of this, the usage of free text may impede the correct identification and usage of these systems. For example, it is important to identify in a system if the title of a drawing can change between revisions. If this is the case, then free text can be used as descriptive title metadata, but it can not be used as a key for selecting drawings. If, on the other hand, the title can only be used if it is maintained in a drawing register, which all revisions share, then this can be done. Strict application of fixed value syntax can make systems more restrictive, but it can allow them to be better integrated.

The language and character set of metadata presentation also needs to be defined, as this can hinder data exchange. Unless a specific language is defined, the representation and semantics may restrict interoperability (Baker 1997). An example of this, is the representation of dates, which will vary between countries if free text is used.

Syntactical issues are normally formalised as part of the structure of metadata and hence no specific standard is described here. However, it is important to highlight that metadata syntax is considered separately from its structure, as it is independent of the layout that the data will take when it is represented. This allows flexible data interchange between different approaches to structuring metadata.

4.2.3 Structure

Though the meaning of information can be defined in accordance with a standard and consideration given to the syntax that is permitted, it is still necessary to output metadata in a way that applications can interpret. The layout of metadata within a file or data stream is the structure that represents it.

The format that the data will take, the actual names that the attribute tags will have, the implementation of the syntax and other presentation details can be defined in a standard. This presentation format stipulates the constraints that metadata must adhere to in order to be considered valid. If the structure is correct, a parser may be created in any application to interpret and disassemble metadata, therefore understanding how it is composed. This does not necessarily mean that the semantics are correct, but just that the data is laid-out in the correct way using appropriate identifiers and sequences in accordance with the chosen standard.

Numerous standards, known as mark-up languages, have been defined to structure information and amongst the most important with the construction industry are STEP, SGML and XML.

STEP (Wilson 1998), the Standard for the Exchange of Product Model Data, has been ratified as International Standard ISO 10303: *Industrial systems and integration - Product data representation and exchange,* and is widely used throughout the engineering world. Important collaboration initiatives within the industry, such as CimSTEEL or the IAI Industry Foundation Classes, have been defined using STEP. Though it is still very popular, it is now being overshadowed by other mark-up languages.

One of the most versatile mark-up languages that is exists is SGML (Goldfarb 1990), the Standard Generalized Mark-up Language. It is an international standard (ISO 8879) which was defined in 1986 and specifies a common method of describing the structure of the information. This is one its greatest strengths and because of this, has been used as way of unifying the increasing number of mark-up languages that are in existence. By using SGML, it is possible to re-express essential information from one language into another so that it can be transferred between applications (Burnard 1991).

SGML supports an infinite number of structures and allows the representation to be independent of any software or hardware constraints, and since it is defined as international standard, the result is highly portable. SGML deals with information in three layers: structure, content, and style. The structure is defined by a Document Type Definition (DTD) which, like a database schema, shows the relationships between the elements to be represented. The content is the information itself, which is surrounded by tags forming the elements. Tags mark the beginning and end of each element's content. Style is related to the way in which the information will be presented to the user. More often that not, SGML does not concern itself with this and it is left to proprietary applications to display information as they find appropriate. Nevertheless, in an attempt to provide some standardisation, the Document Style Semantics and Specification Language was created for use with SGML information. The main benefits of SGML that can be highlighted are reusability, information longevity, portability, improved data integrity and its capabilities for being shared (Arbortext 1995).

The fact that SGML is such a generic and powerful language means that any information can be represented in any way. However, this flexibility has also made it complex. The result of this, is that it can sometimes be difficult to interpret a lot of the information that is represented, requiring complex and elaborate parsing techniques. One solution to this was to define a simpler subset of SGML.

XML, the Extensible Mark-up Language (Graham and Quin 1999), was defined in 1996 to provide a simpler alternative to SGML. Guided by their experiences with HTML, the SGML subset for web hypertext mark-up, the designers added additional constrains to the way in which information can be represented, making it much easier to use, learn and parse than SGML. They kept the best features of the language, such as structure, validation and extensibility to produce a simple yet powerful solution.

One further development has been the introduction of XML schemas; these are an alternative form of SGML DTD files, but with numerous advantages:

- Schemas are extensible, allowing others to add properties to an existing definition.
- They are written in XML, making them easier to understand and create.
- They support complex data types and namespaces.

Because of the ease with which XML can be created and used, it has now become the most widely used format for exchanging information between systems, and is considered to be the de-facto standard language for data exchange. This is also in part due to the proliferation of standard schemas, which are easily shared.

Though schemas have been defined within the construction industry to ease information exchange, they typically focus on a particular aspect. AecXml (AecXML 2001), for example has been defined by AutoDesk as a way representing information in CAD drawings but, justifiably, deals with the complex details of describing each building element. Equally, DesignXML, defined by Bentley, has appeared as a proposal from a rival vendor. In both cases, none of these solutions offer a simple schema to allow basic Dublin core attributes to be exchanged and it is difficult to find a suitable schema that allows all types of documents to be shared simply. A similar scenario can be found when looking at schemas proposed by other vendors in the purely document management, document control or project hosting fields.

4.2.4 Compound Formats

Though a particular semantic, syntactic and structure standard may have been chosen in a metadata definition, it may not always be enough to use a single format. In order to allow different providers to add their own content or to store metadata that is defined in accordance with different schemas, the use of multiple metadata sets is required. To assist with this, a number of formats have been defined which act as containers for different streams of information. These can be used to package up descriptive metadata from different sources alongside the documents.

One well known standard is the Warwick Framework, which originated from the work on the Dublin Core. As the developers established, there was a need to find a way of defining how the Dublin Core could be combined with other metadata sets. A core requirement was to preserve each individual item's integrity, whilst separating the realms of responsibility of the distinct data. The resulting container architecture, the Warwick Framework, allows for aggregating logically and physically distinct packages of metadata. (Lagoze 1996).

Though the Warwick Framework represents an important way of externally gathering metadata together, other standards do exist. For example, a format, which became popular over the Internet is the Channel Definition Format (CDF)(Castedo-Ellerman 1997). Developed by Microsoft and its partners, it is an application of the XML standard providing a mechanism for defining groups of objects that can be pushed or pulled as a unit, unifying the pieces of a document into a single entity rather than separate component files.

4.3 Metadata Creation and Storage

In addition to looking at how metadata can be defined and formatted, it is also important to consider how it is created and stored.

Generally, metadata is either created either by the application that handles the data file or by a different piece of software. Some applications maintain basic information about the document such as the title, originator or subject which are semantically defined by the Dublin Core. However, in addition, other information is typically held which is specific to that software. Examples of these include the page count and template name which are kept in a word processing document. This information, though valuable when considering other word processing documents, may be of limited use when searching alongside other types of files.

Many applications create metadata for their own internal use and expect that it will not be read, shared or modified by others. Unless the file format has been published as an open standard, it is typically necessary to reverse engineer how the information is stored. However, it is important to consider that the Digital Millennium Copyright Act (DMCA 1998) in the United States may consider reverse engineering of file formats to be criminal offence. This may discourage access to information which is held in unpublished file formats and reduce interoperability between applications.

Metadata can be created or managed by applications other than the document creator. These can be in the form of an add-on customisation module or a completely detached application. In these cases, it is important that the software responsible for the metadata is accessible when required as otherwise this can create problems with document handling.

There are two generally accepted approaches when linking metadata to documents (Hillmann 2000). The first is embedding the metadata within the resource itself and the second is to maintain it in some external form. Many metadata standards do not specify the linkage, allowing the implementer to store information in either way.

An example of embedding metadata elements within the resource, is the way in which a Microsoft Word document holds document information such as the document title, subject, author and category. In this case, it is held in the Structured Property Data format, also known as Compound Document Properties (Microsoft 1998).

The major benefit of having the information within the document is that it can not become detached and mislaid when used outside of a document handling system. This is a great advantage when sharing information between dissimilar systems. The disadvantage is that the information is generally held in proprietary formats which may hinder information sharing.

The second approach to storing metadata, is to keep it separate from the document itself. This is typically the case when a document is under the control of a

Document Management System, where the information may be held in a database which keeps a link or reference to the object being catalogued. The advantage of this system is that information is entered, maintained and queried independently of the capabilities of the editing application.

However, when it is necessary to share documents, it can be difficult to ensure that the metadata is correctly transmitted with the document. Additionally, it is also important to ensure that it is constantly updated and paired with the correct document. Moreover, the actual location of the metadata should be considered. Though metadata that is created by the application that edits the document tends to be stored in the same document file, this is not always the case and it could, for example, be stored in an external database. Equally, metadata defined by an add-on or external application may be written within the document itself if the file format permits it.

It is also important to consider that as metadata is moved between participants and different document handling applications, this can result in metadata being transformed from one storage scheme to another. This can result in syntax and structure changes but as the semantics remain equivalent, it allows the essential value of the properties to be preserved between systems.

4.4 Unified Metadata Solution

The idea suggested in this work is to use metadata as way of linking together document management, document control and project hosting systems when used within an organisation or between project participants.

Though a single metadata format could be defined, this is not always ideal as a certain amount of flexibility should be allowed. The ideal way to achieve this, is to allow various structural and syntactical definitions to be used, whilst maintaining a core set of semantically equivalent properties.

This section describes a solution based on the Columbus application, whilst maintaining a generic approach.

In addition to linking all document handling technologies, Columbus uses metadata to keep track of the document editing history, the production of transmittal records and archives of historical data.

4.4.1 Semantics

Semantically, the approach taken in Columbus is to support the Dublin Core standard metadata elements as a fundamental common set. In this sense, the Dublin core is implemented following the minimalist rather than the structuralist approach as defined in section 4.2.1. Examples of the attributes used include the title, creator or identifier. Though their use is not compulsory, they should be considered as the only basic attributes that can be shared across all applications, document types and application types.

The suggested approach is to derive a model of metadata usage that is simple enough to allow all applications to understand the core set of semantic elements, yet rich enough to represent the wide variety of attributes that different documents may hold.

Any additional metadata outside this set is specific to a task or application and is considered to be enhanced metadata. For example, routing information might only exist on document transmittals and a layer table will only be defined for a CAD drawing. Additional metadata elements can be defined in accordance with other semantic standards as chosen by each application.

As a document manager, Columbus presents the user with a rich collection of metadata for each document. Regardless of the document type, Dublin Core metadata is available for all of them and additional specific information is shown for any of the other document types that it supports. This is one of Columbus' key features, to present as much application specific metadata as possible when appropriate. This can clearly be seen when an AutoCAD drawing is selected, as a substantial amount of information is presented which is not available for other document types.

4.4.2 Syntax and Structure

The syntax and structure of the information will vary according to the method used to hold it. Typically, externally stored metadata will be held in a compound format, allowing basic and enhanced metadata to be segregated. In this way, applications can ignore any data that they are not interested in and use what they can interpret. Internally stored metadata will usually have to be constrained by any restrictions imposed by the document's file format. Equally, externally held metadata may also be limited by the chosen schema or format.

Though any structure and syntax could be implemented to hold the information, this could result in a very large number of formats being used. This might be impracticable and to make this more manageable two solutions are proposed:

The first, is to allow any structure and storage format to be used and to produce plug-in modules for each format. These, which can be referred to as "metadata enablers" or "metadata access filters", allow any application to extract the semantics from a custom format. The advantage of this is that applications do not need to understand the storage format used and can just access the data. However, there are a number of limitations; one of them being that hardware specific software needs to be distributed (e.g. a Win32 plug-in may not work under Unix) and the robustness of the software may not always be considered adequate. Though Columbus includes a number of metadata plug-ins, in practice this option is only used when information is embedded in a non-standard file format.

The second solution, is to use common predefined file formats. A format based on XML is supported by Columbus and is suggested as a standard for external metadata usage. However, as Columbus was originally designed before XML was in common use, a format known as DIN (Document Information) had already been defined by myself and is in widespread use throughout the Columbus user base.

DIN files are just made up of simple text which holds attribute names and values in discreet sections. They are extensible by any metadata writer to include new sections and attributes, whilst maintaining a common set of properties. These are typically from the Dublin Core, but can be defined in accordance with other standards too.

One other suggested format, which all applications should support, is the Microsoft Structured Property Data format as mentioned in 4.3. Though this may seem biased towards one software vendor, the reality is that many others are also using this format to store its metadata. This type of metadata is found in a number of very common Microsoft Windows applications such as Microsoft Word, Microsoft Excel, Microsoft Visio and AutoDesk 3D Studio.

4.4.3 Columbus Schemas

Columbus is able to use metadata that is stored in a number of different ways as has been suggested. It can be embedded in the document, kept externally or exchanged as part of an information stream. In addition, the metadata itself can be represented as simple text or encoded in a proprietary format if a metadata enabler is provided. In order to present an overview of the metadata structure and semantics that are supported, one specific format has been selected, which is XML.

Despite the fact that the most common format for storing document information at present is the simple text DIN file, it is better to describe the XML format. This is because the semantics and syntax are very similar, the format will soon supersede others for external use and it is a widely know language. XML also has an excellent mechanism for defining how the information can be checked for conformance, the schema.

The best way of describing how XML is used to store Columbus metadata is to look at an example and describe its contents. After this, the schema definition, which provides a formal definition for its usage, can be presented and discussed.

The document information example that is presented in Figure 4.1 relates to an AutoCAD drawing, but the metadata is structured in the same way as it would be for any other document format. It is made up of a number of sections, which provide separate sets of information about the document.

One important thing to consider is that Columbus has been designed as a system capable of displaying a rich collection of metadata whilst maintaining a common set of properties. In order to achieve this, different document types will need to have a common definition for certain metadata items and a specific one for the information that is particular to that document type. In Columbus' XML document information, the dublinCoreProperties, extendedProperties, jobDetails, references and previewImage sections are common to all documents. Though their use is generally optional, if they are present they have to be written in accordance with the required common layout. Conversely, the contents of the typeSpecification and formatSpecification sections are defined by each different document category. This is also the case for the transmittalDetails section, which is defined according to the mechanism used to exchange or share the document.

Sample XML Document Information


```
<?xml version="1.0"?>
<documentInformation
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="cbs-format-acad.xsd" >

  <dublinCoreProperties>
      <identifier>ze1g-115.dwg</identifier>
      <type>drawing</type>
      <format>acad</format>
      <title>Mechanical Services 2nd Floor  Stair Riser</title>
      <subject></subject>
      <description>d</description>
      <date>25-MAY-2001</date>
      <creator>Juan Herrero</creator>
      <publisher>ARUP</publisher>
  </dublinCoreProperties>


  <extendedProperties>
      <number>M10-212</number>
      <status>construction</status>
      <revision>A</revision>
      <generator>CADtools</generator>
  </extendedProperties>

  <jobDetails>
      <name>8 Westminster Avenue</name>
      <number>51221-00</number>
  </jobDetails >

  <references>
      <itemIdentifier>ep1g-115.dwg</itemIdentifier >
      <itemIdentifier>xk10-100.dwg</itemIdentifier >
  </references>

  <previewImage>
      ZjMzAJkzMwDMMzMA/zMzAABmMwAzZjMAZmYz5mZAADMmQAzzJkAZsyuW
      XjPMAJkzzADMM8wA/zPMAABmzAAzZswAZmbMAJlmzADMZswA/2bMAACZ
      DMAAAzzAAzM8wA/MAJn/zADM/8wACAzmcwAZpnMAJmZ8zMAAD/zAAz/8
      zADMmcwA/5nMAADMzAAzzMwAZszMAJnMzADMzMwA/Avr6+vuy+vtzc3N
  </previewImage>

  <transmittalDetails>
      <issueBy>Juan Herrero</issueBy>
      <issueTo>Martin Smith</issueTo>
      <purposeOfIssue>For Review</purposeOfIssue >
      <sharedRepository>ftp://projects.westave.com/</sharedRepository>
      <issueIdentifier>ISS44399922AY</issueIdentifier >
      <respondBy>12-MAY-2003</respondBy>
  </transmittalDetails>


  <typeSpecificInformation>
      <layers>
          <layer>Z010LX000</layer>
          <layer>Z010HX000</layer>
      </layers>
  </typeSpecificInformation>


  <formatSpecificInformation>
      <blocks>
          <block>SBC-KEY</block>
          <block>SCALEBAR</block>
      </blocks>
      <objectEnablers>
          <application>Architectural Desktop</application>
      </objectEnablers>
  </formatSpecificInformation>
</documentInformation >
```


Figure 4.1 Sample XML document information

In the sample XML document information presented in Figure 4.1, there are eight distinct metadata sections, which are:

- **Dublin Core Properties:** This represents the basic Dublin Core metadata properties. These are: identifier, type, format, title, subject, description, date, creator and publisher. The identifier is a string such as the filename that can identify the document, the type classifies documents under a general category and the format defines the application that created the document. These and other elements in this section have a predefined semantic definition in accordance with the Dublin Core and as such can be used easily to exchange information with systems unrelated to Columbus.

- **Extended Properties:** This contains other common properties that are not covered by the Dublin Core. They are the number, status, revision and generator of the metadata.
- **Job Details:** Lists the name and number of the job to which this document belongs.
- **References:** This section enumerates any other resources that this document makes use of. In the case of AutoCAD drawings these are any external reference files (i.e. the Xrefs).
- **Preview Image:** This is a binary piece of data that holds a bitmap representation of the document. For AutoCAD drawings, this is the thumbnail image that is taken as a screenshot when the document is saved. The binary size of this image is normally about 4 kilobytes, which can be one hundred lines of base64 encoded data. In the sample, it has been cut down to a few lines to simplify the presentation.
- **Transmittal Details:** This is intended to hold information to assist in sending documents to other participants. It can be used to link to document control systems and help with document archival. The format of the data is defined according to its particular usage and cannot be assumed to be common to all documents.
- **Type Specific Information:** This sections contains information that is common between documents of the same type. For example, drawings from two distinct CAD formats are of the same type (i.e. drawing) and share some common information that can appear here. The format of this section has to be defined by a type specific schema.
- **Format Specific Information:** Holds information that is specific to this data format. In this case, AutoCAD drawings may have a block table and require additional add-on application object enablers, which are listed here.

The sample presented provides a clear description of how XML metadata can be presented. However, in order to provide a formal definition of its usage, it is necessary to provide a metadata schema. This defines the structure that the data must follow to be considered valid XML Columbus document information.

In order to provide the flexibility to allow different document types to maintain their own definitions, different schemas are required. So, AutoCAD drawing metadata is written in accordance with an AutoCAD schema and Microsoft Word document metadata will follow a different one. However, as the majority of metadata is common to all document types, it makes sense to share these definitions. This is done, as shown in Figure 4.2, by providing a Common Document Information Partial Schema. Though it cannot be used on its own to validate XML document information, as it omits the transmittalDetails, typeSpecification and formatSpecification sections, it can be included as a common definition in all other schemas.

Common Document Information Partial Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<!--
 This PARTIAL SCHEMA defines the basic common elements that all documents can share
 With the exception of dublinCoreProerties/identifier, all others elements are optional
 Transmittal details, typeSpecificInformation and typeSpecificInformation are refered to,
 But need to to be defined by type and format specific schemas.
-->
 <xsd:element name="documentInformation" type="DocumentInformation"/>

 <xsd:complexType name="DocumentInformation" >
     <xsd:sequence>

        <!-- Dublin Core Properties
             These are defined semantically by the Dublin Core standard -->

        <xsd:element name="dublinCoreProperties" minOccurs="1" maxOccurs="1">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="identifier"  type="xsd:string" minOccurs="1" maxOccurs="1" />
                    <xsd:element name="type"        type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="format"      type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="title"       type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="subject"     type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="description" type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="date"        type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="creator"     type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="publisher"   type="xsd:string" minOccurs="0" maxOccurs="1" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>


        <!--    Extended Properties.
                Additional optional properties that go beyond the Dublin Core. -->

        <xsd:element name="extendedProperties" minOccurs="0" maxOccurs="1">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="number"     type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="status"     type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="revision"   type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="generator"  type="xsd:string" minOccurs="0" maxOccurs="1" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>


        <!-- Job Details -->
        <xsd:element name="jobDetails" minOccurs="0" maxOccurs="1">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="name"      type="xsd:string" minOccurs="0" maxOccurs="1" />
                    <xsd:element name="number"    type="xsd:string" minOccurs="0" maxOccurs="1" />
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
```

```
        <!--    References
                All dependent documents such as xrefs are listed here by item identifier -->
        <xsd:element name="references" minOccurs="0" maxOccurs="1">
            <xsd:complexType>
             <xsd:sequence>
               <xsd:element name="itemIdentifier" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
             </xsd:sequence>
            </xsd:complexType>
        </xsd:element>

        <!--    Preview Image.This is a BASE64 encoded preview image of the document -->
        <xsd:element name="previewImage" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>


        <!-- The following definitions exist so that any format specific xml document
             can be validated by the generic schema -->

        <!--    Placeholder for any TRANSMITTAL information.-->
        <xsd:element name="transmittalDetails" minOccurs="0" maxOccurs="1" type="TransmittalDetails"  />

        <!--    Placeholder for any information that is specific to this document TYPE.-->
        <xsd:element name="typeSpecificInformation" minOccurs="0" maxOccurs="1"
                                                                    type="TypeSpecificInformation" />

        <!--    Placeholder for any information that is specific to this document FORMAT.-->
        <xsd:element name="formatSpecificInformation" minOccurs="0" maxOccurs="1"
                                                                  type="FormatSpecificInformation" />

      </xsd:sequence>
    </xsd:complexType>

  </xsd:schema>
```

Figure 4.2 Common Document Information Partial Schema

The Common Document Information Partial Schema states that it is compulsory to have a dublinCoreProperties section, which must have an identifier element. This section has a standardised semantic definition that makes it possible to share metadata with applications unrelated to Columbus.

All other elements and sections that are defined are optional. Most are just straightforward complexType definitions that allow string values for each element. However, previewImage is different, it is encoded as base64binary object, which is an efficient ASCII representation of binary data, as commonly used in email systems. The sections for typeSpecificInformation, formatSpecificInformation and transmittalDetails are referred to but not defined. It is left to each implementation schema to provide a definition for these sections.

Allowing for each document type to define its own schema provides a lot of flexibility but can also result in an undefined number of schemas being required. In order to validate an XML file, it is necessary to use an appropriate schema, which might not always be available, particularly when used for long term archival. The xsi:noNamespaceSchemaLocation="cbs-format-acad.xsd" entry in the XML sample document states that it is written in accordance with the schema called cbs-format-acad.xsd. However, as emphasised in the XML language definition, this is only hinting at what schema to use and where to obtain it, it is not compulsory to use that particular one. Based on this, Columbus may choose to ignore the named schema and validate against its own simple generic Columbus schema. This is useful if only basic common properties are required. Figure 4.3, presents a generic schema that includes the common definitions and allows any data to appear in the typeSpecificInformation, formatSpecificInformation and transmittalDetails sections.

Generic Document Information Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <!-- Bring in the Common Document Information Partial Schema -->
  <xsd:include schemaLocation="cbs-common.xsd" />

  <!-- TypeSpecificInformation -->
  <xsd:complexType name="TypeSpecificInformation" >
      <xsd:sequence>
          <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax" />
  </xsd:complexType>

  <!-- Format Specific Information -->
  <xsd:complexType name="FormatSpecificInformation" >
      <xsd:sequence>
          <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax" />
  </xsd:complexType>

  <!--TransmittalDetails -->
  <xsd:complexType name="TransmittalDetails" >
      <xsd:sequence>
          <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax" />
  </xsd:complexType>

</xsd:schema>
```

Figure 4.3 Generic Document Information Schema

Columbus' schema definitions permit developers to extend the metadata requirements as they wish. However, the Generic Document Information Schema provides the possibility of reading the common properties from any XML document information file. This can be done without the need to obtain the specific schema that was used to create it. To achieve this flexibility, it uses the processContents="lax" with the "any" and "anyAttribute" definitions, which allow any well-formed XML to be considered correct when validating the document information.

This approach resembles the object orientation notion of polymorphism, where there is only a need to know about the base class definitions of objects if base class data or methods are only required.

If Columbus or any other application needs to use any of the data that is format or type specific, then they can use the specific schema rather than the generic one to validate the document information. The suggested approach is to provide a hierarchy of definitions, where another common schema is defined for all documents of the same type and format specific information is described in yet another schema. Alternatively, a combined definition for format and type can be presented in a single schema as seen in Figure 4.4

AutoCAD Format Document Information Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <!-- Bring in the Common Document Information Partial Schema -->
  <xsd:include schemaLocation="cbs-common.xsd" />

  <!-- Type Specific Information -->
  <xsd:complexType name="TypeSpecificInformation" >
      <xsd:sequence>

          <!-- Layer table -->
          <xsd:element name="layers" minOccurs="0" maxOccurs="1">
              <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="layer"    type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
                  </xsd:sequence>
              </xsd:complexType>
          </xsd:element>

      </xsd:sequence>
  </xsd:complexType>


  <!-- Format Specific Information -->
  <xsd:complexType name="FormatSpecificInformation" >
      <xsd:sequence>
          <!-- Block table -->
          <xsd:element name="blocks" minOccurs="0" maxOccurs="1">
              <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="block"    type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
                  </xsd:sequence>
              </xsd:complexType>
          </xsd:element>

          <!-- object enablers table -->
          <xsd:element name="objectEnablers" minOccurs="0" maxOccurs="1">
              <xsd:complexType>
                  <xsd:sequence>
                     <xsd:element name="application" type="xsd:string" minOccurs="0" maxOccurs="unbounded" />
                  </xsd:sequence>
              </xsd:complexType>
          </xsd:element>

      </xsd:sequence>
  </xsd:complexType>


  <!--TransmittalDetails -->
  <xsd:complexType name="TransmittalDetails" >
      <xsd:sequence>
          <xsd:any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:anyAttribute processContents="lax" />
  </xsd:complexType>

</xsd:schema>
```

Figure 4.4 AutoCAD Format Document Information Schema

The AutoCAD format document information schema presented in Figure 4.4 includes the common partial schema and defines as type specific information the list of layers in a drawing. This, for example, can be used by applications to confirm that drawings comply with layering guidelines. The schema also allows a block table and application list to be presented in the format specific section. Finally, transmittal details are included for completeness, but are set so that they are ignored when validating document information with this schema.

Overall, the schema nesting arrangement that is suggested allows for flexibility in defining metadata, whilst maintaining a core set of standard properties that can be used by any application. The generic schema approach, which allows any document information to be validated, guarantees that common data will be readable regardless of the type or format specific information that is added. This, is a particularly important design aim of Columbus, as data must be accessible easily at all times and will normally be required many years after a project has been completed. This must be achievable without requiring any additional software or schema definitions.

4.4.4 Columbus Metadata Usage

Having suggested how metadata could be defined, it is important to highlight how it is used by Columbus alongside each of the document handling technologies.

- **Document Management:** Metadata is used to present detailed information about documents to the user before editing or viewing. It also may be used to record document activities such as editing, printing and saving of documents. Document searching, cataloguing and filtering are enhanced when document properties are used. Columbus also includes document and history reporting capabilities which gather information from metadata. Document archival and retrieval facilities also make use of metadata to record document properties, archival details, storage location and format and to assist with future accessibility. Multiple versions of documents are also managed by using property metadata and appending information to their history logs. Workflow processes also use metadata to record the document's progress through the system. Metadata is also required to confirm information about a document and to assist with how it is subsequently used. For example, plot files contain detailed metadata about the way the plot was created and can validate that the correct pen mappings and paper size have been selected. Moreover, it is easy to check if internal or externally received documents conform to agreed standards such as layering or naming standards.
- **Project Hosting:** Detailed metadata can be used with project hosting sites to give the user a full preview of information that is available on the shared repository. Metadata makes it easy to correctly identify and track the origin and versions of documents. When using compound document formats, such

as reference files, it also helps to keep files together as a single unit. When documents are uploaded or downloaded from a project hosting site, any metadata that is held can normally be transferred between systems. This avoids having to re-enter property information and represents a big cost and time saving, whilst also reducing errors.

- **Document Control:** Metadata can be used to keep track of information sent to or received from other parties and for the management of historical records. Columbus is able to directly export document transmittal reports into document control systems, therefore reducing any data re-entry and reducing errors. This information is also used to link the correct archive copies of documents with records of transmittals that are maintained within the system. When incoming document are received either electronically or scanned in from paper, they need to be tagged electronically and recorded in a document control system. The use of metadata allows these two activities to be coordinated.

From what has been presented, it is clear that great efficiency gains can be made when using metadata to enhance document sharing. Columbus, the application used to demonstrate these issues, provides a number of modules to work across all stages of document handling and metadata is crucial to the way in which the application works. The next chapters will present Columbus in detail, both from a user functionality and software design perspectives. However, it is important to emphasise that metadata is the underlying technology used to achieve successful information exchange.

Chapter 5

Columbus: A Working Solution

This chapter focuses on the software that has resulted from this research, the Columbus application. The first section presents an overview of the product, highlighting the design aims and how the product addresses these needs. In the second section, Columbus' major component modules are described in detail, discussing how the product deals with some of the major issues highlighted in earlier chapters. In the final sections of the chapter, metadata creation, installation, configuration and deployment issues are considered.

5.1 Columbus Overview

As discussed in previous chapters, in order to handle information management efficiently, a unified solution to document management, project hosting and document control has been suggested. Ove Arup's need to address some of these issues led directly to this research and the development of the Columbus product.

5.1.1 Columbus Design Aims

Columbus has been designed to fulfil a number of key requirements which were discussed in earlier chapters. The most important of these, and how Columbus addresses them, are as follows:

- **Ease of access to information:** An important goal of Columbus is to allow users to create, find, edit, view and track documents in a simple yet powerful manner. The metadata richness that Columbus supports thorough its navigator makes cataloguing and identifying documents very easy. The built-in viewing capabilities mean that anyone on a project can view and print documents without having to purchase or install the applications that created the files. Though originally designed as a tool for managing complex CAD drawing assemblies, Columbus now supports most office document formats and can be enhanced by add-on plug-in modules. This, together with the reporting facilities available, makes the product a powerful document management system.

- **Distributed working:** Though Columbus has the flexibility to work with documents from numerous sources and protocols; the current emphasis on concurrent engineering (Amor et al. 1997) is to encourage teams to directly share electronic information. This can either be to maintain design data at a remote shared location or to publish information seamlessly between them in a peer to peer arrangement (Oram 2001). Columbus' navigator module unifies the view of resources and can present information from different sources in a single location. Its project hosting features allow information to be published to other participants in a straightforward manner using simple protocols such as FTP or HTTP or by linking to external hosting sites. Columbus also has tools for acquiring information from remote locations and scanning in paper documents.
- **Information recording:** To assist in controlling the exchange of documents, Columbus maintains records of all transactions and detailed information about documents. This information can be linked to document control systems or passed to other parties in order to assist with document exchange. In addition, it can maintain record copies of documents to provide a reliable archive of information for the future. Columbus also has built in reporting facilities to provide document and history reports of documents.
- **Open format:** A core requirements of Columbus is that information should be held in an open and externally accessible file system, supplemented by metadata. This requirement guarantees that at a later date, documents will still be accessible without the need to use Columbus, making it 'future proof'. Any documents and metadata can be archived to tape or optical media and read back many years later without requiring specialised applications.

5.1.2 Product Overview

As a piece of software, Columbus can be described as a combined document management and project hosting client application that easily links to document control systems. The application allows users to create, manipulate and share information in a flexible and unified manner. Columbus is made up of a number of functional component modules as shown in Figure 5.1.

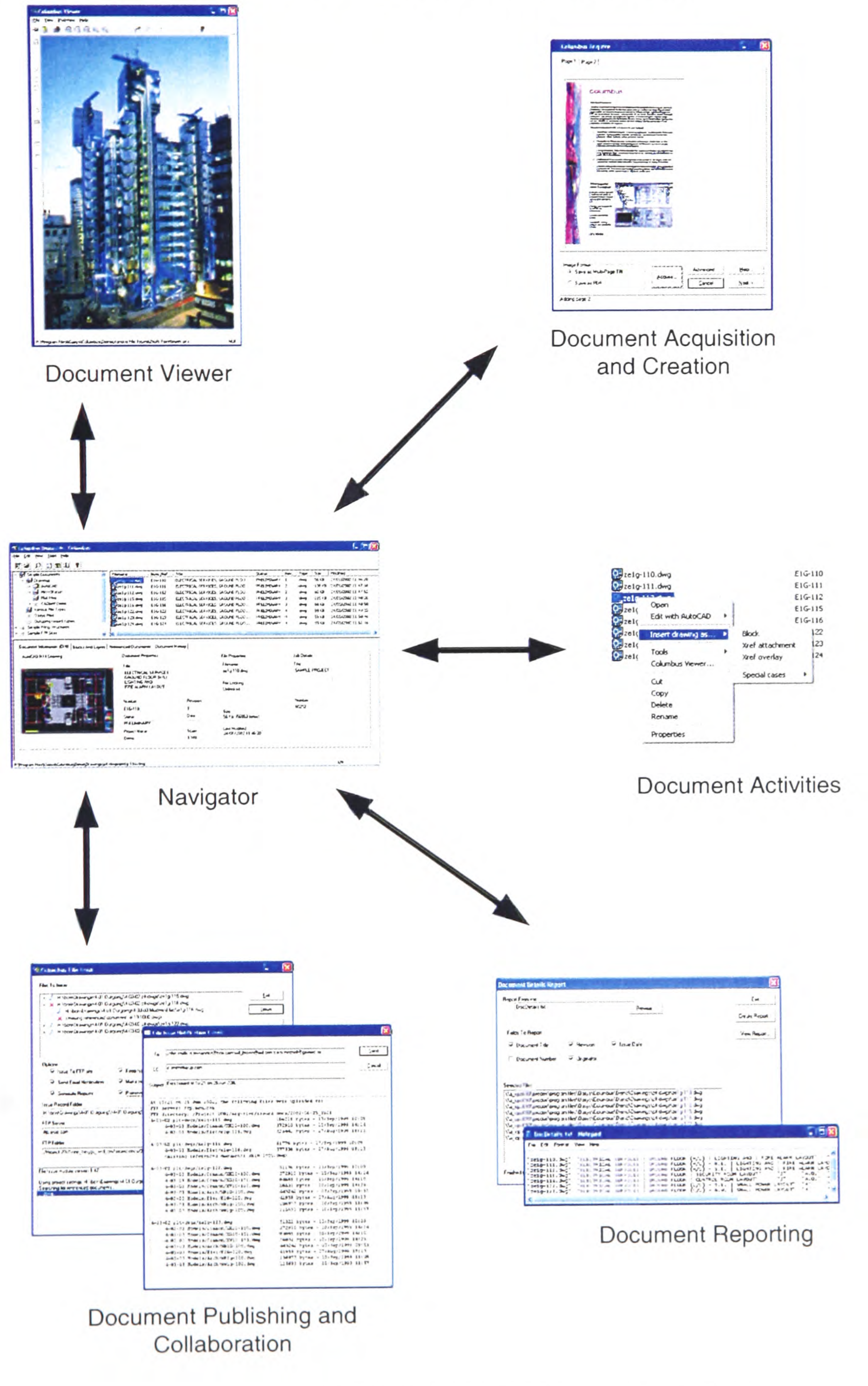

Figure 5.1 Columbus Component Modules

Each of these component modules is relatively independent, though as can be seen in the diagram they interact through the navigator. Briefly, the role of each component can be described as follows:

- **Columbus Navigator:** Allows the user to see all configured project areas in a simple and unified way, regardless of where they are stored or how they are accessed. They can list the documents at a location according to a filter and see any metadata associated with a particular document.
- **Document Viewer:** Is used to view any document regardless of whether the native application that created the document is present on the machine or not. For certain file types, it also allows the user to "red-line" or "mark-up" the document with comments.
- **Document Publishing and Collaboration:** This module's role is to share documents with others. Documents can be uploaded to project hosting sites or shared on any media. Optionally, other project participants can be notified via email. Metadata is critical to identifying the information shared and to record its transmission in a document control system. The module is also responsible for the long term archival and storage of information in a simple and easily accessible format.
- **Document Acquisition and Creation:** Is used to obtain documents from external sources. They may be downloaded from project hosting sites, imported from physical media or acquired from paper using a scanner. Metadata is once again an important element and, if available, allows documents to be processed more effectively. When creating documents, they can be based on pre-set templates and users are helped to place them at a suitable location. Documents can also be named in accordance with a predefined convention and can have appropriate metadata attached. In addition, the module is also responsible for the set-up and creation of project areas.
- **Document Activities:** Implements most of the file manipulation commands. It brings the familiarity of Microsoft Windows Explorer to Columbus, implementing commands such as open, print, cut, copy and paste. Most of these commands can be invoked with a right click or pull down menu. This module also acts as a proxy for other modules in order to

unify the user interface menu. As such, it has the responsibility of receiving most menu-driven input and passing it directly to other modules.

- **Document Reporting:** Is responsible for reporting metadata information from a group of documents and producing history reports for activities such as edits. The reports produced can be imported and post-processed by applications such as Microsoft Excel.

5.2 Detailed Description of Columbus

From the overview of the Columbus modules that was presented in the previous section, it is clear that these components are responsible for distinct aspects of the software. Because of this, it is worth looking at them individually in more detail.

5.2.1 Navigator

This is the core Columbus component which represents the primary interface with the user. Though other modules can interact with the user, the navigator acts as the main entry point to all other modules as they must be invoked and initialised from this module. The navigator is mainly responsible for presenting the user with the location, filtered file list and the metadata for all documents.

As can be seen in Figure 5.2, the navigator is composed of three splitter window views. These are the tree view at the top left, the list view at the top right and the document details view at the bottom.

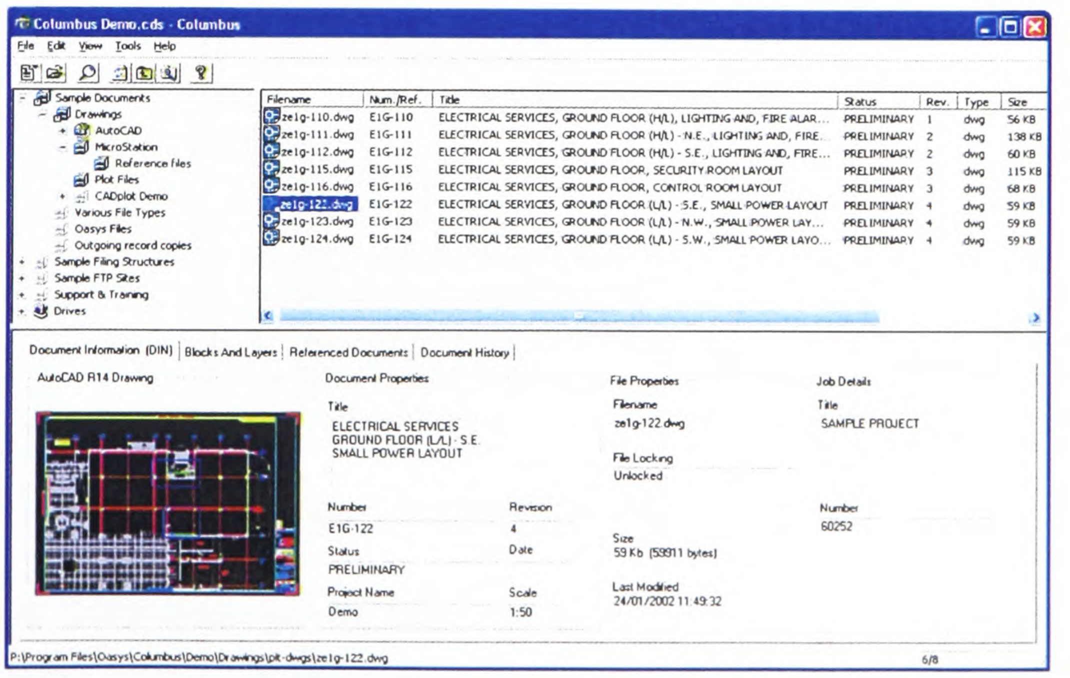

Figure 5.2 Columbus Navigator

The user is able to select a location or filter in the tree view and any documents found at that location are then presented in the list view. A document can then be chosen and detailed information about it is displayed in a series of tabs within the document details view. In addition, a number of tools are available from pull down or right click menus. Each of these views can be described as follows:

Tree View: In this view, the user can navigate through any of the configured project areas and locations. These are specified as a filter which can be a wildcard pattern, which limits the number of files displayed or selects documents using any of the supported protocols (e.g. FILE, FTP or HTTP). This is a very powerful feature, as it allows data to be grouped in a way that is natural to users, independent of physical server location or protocol. For example, directories on a file server in Manchester can be shown next to an FTP server in Hong Kong if they are part of the same project. In the example shown in Figure 5.3, drawings located in Coventry are accessible alongside drawings from Glasgow, as they are part of the same project. Because of this, project data can be seamlessly distributed across file systems, extranet sites, archive media, etc.

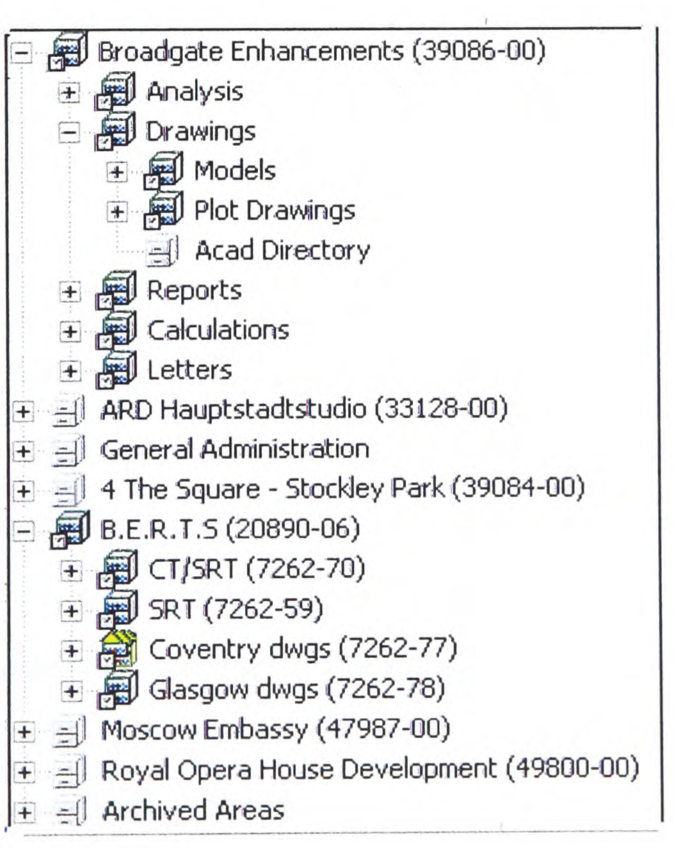

Figure 5.3 Navigator Tree View

A Columbus Data Structure (CDS) file determines the hierarchical tree layout displayed in the view and the links and filters used for finding documents. CDS files are very powerful, allowing information to be gathered from different sources, filtered according to a pattern and assembled from other CDS files into complex nested structures.

List View: Presents basic descriptive information about all of the documents at the selected location and according to the filter specified in the CDS file. The information shown, as can be seen in Figure 5.4, comprises the filename, document number, title, status, revision, type, size and date of the document.

Filename	Num./Ref.	Title	Status	Rev.	Type	Size	Modified
ze1g-110.dwg	E1G-110	ELECTRICAL SERVICES, GROUND FLOOR (H/L), LIGHTING AND, FIRE ALARM LAYOUT	PRELIMINARY	1	dwg	56 KB	24/01/2002 11:46:20
ze1g-111.dwg	E1G-111	ELECTRICAL SERVICES, GROUND FLOOR (H/L) - N.E., LIGHTING AND, FIRE ALARM LAYOUT	PRELIMINARY	2	dwg	138 KB	24/01/2002 11:47:16
ze1g-112.dwg	E1G-112	ELECTRICAL SERVICES, GROUND FLOOR (H/L) - S.E., LIGHTING AND, FIRE ALARM LAYOUT	PRELIMINARY	2	dwg	60 KB	24/01/2002 11:47:52
ze1g-115.dwg	E1G-115	ELECTRICAL SERVICES, GROUND FLOOR, SECURITY ROOM LAYOUT	PRELIMINARY	3	dwg	115 KB	24/01/2002 11:48:20
ze1g-116.dwg	E1G-116	ELECTRICAL SERVICES, GROUND FLOOR, CONTROL ROOM LAYOUT	PRELIMINARY	3	dwg	68 KB	24/01/2002 11:48:50
ze1g-122.dwg	E1G-122	ELECTRICAL SERVICES, GROUND FLOOR (L/L) - S.E., SMALL POWER LAYOUT	PRELIMINARY	4	dwg	59 KB	24/01/2002 11:49:32
ze1g-123.dwg	E1G-123	ELECTRICAL SERVICES, GROUND FLOOR (L/L) - N.W., SMALL POWER LAYOUT	PRELIMINARY	4	dwg	59 KB	24/01/2002 11:50:46
ze1g-124.dwg	E1G-124	ELECTRICAL SERVICES, GROUND FLOOR (L/L) - S.W., SMALL POWER LAYOUT	PRELIMINARY	4	dwg	59 KB	24/01/2002 11:51:16

Figure 5.4 Navigator List View

Most of this information is obtained by reading the document's metadata. In order to present the list as quickly as possible, metadata is only retrieved as the user brings documents into view, which explains why scrolling is initially slower. The list may be also sorted on any of the fields, but as metadata is read for all documents at that location, there may be a delay depending on the field chosen. Documents may be selected in this view and passed to other Columbus modules such as the document viewer.

If the users right-click on a document in the list view, they are presented with a context sensitive menu of operations, which is specific to the document type. Most of these events are directly passed onto other modules for processing.

Document Details View: Shows detailed information about the selected document in a number of tabbed property pages. This information is obtained from various metadata sources and varies according to the document type and configuration. For drawings, as seen in Figure 5.5, it is very detailed and includes an image preview, full document and job details and file properties. Other tabbed pages display block and layer information, reference file details and a full history of the document.

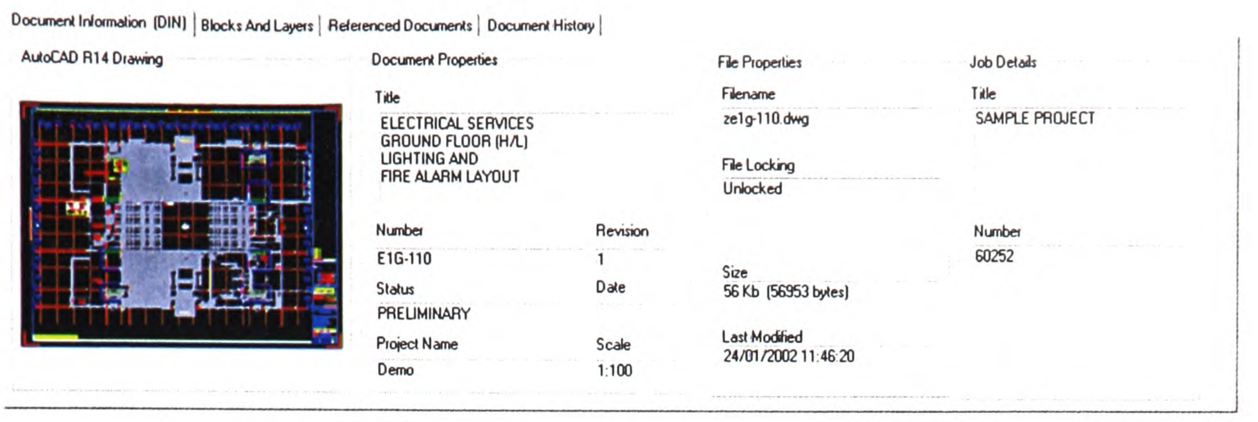

Figure 5.5 Navigator Document Details View for a Drawing

The appearance and number of tabs displayed will vary according to the type of document. This is one of Columbus' most powerful features, allowing the application to display as much or as little metadata as it can find and customised to each document type. In the case of a Microsoft Word document, as seen in Figure 5.6, Columbus is able to display most of the information that is directly generated by Word. This is normally embedded metadata that is directly held in the document as Structured Property Data (Microsoft 1998).

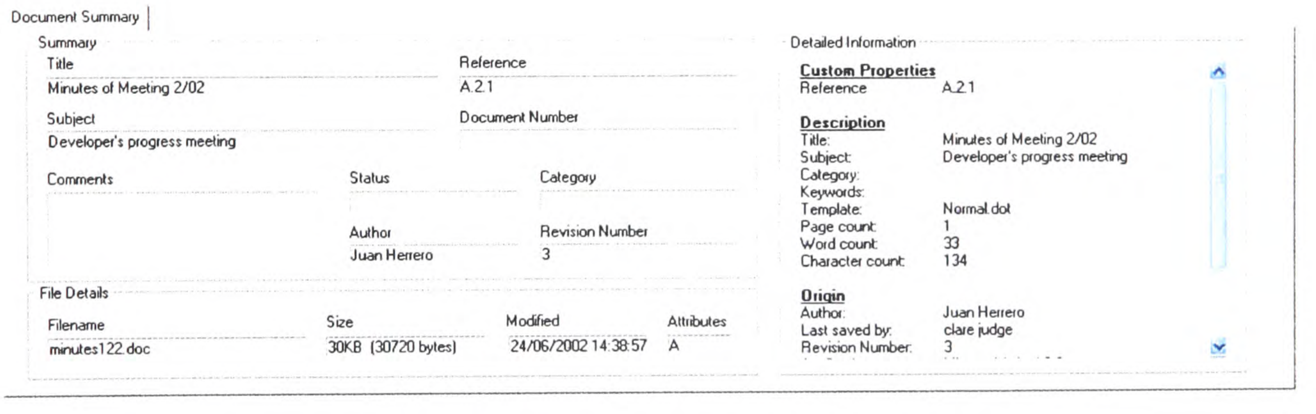

Figure 5.6 Navigator Document Details View for a Word Document

5.2.2 Document Viewer

Columbus allows project participants to view documents without having to purchase a copy of each application that generated the data that is shared. This is done using the Columbus Viewing module, which encapsulates a number of viewing engines. The two principal engines used within the module have been licensed from third party vendors. These companies, Stellent (Stellent 2002) and Rasterex (Rasterex 2002), specialise in selling libraries for viewing many different file formats. Each of them is particularly strong in handling certain document types and weaker with others. The Stellent engine, for example does not excel at viewing

CAD file formats, but is very good with office documents. The Rasterex library, on the other hand, has a more limited range of formats, but is excellent with CAD files. In a rather unique negotiation process, we have been able to offer this technology free with Columbus to the whole industry. Though the Stellent viewing engine supports many file formats, it is rather limited in the facilities that it offers for navigating within the document. Examples of this include zooming, panning and rotating where, as shown in Figure 5.7, the buttons are disabled when this engine is active.

Figure 5.7 Columbus Viewer Using the Stellent Engine

In addition to viewing the most common file formats, the Stellent engine can look at and extract files from compressed archives as seen in Figure 5.8.

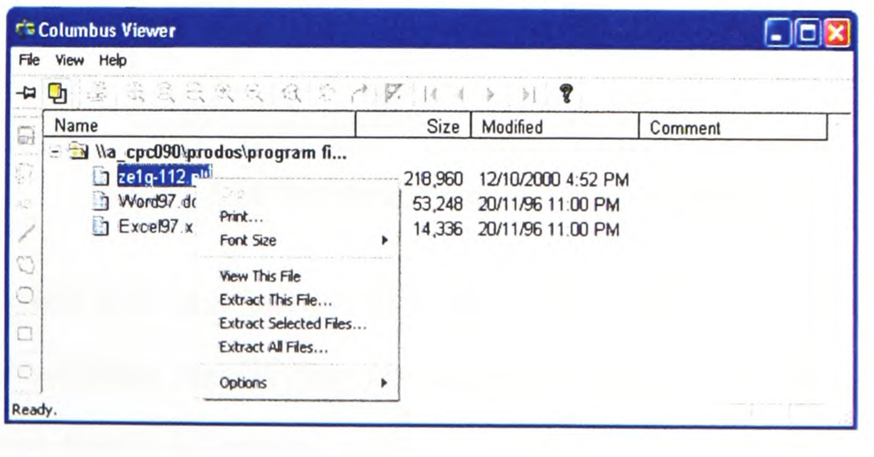

Figure 5.8 Viewing a Compressed File

The Rasterex viewing engine is used for CAD formats such as AutoCAD, MicroStation and the Hewlett Packard Graphics Language (HPGL). It offers more facilities for navigating within documents and, as can be seen in Figure 5.9, the buttons that have been implemented allow amongst other things zooming, panning and rotating the image.

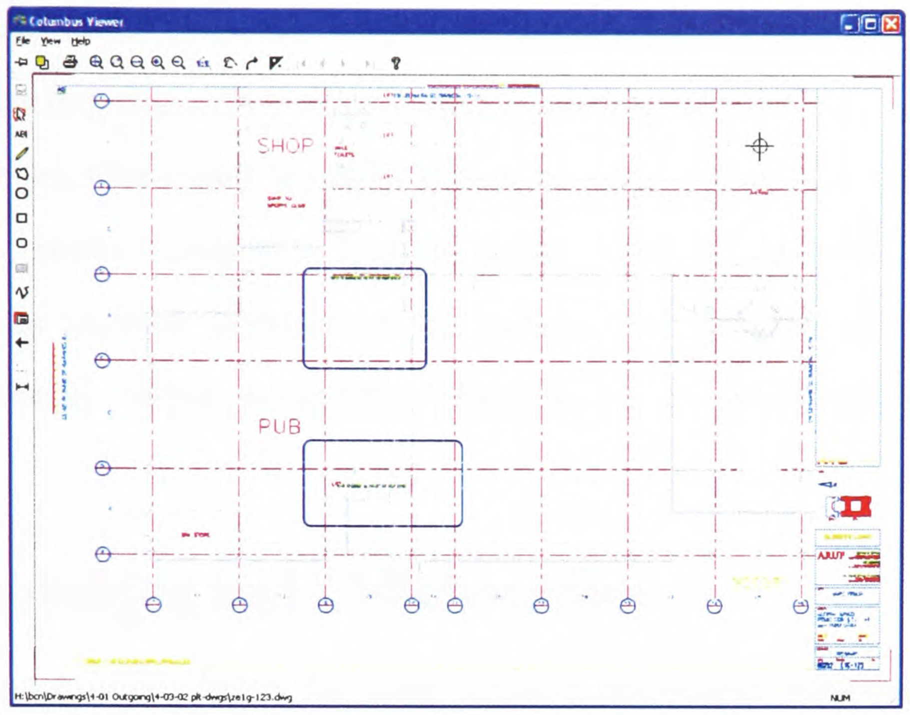

Figure 5.9 Columbus Viewer using Rasterex Engine

Printing facilities are superior, offering the option of scaling the image, printing a portion of the document and seeing a preview image, as seen in Figure 5.10.

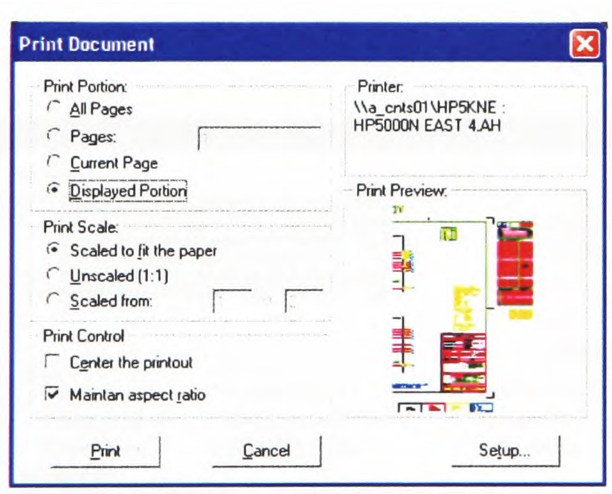

Figure 5.10 Rasterex Print Document Dialog

The Rasterex engine also supports redlining, which allows comments to be added to the document without modifying the original file. The comments can be just simple text or use basic geometry such as squares, circles, lines and bubbles to encapsulate the information.

The Columbus viewer does a number of things before a document is viewed. First, it downloads the main document and associated files from the remote location. This can be through any supported protocol such as FTP, HTTP or simply the file system. Secondly, it adjusts the reference paths to be able to load any referenced or red-line files. Finally, it reads the metadata for the document, which will determine what viewing engine to use and how to handle the document.

As the user selects different documents in the navigator, the viewer automatically shows the selected one. This allows quick an easy identification of any project information, even when they are held at remote sites. However, if the user chooses to "pin" the viewer using the toolbar button resembling a drawing pin, it is possible to start another instance of the viewer and compare documents. This is particularly useful when reviewing changes or redlined comments between drawings.

5.2.3 Publishing and Collaboration

This Columbus module helps the user publish and share information with other parties and collaborate using project hosting sites. The module comprises the file issue tool and the pack and show facility.

File Issue Tool: This tool, as seen in Figure 5.11, allows the user to publish information to a shared repository, notify other project participants, keep archive copies of documents and route files as part of a workflow process.

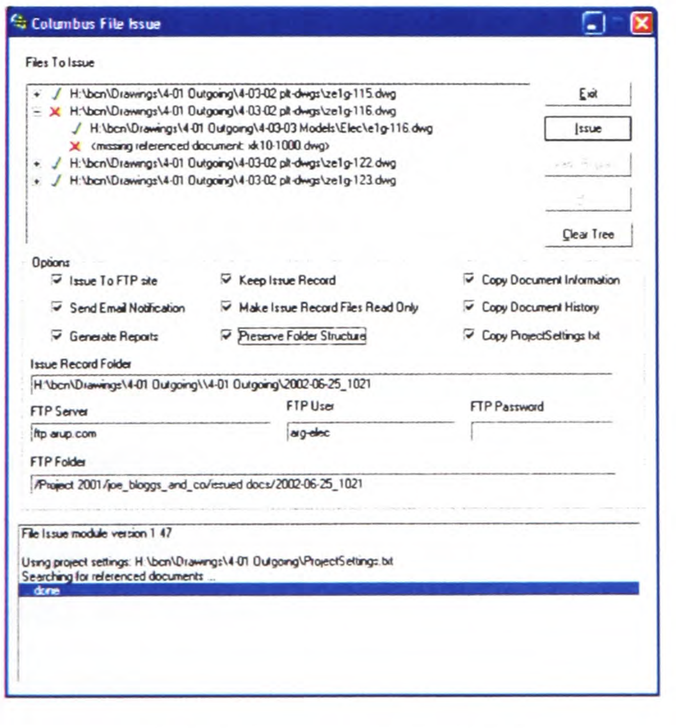

Figure 5.11 File Issue Tool

The capabilities of the File Issue tool include:

- Uploading Documents: The tool can copy a group of files from a project source area to a target location. Typically, when documents are issued as part of an internal workflow process, the standard file-system protocol is used, but when issued externally to a project hosting site it is done using the FTP or HTTP protocols. By using additional plug-in modules, it can upload documents to other project hosting systems.
- Compound Document Relationships: One of the most challenging problems when dealing with complex document types such as drawings, is to check the integrity of compound documents. This is particularly critical in the case of CAD models, as documents should be considered as an assembly or collection of drawing files. As shown in Figure 5.11, the file issue tool will indicate if any of the dependent files are missing with a red cross or display a green tick if all have been found. The facility to check document integrity alone is one of the major benefits that this tool provides.
- Issue Record: The file issue tool can make a permanent archive record of the files sent. It is of vital importance, as discussed in chapters 2 and 3, that a historical record is kept of any documents that are issued to other parties. It is also important that the correct master to reference file relationship is preserved. Many document management systems fail to correctly match a master file with the correct reference files that were issued. This tool has been designed to do this, making it an ideal solution to show the integrity of an archive.
- Preserve Directory Structure: This allows the directory structure at the destination to mirror the structure in the source area. If this option is not selected, all files will be copied into the target directory. The choice of which of these two options should be use is normally agreed before the commencement of the project.
- Metadata Handling: The user can control if metadata information such as DIN and HIS files are copied to the remote location. Sending these files lets the recipient import the information into their document management

and document control systems and allows project-hosting sites to record the information that has been uploaded.

- Reporting: The module can produce reports stating the details of the files uploaded. This is normally a simple text file for human consumption or a report that includes metadata for processing by other applications. It can also provide suitably formatted transmittal documents in Microsoft Word containing company logos and nicely tabulated document lists.
- Notification: It is possible to notify a group of users via email that the document issue has occurred. This can be done if the user has a MAPI (Grundgeiger 2000) compliant email client, such as Microsoft Outlook. This is a better solution that relying on a proprietary notification system and provides the greatest flexibility as it can work with almost any email system. The dialog box that is presented is shown in Figure 5.12 and contains, as the main body of the message, a list of documents that were uploaded. The recipient list is obtained from a predefined settings configuration for the project.

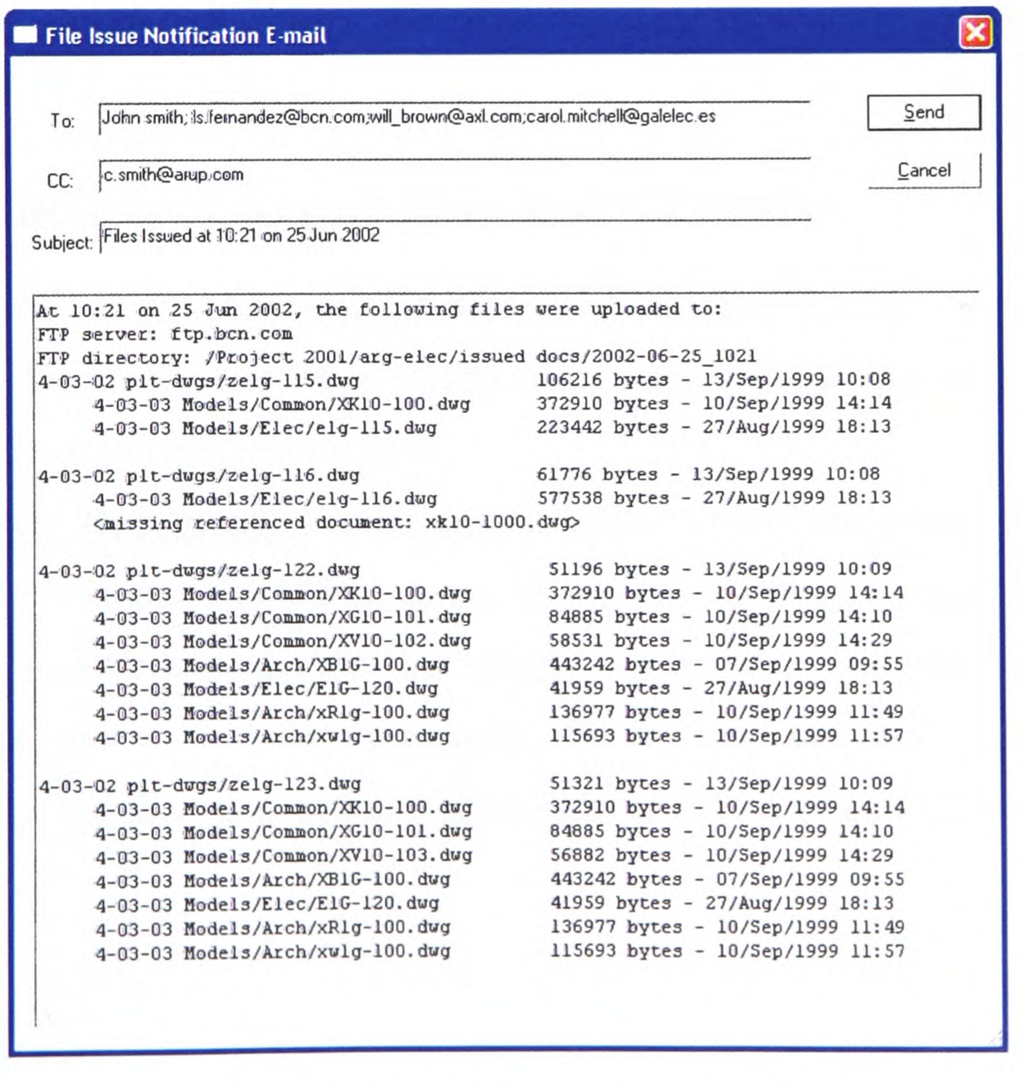

Figure 5.12 File Issue E-mail Notification

Columbus Pack and Show: This is a facility that allows users to create a runtime version of Columbus on removable media such as CD-ROM. Typically it is used to send information to a third party that does not normally use Columbus. When the recipients insert the CD-ROM into their machines, Columbus is launched and using a predefined Columbus Data Structure (CDS) file, lets them navigate and view the contents of the CD-ROM. Rich metadata is typically shown for each document and together with Columbus' other capabilities can help to improve the way in which information is distributed between organisations.

One other important use is document archival, as emphasised in chapter 2. Since there is no guarantee that Columbus will still be in use in many years time, the user will still be able to run Columbus directly from the archived media rather than having to install it on their machine. Moreover, even if future operating systems were to prevent old versions of Columbus from running, the data on the CD-ROMS would still be accessible directly, as it is stored in an open rather than proprietary format. The associated metadata is held in simple text files thereby ensuring its longevity.

"Pack and Show", also acts as way of promoting Columbus. Various organisations that have received CD-ROMS produced in this way have become regular users of the software.

5.2.4 Document Acquisition and Creation.

The document acquisition and creation module allows users to create new documents, download them from remote sites, acquire them from devices such as scanners and set-up new project areas.

Creating New Documents: An important feature of any document management system is the creation of new documents. Columbus has two standard mechanisms for creating new documents: the first simply creates an empty file in the current folder, with the user being prompted for a filename or generating one automatically; the second is to use a predefined template, as can be seen in Figure 5.13.

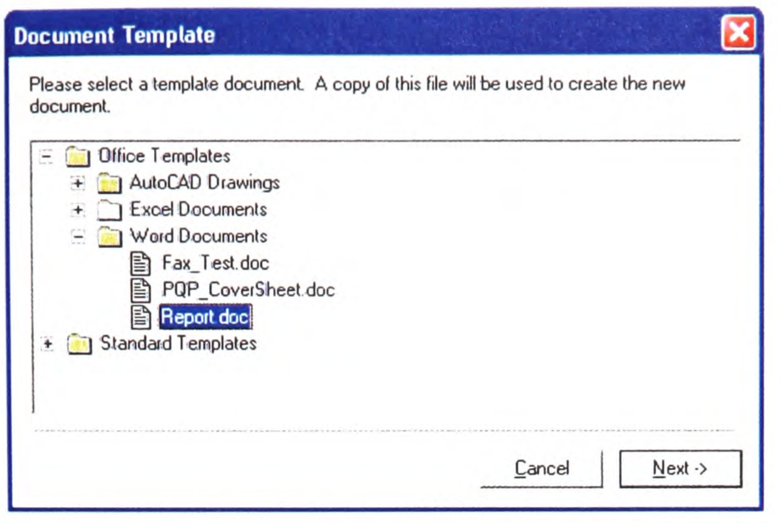

Figure 5.13 Document Template Selection

This option allows the user to select from a number of standard, project or office specific templates for different document types. A copy of the template file is then made and the user is prompted to enter some metadata about the document. Finally, the option of opening the document is given to the user. The ability to name, place and open a document in one step is considered a useful feature that simplifies the document creation process. Columbus uses the dialog box in Figure 5.14 to prompt the user for a filename or use one that is automatically generated. However, this can be replaced by a separate plug-in to support other corporate file-naming conventions.

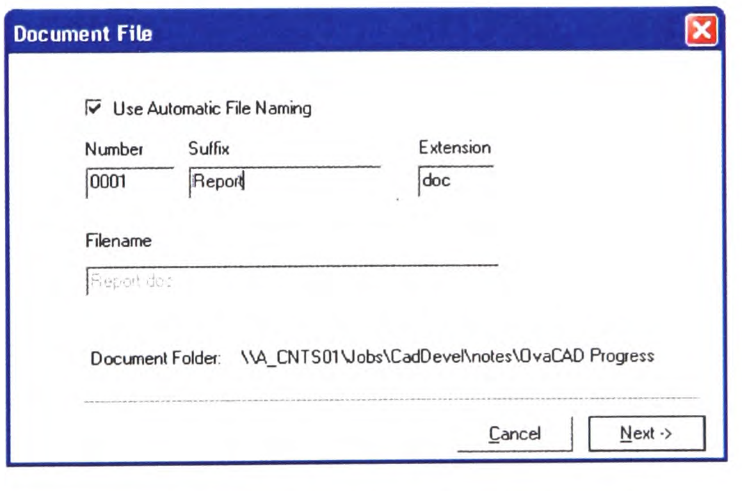

Figure 5.14 Automatic File Naming

Though these two options allow the user a certain amount of flexibility, there is still scope for improvement and the option to have a "New Document Plug-in" is available. Basically, a third party developer can produce a tool for creating new documents that will appear within the Create New Document menu. The internal Arup version of Columbus ships one such plug-in, which I developed to interface with our in- house customisation of Microsoft Word known as OvaWord.

This tool, as can be seen in Figure 5.15, checks that OvaWord is installed, obtains a list of document creation macros (e.g. letter, fax, meeting, agenda, memo, etc.) and presents them to the user. When the user then selects one of these menu options, it connects to Microsoft Word via ActiveX and runs the appropriate macro. It is then the macro's responsibility to create the document, add metadata, select a document name and save the file to the directory specified.

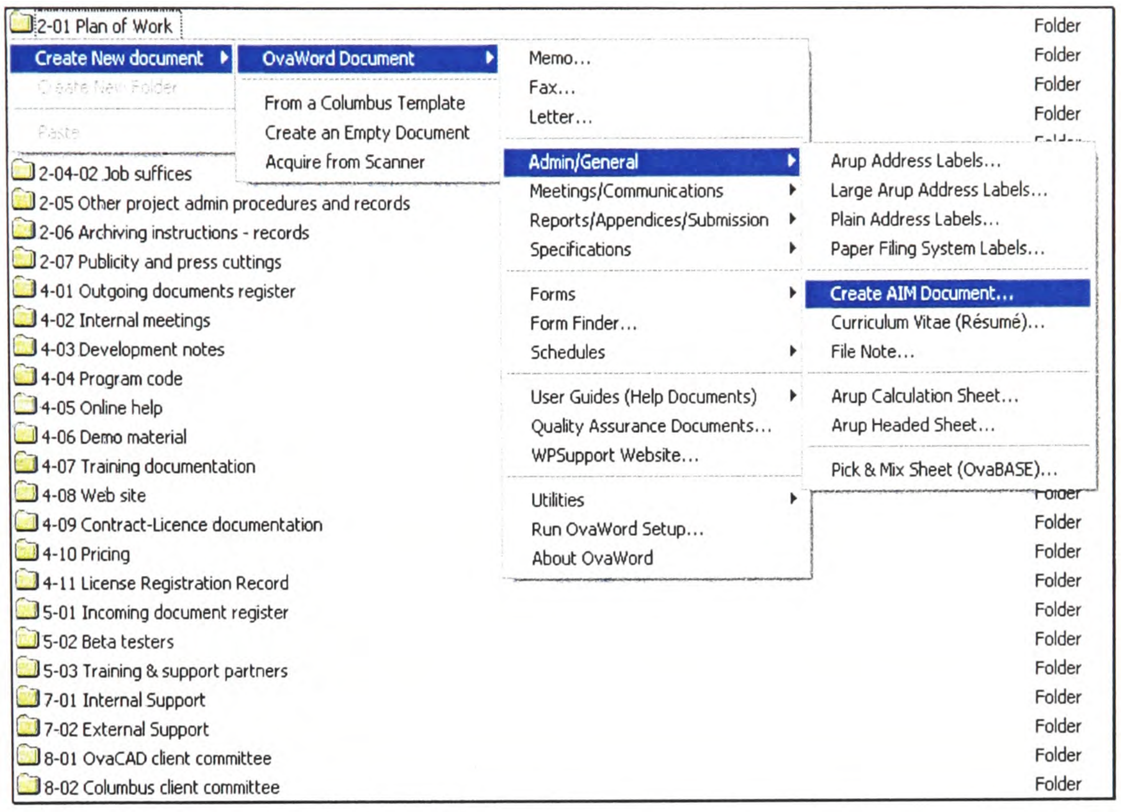

Figure 5.15 OvaWord New Document Plug-in

<u>Columbus Acquire:</u> Columbus Acquire is a Columbus add-on component for scanning multi-page or single-page documents from any TWAIN compatible device. TWAIN (2000), the "Technology Without An Interesting Name", is a standard protocol to allow any imaging device to talk to an application. These are typically scanners, but other devices such as digital cameras are supported. The module saves images as multi-page TIFF or PDF files and there is no need to purchase additional software such as Adobe's Acrobat writer, which can represent a big saving.

Typically, the module is used as part of a Create New Document process, and documents are saved directly to a folder in the Columbus structure with document metadata added at the time of scanning. The main dialog, as seen in Figure 5.16, allows the user to capture various pages before being prompted for a filename and adding descriptive metadata about the document.

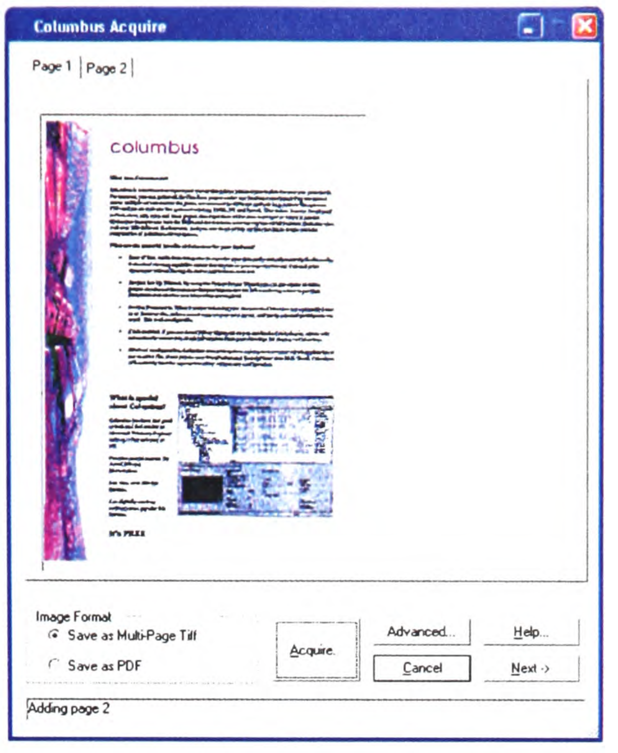

Figure 5.16 Columbus Acquire

This module uses the highly efficient CCITT Group 4 (Murray 2002) algorithm for compressing black and white images. Typically, an A4 uncompressed black and white A4 page will be over 1 megabyte in size, yet by using CCITT G4 encoding, this can be reduced to under 50 kilobytes.

Document Download: One other way of gathering documents is to download them from a project hosting site or other remote location. This module, which can be seen in Figure 5.17, contains the code to initiate a document download from a number of protocols such as FTP, HTTP or any of the plug-in file systems that are supported. If the document has been downloaded from a location that includes external metadata, then it is downloaded too.

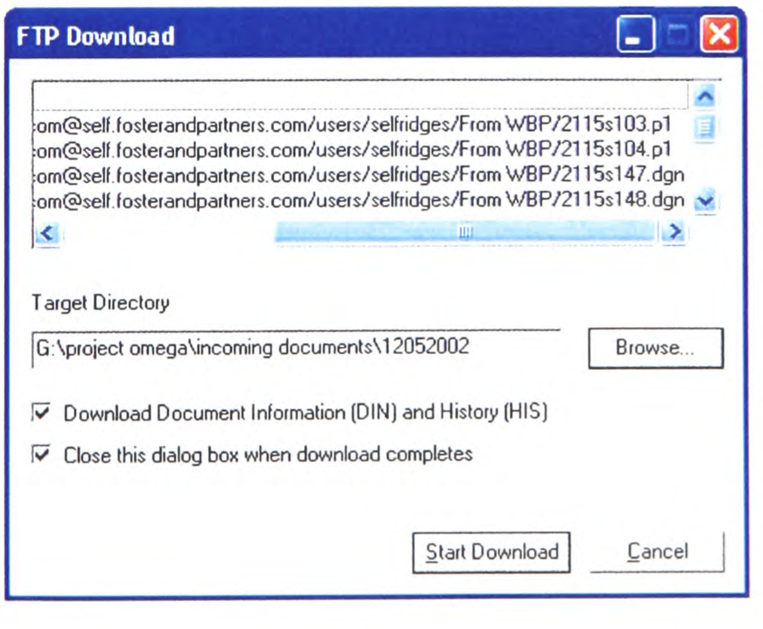

Figure 5.17 FTP Download

Setting up Projects: As filing structures tend to be very similar between projects, Columbus includes a tool called the Project Set-up Wizard to help with their creation and configuration.

As seen in Figure 5.18, it guides the user through the process of creating a new project directory hierarchy from a predefined template, creating the complementary Columbus Data Structure (CDS) file and linking it into the main office CDS file. By using this tool, it is easy to ensure that projects are set-up in a consistent manner. The templates that are used, are typically defined in accordance with a project or organisation's filing convention. In the case of Arup, the Arup Information Manual (AIM) filing convention is used. This structure allows folders to be created as a small, medium or large structure to suit the particular requirements of each project.

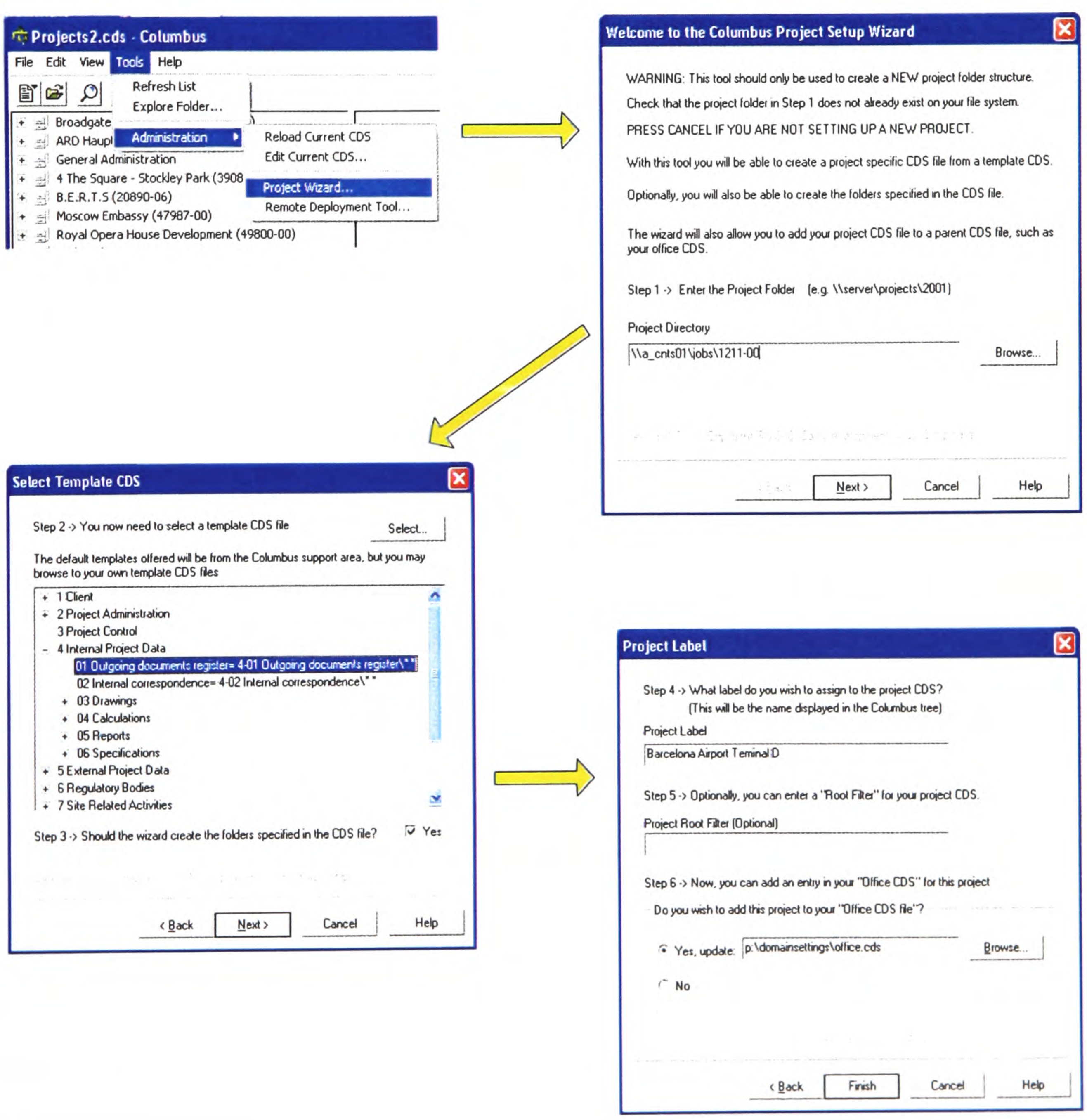

Figure 5.18 Project Wizard

5.2.5 Document Activities

As previously described, though the user mainly interacts with the navigator, activities other than browsing locations, listing documents and displaying metadata are normally handled by other modules. The document activities module, handles most actions that are carried out on the documents themselves and are all invoked directly by the navigator as the result of a context sensitive menu selection. The activities that can be requested are dependent on the type of document and additional software that is installed and include editing documents, explorer facilities, document property editing and miscellaneous tools.

Editing Documents: Once the navigator has found a document, it is usually necessary to be able to edit it with the correct tool. Using Microsoft Windows Explorer, documents are normally just edited by double clicking or selecting the "open" right click menu option. This is also the way in which Columbus works, however there is a need to have far greater control over the way in which certain document types are opened for editing. This is particularly important in the case of drawings files, where, for example, there may be a need to open the drawing with different versions of AutoCAD (e.g. R14 or 2000), using different configurations (e.g. Mechanical Desktop or Architectural Desktop) or with a different CAD package altogether (e.g. IntelliCAD). The way that Columbus does this, is by looking in the registry to see if any of the known applications have been installed and adding them to the menu if they are there. Additionally, it is important to have the option of opening documents in different ways according to the application. So, for example, in the case of AutoCAD, it is possible to open the drawing in a new session and keep the current drawing open or to open it in the same session and close the current drawing.

Explorer Facilities: An important design requirement for Columbus is that it has to be as user friendly as possible. The way in which documents are manipulated has been modelled on Microsoft Windows Explorer. Copying, deleting and renaming documents are similar to Explorer and are invoked from the right click or pull-down menu as seen in Figure 5.19. However, Columbus will additionally ensure that any external metadata and associated files are also taken into account and handled appropriately. To help integrate Columbus with other applications, this module also implements the "drag and drop" facilities.

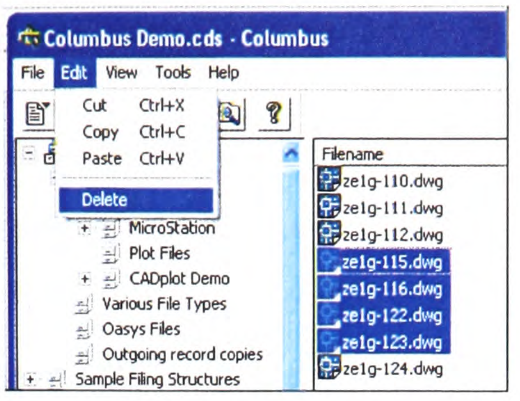

Figure 5.19 Explorer Facilities

Document Property Editing: Columbus has so far been described as a navigator that can read and view document metadata. However, as seen in Figure 5.20, this module is also capable of creating and editing basic descriptive metadata if it is stored in a known format. This can be in standard DIN, XML, Structured Property Data or through plug-in modules which support custom format metadata reading and writing.

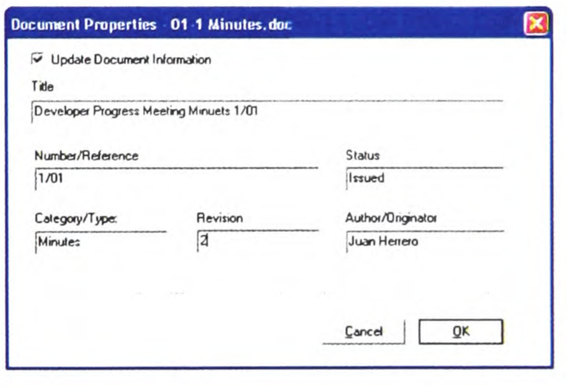

Figure 5.20 Editing Document Properties

Miscellaneous Tools: This module also implements a number of tools that assist the user with general document handling. These include refreshing the displayed document list, opening a Microsoft Windows Explorer session at the current location and editing and reloading the active Columbus Data Structure (CDS) tree. These commands are accessible from the pull-down menu as seen in Figure 5.21

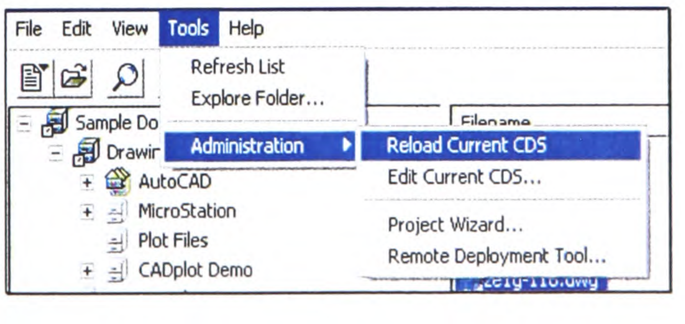

Figure 5.21 Miscellaneous Tools

5.2.6 Document Reporting

In order to allow Columbus to manage documents effectively, interact with other systems and provide a mechanism to archive information in a neutral format, two reporting utilities are available to assist the administrator: the History Report and Document Details Report generation tools.

History Report: The History Report utility, seen in Figure 5.22, allows the user to calculate an edit count and total edit time for a group of documents. With certain types of contracts, it is necessary to charge clients for any time spent working on drawings and the utility can report the total edit time for the lifetime of a document or with a date range.

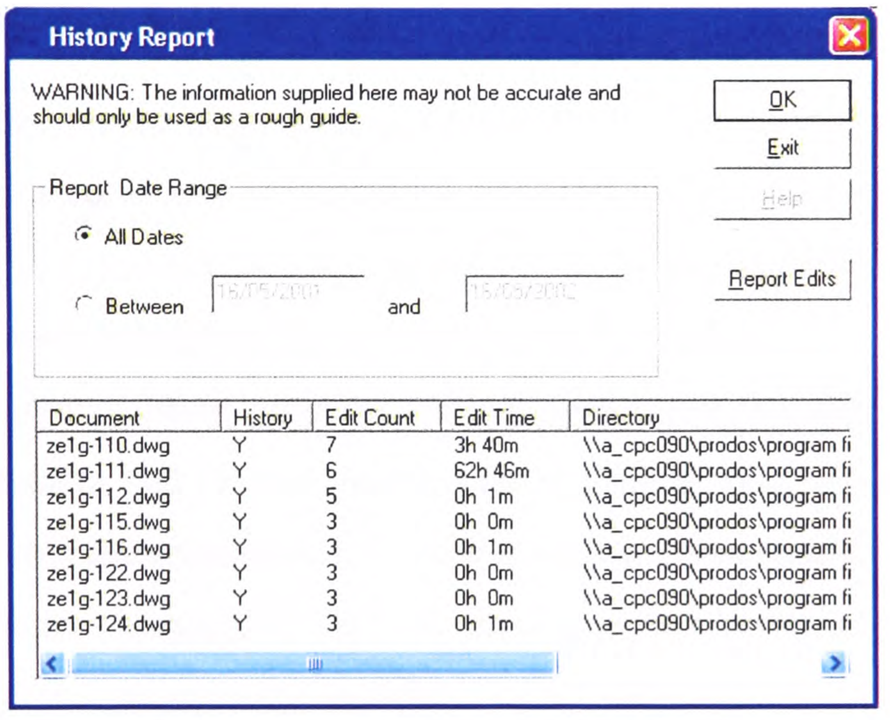

Figure 5.22 Document History Report

Document Details Report: The Document Details Report utility, shown in Figure 5.23, exports the chosen metadata items from a selection of documents. This information is written to a text file which can then be processed by other applications. Typically, it would be imported into Microsoft Excel to produce formatted reports of the selected document metadata. The tool is also useful for producing transmittal information when sending data electronically to other parties that do not use Columbus.

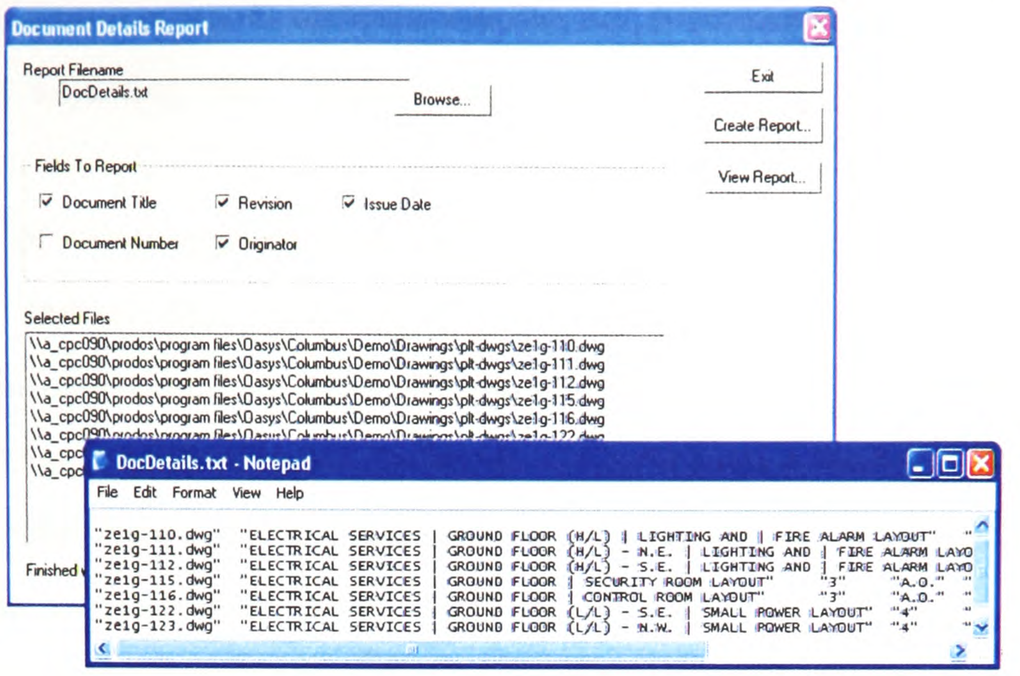

Figure 5.23 Document Details Report

5.3 Metadata Creation

Columbus is able to use and produce metadata in various formats. Metadata is very important when used as part of a process such as issuing documents, generating reports or keeping an edit history. However, it is the creation of descriptive metadata about each document that is particularly of interest.

With the exception of basic property editing, Columbus is really a consumer of this type of metadata and expects that it should already exist for each document. The creation of this information is left to other applications or add-on modules. As long as information appears in one of the supported formats or a plug-in is available to interpret proprietary metadata, then Columbus can directly make use of it.

One example of an external metadata creation utility is Columbus.arx, which extracts title block information from AutoCAD drawings. Columbus.arx is installed if AutoCAD is present on the machine when Columbus itself is installed. It finds the title block information by looking in a project configuration file for the names of the blocks and if a match is found in the drawing, will write the data out in a metadata file. Included amongst the information written are Dublin Core and other properties such as: document title, document number, job title, job number, originator, scale, revision and status. Also, sections are written listing the external dependencies, block table and layer table. Finally, an encoded version of the thumbnail preview image is also added. In addition to extracting the document metadata, Columbus.arx is also responsible for recording the document's editing history and adjusting the reference file paths according to the existing project settings.

A number of other applications have been produced by third-party developers which can extract data from other file types. Amongst them are macros for extracting information from Microsoft Word, Microsoft Excel and the SolidWorks design package. Some of these applications have been created by Columbus users and have been made freely available on the Columbus web site. Others, are available as part of third party applications which use Columbus to solve a particular document handling problem.

5.4 Installation and Configuration

Columbus has been designed to make the process of managing documents, working with project hosting sites and interacting with document control systems as simple as possible. Because of this, it was considered essential that deployment and management of the product would not impose a burden within large organisations. Zero Administration features such as the Network Install Setup and the Enterprise Deployment Tool were created to assist system administrators with rolling-out the product.

5.4.1 Network Installation

At its simplest Columbus can be installed, just as most other applications, on the hard disk of every workstation. The requirement for this, is that the system administrator logs-in to each individual machine and installs the software. In a large organisation, this can prove extremely tedious, as machines may only being accessible "out of hours", therefore proving impractical, inefficient and costly. Furthermore, when an update to the software occurs, the whole process would need to be repeated.

The solution to this problem within Columbus, is to provide a Network Installation, which is also known as a File Server Installation. This places the application files on a server which is accessible through a network drive or share, making it easy to install and update centrally at a single location. Figure 5.24

shows the process of installing Columbus for both standalone and server installations.

By installing the software somewhere that will be seen by users as a read-only location, the system administrator can also guarantee the integrity of the application. However, placing the files on a network server only partially solves the problem, as Columbus does not just consist of application files; it requires start-up shortcuts, ActiveX/COM registry entries to be configured and dynamic link libraries to be installed on the workstation. Columbus handles this by following a network installation with a workstation configuration on each machine. Though this can still seem tedious, it is a one-off task as updates on the central server will not normally require this task to be repeated.

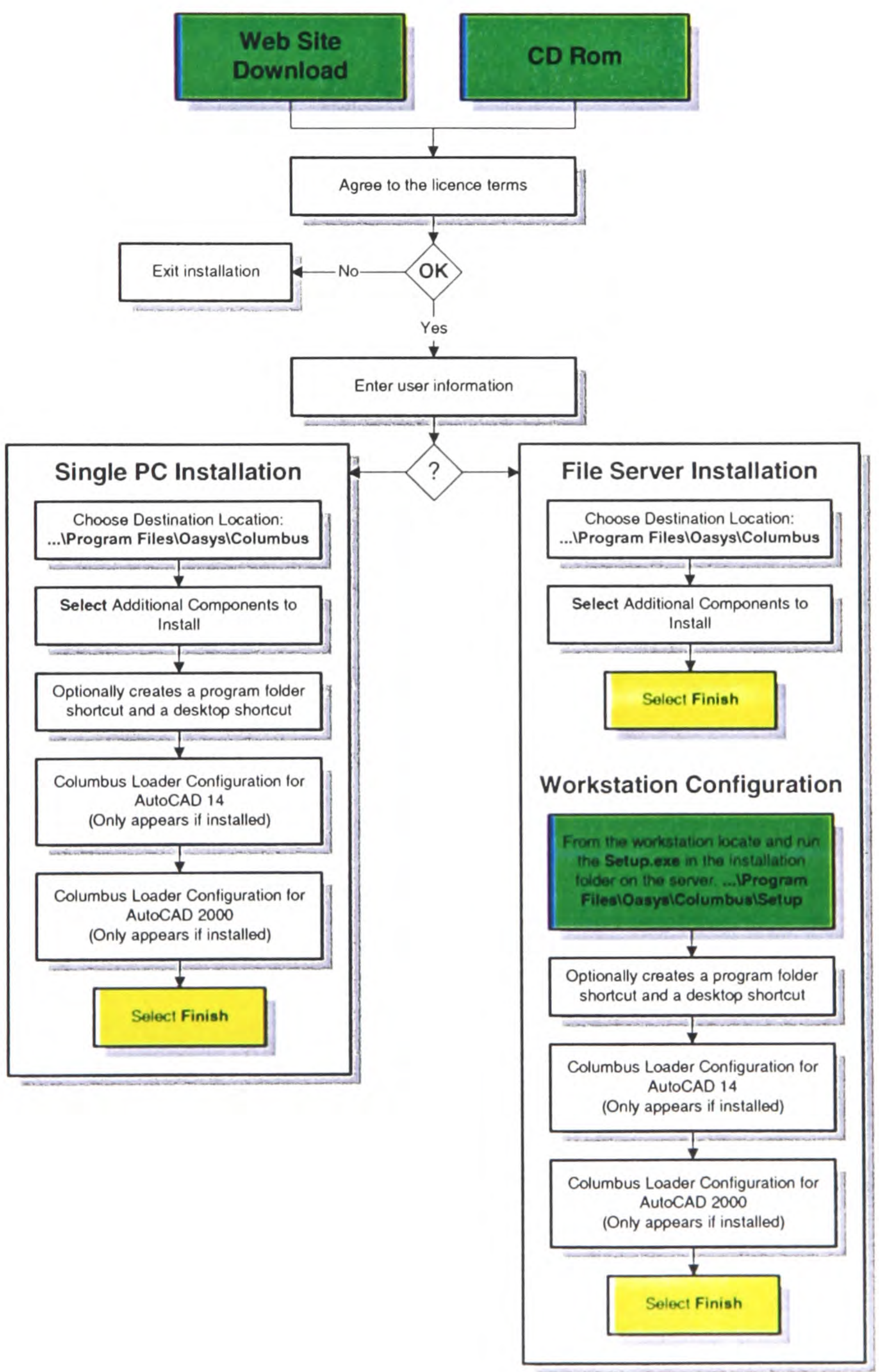

Figure 5.24 Columbus Installation Flowchart

It is also important to consider what happens when Columbus is updated on a server to a newer version. As it is a shared resource, there is a chance that the product might be in use and files could be locked. This would prevent files with the same name from being deleted or overwritten, halting or invalidating the installation. To overcome this, Columbus uses numbered component files such as "cbs#1023.exe". When a revised version of component is issued, it will have a different file name such as "cbs#1024.exe". When Columbus starts, a loader application will look for the highest numbered component to run. Generally this solution works well, but requires additional handling when the component is an ActiveX object and is referenced by its filename in the registry. The way that Columbus overcomes this is to re-write the component's filename to the registry if it has changed, when the application is restarted.

5.4.2 Enterprise Deployment Tool

Though a network installation can simplify the installation and updating process, it still requires administrators to visit each workstation at least once. In addition, many organisations favour installing software locally on each workstation. This is because the majority of software applications are unsuitable for being installed on a central server or for performance reasons. Because of this there is a utility available for Columbus called the Enterprise Deployment Tool, which makes it easy for System Administrators to install or upgrade applications on machines remotely. This can even be done whilst the users are logged in and working. The tool works by connecting to the registry and hard disk remotely, updating the software, registry entries and shortcuts.

As seen in Figure 5.25, when the tool is run, Administrators are presented with a tabbed dialog box interface which allows them to push out and configure Columbus on each workstation to run from a central server or be deployed to the local hard disk. Additionally, Columbus specific settings can also be pre-set and there are options to indicate whether shortcut icons should be created and if a notification should be sent to the user on the target machine. A special status window can also be launched to check machine settings. This last item on its own makes the tool extremely useful and can be used to determine the operating system service pack level, bios settings and general information from all computers on a network.

The tool works in a non-intrusive manner and does not require the current user on the target machines to log out or even exit Columbus. If some shared files needed updating and are found to be in use, then users are prompted to reboot. The computers tab allows the administrator to select the target machines for deployment; which can be done by typing them in manually, loading them from a pre-configured list or browsing the network. Once this has been done, the deployment process can be started. If there are any failures, these are reported without halting the process and a button is also available to retry failed machines.

The Enterprise Deployment Tool can be run at any time to ensure that all workstations are up to date and it greatly improves the process of deploying Columbus within any organisation.

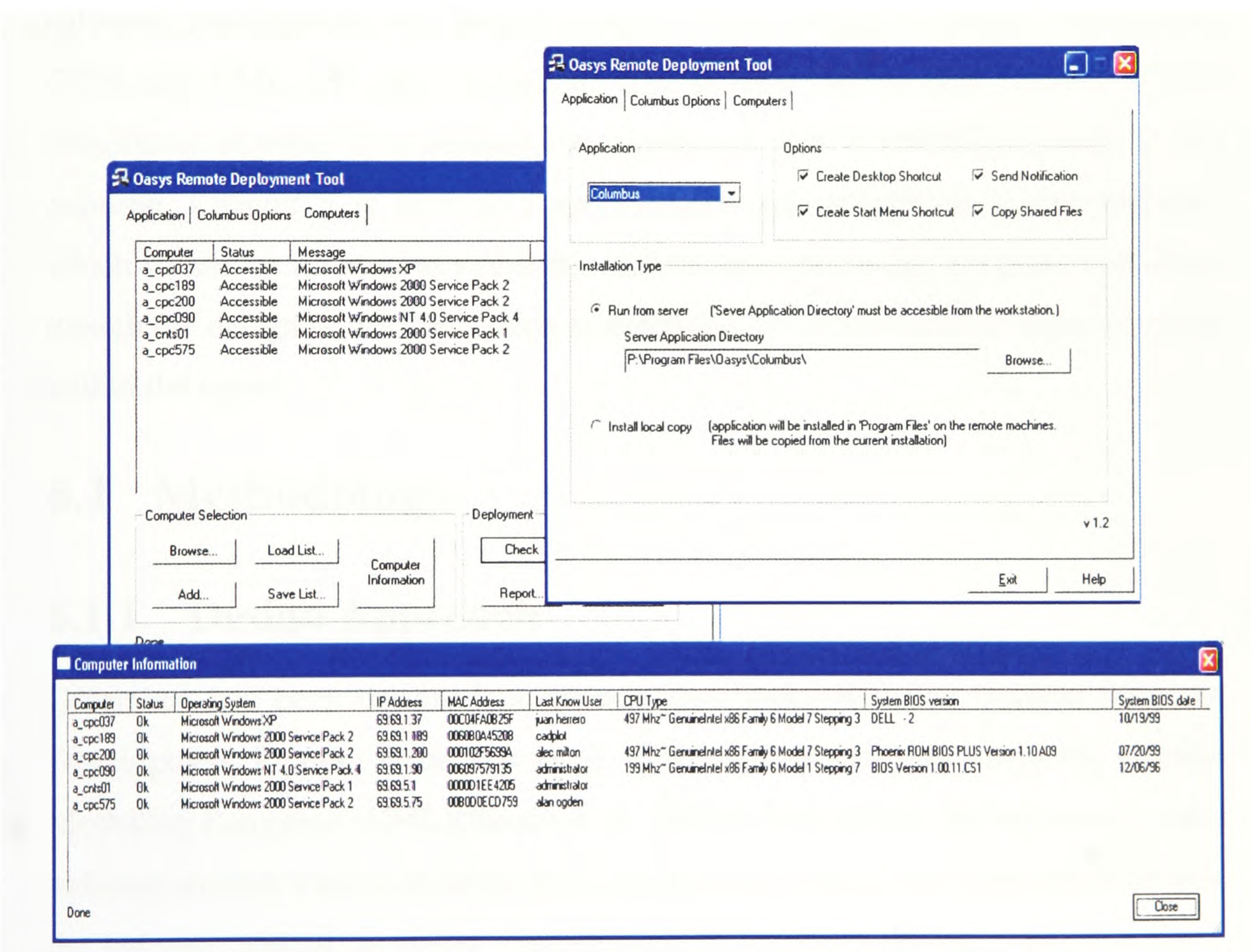

Figure 5.25 Enterprise Deployment Tool

Chapter 6

Columbus Architecture and Design

This chapter covers the architecture and design of Columbus. A description of the methodology used is presented first, followed by a high level description of the main Columbus modules. The rest of the chapter covers the design of these modules in greater detail.

It is assumed that the reader has a background knowledge of object oriented software development and programming as terms such as classes, inheritance, COM and UML are used throughout. The chapter can be read superficially or completely skipped if a detailed description of how Columbus is built is not required. Chapter 5 is used as a specification and description of the software, which should be referred to whilst the architecture and design are presented. Even though the design of the application is described, no source code is made available within the report.

6.1 Methodology

6.1.1 Design Approach

Columbus has been designed in general accordance with the Unified Software Development Process (Jacobson et al. 1999) and specifically using the Unified Modelling Language (UML)(Booch et al. 1998). UML allows the analyst to model software to such a level of detail that a programmer has a clear description of how the software should be coded. Nevertheless, it has been decided that in order to present the research more effectively, the design will be limited in scope. This is because of two reasons: firstly, there is a limit on the size of the report, and secondly, it would become extremely repetitive and tedious to the reader. If this approach were not taken, then simplistic and mundane aspects of the software would have to be meticulously described. The approach taken is to provide an overall view of the application, followed by a description of each module's most critical design aspects. This, though only partially complete will have covered any particularly challenging characteristics of the design.

6.1.2 Architecture Views

Using the Unified Software Development process, it is possible to consider the architecture as different views. The approach used here is the "4 + 1" view model (Priestley 2000), which separates the architecture into five concurrent views, each addressing a particular concern. The "4" views are the design, implementation, process and deployment. The other view, called the use case view, has the role of integrating the other views, which is why it is known as the "+1" view. Briefly, they can be described as follows:

- **Use Case View** : Is where problem domain is developed, presenting a description of what the system must provide to its end users in order to satisfy the functional requirements. As other views are constrained by these requirements, this is why it has a central role driving the design. No internal architecture or details of how the application works is covered here, just external interactions. In this view, static modelling consists of use case diagrams and dynamic modelling consists of interaction diagrams. Interaction diagrams in this work are presented in the form of sequence diagrams.
- **Design view** : Also known as the logical view, presents the detailed object model, containing the program components such as classes and interfaces. The information in this view is directly relevant to programmers, as it provides all the details of how the application should be coded. Design view modelling is based on class diagrams.
- **Implementation view** : Addresses issues regarding the physical files and the process of assembling these to produce a running system. It also addresses any configuration management issues. The view is modelled as a component diagram.

- **Process view** : Covers additional system issues regarding concurrency, synchronisation, threads, processes, performance, scalability and throughput.
- **Deployment view** : Is concerned with the system architecture, hardware, distribution, delivery and installation. It encompasses the nodes that form the system's hardware topology on which the system executes. The view is presented as a deployment diagram.

Each of the five views is a projection into the organisation and structure of the system, focusing on a particular aspect of the application. Each of them can stand alone so that different issues of the system's architecture can be analysed independently.

In the detailed design that follows, the use case and design views will be looked at for each application package, discussing how they interact with the user and relate to the other packages. Subsequently, the implementation and deployment views for the whole system will be presented. Though the application is multi-threaded and runs various simultaneous processes, presenting the process view is beyond the scope of this report and is not included, though it can be discussed separately.

6.1.3 Class Diagram Notation

In order to keep the report reasonably compact, class diagrams are presented in condensed form. Generalisation and association/aggregation relationships are shown, however no class members or methods are presented in the class diagrams as they would clutter-up the design. Where a class is defined in a standard framework, its name is followed by a bracketed acronym to indicate this. For example, classes from the "Microsoft Foundation Classes" framework are followed by "(MFC)". In these cases, no further details are given about the design of those classes, though a reference may be provided for more information.

6.2 Environment

6.2.1 System Requirements

Columbus is designed to run under all current Microsoft Windows operating systems (Windows 98, ME, NT, 2000 and XP) on Intel's x86 hardware. New versions of Columbus are tested on all these operating systems directly or using a Connectix Virtual PC (Connectix 2002). In addition, Columbus has been reported to work on the Apple Macintosh using the windows emulator. Tests have also been carried out under the Linux operating system using the WINE (WINE 2002) environment and it is listed as a supported application in their database. A prototype has also been developed using the WINE libraries to create a native Linux application.

6.2.2 Development Environment

Though not strictly part of the design stage, it is important to highlight the technology that has been used in building the application. As already stated, Columbus has been predominantly designed using the Unified Modelling Language. This was done for the earlier versions as hand drawn or Microsoft Visio sketches and more recently using the Rational Rose Development Environment (Rational 2001). As a programming environment, all modules are written in C++ using Microsoft Developer Studio, and are heavily dependent on the Microsoft Foundation Classes. A good in depth reference for the MFC architecture is Programming Microsoft Visual C++ (Kruglinski 1998). It also makes extensive use of the Standard Template Library (Ammeraal 1997) and the Advanced Template Library (Armstrong 1998) for COM connectivity between internal modules.

Additionally, the following libraries are used within the application:

- **AutoDesk ARX**: For working with AutoCAD .
- **OpenDWG**: For direct access to drawing.
- **TWAIN**: Standard toolkit for access to scanning devices.
- **PandaPDF**: PDF file writing library.
- **LIBtiff**: TIFF file manipulation.
- **ZLib**: Handling compressed files.
- **Expat**: XML reading and writing.
- **Xerces**: XML validation.

Moreover, the viewing engines are based on libraries licensed from Rasterex and Stellent.

6.3 High Level Application Architecture

This section presents a high level overview of the Columbus application. Architecturally, Columbus can be separated into user packages and Columbus Services. A general description of each of the components and how they relate to each other will be given.

6.3.1 User Packages

Functionally, from a user perspective, Columbus is internally separated into six core functional constituent parts as seen in Figure 6.1. These components are referred to using the UML term "packages", though this may seem confusing to those accustomed to referring to a suite of applications as a package.

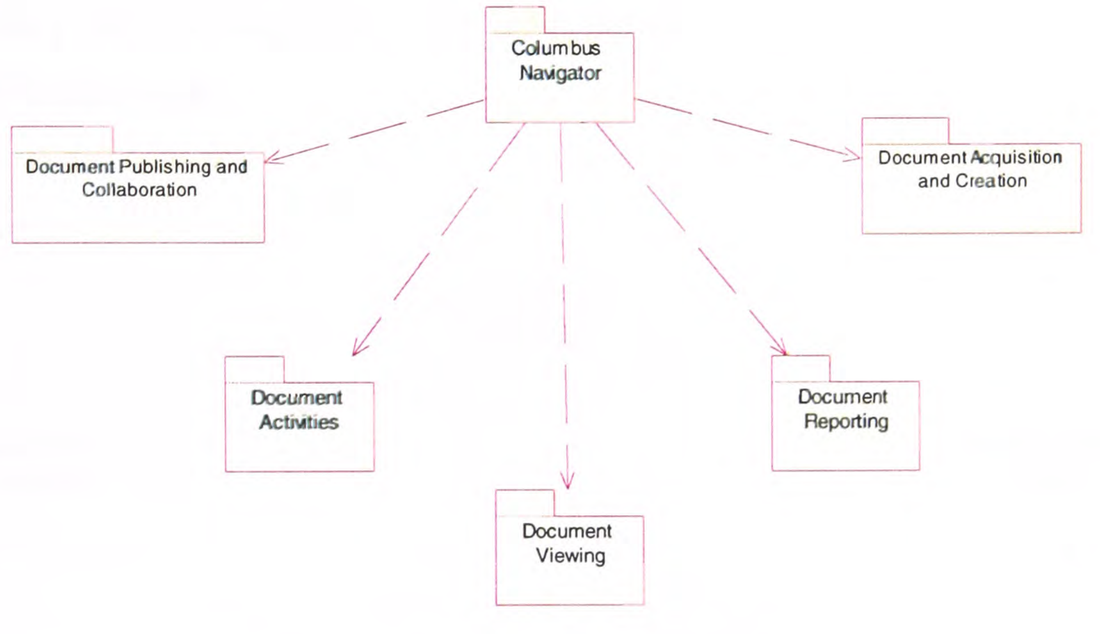

Figure 6.1 High Level Columbus Architecture

There is a directly correspondence between these packages and the component modules presented in chapter 5. As they were previously described in detail, only a listing with a brief reminder is now given. The user packages are:

- **Columbus Navigator**: Browses projects areas, lists documents and shows metadata. Also acts as the main user entry point to all other packages.
- **Document Viewer**: Used for viewing documents and allows users to produce red-line comments.
- **Document Publishing and Collaboration**: Assists in sharing information with other project participants.

- **Document Acquisition and Creation**: Is responsible for obtaining documents from external sources, creating new documents and setting up project areas.
- **Document Activities**: Implements most of the document manipulation commands.
- **Document Reporting**: Provides information and status reports on documents.

6.3.2 Columbus Services

In addition to the user packages, there are a group of modules that do not interact directly with the user but provide important facilities to meet some important functional requirements. As shown in Figure 6.2, these are the input-output handler, file-naming service, metadata service and the configuration manager. Though they will be described briefly, a detailed design for them will not be presented in this report.

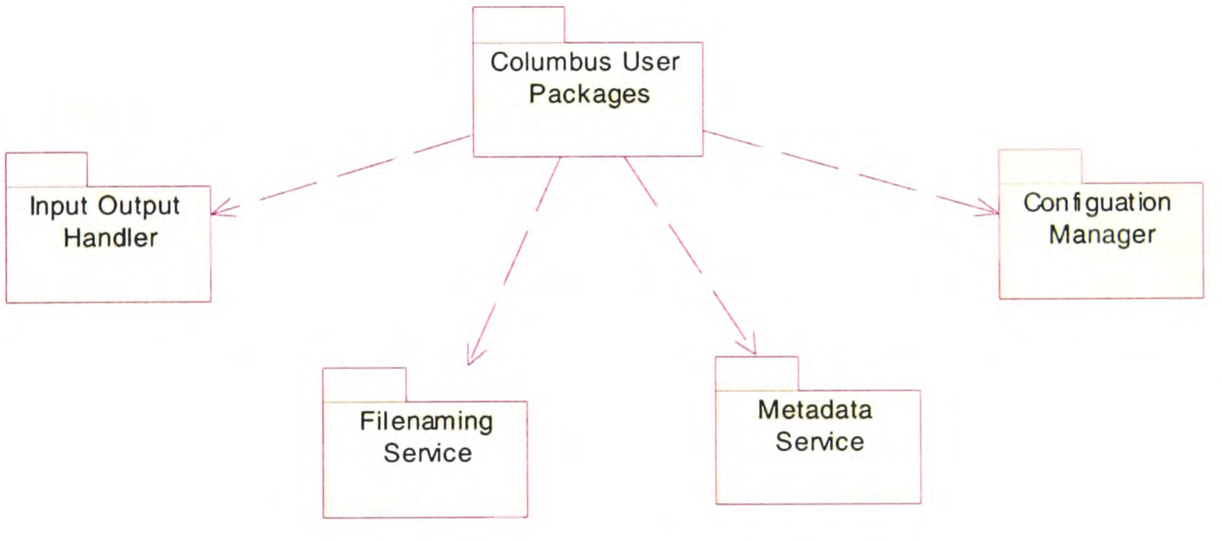

Figure 6.2 Columbus Services Packages

Input Output Handler : This package is responsible for providing all other packages with seamless access to documents regardless of the protocol or security mechanism used. Currently, the following protocols are supported in Columbus: file-system, FTP and HTTP (prototype using WebDav). The module also handles security issues, so that passwords are transferred across component boundaries to produce a seamless security model.

File-naming Service: Provides other packages with a centralised method of naming documents. Typically, different automatic file-naming methods are used which can allocate unique names and number to documents. The file-naming scheme is also extensible by third parties.

Metadata Service: Allows centralised access to document metadata. The module can read metadata in various different formats. For example, metadata can be extracted from drawings, Microsoft Office documents, directly from DIN and XML files or via third party add-on handlers.

Configuration Manager: Has the responsibility of managing access to Columbus' various component modules. The package is responsible for registering, re-locating and changing the path of the various COM objects that Columbus requires. It achieves this regardless of the permissions that the user has, or the type of installation (i.e. workstation or server). The package also includes the Enterprise Deployment Tool, which allows the administrator to remotely install Columbus on other workstations using remote registry and disk access functions.

The package also allows third-party extensions to other packages to be managed in a seamless manner.

6.4 Navigator Package Design

Having provided an overall description of the architecture of the application, this section now presents a detailed design of the first component, the navigator package. Subsequent sections will present the remaining packages in detail.

The Columbus Navigator package forms the core of the application. It is the entry point to most other packages and as such is the central hub for initiating all user activities. A description of the use cases, sequence diagrams and logical view is presented.

6.4.1 Use Case View

Though the navigator is the central Columbus module, its core responsibilities can be summarised quite simply in the use case diagram shown in Figure 6.3. The use cases cover loading a Columbus Data Structure (CDS) file into a tree structure view, navigating the tree, populating a list with the documents defined by a tree node, selecting an individual document and showing its metadata.

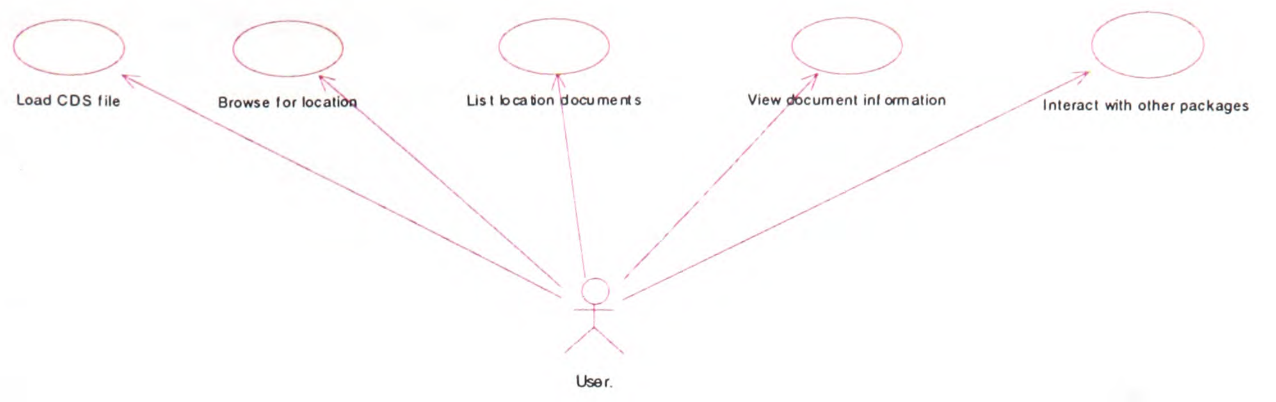

Figure 6.3 Columbus Navigator Use Case diagram

The sequence diagram presented in Figure 6.4 shows the objects required to implement these use cases and the interaction messages that are sent between them. To simplify the analysis, a single unified diagram is presented covering the use cases.

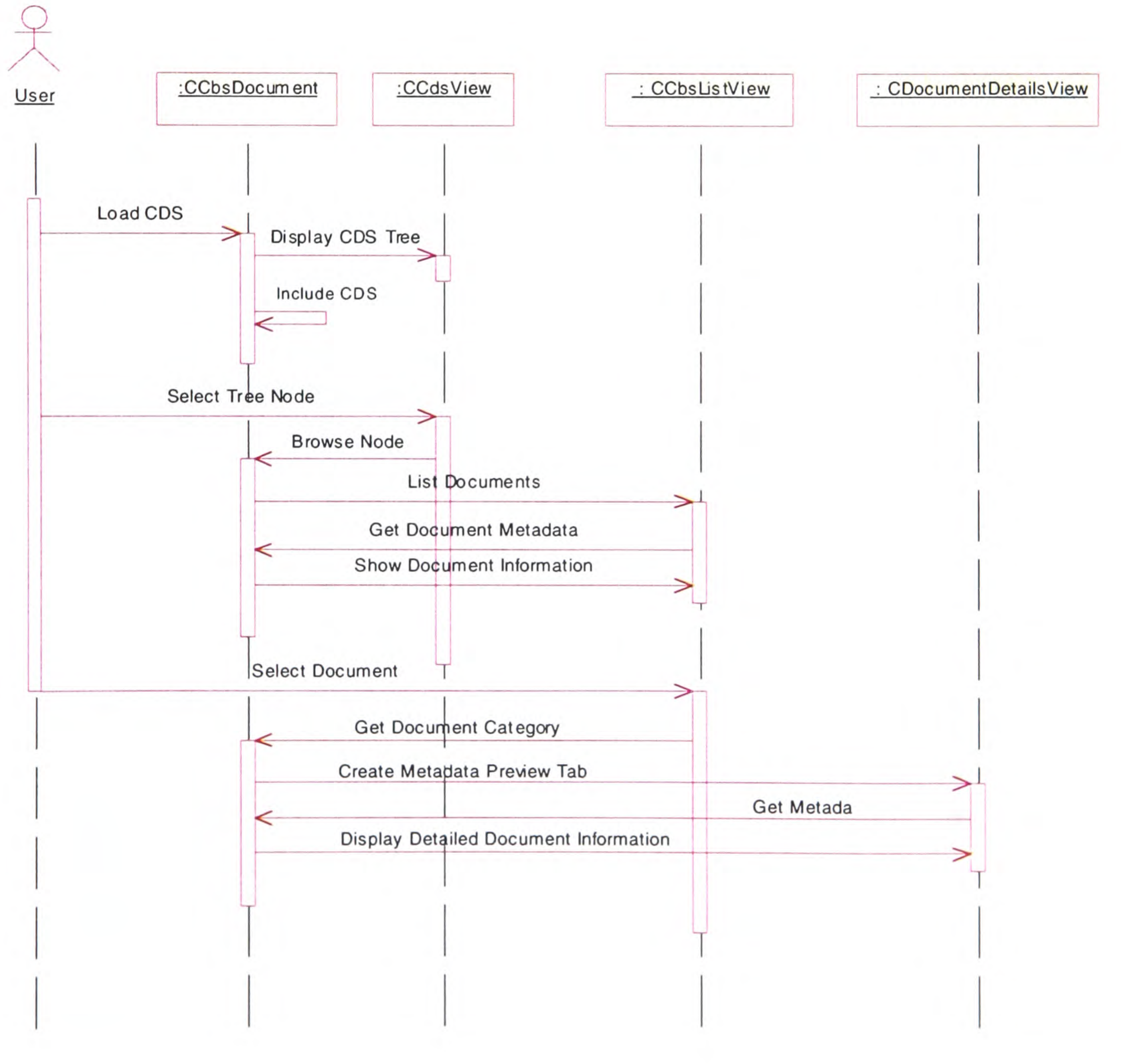

Figure 6.4 Columbus Navigator Sequence diagram

The navigator allows the user to load a CDS file into the CCdsDocument, which is then presented in a tree view structure. The CCdsDocument object acts as the coordinating class which links all the browsing activities. The CDS file contains details of where each node should point to, how the information should be filtered, the label that will be displayed in the tree and how it will appear within the hierarchical structure. It is possible to have a reference to another CDS file using the include keyword, which will load it automatically. The CDS tree structure is loaded into the CCdsDocument object and is presented to the user in the CCdsView object as seen in Figure 6.5.

When a node is selected, this will cause a list of all documents that match the required pattern to be displayed. This is presented to the user in a list view that is managed by the CListView object. As a background activity, in another thread, metadata such as the title, document number and status will be retrieved for each document and added to the list.

Once the list has been populated, the user can select a particular document and detailed information about it is presented in the CDocumentDetailsView object. In addition to the items shown in the sequence diagram, the navigator can invoke any of the document activities or tools offered by other Columbus modules.

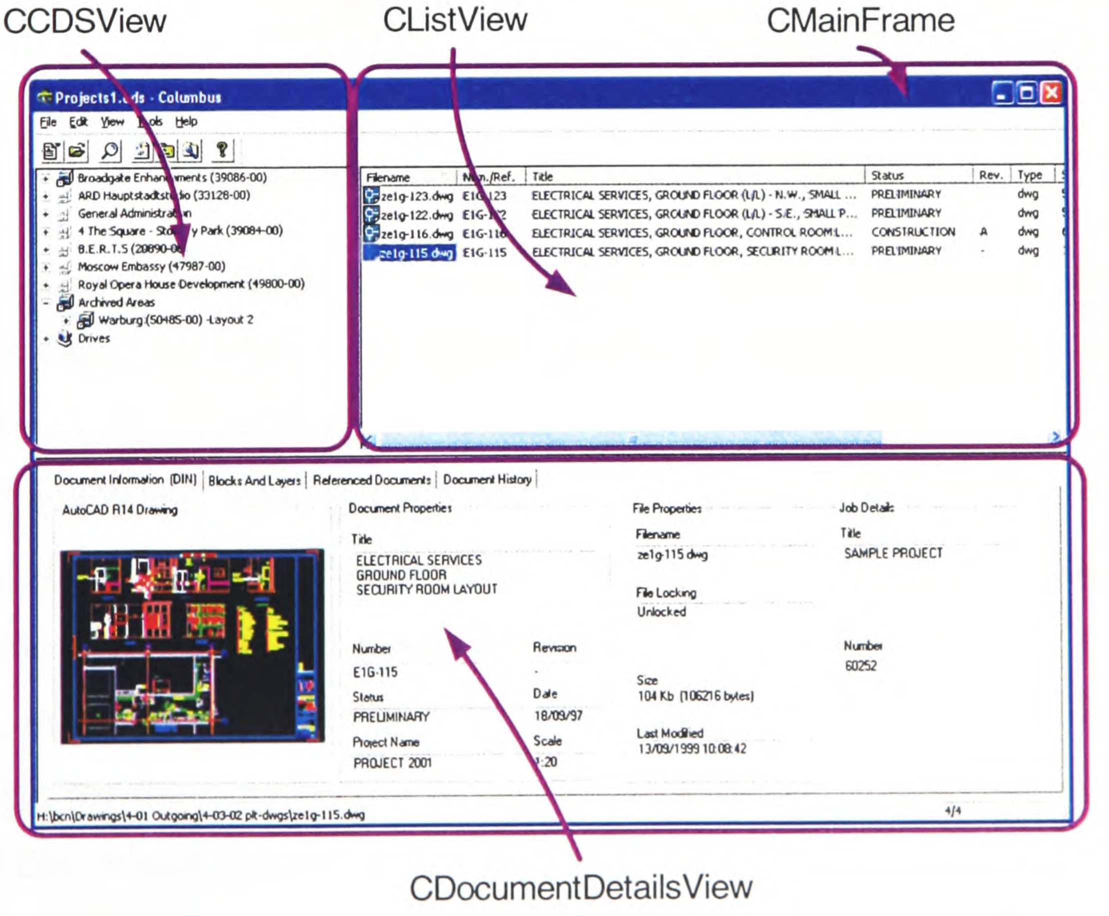

Figure 6.5 Navigator Object Appearance

6.4.2 Design View

The Columbus Navigator architecture is based on a document-view architecture, using Microsoft's Foundation Classes framework (MFC). The top-level class design for the navigator can be seen in Figure 6.6.

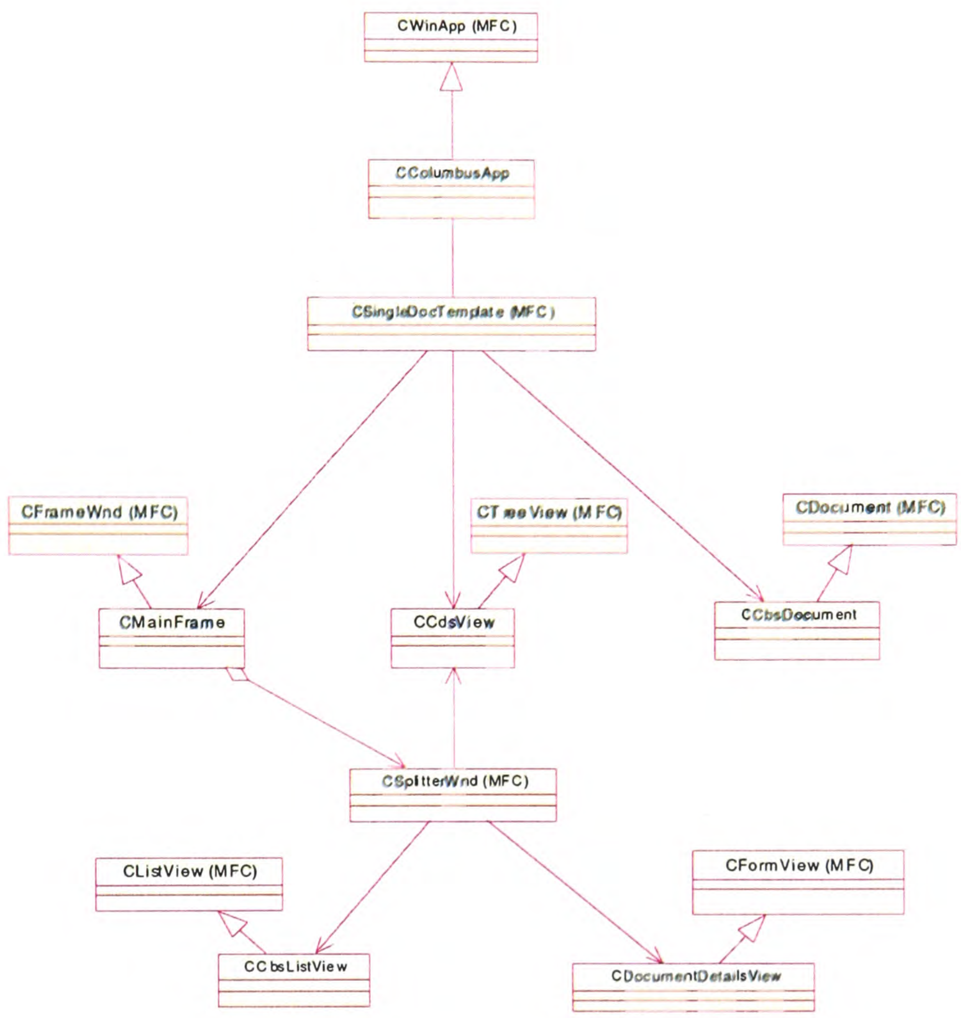

Figure 6.6 Navigator Top Level Class Design

The Navigator is a Single Document Interface (SDI) application, defined in CColumbusApp, an MFC CWinApp derived class. The application contains an MFC CSingleDocumentTemplate that has a frame (CMainFrame, derived from MFC CFrameWnd), view (CCdsView, derived from MFC CTreeView) and a document (CCdsDocument, derived from MFC CDocument). It is important not to confuse the MFC notion of a document with the document files that Columbus manages. In the MFC document-view architecture, a document is the coordinating class where the main processing engine is coded. In this design, the document is the CDS file, which the navigator window opens and presents its contents in the CCdsView.

Though CCdsView is a tree view that shows the CDS structure in a tree style, it is linked to the CCbsListView and CDocDetailsView though an MFC CSplitterWnd, which is a container for all the views and is contained within the application's window frame, CMainFrame. The CCbsListView is responsible for presenting the document list and the CDocDetailsView displays detailed metadata for each one. In the earlier diagram, in Figure 6.5, it was shown how all these classes appear to the user.

As described in the use case view, selecting a node in the CCdsView causes a list of documents to be shown in the CListView. In a background thread this populates the list with any relevant metadata. When a particular document is selected in the list, detailed metadata is presented in CDocumentDetailsView. This class and its associated classes are of particular interest and their design can be refined further. CDocumentDetailsView is responsible for presenting a detailed view of all the metadata and information about a document. This is normally presented as a series of tabs within the view. Figure 6.7 shows the design of the classes associated with CDocumentDetailsView.

CDocumentDetailsView is derived from a standard MFC CFormView and has a CPropertySheet object that acts as a container for various CPropertyPages. CPropertyPage objects are standard MFC objects, which are used as tabs to present different categorised metadata to the user. They are created dynamically as and when required and according to the document type. So, metadata relating to an AutoCAD drawing may be presented very differently from the metadata about a Microsoft Word file. CPropertyPages can either be implemented as objects based on the CMetaDataPreview or CSupplementaryInfo classes.

There must always be exactly one CMetaDataPreview tab for each document, showing its primary metadata. If additional information needs to be shown, then an unspecified number of optional CSupplementaryInfo tabs may be created. For example, a history tab will be shown if a document has a history log file or the blocks and layers table may be presented for an AutoCAD drawing. Both CMetadatapreview and `CSupplemetaryInfo are abstract base classes, which generalise specific tabs such as CInfoPreviewPlot, CInfoPreviewOffice, COasysPreview, CInfoPreviewUstn or CDwgPreview, to mention five of different document types supported.

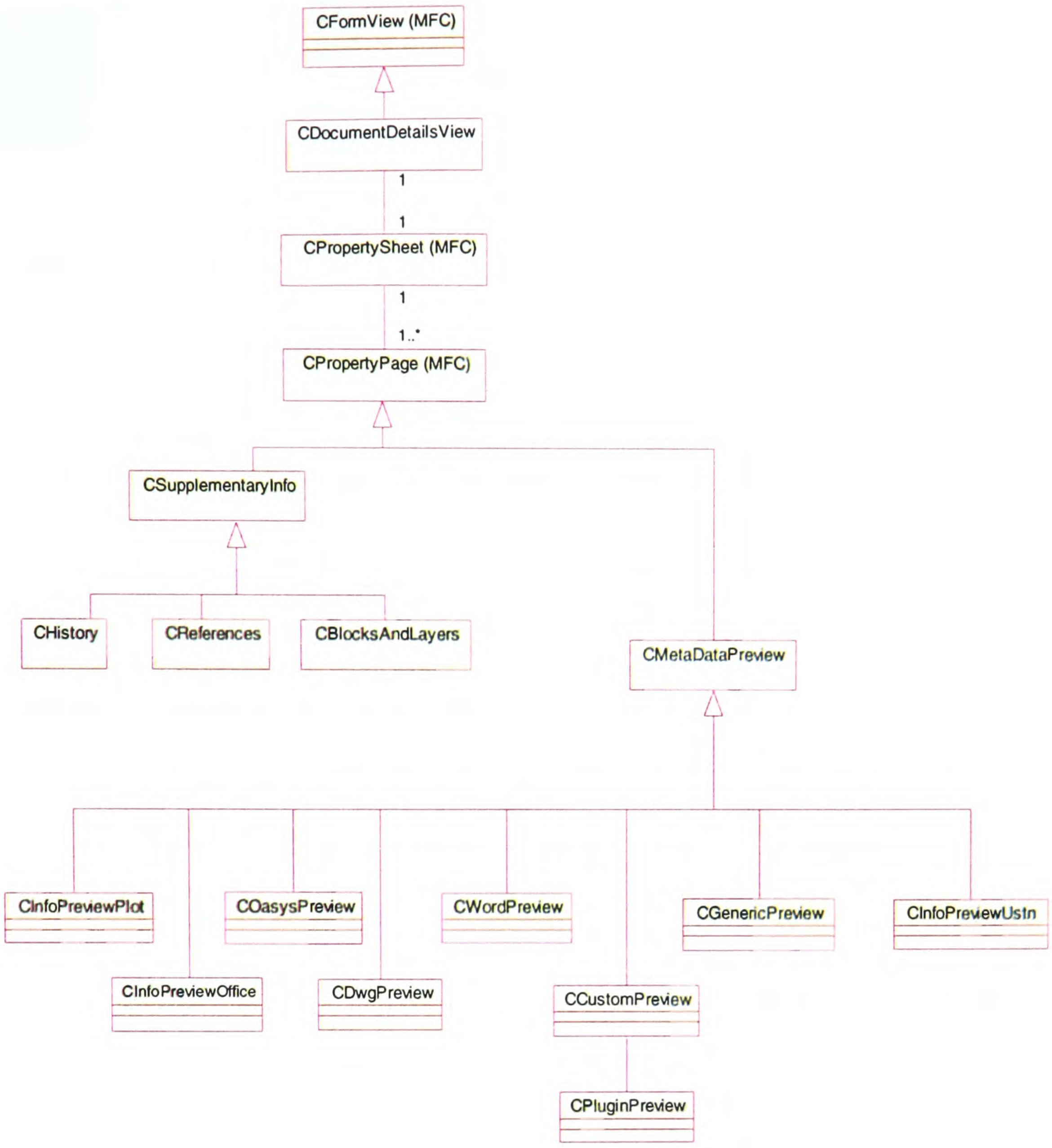

Figure 6.7 CDocumentDetailsView Class Design

Figure 6.8 shows some examples of how CMetadataPreview tabs may appear to the user as different document types are selected. In this case, the tabs shown would appear when selecting an AutoCAD drawing, Microsoft Word file, Oasys document, a plot file or a generic document. They clearly show that different documents have distinct metadata requirements and Columbus is able to handle them in a seamless manner. One of Columbus' most powerful features is the way in which it customises how each document's metadata is presented to the user.

In addition, not only are there a large number of built in document types supported, but a special third party developer tab known as a CCustomPreview is available where others developers can produce their own plug-in tabs using COM to implement a CPluginPreview object. These can be written in C++, Visual Basic or any other language capable of generating COM objects.

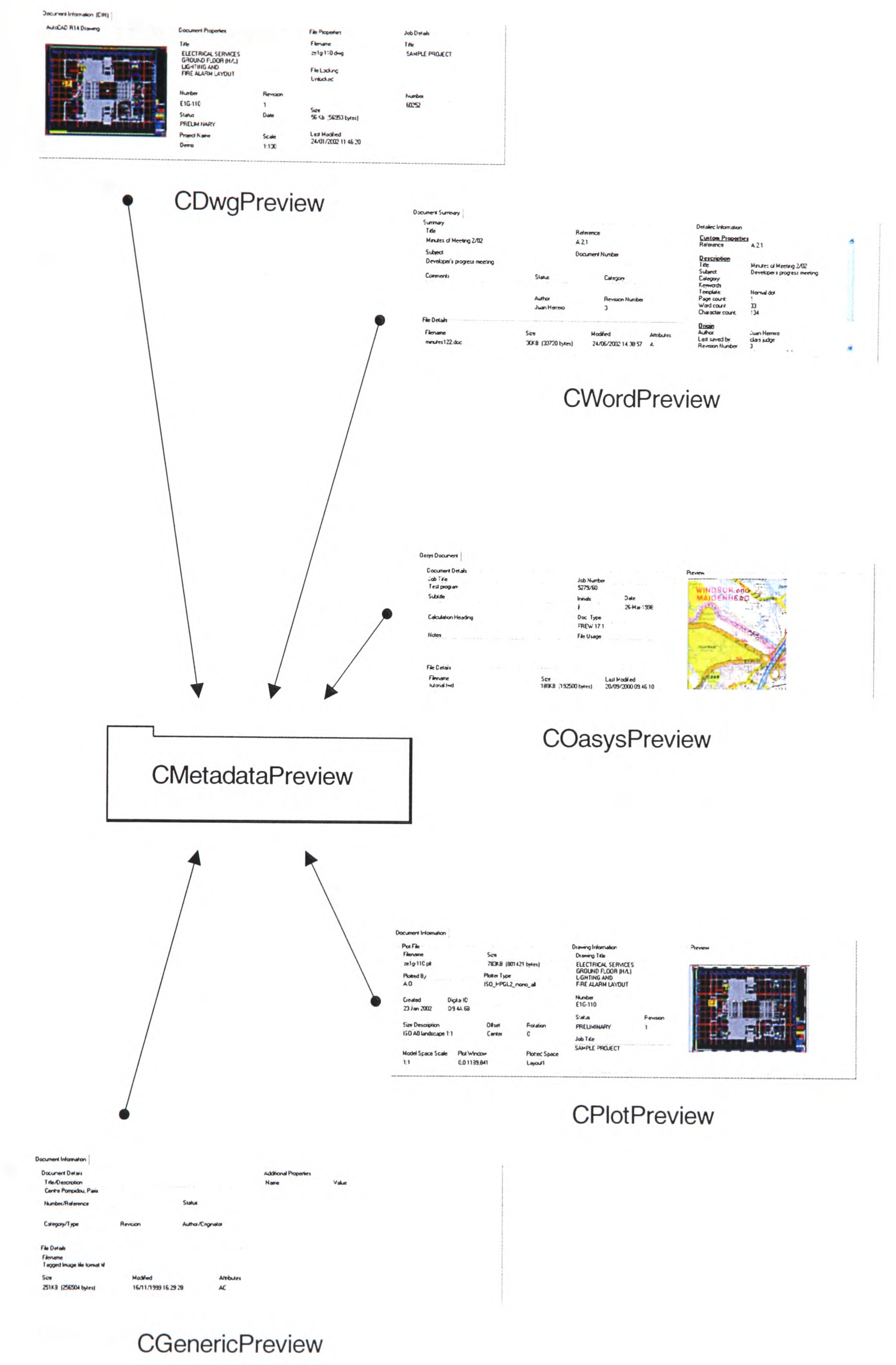

Figure 6.8 Metadata Preview Tab Inheritance Appearance

6.5 Document Viewing Package

This module is responsible for viewing and redlining documents. The viewer provides an outer shell linking specific viewing engines to the Columbus Navigator.

6.5.1 Use Case View

Despite the fact that the viewing package is quite complex internally, its interface with the user is quite simple. As seen in Figure 6.9, the user interacts directly with the document or can pin the viewer.

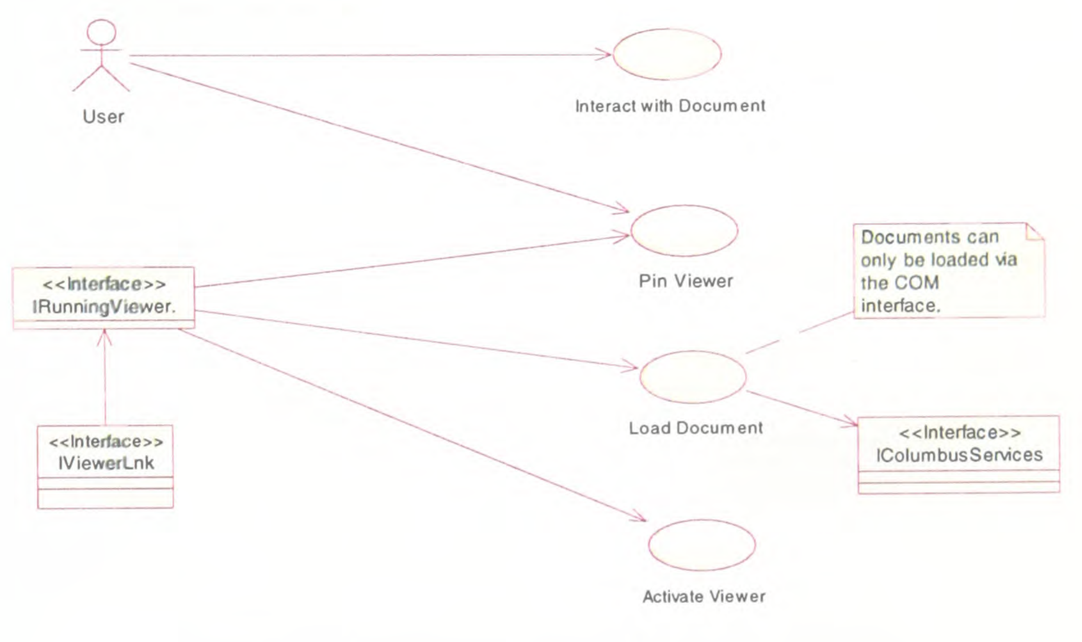

Figure 6.9 Document Viewer Use Case Diagram

Interacting with the document encapsulates many activities such as zooming, panning, printing and redlining. These are all be grouped in one category as there is no direct relationship between them and the rest of the application. The module is directly linked to the Columbus navigator module via COM interfaces and also makes use of services provided by other Columbus modules. These include accessing metadata information about each document, adjusting project paths for resolving reference file dependencies, loading documents from different file-system protocols and updating redline information after commenting.

These interfaces tie the modules together forming a loosely coupled application. The Iviewer interface allows Columbus to start the viewer if it is not running, load a document for viewing or, if requested though the navigator, view a document in a new window. The IRunningViewer interface is used by Columbus to connect to a viewer which is already running and exposing this interface. The interface can be used to load a new document, pin it or activate the viewer if it is minimised. A new viewer process cannot be started through this interface and can only be done through the Iviewer interface, which for simplicity is not shown in the diagram. When the viewer is pinned, the IRunningViewer interface is revoked and Columbus is forced to instantiate a new viewer process when requested to view another document, as seen in the sequence diagram in Figure 6.10.

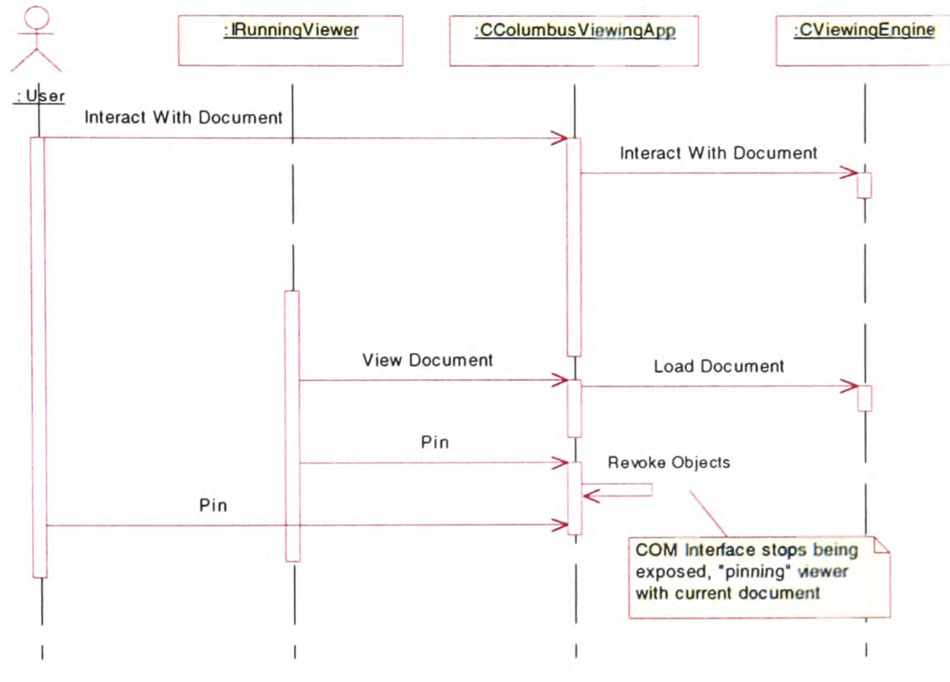

Figure 6.10 Document Viewer Sequence Diagram

It is important to note that the only way to load a document is via the COM interface. If it were possible for the user to load a document directly, this module could be used as a standalone viewer, possibly compromising our licence agreements with the viewer technology vendors.

It is also worth stating that loading a document does not explicitly request the use of a particular viewing engine and it is decided internally within the module which particular engine is best suited for viewing a document.

6.5.2 Design View

The module is created as a Microsoft Visual C++ MFC project as an application-frame-child view application rather than using the more heavyweight document-view architecture. This is because the view is just a simple link to the engines, which handle viewing in their own way. The class diagram presented in Figure 6.11 shows how the module is structured internally.

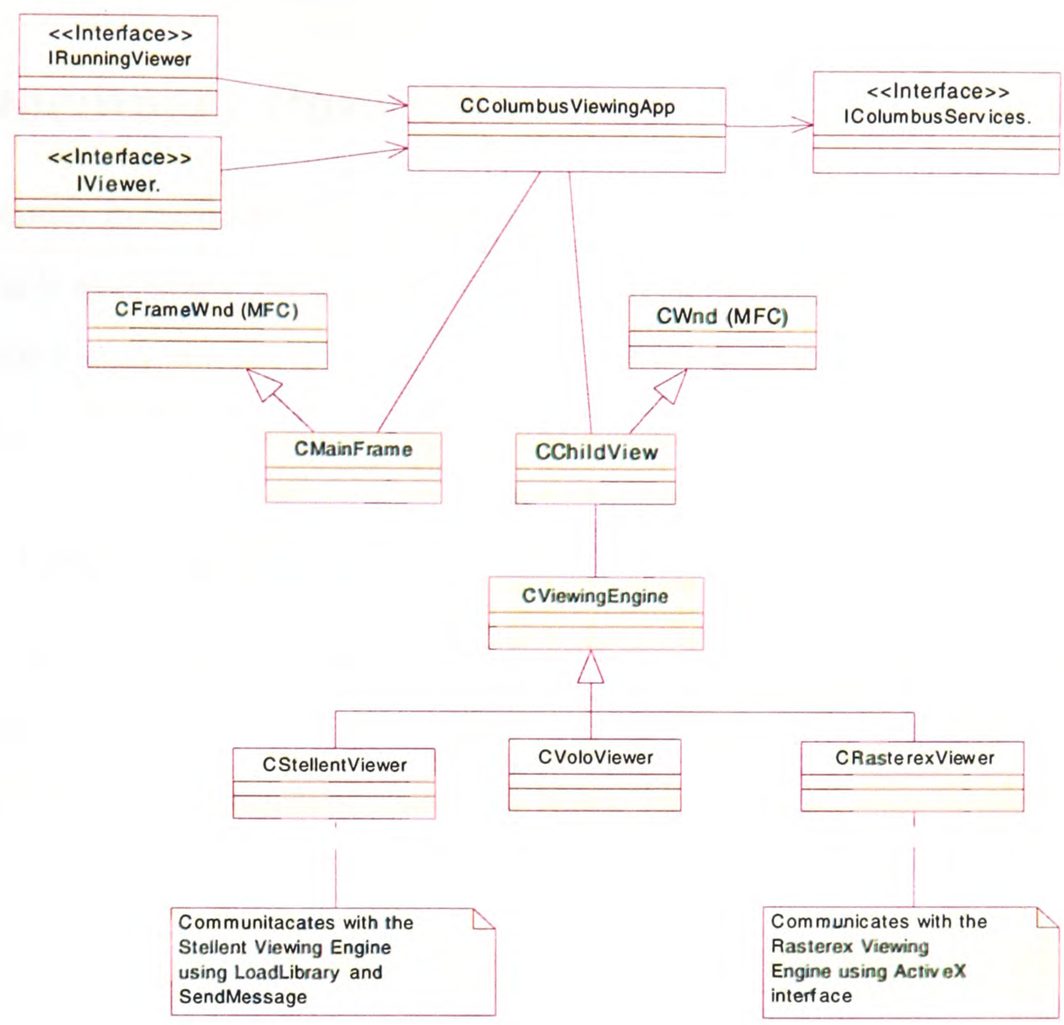

Figure 6.11 Document Viewer Class Diagram

The module consists of an application class (CColumbusViewingApp) that contains a frame window (CMainFrame) and view class (CChildView). The application class is responsible for the initialisation of the module, implementing the IRunningViewer and IViewer COM interfaces for the Columbus Navigator to use and to access services provided by other modules. CChildView is derived from a simple CWnd window that allows the module to interact with the user by instantiating one of the CViewingEngine classes. CViewingEngine is an abstract base class that allows different viewing engines to be plugged in and present a common interface to the CChildView. CRasterex is one of these classes and communicates directly with the RxHighX ActiveX component supplied by Rasterex. The other one is CStellentView, which talks to the Stellent viewing engine by loading a dynamic link library and using the Windows SendMessage API. The choice of engine is made by the application class based on the file format of the document. Any viewing engine can be incorporated into the module by providing an interface in the form of a class derived from CViewingEngine. Though inactive in the current version, a further engine based on Autodesk's Volo View technology has been coded in the CVoloViewer class.

6.6 Document Publishing and Collaboration

The document publishing and collaboration module comprises the File Issue tool and the Pack and Show facility. Pack and Show is not really a software component and therefore will not be covered in this section, only the File Issue tool will be presented.

6.6.1 Use Case View

The file issue tool can gather documents and publish them to remote locations. As can be seen in the use case diagram in Figure 6.12, the user can select a number of options, choose the documents and then issue them.

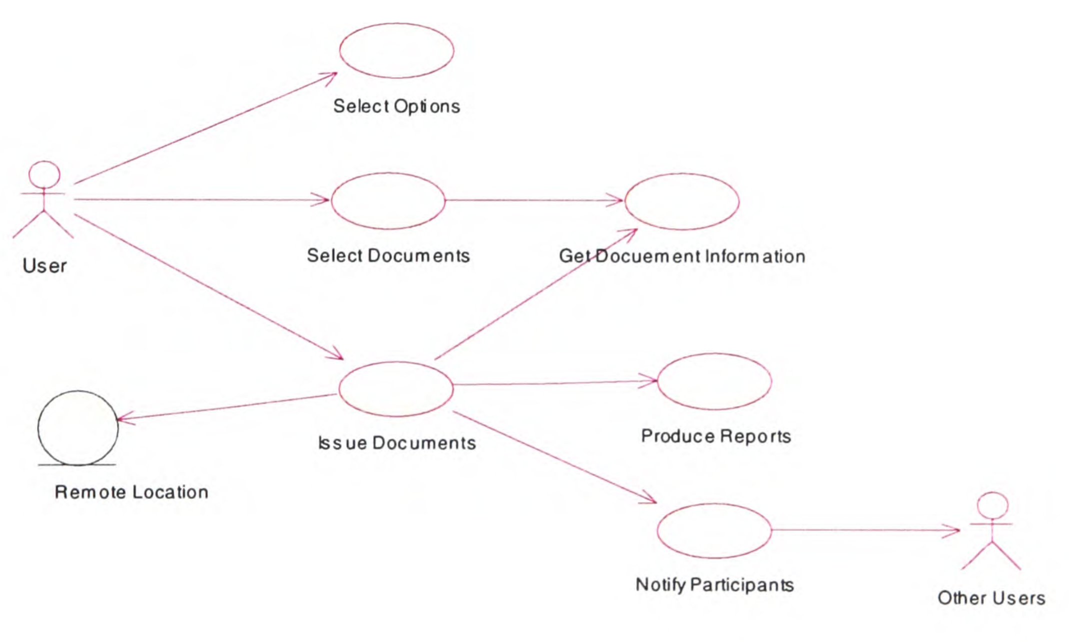

Figure 6.12 Document Publishing Use Case Diagram

As seen in more detail in the sequence diagram in Figure 6.13, when documents are issued, the main application reads the settings from the options dialog and starts the document upload process to the remote site. Once this has been completed, archive copies of the data are made, transmittal reports are produced and other project participants are notified via email.

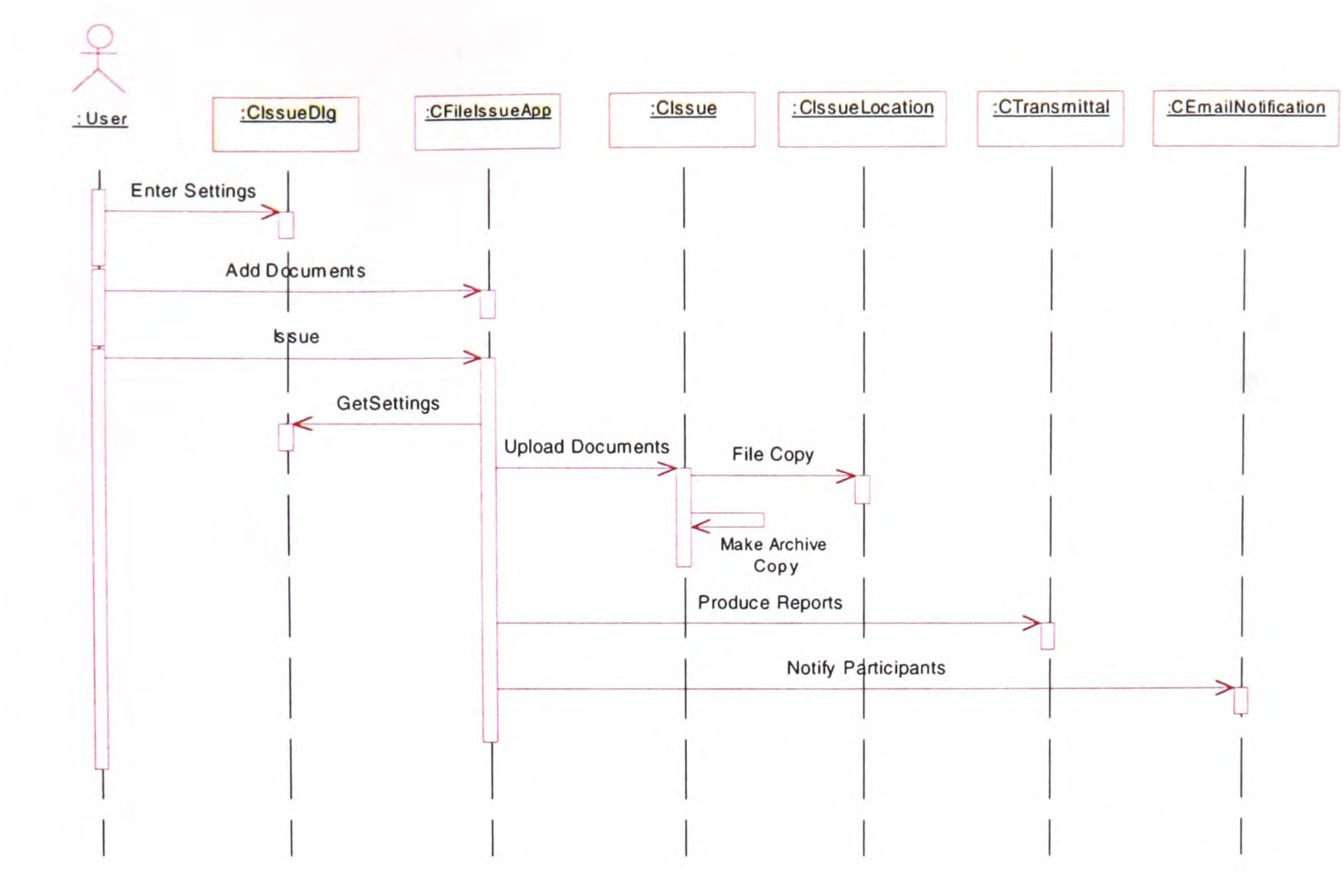

Figure 6.13 Document Publishing Sequence Diagram

6.6.2 Design View

Figure 6.14 shows the class diagram for the Document Publishing module. It is a dialog based MFC application, which has a CIssue class that is responsible for the whole publishing process. The way in which users select documents to issue, is through the IFileLocader interface, which ties the application to the Columbus Navigator. This means that the tool can not be used without Columbus.

The file issue tool uses IColumbusServices to obtain metadata about each document and a list of reference files, as these will also be published alongside the main documents. The protocol used to access the location where documents are issued to is encapsulated in one of the classes derived from CIssueLocation and currently can be FTP, HTTP or the file-system protocol. Reports are produced by the abstract base class CTransmittal, which is implemented in the different transmittal generation derived classes: CXmlReport, CMsWordReport, CTextReport and CCustromReport; this last one can interact with an external IDocumentReport module or custom plug-in. Notification is implemented in CEmailNotication either using Microsoft Outlook in CMsOutlook or directly using MAPI in CMapi.

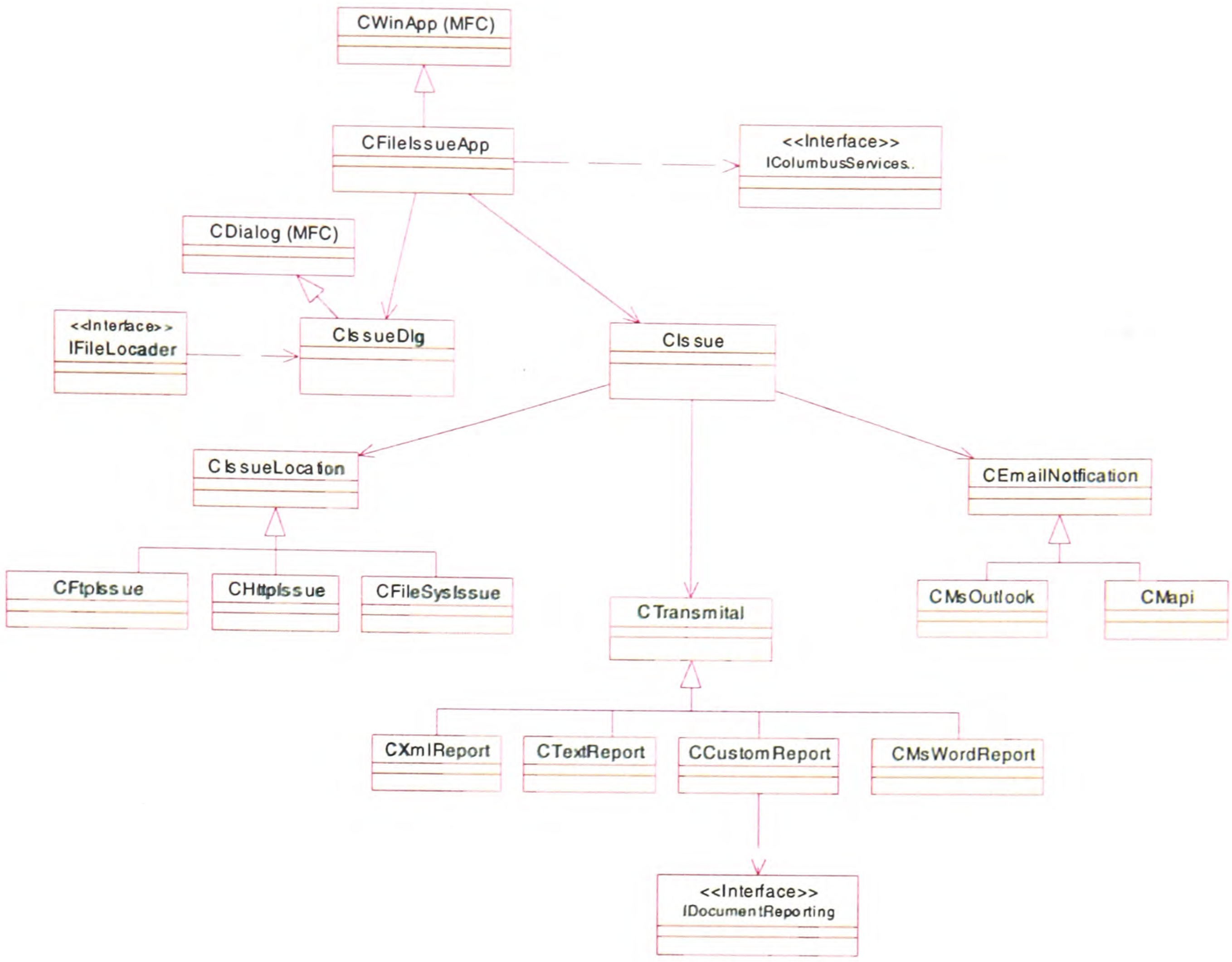

Figure 6.14 Document Publishing Class Diagram

6.7 Document Acquisition and Creation

This module is responsible for acquiring documents from external sources, creating new documents and setting up project areas.

6.7.1 Use Case View

The use case diagram in Figure 6.15 shows the three main activities that this component is responsible for: project setup, document creation and acquiring documents from scanning devices.

Even though the diagram may be quite simple, there is a great dependency on a number of Columbus services. These include file system access, plug-in extensions and file-naming.

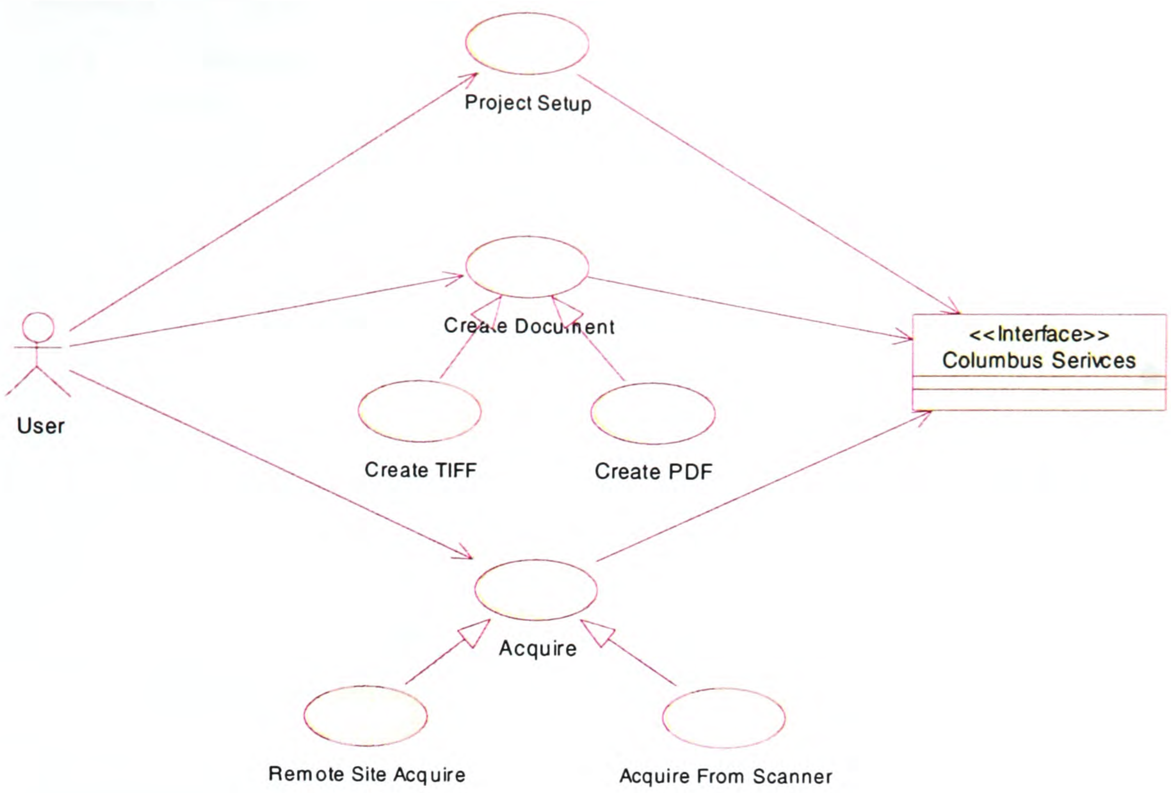

Figure 6.15 Document Creation and Acquisition Use Case Diagram

The user can acquire documents from a remote location such as a project hosting web site, ftp location, import from other media or acquire it from paper using a scanner. The project setup activity is quite straightforward and involves a simple wizard that guides the user towards creating a folder structure at a new project location. Document creation is similar to document acquisition except that documents are copies of a template rather than obtained from a scanning device. To simplify this report, only the sequence diagram and design of the scanning document acquisition sub-component is presented.

As seen in the sequence diagram in Figure 6.16, the process of gathering the scan data and saving it to a file involves a number of high level objects.

Acquiring a document from a scanner is coordinated by the CCbsAcquireApp object. After obtaining any required settings from the user interface dialog (CCbsDlg), it communicates with the device using the standard TWAIN protocol, which returns the information as a stream of raster data. This is then saved to a file by the CSaveDocument object. The name of the document is established by the CFileName object, which makes use of Columbus services to invoke an automatic file-naming wizard, custom naming plug-in or just a standard file browser.

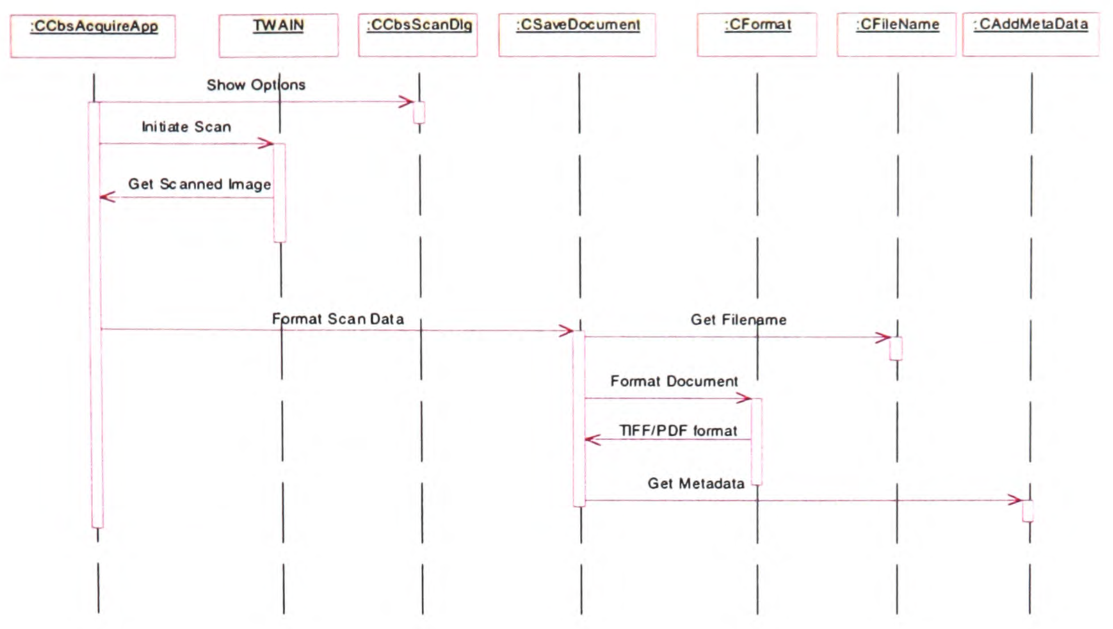

Figure 6.16 Document Acquisition Sequence Diagram

6.7.2 Design View

The module is designed as a simple dialog based MFC application as seen in Figure 6.17. The CCbsAcquireApp is a CWinApp (MFC) has a CCbsScanDlg and uses the TWAIN interface to scan the document and CSaveDocument to save the file.

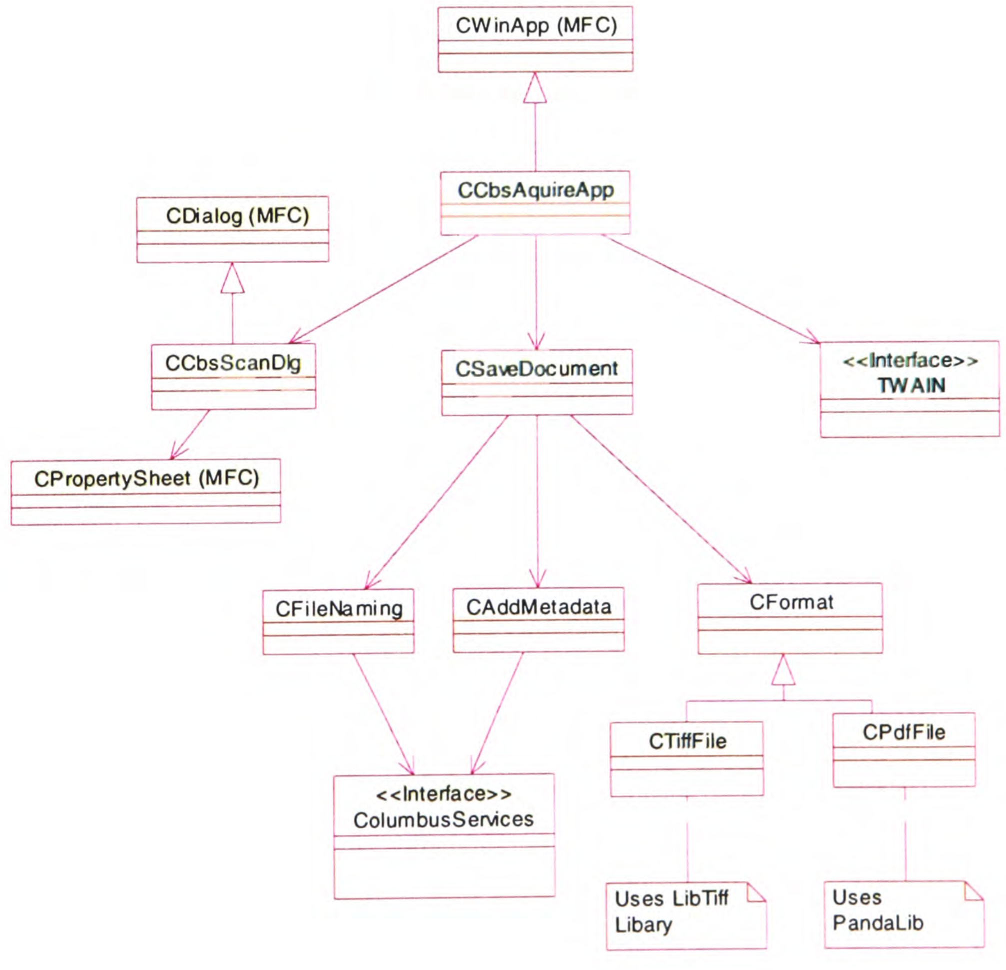

Figure 6.17 Document Acquisition Class Diagram

CCbsScanDlg is an MFC CDialog which contains a number of CPropertySheet (MFC) objects which display a preview of each page that is scanned in.
The format that the data is saved is defined by CFormat object. This is an abstract base class which is implemented in CPdfFile and CTiffFile. CPdfFile produces an Adobe PDF document using a library known as PandaPDF and CTiffFile saves the document as a multi-page TIFF document using the well know LibTiff library.

6.8 Document Reporting

This module is responsible for producing reports on the status of documents. It comprises the history and document details reporting tools.

6.8.1 Use Case View

The use case diagram in Figure 6.18 shows how the user can invoke both tools. The history report is presented directly to the user and saved in a file, whilst the document details report is only written to disk. As both tools are quite similar, only the document details report will be discussed further.

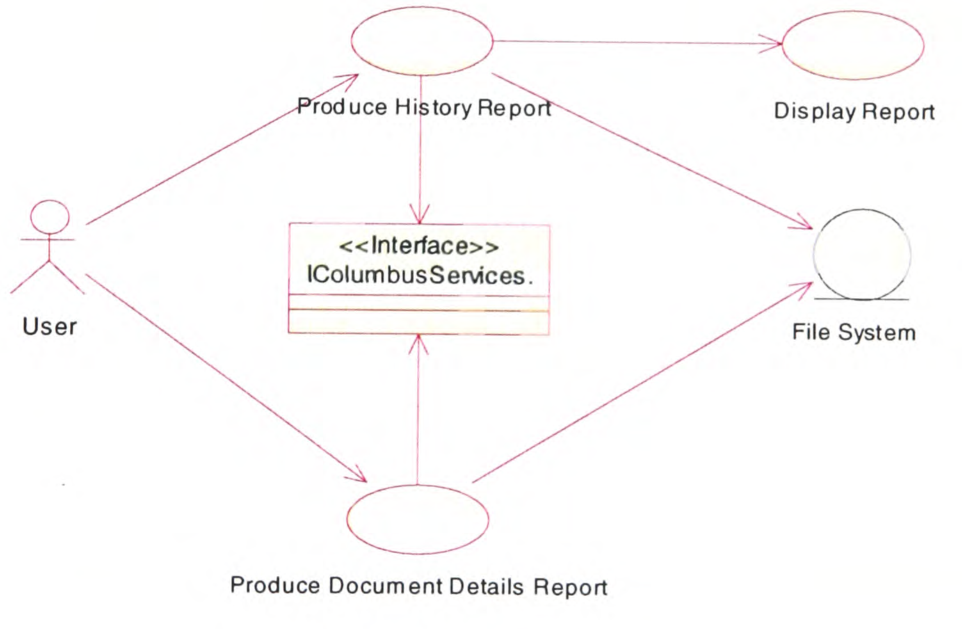

Figure 6.18 Document Reporting Use Case Diagram

The sequence diagram for the document report shows how the application class CDocumentReportApp obtains any settings from the main dialog. It then uses CReadMetadata to get the document details through the Columbus metadata service. Finally the report is generated in one of the supported formats.

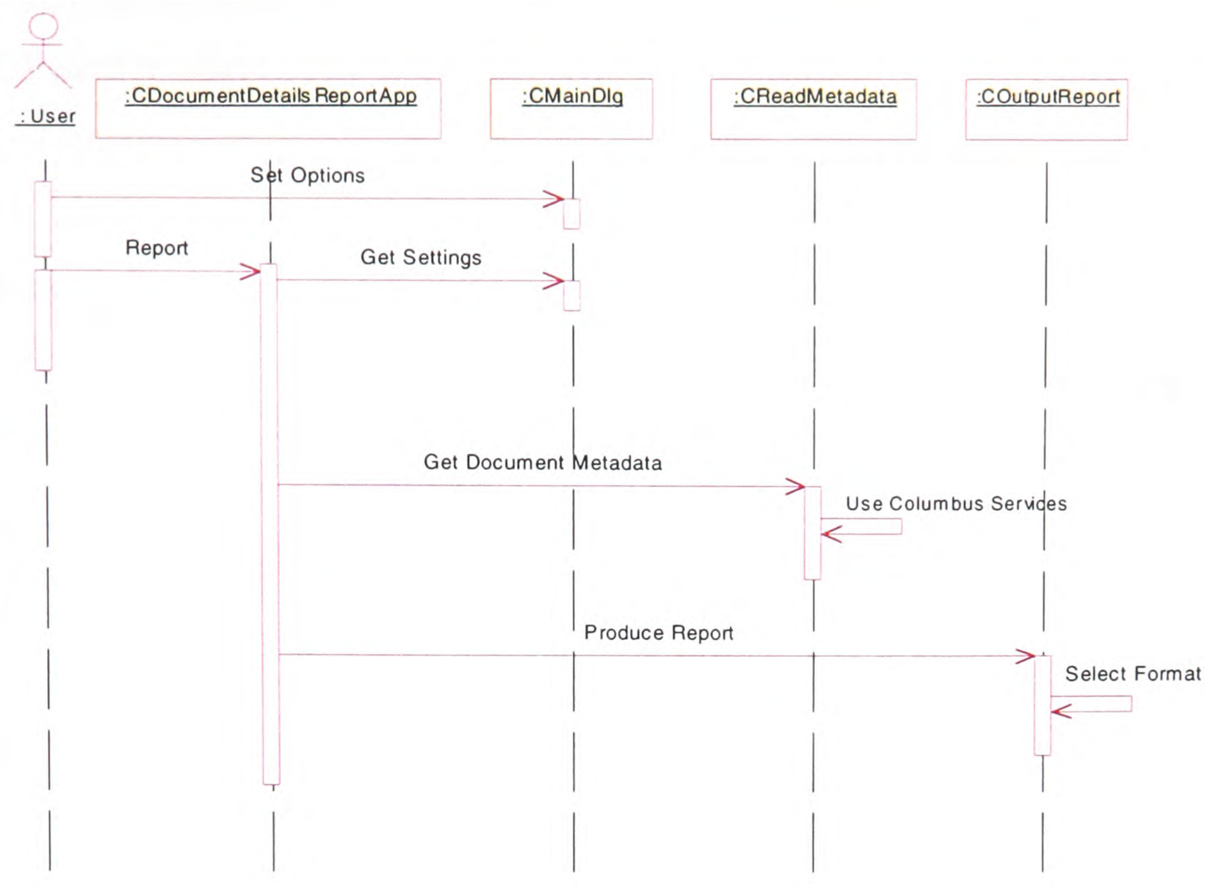

Figure 6.19 Document Details Reporting Sequence Diagram

6.8.2 Design View

The document details report tool is a simple MFC dialog based application as seen in Figure 6.20, which is implemented in CDocumentDetailsReportApp.

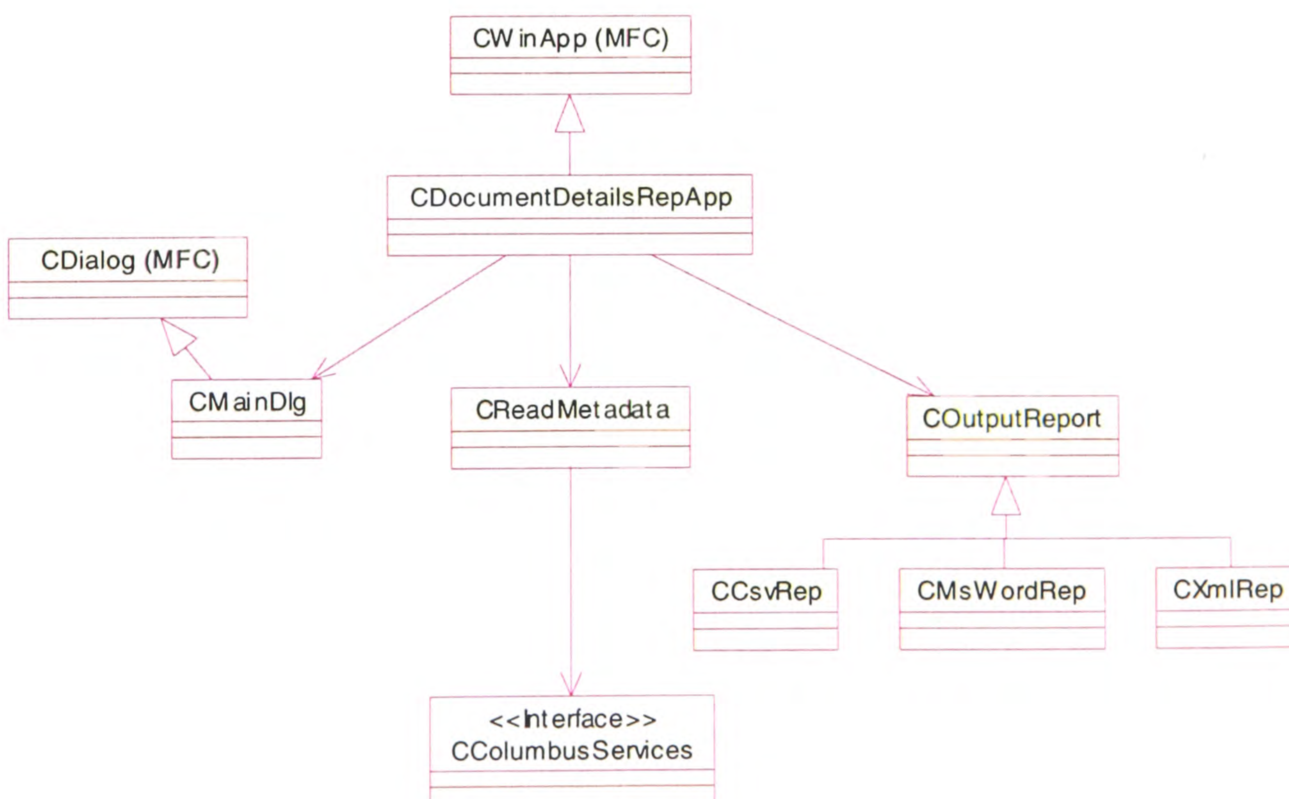

Figure 6.20 Document Details Reporting Class Diagram

The application interacts with the user in CMainDlg, which is an MFC CDialog. It uses a CReadMetadata class to gather metadata about each document, which relies on CColumbusServices as provided by the rest of the application. The report output is produced by COutputReport. This is an abstract base class which is implemented in one of three different classes, giving it the ability to output information in a simple comma delimited format using CCsvRep, XML in CXmlRep or into a Microsoft Word table using the CMsWordRep class.

6.9 Document Activities

This package is responsible for the activities that are carried out with documents. Typically, they are initiated by the Navigator package with a right button mouse click and a menu is presented to the user where a selection can be made. Examples of document activities include opening, printing, copying, moving and deleting documents or starting a number external tools.

6.9.1 Use Case View

The use case diagram presented in Figure 6.21 shows in a simple way how a command is initiated and executed.

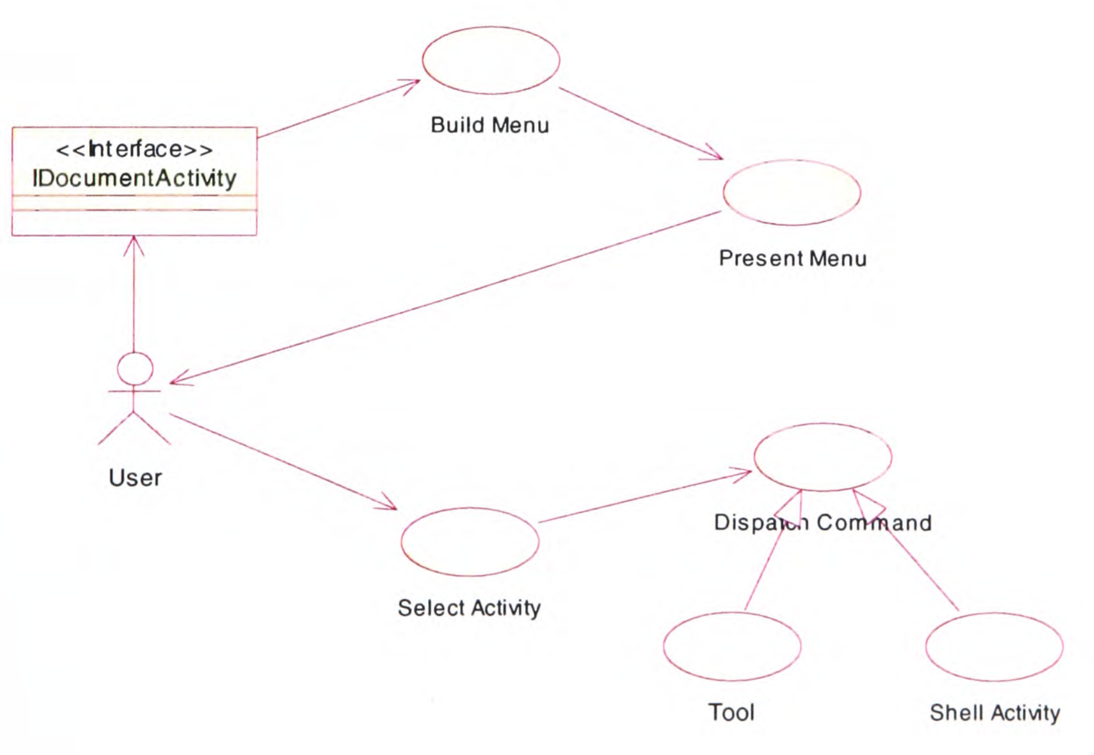

Figure 6.21 Document Activities Use Case Diagram

As the sequence diagram in Figure 6.22 shows, the user starts an activity in the navigator by using the IDocumentActivity interface to tell the component to build a menu and present it to the user. When the user selects an activity, it is dispatched to the appropriate call-back handler. These handlers can either be shell activity handlers that work on selected documents or general tool handlers that are not document specific.

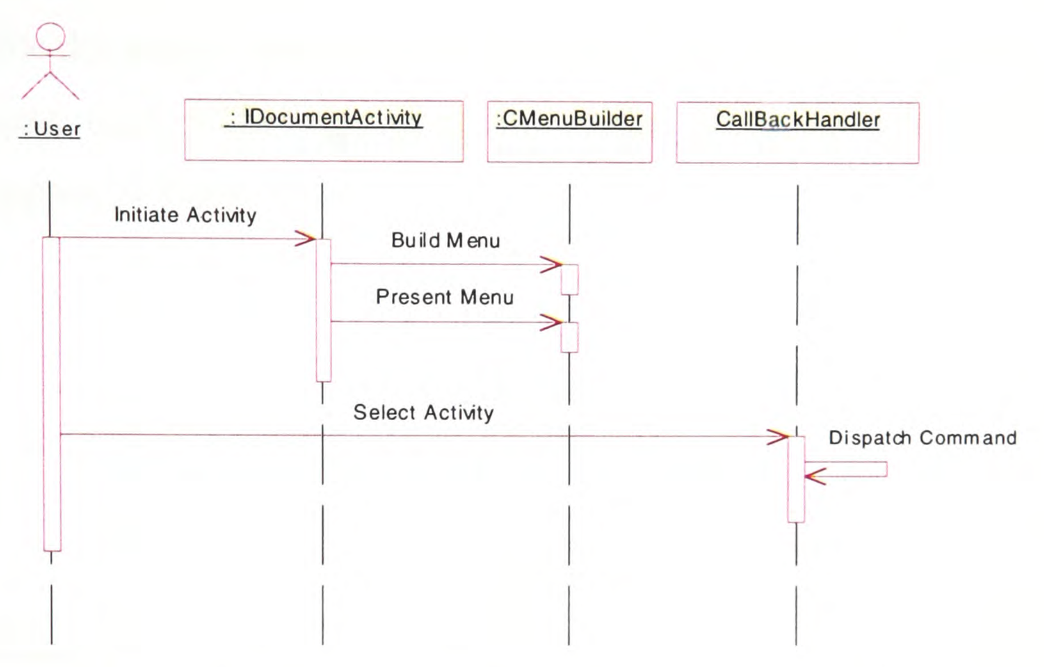

Figure 6.22 Document Activities Sequence Diagram

6.9.2 Design View

The design of this module is quite simple. As seen in Figure 6.23, the only particularly noteworthy item is the CCallbackHandler object, which can be implemented as a CShellActivity or CToolsHandler depending on what action has been requested by the user.

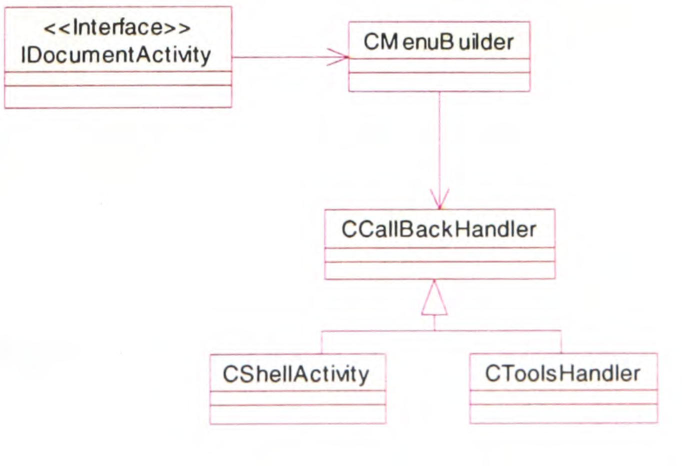

Figure 6.23 Document Activities Class Diagram

6.10 Implementation View

Having presented the use case and design views for each module individually, this section now gives an overall implementation view of the whole application. It describes the physical files that make up each of the modules and how they are referenced by each other.

Despite the fact that the implementation view could go into great detail and show each C++ source code file that makes up each module, this was considered to be too detailed for this report. Instead, since the application is architecturally made up of COM objects, each of these are presented as the individual components shown within the diagram in Figure 6.24.

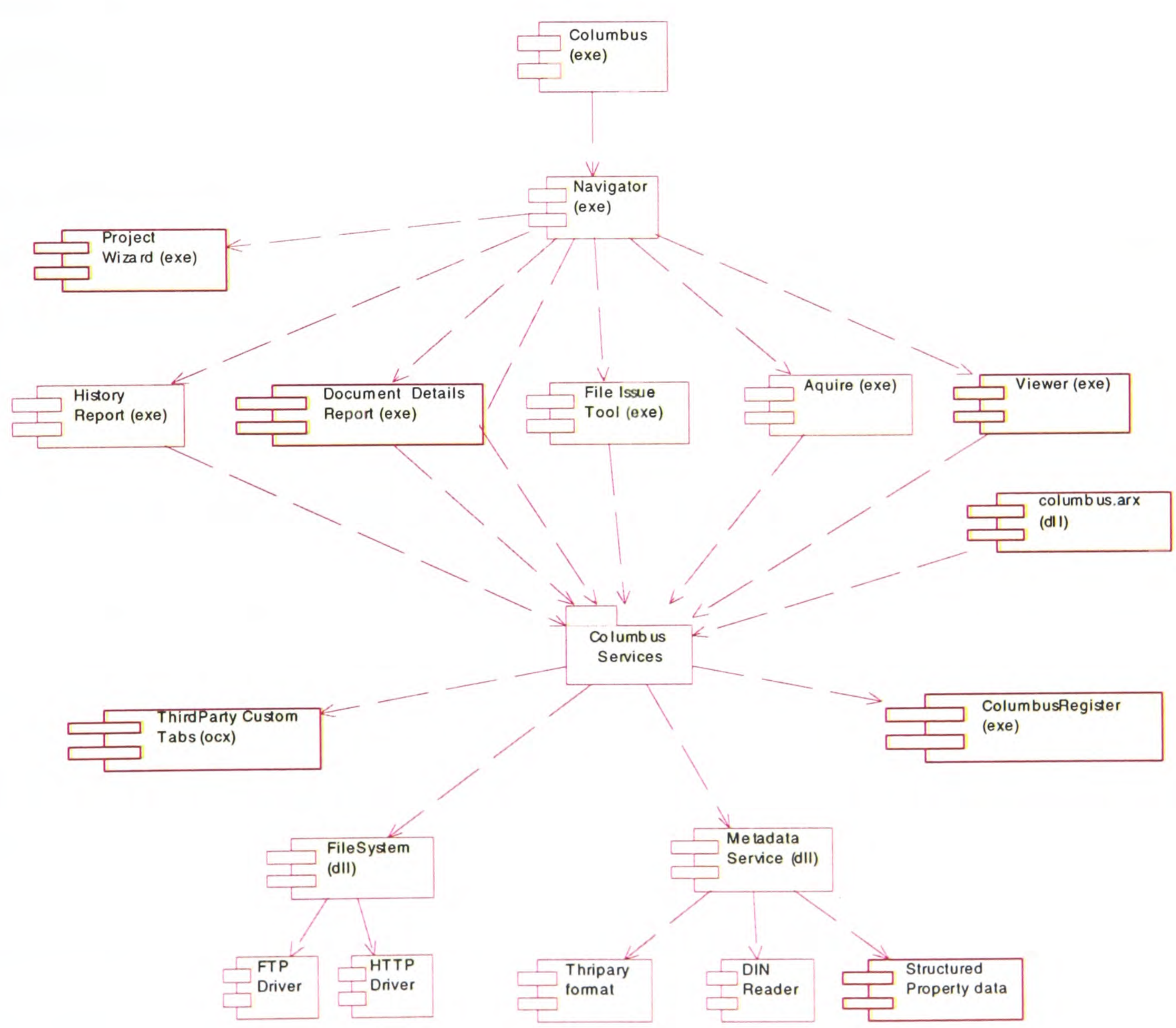

Figure 6.24 Columbus Component Diagram

The user entry point to Columbus is Columbus.exe, which is a stub loader application that starts the navigator module. The navigator is the main user interface package and is responsible for invoking all the others.

All user packages have a direct correspondence to specific components and are implemented as out off process EXE modules using COM; these include: project wizard, history report, document details report, document acquisition and the viewer module. Columbus.arx, which is an AutoCAD application, is shown as a separate component as it does not interact with the navigator.

All modules make direct use of Columbus services, which were described in section 6.3.2. To simplify the diagram, these are shown as a single Columbus Services package. Each service is generally implemented as a DLL or OCX component and is loaded in-process by each package as required. The services include add-ons such as third party tab customisations, file system extensions, metadata services and Columbus registration. This last one is actually implemented as an out of process component, which is solely responsible for maintaining the registry updated.

The diagram shows how the file system component can make use of different drivers, in this case FTP or HTTP to provide the required service. Equally, the metadata service uses the DIN reader, which can access external metadata in DIN or XML formats, internal metadata stored as Structured Property Data and custom formats using a third party format driver.

6.11 Deployment View

The final view of the system that is presented, it the deployment view. This describes how the system interacts with other processes that are external to the application. The deployment diagram, seen in Figure 6.25, is quite simple as there are only a handful of these processes.

From a document management perspective, Columbus interacts with an internal file system, which it uses to hold both the documents themselves and the metadata about them. The application carries out a number of activities on the file system such as cataloguing, searching for information and editing documents.

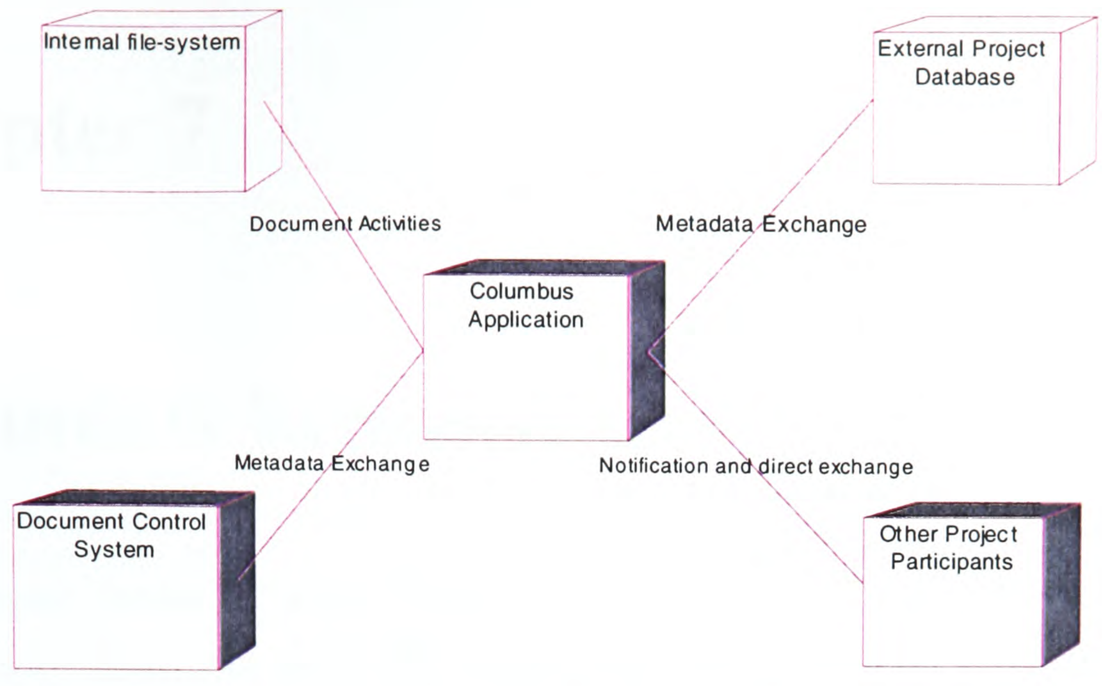

Figure 6.25 Columbus Deployment Diagram

Columbus' project hosting capabilities allow it to cooperate directly with external project databases. This could be through a project hosting service or directly on an FTP site. The application is able to upload and download documents and metadata which can be imported into other systems if required.

As external project databases are typically used to share information with other project participants, Columbus can directly notify them of any document transactions. Alternatively, information can be sent directly to other participants without placing it in a shared repository.

As part of a unified solution, Columbus is also able to keep records of transactions and detailed history and archival logs by keeping its own records, producing reports or linking to Document Control Systems using metadata exchange.

Chapter 7

Columbus Released

This chapter looks at how Columbus was released, first within Arup and subsequently across the rest of the construction industry and the impact that it has had. A description is presented of how a community of users has been built by providing a number of direct support mechanisms. This is followed by a review of how the use of various Internet download sites and forming alliances with external partners has helped to make Columbus a widely known application. Also, by looking at the method used to register the product's usage, the level of uptake of the software is also described. The industry feedback that Columbus has received is then presented by looking at journal reviews, user comments and the awards that it has won. An analysis of its usage within the construction industry and beyond is then covered with a series of case studies. Then, the side effect of how Columbus has been used to promote the sale of other products and services is discussed and finally there is a review of the alliances formed with software houses and academic institutions.

It is important to reiterate that, as described in Chapter 1, though I was responsible for the research, design and programming of the Columbus product, most of the web site development, documentation, promotion and marketing were done by other team members, even though I had a major input to these processes.

7.1 Release History

7.1.1 Arup Releases

As described in chapter 3, soon after work on the development of the Columbus project began, various versions of the software were made available to interested Arup users. The feedback received was very good and in 1996, this provided a justification for Arup to continue funding the research. Within a few months, a preliminary version was ready and beta testing began. The feedback was once again very positive and we were encouraged to make the product widely available across the firm. That release, labelled as version 1.0, was purely aimed at providing drawing management facilities for CAD operators and lacked many of the features that are at present available in the product. The only metadata that was available was for AutoCAD drawings and was generated by a module in OvaCAD, the in-house customisation of AutoCAD. It had no built in viewing engines, but the AutoCAD preview thumbnail image was there. At this stage, Columbus was solely distributed as part of the OvaCAD suite.

The feedback from this release was very encouraging and a lot of requests for new features were received. Work started on the file-issuing module to be able to link to FTP sites and support for more CAD facilities were added. At the same time, there was a growing interest from users outside of the CAD field to make Columbus a more general tool, so that it could be used as a general document management system. As a result of this, support for metadata for other document types was added and a viewing engine was licensed from Stellant (2002).

Due to the popularity of Columbus and extent to which it was being used within Arup, in 1998 it became a corporate application. This meant that it would be centrally funded by the firm as a whole, rather than solely by the internal CAD user's committee. As the usage of Columbus grew within the firm, so did the interest from other companies in the construction industry that worked alongside us. Other project participants saw the ease with which Arup users organised their documents and were able to upload them to shared FTP sites with useful metadata. Numerous requests from them to use or buy the software were then received.

7.1.2 Freely Available

As interest grew from external organisations that we worked with, it was decided that a version of Columbus would be made available to the rest of the industry. The obvious way to do this would have been to produce a commercial version and sell it, but this had its drawbacks. Despite being well known as leading consulting engineers, we were not a top player in the competitive document management market, and it would be difficult to generate enough sales to even cover the marketing costs.

A powerful Columbus feature that was identified early on was its ability to share document metadata between project participants. This represented a big saving in time and manpower by not having to re-enter information into our document control systems. However, information received from external users still had to be typed in from paper transmittal slips, and the statement: "If only they had Columbus" was often made. Because of this factor, the team leader and I took the view that we should convince the firm's board to allow us to place the software in the public domain. Information exchange within the industry would improve and Arup would benefit from the usage of standard metadata when sending and receiving electronic documents to and from other project participants.

One other important consideration was that the primary costs of developing Columbus had already been paid for, and would still be paid for even if it were solely developed for internal use. This is very different to other software developments, where the expected revenue from software sales must justify the development cost.

A secondary factor that was also presented to the board was the fact that our group intended to grow as a software house and sell other products and services. The publicity and user base that Columbus would generate would not only benefit Arup as whole, but also enhance sales of our other products.

After careful consideration the board agreed to our request and a version of Columbus for external release was prepared.

7.1.3 External Releases

By mid 1999, pre-release external versions of Columbus had been made available to other organisations, reviews had appeared in construction industry journals and interest was building up. In December 1999, the first general public release, version 2.0, was made. It incorporated only minor enhancements, as most of the work involved was to make it non-Arup specific.

The decision was also taken to make Columbus available for download via the Internet on a custom built site. As the installation file was quite large, over 20 megabytes, and at the time many organisations were still using 56Kbit modems to access the web, it was decided to also offer Columbus on CD-ROM, but a charge would be levied to cover the cost of production. With regards to user support, a forum was made available on the web site, a Columbus support email account was created and a premium rate telephone line was set-up for those who required immediate help. When Columbus was released, it was a great success, with over one thousand companies registering it within the first month. Subsequent releases of Columbus have taken it up to version 2.5, released in September 2002. Many enhancements have been made to the product since the first version was released, making it now more popular than ever.

7.2 User Community

7.2.1 Web Site

In order to promote the usage of Columbus, make it available for download to the rest of the community and provide a contact point for support, a purpose-built web site was set-up at *http://columbus.arup.com.* After various incarnations, the site currently appears as shown in Figure 7.1. An outline of the main facilities that the site provides can be described as follows:

- News: Information on the latest releases and a constantly updated count of uniquely registered companies and their countries of origin is presented.
- Promotional Material: Reasons for using Columbus, access to a product brochure, who are we are and general information on the usage of document management.
- Technical Material: Frequently asked questions, software and hardware requirements, Teach Yourself Columbus and Advanced Configuration training manuals.
- Download Area: Columbus application software, updates, hot-fixes, old versions of Columbus, third party utilities and product documentation.
- Support: A number of public forums are available, where users can ask questions about Columbus or other software. In addition, an email support address is offered (*columbus.support@arup.com*) and links to registered Columbus training partners.

- Other Products: Links to sub-sites where some of our other software is available.

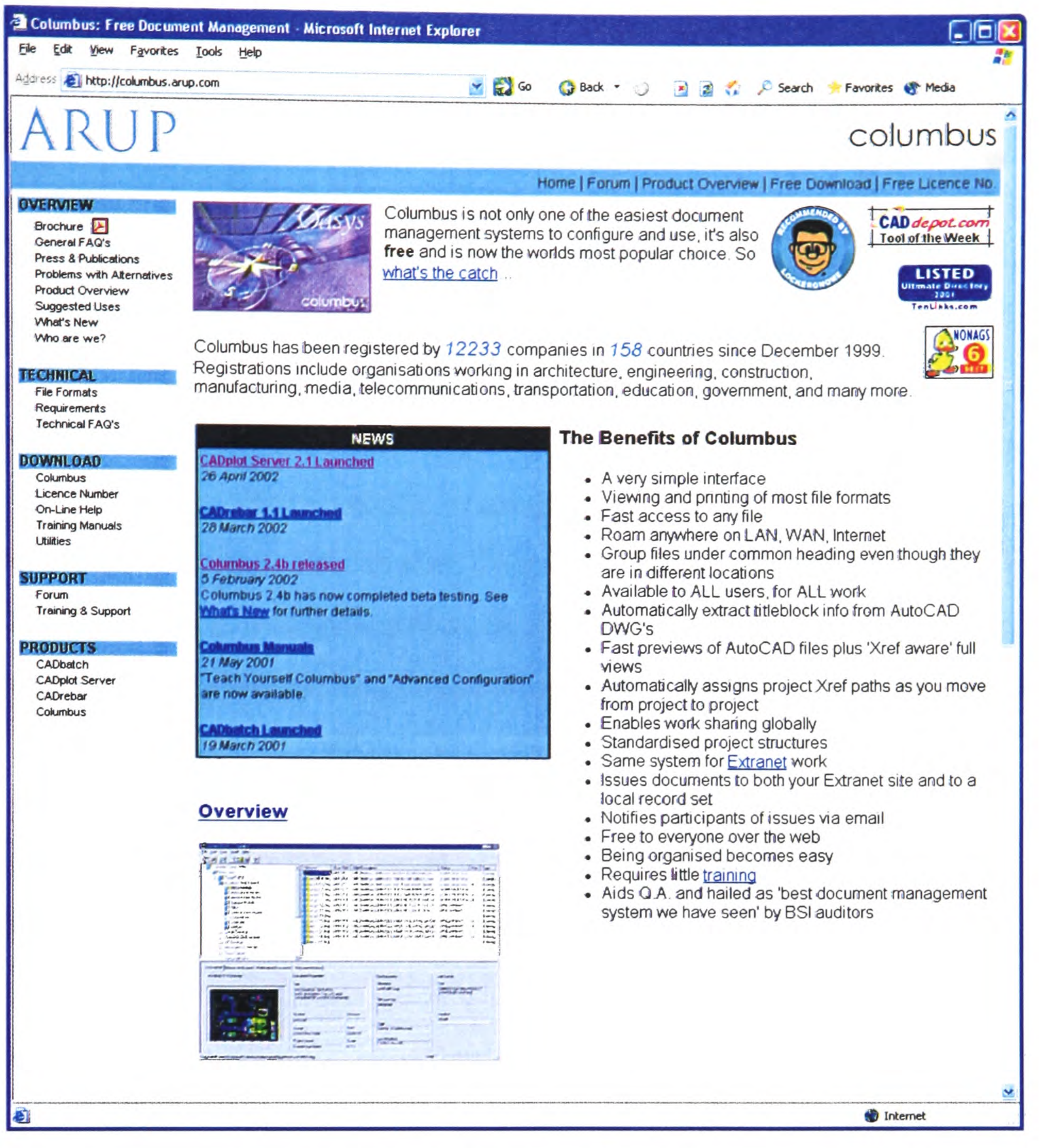

Figure 7.1 Columbus Web Site

7.2.2 Company Registrations

When the decision to publicly release Columbus was made, it was realised that it was important to keep track of how many organisations were using it and where. In order to do this, we made it compulsory for a form to be filled in before being able to download the software. Being well aware of how tiresome some of these forms could be, we decided to ask only two compulsory questions: an email address and a country. Upon completing the form, a licence number would be emailed to the user.

A very important point to consider is that Columbus has been made an extremely easy application to deploy across an organisation. Because of this, with only one download, one registration and one installation on a file server, a whole organisation could be using Columbus. This meant that we would not be able to count how many users we had, but only how many companies had registered the product.

Within a few months, Columbus had become much more popular than we expected and we decided that it could be offered for download from other sites or included on CD-ROM by other parties to reduce the load on our servers and further promote the product. However, if this alone had just been done, we would not be registering Columbus' full usage. In order to overcome this, it was decided to make Columbus downloadable without registration, but to request a serial number when first used. This is now done with a "nag screen", as seen in Figure 7.2. This is a dialog box that cannot be dismissed until an ever-increasing time delay has elapsed. Though the product is fully functioning without registration, it is an irritant if constantly used. One major advantage of this is that only those organisations that use Columbus actively will request a licence number and the "company registration total count" will not include those that that have downloaded it solely for evaluation purposes.

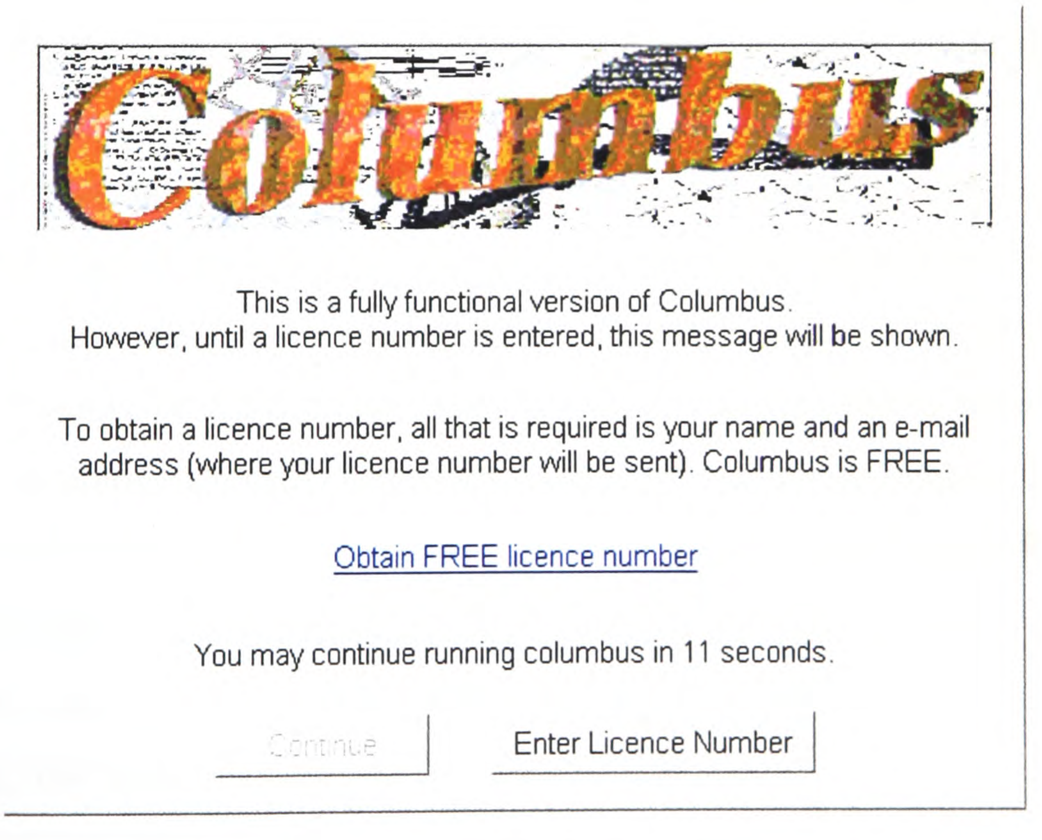

Figure 7.2 Columbus "Nag screen"

As Figure 7.3 shows, over the past two years, registration growth has been almost exponential. As of September 2002, the total unique recorded company registration stands at over 16000 from 165 countries.

Currently, Columbus is self-promoting; numerous postings have appeared on Internet newsgroups and it seems to spread by "word of mouth". One other factor that can results in a daily peak for a particular country is the appearance of a review in a local journal.

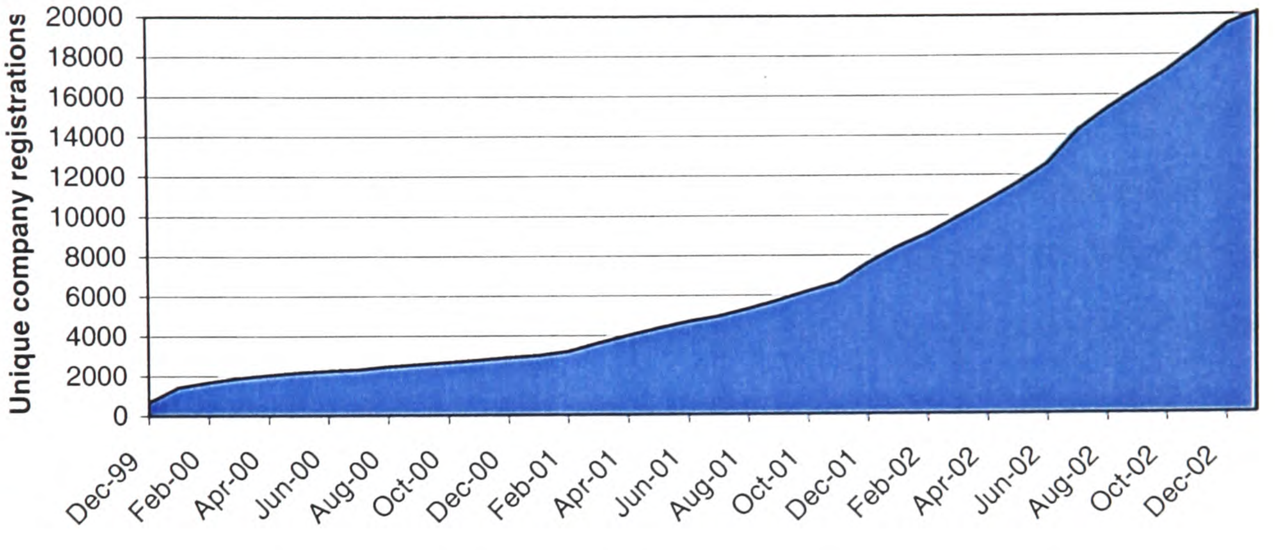

Figure 7.3 Total Columbus Unique Company Registrations

7.2.3 Distribution

Columbus is distributed in a number ways; most people obtain it over the Internet, but it can also be obtained on CD-ROM.

Download: Most users get Columbus by directly downloading it from our server. The majority visit our web site, but others come re-directed from a number of cataloguing sites that exist. Columbus is currently listed in over thirty sites, including Nonags, Winfiles, Softseek and Zdnet, all of which have helped to promote Columbus. At some of the sites, copies of the downloadable file are kept, but due to its large size, most just link directly to our server. Having Columbus listed on so many sites has made it a much more international product and taken it beyond the construction industry, for which it was originally intended.

CD-ROM: The size of Columbus has been a problem for a number of individuals or organisations that have slow links to the Internet. Specifically for them, we offered during the first year of Columbus' release a version on CD-ROM.

However, to cover production costs a charge was levied, but this was still not enough to make it feasible and CD-ROM releases were discontinued. Despite the fact that Columbus cannot normally be obtained on CD-ROM from us now, we do occasionally still produce them for handing out at conferences, trade shows and special events. More interestingly though, is the fact that we are quite often approached by organisations seeking permission to distribute Columbus on their own CD-ROMS. The current licence agreement states that Columbus can be freely passed on to others if it is not done for profit, which is the case in most situations. However, when it is distributed for commercial gain, such as to enhance a commercial software suite, it is necessary to sign a separate agreement with us. One of the most successful distributions of Columbus is on the CADopia IntelliCAD CD-ROM (CADopia 2002). IntelliCAD is a very popular alternative to AutoCAD that uses the same file format. As part of the agreement, we receive regular royalty payments, which help to offset the cost of developing Columbus. Detailed information on how CADopia use Columbus can be found on their web site and has also been described by Harrod (2002) in a detailed review.

7.2.4 User Support

As with any software product, user support is very important. Even when Columbus is considered relatively easy to use, there are undoubtedly situations that arise where assistance is required. In order to address this, a number of support mechanisms were put in place:

- Forum: As already mentioned, one of support options that we have made available is a forum on the web site, which is categorised into various topic areas. As no income is received from the forum, a quick response cannot be guaranteed. Nevertheless, we always try to provide an answer to all queries and on many occasions it acts as a self-help group where users respond to each other's questions. The forum has been very useful in identifying bugs that are not easy to detect. One example where is has been particularly effective, has been in identifying incompatibilities between different language versions of operating systems, thanks to the international user base.

- Support Line: Telephone support has always been provided to internal users of any of our software. This is typically part of the service that the firm expects to receive from us. So, when Columbus was released externally, a telephone support line was set-up for external users. Despite being charged at a premium rate, it did not prove profitable and was discontinued.
- Training Courses and Set-up sessions: Since the first release of Columbus, we have offered training courses to users. These are typically aimed at administrators and include sessions on how to configure Columbus on projects, good practice on filing and document exchange. On many occasions, a full day on-site training course will also include setting up the product so that the whole office can be up and running with Columbus.
- Training and Support Partners: As the popularity of Columbus increased, we found that our resources were not sufficient to provide the required training facilities, so the decision was taken to have external training and support partners. They do not receive any income from us, but directly from the user, and typically also use Columbus as a mechanism to further promote their own dealership activities. In order to provide training, they are required to attend one of our dealer courses and are included on the list of Columbus Training and Support Partners on our web site. The scheme is also intended to enhance the level of support that we provide overseas.
- Training Manuals: As part of the training courses, two manuals entitled Teach Yourself Columbus and Advanced Configuration were produced and distributed to attendees. These were also available for purchase directly from the web site and were particularly popular with overseas locations where local training was not available. The manuals are no longer being sold, as the activity was not profitable and can now be freely downloaded from the web site in Adobe PDF format.
- Seminars: As far as Columbus promotion activities are concerned, a series of one-day seminars are occasionally presented to users. The response has been very positive and is seen by users as a way of talking directly to the developers. The seminars have highlighted a number of user concerns and requests for new features. To some, they are a way of learning about

Columbus for the first time, whilst other more advanced users are interested in learning about forthcoming features.

- Maintenance: A more recent support mechanism that is now being introduced is the maintenance contract. Though the software is free, some organisations feel uncomfortable using the product unless a mechanism is in place to ensure that they receive specific help. The maintenance agreement provides the user with unlimited support for Columbus on a private forum, via email or telephone and also includes free licences of some of our commercial software.

7.3 Industry Feedback

Since the release of Columbus, feedback from the industry has been very positive. This has not only been in the form of direct comments, emails and phone calls from other organisations, but from reviews and articles published in industry journals and magazines. In addition, it has also been very encouraging that Columbus has won a number of awards.

- Publications: Even before Columbus was publicly released, it had been featured in a number of industry journals. Appendix A shows some quotes from what has been published. Some articles have been product reviews whilst others have described the way in which Columbus has helped companies organise themselves better and improve collaboration.
- User Statements: Many users have privately commented on Columbus at presentations, seminars and meetings or just in telephone calls. But, many others have made public statements that have helped promote the product and encourage others to use it. Appendix B shows some of the user feedback received from all over the world. In particular, it was very satisfying to know that Sir John Egan, author of "Rethinking Construction" (Egan 1998), had some very encouraging words to say.
- Awards: In July 2000, Columbus was awarded a "Special Award" by the London Association of the Institution of Civil Engineers, as described in Appendix C. This award provided a degree of recognition within the industry and has been a great source of pride. As a result, there was a further boost in downloads and registrations. In addition, Columbus has

received a number of Computing awards. It was named "CAD tool of the week" by CADdepot (2001), was awarded 6 out of 6 by Nonags (2001) and was listed in Tenlink's (2001) "Ultimate CAD Directory 2001".

- Press Releases: There have also been numerous press releases about Columbus. These have typically been announcements of new releases or updates from us. However, some of them have been made by other parties, as was the case with the announcement by CADopia of the inclusion of Columbus in their IntelliCAD release. This is described in detail in Appendix A.

7.4 Industry Case Studies

When analysing the benefits that Columbus has provided to the construction industry, it would be easy to review the use of the product with Arup, but that would not show that it has had an impact externally. The best way of reviewing its success is to look at some independent case studies.

7.4.1 Shepherd Robson

Shepherd Robson are a UK based architectural firm that use their own "in-house" Document Control System called Drawing Manager. Before learning about Columbus, they were about to develop or purchase a document management solution. As Arup had previously experienced, they also found that the job of entering incoming document's metadata was a time consuming task and believed that Columbus would help. For example, four technical clerks were employed at one of their offices on a full time basis to enter the title block details of the multitude of drawings that were received every day. After a short evaluation, they decided to use Columbus and to try and integrate it with their own system.

Shepherd Robson have developed a customisation that includes the ability to import and export document metadata in Columbus format and has the ability to interact with Columbus to view the files and data being issued. To export information, their system pre-issues the documents to a folder and creates Columbus metadata. Columbus is subsequently used to validate, package and upload the information to a project hosting service.

Their view is that the metadata format used by Columbus is adequate, but it would be ideal to have an industry standard schema. None exist yet for this type of document exchange and it would be good if Columbus scould set a de-facto standard. Their main developer likes Columbus because it is a simple application of metadata exchange rather than a monolithic database. He believes that document databases should be provided at each organisational end of the communication link, with Columbus acting as the common glue. Currently they use the application to export documents from their system, to import documents from third parties and for internal document management.

Shepherd Robson have actively promoted the use of Columbus amongst their project partners. For example, construction management firm Mace have also produced a tool for importing Columbus format metadata information into their own document management system and contractor Sir Robert McAlpine has purchased Shepherd Robson's Drawing Manager as has been stated by them, it is a requirement of the Birmingham Bull Ring Redevelopment project that information is exchanged in Columbus metadata format.

What is most interesting about this case is that these software developments were all done quite independently from Arup. Now, however, as Shepherd Robson and Arup team up to work on the "Fitzrovia Redevelopment" project, the true benefit of integrating common systems for data exchange is of immediate benefit for Arup too.

7.4.2 Building Research Establishment

In 1999, Columbus and the use of Document Management within Arup was the subject of the Building Research Establishment and Construction Industry Computing Association case study (CICA 2000) in which the development and use of Columbus was investigated.

The study concluded that Columbus was being used effectively by project teams for internal document management and for working between different offices. Staff in multiple offices can now collaborate on larger projects and Virtual Project Teams are a reality, with Columbus allowing flexible, comprehensive and efficient multi-user access to all document data. The example of how the Manchester Stadium project was set-up and managed as a single common model was quoted as

a success story. The way in which Arup has used Columbus to achieve ISO 9000 quality assurance accreditation was also mentioned as a major benefit. Using the product to organise the filing, exchange and production of documents was recognised by a British Standards Institute auditor as the best tool he had seen.
The report also attributed Columbus' success to the fact that the development team were able to consult users, establish their requirements and match them to the available technology.

7.4.3 Chiswick Park Project

The Chiswick Park project is a very good example of where Columbus metadata has been used to integrate project participants. Construction management company Mace in agreement with the client, property developer Stanhope, imposed a contractual requirement to exchange document information in Columbus metadata format. The Chiswick Park project brought together main contractor Bovis Lend Lease, Architect Richard Rogers, Consulting Engineer Arup and numerous other project participants. One of Mace's requirements was to ensure that all project information exchange was appropriately coordinated and chose Columbus as the way to achieve it. They did not force the use of Columbus, only the metadata format. Indeed, they themselves are not big users of Columbus as they are not information producers. Mace maintained a central server where, in accordance with strict guidelines, participants would upload their information. Appendix D shows an extract of the guidelines of how Columbus was used to achieve this. As part of the process, Mace developed a database application to run on the server and process information based on the metadata. In fact, it was rather overzealous, removing any uploaded information if the metadata was not in the correct format. The process handled the transfer of information for comment, information, construction and documents for use by other parties. It allowed the transfer of source design files or publishable data. The guidelines clearly make a distinction between the processes for information transfer between different project participant categories (i.e. consultants, contractors and trade contractors), handling the different needs of each group.

7.4.4 Swanke Hayden Connell

Swanke Hayden Connell (SHC 2002) is a UK and US based firm of architects. SHC had evaluated and tried a lot of document management systems, finding that much of them were "oversold", crude and resulted in failures on projects. As a consequence, they then began to develop their own solution, but came across Columbus. After establishing that it addressed most of their needs, they changed direction and rolled-out Columbus across the firm. SHC were one of the early Columbus adopters, and soon became prominent within the Columbus User group. In October 2000, they made a presentation at a Columbus Seminar describing how they had replaced their whole document management and filing system with Columbus. SHC designed their own custom office data structure to use Columbus and now achieve consistent storage across all projects, apply quality management methodology to their work, handle external document control processes and save enormous amounts of employee time when searching for information. In SHC's own words, "Columbus sets a landmark by providing a system relevant to the construction process".

7.4.5 Local Authorities

From the list of companies that have registered Columbus, we are aware that many local authorities are using the product. Portsmouth City Council, for example, have deployed Columbus throughout their planning department and have taken a number of our training courses.

One other local authority that is a major user of Columbus is Glasgow City Council. They use Columbus as an internal document management, external project hosting system and link to a Document Control application.

According to their internal Columbus usage documentation, because of security issues, they use a private leased line to connect three categories of user to a shared repository. These categories cover the Document Controllers, the Appointed Agents and the Project Design Team. Each of these groups has a different responsibility for how information flows and as such has appropriate permissions within the shared repository.

7.4.6 Other Industries

We are also very much aware that Columbus is being used outside of the construction industry. Though it was never intended to go this far, it has proved to be a popular solution in various other environments.

Columbus is being used by one of the world's largest manufacturers of communications equipment and integrated circuits. They are particularly interested in licensing Columbus for use as a run-time document and metadata viewer for distributing and cataloguing application notes.

Columbus is also being used quite actively in the petrochemical industry. One of the world's largest oil corporations is a big user of the application, which they use for design and development. Interestingly, one of their main concerns about the application has been that they did not feel at ease using free software and are considering taking out a maintenance agreement to meet their support requirements.

Still in the petrochemical industry, Moving Parts (2002) are a "system integrator" based in New Orleans, USA, with a mixed client base of offshore oil and gas and onshore manufacturing. They take a combination of off-the-shelf mechanical and electronics hardware and combine them with custom mechanical designs and purpose written software in the development of robotic automation, in particular underwater robots. Moving Parts first became interested in Columbus because of its flexibility and cost. They have customers all over the world and can set up project hosting sites and manage them all within Columbus without resorting to expensive and cumbersome collaboration systems. They also use suppliers that have FTP sites with electronic parts files and can set these up for easy access as well. What they have found in Columbus is a product document management system without the constraints of a typical product document management system. They believe it is fairly easy to convince customers to use software that is freely available and has worked very well so far. Moving Parts use SolidWork's three-dimensional mechanical CAD package to do all their design and have written custom modules to populate the title block information and preview image in Columbus. They are one of the independent third party developers who have made an active contribution to the project and promoted it within their industry.

7.5 Software and Service Promotion

Columbus was never released to make money. Its goals have always been to help Arup improve data exchange and as a consequence, improve the way in which information is exchanged within the construction industry. Nevertheless, as a side effect, it has acted as a way of promoting Oasys, the Arup software house, helping the sales of other software and services that we provide.

7.5.1 Columbus Enhancements Packages

Even when Columbus is a free product, some of the enhancement modules and tools that have been added recently are only available if purchased. Currently, these are the "Enterprise Deployment Tool" and the "Productivity Pack".

- Enterprise Deployment Tool: This tool allows the system administrator to deploy and maintain Columbus across a whole organisation as described in detail in Chapter 5. It includes free facilities such as checking the operating system and hardware configuration of remote machines (e.g. version, total memory, free disk space, etc.). It is not expected to sell in high volume, as only one copy is required per administrator. However, we believe that the free facilities that are included will encourage system administrators to purchase it, as it will show them the potential benefit of the full product. The tool is currently being tested internally and is expect to be available for sale in the last quarter of 2002.
- Productivity Pack: This is a collection of utilities for Columbus, which will be sold as a single package. Included in the pack, amongst other things, will be the scanning module CbsAcquire, a Microsoft Word document creation application and a tool for changing title block information in AutoCAD drawings from Columbus. The pack is currently being tested internally and is expect to be available for sale in the last quarter of 2002.

7.5.2 Other Software

In addition to selling Columbus enhancements packages, one of the other benefits of distributing Columbus freely, is that it promotes the sale of our other software which is unrelated to Columbus. Though ultimately it is expected to promote all the software that Arup sell through Oasys, we are initially focusing on the packages produced within our development team. Amongst these are:

- CADrebar: (Herrero 2002), is an AutoCAD based reinforced concrete detailing package.
- CADbatch: is an AutoCAD add-on that allows repeating a task on a set of drawings.
- CADplotServer: (Conlon 2002), is an application that allows client workstations to connect to a central server to plot AutoCAD drawings.

Other software packages produced by Oasys that will be promoted by Columbus are in the fields of structural, mechanical, electrical and geotechnical engineering.

7.5.3 Promoting IT services

Columbus has also been influential in promoting some of the IT services offered by Arup. At the top of the list, is the joint venture between Arup and Causeway Technologies (Causeway 2002), known as Causeway Collaboration. This product, formerly known as Integration, achieved great success early on, based on the Arup presence and interest from the Columbus user base. In addition, other contracts in the area of email configuration, systems management, network management and special project developments is being awarded to other Arup groups, as a result of the contacts established through the Columbus user community.

7.6 Alliances

We have formed a number of relationships with external organisations as part of the development of Columbus. Some have directly benefited the development of the product, whilst other alliances have improved the perception of Arup within the industry.

7.6.1 Project Partners

As described in some of the case studies in 7.4, Arup has entered into a number of partnerships where Columbus has been established as a way to exchange data. Moreover, the benefits have not only been apparent to other project participants, but clients themselves have also been aware of the product's benefits. In one example, when Arup were bidding for work on the "Heathrow Terminal 5" project the client, British Airports Authority, recognised Columbus as valuable contribution to the industry.

7.6.2 Software Houses

As previously mentioned, we have formed a number of partnerships to allow Columbus to be distributed by other organisations. In addition to this, we have formed alliances with a number of software houses to make Columbus work with their applications.

Some of the developments have been direct enhancements to Columbus, whilst others are provided within the other party's software. For example, we are working closely with a major software developer in the Civil Engineering field. We intend to include Columbus with distributions of their software and provide specific support for their file formats in metadata view tabs. We are also holding discussions with Felix CAT (Felix 2002), producers of FelixCAD and CeCAD, which are popular mobile and desktop CAD applications. They share our view that there should be powerful free software and we are considering how our products can work closer together. In a field unrelated to the construction industry, we in talks with a software developer that is interested in using Columbus as part of the content management solution for an accountancy application.

7.6.3 Academic Links

In addition to working closely with other organisations within the industry, we have maintained a close relationship with numerous academic institutions. Included amongst these are:

- Greenwich University: As this research is carried out with this university, there has obviously been a great input form this institution. Various presentations have been made and interestingly, as the university has many part-time students, a number of them have been able to introduce Columbus into their work place.
- Harvard University: During the early phase of development, I travelled to Boston, USA to present Columbus at a special event at the Harvard Design School, which is lead by Dr. Spiros Pollalis. Since then, we have worked closely on a number of activities related to project hosting.

- Leeds University: Over the years, we have had a close relationship with Dr. Alastair Watson from the Civil Engineering department. We have been involved in their CIMSteel developments and intend to support the DocLink (CICA 2002a) document transfer specification that they are defining.
- Imperial College: Though we do not have a research and development association with Imperial college, we were able to establish a good relationship with Professor David Nethercot of the Civil Engineering department, in his capacity as head of the awards committee for the Institution of Civil Engineers. Appendix C describes how Columbus was introduced to the college.
- Others: In addition, from emails received or registration logs, we know that Columbus is being used by a great number of architecture and engineering university departments, particularly in the UK and USA.

Chapter 8

Conclusions

In this chapter, a general summary of the thesis is presented. It begins with a brief review of each chapter's findings and is followed by an appraisal of the research's outcome. This focuses on what have been its aims, achievements and failings, together with an assessment of what could have been done differently. Finally there is an evaluation of particular items that can be considered for further development.

8.1 General Summary

The thesis is made up of a number of separate chapters that address quite distinct aspects of the problem. Nevertheless, they all follow a single theme, building up to the working solution that is presented. To summarise, the following was discussed and established in each of the main chapters:

Chapter 2 investigated the state of the construction industry, looking at how project teams are formed and how contractual relationships have put project participants in an adversarial position rather than encouraging team working. Initiatives such as Latham and Egan were reviewed, which established a framework for enhanced collaboration. Improvements in Information Technology were identified as crucial to the success of these imperatives and within this, document handling was specifically highlighted as an important area for improvement. As a result, the focus of the chapter then changed to look at how documents are shared within the industry, highlighting possible pitfalls and failings.

Chapter 3 expanded on previous chapters and looked at the document handling technologies: document management, document control and project hosting. Each one of these was reviewed in detail and key areas for improvement were identified.

As a result of this, the main idea that the thesis investigated was how a unified solution to deal with all these systems could be produced. It proposed that by using document metadata in a neutral format and sharing data seamlessly between applications and participants, substantial gains could be achieved.

Chapter 4 built on the idea of linking systems and concentrated on the topic of metadata. It described what it is, how it is used, the manner in which it is stored and what are the various standards that exist for how it is defined and exchanged. It presented a series of XML schemas, which could be used as the basis for metadata exchange.

Chapter 5 presented a detailed description of Columbus, the product that was developed as a result of this research. It looked at how it addresses some of the key issues in information exchange by providing a unified document handling solution, yet still maintaining compatibility with project participants that do not use the product. It emphasised how data can be stored in open systems which are easy to manage and archive for future use. The chapter looked at each module in the Columbus application, describing its main features and usage from a user and commercial view.

Chapter 6 is a technical chapter, which was written from a computer science perspective. Using the preceding chapter as a specification, it described the application's architecture and presented a design. This was done using the Unified Modelling Language, resulting in a substantial number of use case and design views of the software. The application was covered from a high level perspective and then each component module was designed individually.

Chapter 7 discussed how Columbus was released within Arup and subsequently across the rest of the construction industry and beyond. It covered the reasons for making it freely available and how this would improve data exchange on Arup projects and throughout the rest of the industry. In addition, the other benefits of making the product freely available were also reviewed. The chapter also looked at how Columbus was made available over the Internet and on CD-ROM, it described the support mechanisms that have been put in place and how thanks to a number of alliances with other partners the product has been promoted further. There was also a detailed review of the uptake and success of the product based on user feedback, journal reviews, case studies and awards received.

8.2 Research Outcome

8.2.1 Review of Aims

The aim of this thesis is to look at how communication between project participants in the construction industry can be improved. It focuses on the use of Information Technology and specifically on how document exchange between applications and amongst organisations can be enhanced. Having established how the different technologies are used, the first practical issue that needed to be addressed was to avoid the repetitive and error prone task of re-entering document titles and numbers from drawings when exchanged amongst participants. The solution suggested was to use a neutral metadata format that makes it easy for applications at either end to interpret. In order to show the practicality and ease of this approach, the Columbus program was then developed, which resulted in a completed application that addresses a number of key issues that affect document handling. Even when the straightforward use of metadata does not require Columbus, the flexibility and capabilities of the software have made it a popular solution. Having produced this application for use within Arup, the next aim was to extend its use to other construction industry organisations, so that Arup and the industry as a whole could obtain the benefits.

The initial scope of the work was focused on CAD and engineering documents, dealing with the complexities of handling CAD information such as reference files. However, it soon became apparent that a unified information handling solution would need to cope with many of the other document types that are used in the industry. This therefore resulted in the need to deal with numerous other document types and cope with different information formats. This, for example required the integration of new viewing engines, being able to scan paper documents and supporting external project hosting sites.

8.2.2 Answers to Research Questions

In chapter 1, a series of questions were pinpointed as the goals for the project to consider and the following was established as answers throughout the research:

- There is a need to improve project participation by looking at the way in which team members collaborate. An important way to achieve this is to enhance the way in which document handling technologies are used.
- Document management, project hosting and document control systems were identified as the key document handling applications. Rather than produce a monolithic solution that would undoubtedly be difficult to adopt by all, it was suggested that a better option would be to have the applications exchange information seamlessly in a common format, resulting in a unified solution.
- Metadata exchange in a neutral format was identified as the best way of linking the various systems in a seamless manner and additionally improving the way in which information is stored when projects are archived. A number of standard schemas were suggested as a possible solution.
- As the most important output of the research, the Columbus application was presented as a possible solution, demonstrating how metadata could be shared to integrate various document handling technologies.
- The report reviewed how Columbus was made available across the industry and looked at how it became widely accepted as a viable solution, which has been adopted so far by more than 20000 organisations. Consideration was given to the approach taken in releasing the application and various case studies and feedback from the industry were highlighted to demonstrate its success.

8.2.3 Columbus Success

Columbus has proved to be the most successful tangible outcome of the research. It is a fully featured application, which contains a number of modules to handle internal document management and link to document control and project hosting systems effectively. Even its built-in project hosting capabilities are very often used as a project extranet in its own right.

The true benefit of Columbus has been in getting other project participants to send metadata with their electronic documents. This has worked very well on numerous projects, saving countless man-hours of data input time. It is important to emphasise that this one single item alone has been sufficient to justify all the investment that Arup have put into the development of this product and research.

One other achievement has been document archival; on various projects where other document management systems had been used, it was found that retrieval of documents was impractical just a few years after the project had been completed. On more that one occasion, the only valid information that was easily accessible to identify documents was the archived Columbus metadata. Because of this, the Arup Facilities Group, who are responsible for the archival of old paper documents have started to scan in all existing material and attach metadata in a Columbus supported format.

From a different perspective, having recouped their investment, Arup decided to make the product initially freely available to other project participants that they worked with. Subsequently, in the spirit of the Latham and Eagan initiatives, this was extended to the rest of the construction industry. This has resulted in a worldwide uptake of the product throughout the construction industry and beyond. It generated an unexpectedly large user base across the globe, with the product having been registered at the time of writing by over 20000 organisations in 165 countries since it was first released.

It is also important to emphasise that Columbus has promoted Arup as a forward thinking organisation, eager to improve the processes in the construction industry and to provide better value for money to their clients and has been recognised as such.

As a result of this, and in the words of the Institution of Civil Engineers: "In recognition of the improvements achieved by usage of the product, the innovative concept presented and facilitating cross-industry communication", this Institution awarded Columbus a special merit award in July 2000.

8.2.4 Limitations and Different Approaches

Having discussed what this research has achieved, it is also important to think about the limitations. This can be viewed both as limitations in the scope of the research and the lack of features within the software.

When this research began, metadata was not widely used for data exchange. It was found that the industry's ability to influence software vendors was minimal when commercial considerations are a factor. Because of this, the only real practical solution was to consider maintaining external metadata for documents. This, even if not ideal, forced the usage of simple text DIN and XML files rather than native metadata held within each document.

Now, there is a greater focus on information exchange and standards have appeared to encourage the use of common metadata formats, yet due to disagreements between vendors the situation is still not ideal. On many occasions, vendors appear not to be interested in data exchange between dissimilar systems. Despite the fact that they are happy for a user to exchange information with other users, they do not make it easy if the other user is using a competitor's product. This has meant that neutral formats for data exchange have not progressed and pseudo-open standards have been defined by each vendor. Even if this allows data exchange when a single CAD package is used, it prevents sharing information with dissimilar systems such as analysis, visualisation or planning software. In addition, it becomes very difficult to export metadata to project hosting sites and document control systems. This limitation still means that external metadata files will be used for some time. Though Columbus has now defined its own standard, close cooperation with other software vendors would have probably resulted in the definition of a standard XML schema before any of the software was developed and the standard metadata format would undoubtedly have been based on this definition.

As far as software features are concerned, there are many improvements that can be made and some are discussed in the next section. The software does not make radical changes to the way that organisations work as it is simply trying to automate existing processes rather than introduce new ones. The lack of a single model database makes searching and cataloguing more difficult, but this was a conscious decision in order to make the application robust, easy to manage and reliable for long term archival and retrieval. One other area of criticism has become more apparent as the Microsoft Windows operating system has evolved; Columbus has been left lacking in some areas compared with Microsoft Windows Explorer. Though Columbus was never seen as a replacement for this application, user's expectations have been that it should be.

If it were now to be developed a new product, Columbus would probably be written as a Windows Shell Extension rather than a standalone application. Columbus was first developed in the early days of Windows NT and 95, when most competing software was 16 bit Windows based, with clunky interfaces. Since then, Windows Explorer has become highly customisable and it is possible to provide some, but not all, of the metadata preview facilities of Columbus by creating shell extensions.

8.3 Further Work

As Columbus is widely used throughout Arup and by many other organisations, there is a constantly growing list of development requests. Even when Arup intends to keep the product as a fully functional free system, some of the more advanced new developments will only be available in a commercial release.

The type of work that is required falls into three main categories: maintenance, new features and future technologies. In terms of development timescales, it is important to emphasise that Columbus is still developed as a one-man application, but pursuing many of these developments is likely to require further programming staff to help with the workload.

From a straightforward maintenance point of view, a substantial amount of programming is constantly required to ensure that Columbus keeps up with existing technology. For example, the introduction of new operating systems, the appearance of new file formats or the requirement to work with other project hosting systems often require significant changes in the software. As the application becomes more complex, the overhead of maintenance grows at an increasing pace. However, re-writing complex areas of the code and making the application more flexible in design has helped to streamline Columbus and make it easier to maintain. The separation of core elements of code into different components and services using COM has been a great help in modularising the application and simplifying development.

In addition to basic maintenance and bug fixes, other development requests are for new features or improvements. For example, the current development list has items such as enhancing the search capabilities, putting the viewing engine within the navigator, embedding Columbus in an Internet browser, supporting live video and adding support for concurrent working using peer to peer technology such as Groove (Groove 2002). Some of these have now become more feasible as a direct result of improved communication links between participants.

As the industry adopts newer technologies, Columbus will also need to change. One major area where work will be required is in the implementation of Object Modelling. A number of industry initiatives are forcing a move from document and file based systems to object oriented databases. For example, in the CAD arena, the usage of objects in MicroStation and AutoCAD is being considered seriously. However, potential issues such as revision control, integrity and access by other participants needs to be taken into account and a product like Columbus can help.

References

AecXML 2001. AecXML home page. Web site: http://www.aecxml.org. Accessed: 5 August 2001.

Ammeraal, L. 1997. *STL for C++ Programmers*. Pub. John Wiley & Sons.

Amor, R., Clift, M., Juli, R., Papaspyridis, A.C., Sparacello, M., Borg, I. and Heikkinen, L. 1996. *ToCEE:WP-G:Document Modelling:Requirements.* ESPRIT Project No. 20587, ToCEE G1, EC, Brussels, Belgium, June, 26pp.

Amor, R., Clift, M., Scherer, R. , Katranuschkov, P., Turk, Ž., Hannus, M. 1997. *A Framework for Concurrent Engineering* - ToCEE, in: Proc. of the European Conf. on Product Data Technology "PDT Days 1997", 15-16 April.

Andreu, R., Ciborra C. 1996. Organisational learning and core capabilities development: the role of IT. *Journal of Strategic Information Systems;* 5, 111-127.

Arbortext 1995. A Guide to SGML and Its Role in Information Management. *Arbortext SGML White Paper.*

Armstrong, T. 1998. Active Template Library: A Developer's Guide. Addison-Wesley Pub Co.

Arup, 1994. *CAD Good Practice Guide.* Ove Arup & Partners, London, UK.

ATLAS, 1994. ESPRIT III Project No 7280.

Austin, T. 1995. Engineering – A Place in Cyberspace, *Civil Engineering (ASCE)* 65(6), pp.40-44, (June).

Autodesk 2001. The XML Revolution. *White Paper.*

Autodesk 2002. AutoCAD 2002: DXF Reference Guide. Web document, http://www.autodesk.com/techpubs/autocad/dxf/dxf2002.pdf, accessed 15 April 2002.

Avedon, D. 1997. New Federal Procedures & Regulations for Document Management. AIIM International.

Baker, T. 1997. Metadata semantics shared across languages: Dublin Cores in languages other than English. *Break-out group report from the Fourth Dublin Core workshop in Canberra.* (March).

Bannan, J. 1997. Intranet Document Management: A Guide for Webmasters and Content Providers. Reading. Mass.:Addison-Wesley Developers Press.

Barlow, J., Cohen, M., Jashapara, A., 1996. Implementing partnering: some common red-herrings in the literature. Paper presented at the Salford-Westminster Workshop on Partnering in Construction, 13 May.

Barlow, J., Cohen, M., Jashapara, A., Simpson, Y., 1997. *Towards Positive Partnering: Revealing the realities in the construction industry.* Final Report to the Economic and Social Research Council.

Bartholomew, D. 1999. The less-paper office. *Building Services Journal.* (October).

Bates, M. 1999. Second Review of the Private Finance Initiative. HM Treasury.

Berman, H. 1999. Understanding Project Delivery Methods. *The Practical Real Estate Lawyer*, Volume 15, Number 2, (March).

Berners-Lee, T. 1993. Hypertext Transfer Protocol (HTTP). CERN, Geneva, Switzerland.

Boddam-Whetham, P. 1998. IT: The Management of Project Information. Implementation of a project-centred management system. CIRIA Project Report 50.

Booch, G., Jacobson, I., Rumbaugh, J. 1998. The Unified Modelling Lanugage User Guide. 1st Edition. Addison-Wesley Pub Co.

Botterill, E. 1992. Cutting The Paper Mountain, *Professional Engineering,* 5(1), pp.30-31, (January).

Box, D. 1997. *Essential COM.* Addison Wesley Professional.

British Standards Institution 1987. *BS 5750:1987. Quality Systems.* London: BSI.

British Standards Institution 1994. BS 7768:1994., Recommendations for management of optical disk (WORM) systems for the recording of documents that may be required as evidence. London: BSI.

British Standards Institution 1998. BS 1192:1998. Construction drawing practice. Guide for structuring and exchange of CAD data. London: BSI.

British Standards Institution 2000. BS 8888:2000. Technical product documentation (TPD). Specification for defining, specifying and graphically representing products. London: BSI.

Building Centre Trust 1999. IT usage in the Construction Team.

Building Centre Trust 2000. Making the connection: a low cost 'extranet' for construction project teams: Damon Lock Grabowski. September. IT Case Study 20.

Building Centre Trust 2001. Effective Integration of IT in Construction.

Bunn, R. 1998. Construction IT in Practice. *Building Services Journal,* 20(10), pp.26-31, (October).

Burnard, L. 1991. What is SGML and how does it help? *Computers and the Humanities,* vol 29, pp.41-50.

CADdepot 2001. *Home Page.* Web document: http://www.caddepot.com Accessed: 5 June 2001.

CADopia 2002. *Home Page.* Web document: http://www.cadopia.com Accessed: 8 May 2002.

Carr, G. 2001. The Big Opportunity. *Workflow and Doc Mgt.* (September).

Castedo-Ellerman, E. 1997. Channel Definition Format (CDF). *Submission request to W3C.* Microsoft.

Castillo, O. 2000. Design and Application of an XML Schema for the Interchange of Documents in the Construction Industry. MSc. Dissertation. London Metropolitan University.

Causeway 2002. *Home Page*. Web document: http://www.causeway-tech.com. Accessed: 7 February 2002.

CICA 2000. BRE/CICA Case Study: Document Management at Ove Arup & Partners. Web document, http://www.cica.org.uk/arup_columbus_edm_case_study/ arup_columbus_edm_case_study.htm. Accessed 7 June 2001.

CICA 2002a. *CICA Bulletin*, No. 81 (Spring). Pub: Construction Industry Computing Association.

CICA 2002b. Guidance on the Introduction and use of Construction Extranets. *Peliminary report on the Current Use of Extranets.* DTI PII. Pub: Construction Industry Computing Association.

CIMsteel 1994. Definition of a Products Library for Steel Structures, Design Task Group.

Cockshaw, A. 1997. The Process of Civil Engineering: From Vision to Reality.

Cole, M. 1992. Express UDO for Heathrow. *Construction Computing,* No.53, pp.12-13, (December).

Compton, A. 2002. How Private Is Your Company Data?. *Design Productivity Journal (Excitech).* Vol .2 , Issue 5 pp. 8-9. (April).

Conlon, A. 2002. CADplot Server. *AEC Magazine*. Vol. 12. July/August Issue.

Connectix 2002. Connectix Web Site. Web Document, http://www.connectix.com Accessed 16 June 2002.

Construction Industry Board (CIB) 1997. Code of Practice for the Selection of Main Contractors.

Construction Industry Board (CIB) 1998. *Good Practice is Good Business.*

Construction Industry Council 2000. A Guide to Project Team Partnering.

COMBI 1995. ESPRIT III Project No 6909.

Contract Journal 1998. From Paper To Electronic Office. *Contract Journal.* 393(6177), pp.32-33, (May 28).

Cramp, M. 2001. *A Reporting Application For the Columbus Document Manager.* MSc. Dissertation. University of Westminster.

Cravens, D., Shipp S. 1993. Analysis of co-operative inter-organizational relationships, strategic alliance formation, and strategic alliance effectiveness. *Journal of Strategic Marketing.* 1 55-70.

Critchlow, J. 1998. Making Partnering Work in the Construction Industry. Oxford: Chandos Publishing.

Day, M. 1998. Defeating the Object – Chapter 2. *CADdesk Magazine.* Vol. 8, Issue 8, pp. 32-38 (September).

DCMI 1999. Dublin Core Metadata Element Set, Version 1.1: Reference Description. Pub: Dublin Core Metadata Initiative. Web Document, http://dublincore.org/documents/dces/. Accessed 2 September 2001.

DMCA 1998. Digital Millennium Copyright Act. (Pub. US Congress). Available on-line: http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname =105_cong_bills&docid=f:h2281enr.txt.pdf. Accessed: 2 August 2001.

Documentum 2002. *Home Page.* Web document: http://www.documentum.com Accessed: 18 September 2001.

Doherty, P. 1999. Site Seeing. *Civil Engineering(ASCE)*, 69(5), pp.38-41,(May).

Dornfest, R., Brickley, D. 2001. The Power Of Metadata. *O'Reilly's Emerging Technology Conference.*

Egan, J. 1998. *Rethinking Construction*, The report of the Construction Task Force to the Deputy Prime Minister, John Prescott, on the scope for improving the quality and efficiency of UK construction, Department of Environment, Transport and the Regions, HMSO, London.

Excitech 1999. Single Model Envinronment. *Design Productivity Journal*, Vol 1, Issue 3, pp 9-11.

Excitech 2000. Project Hosing: Is it the way forward? *Design Productivity Journal*, Vol 2. Issue 1, pp 16-18.

Fallon, K. 1998. Look For Higher Technology Benefits. *Architectural Record.*

Felix 2002. Felix CAT Web Site. Web Document, http://www.fcad.de. Accessed 27 July 2002.

Flanagan, R., Norman, G., Ireland, V., Ormerod, R. 1985. A Fresh look at the U.K. & U.S. Building Industries.186(12), pp.139-142, (December).

Franklin, C. 1997. Emerging Technology: Enter the Extranet. *CIO Magazine*, (May 15).

Frappaolo, C. 1992. The Promise of Electronic Document Management. *Modern Office Technology. (*October).

Gay, W. 2000. Linux Socket Programming by Example. Que Publishing.

Gielingh, W. 1988. General AEC Reference Model, ISOTC 184/SC4/WG1 DOC N 3.2.2.1 Some Business Questions Examined. Proceedings of the 2nd EC-PPM conference, Watford, UK.

Goldfarb, C. 1990. *The SGML Handbook.* Pub: Oxford University Press.

Graham, I., Quin, L. 1999. *XML Specification Guide.* Pub: John Wiley & Sons.

Green, R. 2000. CAD Standards: Making the Case. Cadence, (July).

Grienfeld, M., Grienfeld, A. 2000a. Get Sorted. *Building,* 265(24), pp.79-82, (16 June).

Grienfeld, M., Grienfeld, A. 2000b. Scrap Paper. *Building,* 265(37), pp.99-102, (15 September).

Groove 2002. Groove Web Site. Web Document, http://www.groove.net. Accessed 26 July 2002.

Grout, P. 1997. The Economics of the Private Finance Initiative. *Oxford Review of Economic Policy*, Vol. 13, No. 4, pp53-66.

Grundgeiger, D. 2000. *CDO and MAPI Programming with Visual Basic.* O'Reilly & Associates.

Gruneberg, S. 1995. Responding to Latham: The views of the construction team.

Hannus, M., Heikkonen, A., Laitinen, J. 1996. Internet in Construction Projects and Research. In: Turk, Z. (ed) 1996. Construction on the Information Highway. CIB Proceedings Publication 198. CIB W78 & TG10 Workshop, 10-12 June, 1996. Bled, SI. ISBN 961-6167-11-1. University of Ljubljana, Faculty of Civil Engineering and Geodesy. Pp. 265-272.

Hannus, M. 1998. *Islands of Automation.* Web document, http://www.vtt.fi/cic/hannus/islands.html, accessed 11 November 2001.

Hapgood, F. 1998. Tools for Teamwork. *CIO Magazine*, (November).

Harrod, G. 2002. IntelliCAD 2001 from Cadopia...with Columbus EDMS. *CADinfo.NET (technical newsletter Review).* Web document, http://www.cadinfo.net/reviews/icad2001.htm, accessed 17 April 2002.

Head, R. 1997. Document Management : The Essentials. AIIM International.

Heery, R. 1996. Review of Metadata Formats. *Program.* Vol. 30, No. 4, October issue, pp. 345-373.

Herrero, J. 2002. Concrete Detailing. *Concrete Engineering Magazine.* Summer issue.

Hewitt, G. 1995. Electronic Document Management And Control. *Construction Computing*, no. 49, pp.16-17, (Summer).

Hillmann, D. 2000. Using Dublin Core, Web Document, http://purl.org/dc/documents/usageguide. Accessed 5 October 2001.

Hollingsworth, D. 1994. Workflow Managment Coalition. The Workflow Reference Model. TC00-1003.

Holt, G., Olomolaiye, P., Harris F. 1996. *Tendering procedures, contractual arrangements and Latham: the contractors' view*. Engineering, Construction and Architectural Management. Vol. 3, No's 1 and 2, pp 97-115. Oxford: Blackwell Science.

Høvlück, J. 1994. The Pragmatics of Electronic Data Interchange: The Use of EDI will Unify Business Procedures. *International Journal of Information Management.* Vol 14, No 5, pp 330-343.

IAI 1996. End User Guide to Industry Foundation Classes: Enabling Interoperability in the AEC/FM Industry. International Alliance for Interoperability.

IAI 1999. An Introduction to the International Alliance for Interoperability and the Industry Foundation Classes, IAI Pub., Oakton, VA, USA.

ISO 10303-225 1997. Application protocol: Building elements using explicit shape representation, Industrial automation systems and integration: Product data representation and exchange Part 225. Draft International Standard WG3 N510 (T12).

Jacobson, I., Booch, G., Rumbaugh, J. 1999. The Unified Software Development Process. 1st Edition. Addison-Wesley Pub Co.

Kernon, S. 2000. Project Collaboration Extranets. *New Civil Engineer*, no. 1380, pp.32-34, (7 December).

King, V. 1996. *Constructing the Team: A US Perspective.* CIB W65 (in conjunction with W92) International Symposium for the Organization and Management of Construction: Shaping Theory and Practice, Glasgow, Scotland, pp 414-423.

Knutt, E. 1998. Drawing them into the net. *Building,* 263(45), p.75, (6 November).

Knutt, E. 2000a. IT:Extranets: Learn To Share. *Building*, 265(14), pp.54-55, (7 April).

Knutt, E. 2000b. Mix and Match. *Building*, 265(24), pp.60-61, (16 June).

Koulopoulos, T., Frappaolo, C. 1995. Electronic Document Management Systems: A Portable Consultant. McGraw-Hill.

Krouse, J. 2001. The Real Value of Collaborative Tools. *Engineering Process Journal*, Issue 01-03.

Kruglinski, D., Shepherd, G., Wingo, S. 1998. Programming Microsoft® Visual C++®. 5th ed. Microsoft Press.

Lagoze, C. 1996. The Warwick Framework: A Container Architecture for Diverse Sets of Metadata. *D-Lib Magazine*. July/August.

Laiserin, J. 2001a. Online Collaboration: Coding for Behaviour. *Cadence,* (February).

Laiserin, J. 2001b. Digital Architect: Technology for Design Collaboration. *Architectural Record*, 198(3) pp. 173-174, (March).

Lampretch, J. 1993. Implementing the ISO 9000 Series. New York: Dekker.

Latham, M. 1994. Constructing the team, Final Report of the Government/Industry Review of Procurement and Contractual Arrangement in the UK Construction Industry. HMSO, London 1994.

Levien, R. 1989. The Civilizing Currency: Documents and Their Revolutionary Technologies. Rochester, N.Y.:Xerox Corporation.

Line, L. 1997. Virtual Engineering Teams: Strategy and Implementation. *Electronic Journal of IT in Construction.* Volume 3.

Lorange, P., Roos, J. 1991. Why some Strategic Alliances Succeed and Others Fail. The Journal of Business Strategy. January/February, 25-30.

Maher, K. 2000. Project Extranets give Design Professionals a Reason to Get Wired. *Architectural Record,* 188(9), pp.175-180, (September).

Masinter, L. 1995, Document Management, Digital Libraries and the Web. *In: INET'95 Hypermedia Proceedings.*

McKie, S. 1995. A New Era in Document Management. *DBMS*, (June).

Michalski, G. 1991. The World of Documents. *BYTE*, 16(4) pp.159-170, (April).

Microsoft 1998. HOWTO: Read Compound Document Properties Directly with VC++. *Knowledge Base Article Q186898*. Web Document: http://support.microsoft.com/default.aspx?scid=kb;EN-US;q186898. Accessed 3 May 2001.

Milstead, J., Feldman, S. 1999. Metadata: Cataloging by any other name. *Online,* January Edition, p24-31.

Morgan, D. 1999. Workings of a Virtual Private Network, Part 1. *Linux Journal*, Issue 68, (December).

Mosey, D. 2000. PPC 2000 ACA Standard Form of Contract for Project Partnering.

Moving Parts 2002. Home Page. Web document: http://www.movingpart.com. Accessed 2 May 2002.

MSM 2000. To E or not to be: E/C/O confronts the new economy: A report from the first "E-Business in the Construction Industry" conference. *MSM Magazine*. July.

Murray, J. 2002. CCITT Group3 and Group 4 Encoding. *Graphics File Formats FAQ*. Online: http://www.faqs.org/faqs/graphics/fileformats-faq/part3/section-26.html. Accessed: 20 February 2002.

NASA 2002. *NASA Software Technology Helps Build Industrial Facility*. NASA Public Affairs Press Release 02-51. Web document, http://nctn.hq.nasa.gov/spotlight/articles/article_145.htm, accessed 10 April 2002.

Nonags 2001. *Home Page*. Web document: http://www.nonags.com Accessed: 15 February 2001.

Oasys 2001a. Teach Yourself Columbus Training Manual. Pub: Oasys Ltd.

Oasys 2001b. Advanced Configuration Training Manual. Pub: Oasys Ltd.

ODMA 2002. Open Document Management Application Programming Interface Web Site. Web document, http://odma.info, accessed 5 June 2002.

Oracle 1999. Application Developer's Guide - Large Objects, Release 2. Oracle Corporation.

Oram A. 2001. Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology. O'Reilly & Associates.

Orange, G., Boam, J., Dixon, P. 1998. A Networked System to Facilitate Organisational Learning Within the Construction Industry. *The 4th EATA International Conference on Networking Entities. October 1998.*

Orr, J. 2000. Extranets: Likker App or Overblown Fluff? *CADinfo.NET online journal* (November).

Orr, J. 2002. *Extranet News*. Web document: http://www.extranets.cc, accessed 30 May 2002.

Ouzounis, V. 2001. Analysis of Distributed Technologies for the Usage in the Context of Virtual Enterprises. COVE News 1/2001.

Parapadakis, G. 1996. Document Management System: What is it and should I buy one? *Document Management Avenue*. (October).

Postel, J. 1980. File Transfer Protocol Specification, RFC 765.

Priestley, M. 2000. Practical Object-Oriented Design with UML. Pub. McGraw-Hill.

Rasterex 2002. *Home Page*. Web document: http://www.rasterex.com. Accessed: 21 June 2002.

Rational 2001. Rational Web Site. Web Document, http://www.rational.com Accessed 1 July 2001.

Rein, G., McCue, D., Slein, J. 1997. A Case for Document Management Functions on the Web. *Communications of the ACM*, 40 (9), pp.81-89, (September).

RIBA 1987. Joint Contracts Tribunal for the Standard Form of Building Contract. RIBA Publications: London.

Royal Academy of Engineering 1996. A Statement on the Construction Industry.

Salimandro, J. 2001. *E-Commerce Story: I Was Wrong About the Web Collaboration Thing.* The Electrical Distributor Magazine. (24 July).

SHC 2002. Swanke Hayden Connell home page. Web Document, http://www.shca.com. Accessed 13 July 2002.

Simmons, C., Hulbert, C., Maguire, D. 1995. *Manual of engineering drawing.* London: Arnold.

Simmons, C. 2002. Why Introduce BS8888 and withdraw BS308. *Engineering Designer,* 1(29), pp.12-14, (January/February).

Stellent 2002. Outside In Product page. Web document: http://www.stellent.com/stellent3/groups/public/documents/translatedpage/puba_p2000402.hcsp. Accessed: 2 May 2002.

Suchocki, M. 1998. The Genesis Project – A Manufacturing approach for construction. CIRIA Project Report 51.

Suomi, R. 1992. On the concept of inter-organizational information systems. *Journal of Strategic Information Systems.* Vol 1, No 2, pp 93-100.

Tenlinks 2001. *Home Page.* Web document: http://www.tenlinks.com Accessed: 1 December 2001.

Turk, Z., Wasserfuhr, R., Katranuschkov, P. 2000. *Environment Modelling for Concurrent Engineering, Int.J. of Computer Integrated Design and Construction /CIDAC/ 2(1),* Special Issue on Concurrent Engineering in Construction, SETO, London, UK, 28-36.

TWAIN 2000. TWAIN Specification Version 1.9. *TWAIN Working Group Committee.*

Wager, D. 1998. Document Management for Construction. *Construction Industry Computing Association Report.*

Weibel, S., Kunze, J., Lagoze, C., Wolf, M. 1998. RFC 2413, Dublin Core Metadata for Resource Discovery.

Weibel, S., Iannella, R., Cathro, W. 1997. *The 4th Dublin Core Metadata Workshop Report.* D-Lib Magazine (June).

Wailgum, T. 1998. What are Extranets? *CIO Enterprise Magazine*, (15 March).

Warburton, M. 1998. Introducing an Electronic Document Management System: Case study on the Derby Southern Bypass, CIRIA.

Weisberg, S. 2001. *i-Collaboration - State of the Industry.* CADALYST Magazine (September).

Wilson, P. 1998. STEP and EXPRESS. Contribution at the NSF Workshop on Distributed Information, Computation and Process Management for Scientific and Engineering Environments, 15-16 May.

Wiggins, G. 1999. Shoppers Guide to Document Management Software. *Law Office Computing.* (April/May).

WINE 2002. Wine Development HQ Web Site. Web Document, http://www.winehq.org. Accessed 2 February 2002.

Wolton, O. 2000. Come Together. *International Construction*, 39(3), p.39, (April).

Woollard, C. 1988. *A Networked Object Oriented Operating System.* PhD Thesis. University of Essex.

Appendix A

Journal Reviews

Since the first version Columbus was released, many articles have appeared in construction industry journals mentioning the product. Amongst them, the following quotes are particularly relevant:

"In 1996, Ove Arup began exploring the uncharted waters of document management systems. The team was unimpressed by what it found. Then an expert from its own computer arm rang. The result, Columbus, is a gift to the industry."
Building Magazine - October 1999

"Ove Arup want to see other firms adopting Columbus to facilitate standard working procedures across the construction industry."
Building Design - October 1999

"One particular advantage to my practice is that Columbus allows peer-to-peer networks to see all information for a particular project as a seamless series of files, even though the material is held on a series of PCs in geographically dispersed offices." - Quoting Mervyn Hill, Archimedia Consulting.
The Architects' Journal - February 1999

"Columbus is now thought to be the most widely used document management system in the UK and is rapidly gaining acceptance across the globe. Around half of the downloads are from the USA."
PaperSpace – December 2000

"In short, Columbus provides a very good compromise between the insufficient organization of files provided by standard system tools, and the over complex controls provided by some of the big document management systems. Those big systems are also extremely expensive, while Columbus is essentially free! But that should not be taken to mean it is ineffective or too limited. It is used on some very big projects around the world, and is a most valuable extra with IntelliCAD 2001."
CADinfo – January 2002

"Columbus can view over 200 commercial file formats - showing a thumbnail view of the document at the bottom of the screen. No great surprise there. The shock comes when you see the thumbnails appear almost immediately when you put the cursor on a file. At last here's an end to the infuriating wait while your computer opens a new application each time you need to view a document."
Building Services Journal – April 2000

"And so Columbus...
In January of 2000, I first started my explorations of Columbus and began to put the product through it's paces.
Several things struck me initially, Columbus was a document manager designed for the Architectural, Engineering and Construction community and it is not simply a drawing file manager. Our administrative, marketing, and project management staff are beginning to use Columbus to manage their documents as well. Also, Oasys had not made the mistake that I had seen in many of the commercial products, which was to include features just because they could, Columbus is a basic, rugged, and well programmed document manager."
PaperSpace – December 2000

"This document-management system (reviewed in Architech, AJ 24.2.00), is a brilliant spin-off from Arup's preoccupation with the nuts and bolts of handling documents.
...The free program is a heavyweight application which has won awards, has a lively website and an active user forum. There is a good chance it will become the industry standard document-management tool. Thousands of people worldwide like it and Arup has a commanding influence over the construction values of many countries."
The Architects' Journal – March 2001

"...allows firms to access internal documents and extranet project data through the same interface. Columbus operates like the Windows Explorer interface, allowing clients with limited design knowledge to comprehend the system."
The Zweig Letter –March 2000

"The ability to view, print, amend and issue a variety of documents and files without access to a plethora of specific software seems like a dream, and an expensive one.
Thanks to Arup - one of the world's pre-eminent engineering design consultancies, one of the UK's largest CAD users and AutoDesk's only VIP customer in the UK - the software, at least, is free.
As well as their prestigious engineering record, the practice has a strong history of software design for the engineering market, sold through their subsidiary Oasys Ltd. In 1996 the practice saw a need for a system that would enable them to manage documents and drawings for large design teams, who often worked in a number of locations. A search for a commercially available solution was fruitless as costs proved astronomical, so the in-house development team set about designing their own software. The result was Columbus.
The program soon proved itself across all parts of the organisation and moves were made to sell it alongside other software in the Oasys stable. However, they were quick to realise that they had more to gain by giving it away to the industry."
Design Productivity Journal (Excitech) – February 2001

"Columbus incorporates both the Stellent and Rasterex viewing engines, which between them can view well over 200 different file formats. As a result Arup users say it makes it easier for them to find, view, edit, print and issue project data since they need not concern themselves with the document type, its location, or any database structure.
And Arup points out that because Columbus combines the features of an internal document management system and an extranet system, you can use the same tool to handle both activities. Other features include a very simple interface, fast access to any file, global work sharing support and standardised project structures. Also, users can roam anywhere on LAN, WAN and Internet; it's available to all users, for any work; it automatically extracts title block information from AutoCAD DWGs and provides fast previews of AutoCAD files plus 'Xref aware' full views."
Manufacturing Computer Solutions – March 2001

"All credit to Ove Arup, both for recognising that the product would be useful to the outside world, and for making it available to users free."
PC Pro Magazine – April 2001

In addition to these journal reviews, CADopia made the following press release:

CADopia and Arup announce availability of Columbus 2.4a with IntelliCAD by CADopia

San Diego, CA - December 11, 2001/Software Wire/ - CADopia LLC, developer and distributor of IntelliCAD and Oasys Ltd, the software house of consulting engineers and designers Arup have teamed up to offer the latest 2.4a release of Columbus document management software with IntelliCAD.

Building on the strengths of what is now believed to be the world's most popular document management system of its kind, Columbus 2.4a includes new features that make organising and sharing data even easier. It also incorporates new viewing filters providing support for Microsoft Office XP (including canvas object and ESHR graphics), Outlook 2002, Corel WordPerfect Office 2002, Lotus SmartSuite Millennium Edition 9.6, Adobe Acrobat 5.0 and Visio 2002.

Columbus is unique because it is a simple system that can be used by different types of companies of all sizes. Indeed any organisation that needs to unite data from different parts of its working environment without the complexities of traditional document management systems can use Columbus. Arup has teamed up with CADopia, a developer and provider of professional CAD software, to make Columbus 2.4a available to more users.

"Columbus 2.4a will take document management even further," said Alec Milton of Oasys. "Columbus is the system of choice for over 7500 companies, with up to 30 more switching over to Columbus each day. People are finding that its easy document control and data exchange is just what they need."

"Columbus is a very popular document management system without the complexities of the traditional document management system. Adding Columbus to IntelliCAD provides our users with an unbeatable range of powerful solutions at an affordable price," said Surya Sarda, CEO of CADopia. "We are pleased to be working with Oasys to provide CADopia users with Columbus to meet their document management needs without the high cost associated with other document management software."

About CADopia LLC

CADopia is creating the industry standard for an affordable and powerful CAD solution. As a privately owned company with a network of strategic ISV partners, distributors and resellers, and a user base of over 350,000 users from 80 countries, CADopia is one of the world's most popular providers of professional CAD software. The software is affordable for the masses, yet powerful enough for the most demanding projects. The company empowers engineers and architects the world over with the tools needed to create highly successful solutions for their customers. Founded in 1999, CADopia is a privately held California Corporation. For more details about CADopia, visit the CADopia web site at www.cadopia.com.

Appendix B

User Endorsements

Columbus has received a lot of user feedback, and the majority has been very positive. Though a lot of people have commented privately, many others have made their views public. The following are a series of quotes that are particularly noteworthy:

"With what I have seen, you will revolutionise the Construction industry."
Sir John Egan, Author of "The Egan Report"

"I've just got Columbus and am really impressed. Wow! A chance to dig myself out of the chaos I pretend is a system. Even in a 2-6 person set-up things can get chaotic, and without having a really good reason to maintain a decent filing system things just slide. We're going to try bending our system into Columbus and hopefully regain some sanity."
Scott Gideon, Forum Posting

"We run this program on a network with about 100 AutoCAD workstations, so it is a great solution for us to find out who is working on what drawing (and locking it)."
Kjell-Petter Ertesvag
Norway

"Some of our engineers found a link to your Columbus software on the Internet. Now that they've read all about its features, they are quite enthusiastic - they'd like to use it as soon as possible (especially because of its ability to handle MicroStation drawings)."
Hans Palm
ExxonMobile

"…. I feel that the "File Issue" feature is one of the best parts of Columbus. I am really impressed with how it can 'pull' all of the attached XREFs from a drawing and include them in the new issue directory…."
Craig Vaughn, Rogers-Ford
Dallas, Texas, USA

"I am the owner of Felix CAT GmbH ,the manufacturer of FelixCAD and CeCAD, mobile and desktop CAD Software. Our software is available in 10 languages and is used around the world. We share your view that there should be powerful free software and wondered if there is a way to combine our products.
If you want, you can integrate our LT application with your free offerings and I would also be happy to do the same with our offerings.
Please let me know if there is any interest from ARUP to enter into a relationship with us."
Wilfried Graebert, Owner of Felix CAT GmbH,
Germany

"My company has offices world wide, with each office operating differently. Columbus allows us to effectively share information in a way unmatched by our current methods."
Adam Saluk, TKS Group (Taikisha Ltd.)
Tokyo, Japan

"Columbus looks as though it might be the answer (or part thereof) I have been seeking for non-paper management of email correspondence for clients and projects. This is a galloping problem that I'm sure is affecting most businesses.
In our business there are always several engineers active on any given project or corresponding in regards to a particular client's assignments. These individuals exchange numerous messages, copying others for information, always "keeping the loop closed" and therefore generating great replication of information.
All of the individuals are in the habit of saving their own copies with often inconsistent directory structures and naming/filing conventions."
Ian Bainbrigge, Managing Director
Krebs Engineers Pty Ltd.

"Just downloaded and had opportunity to 'tinker' (poor verbage) with your Columbus program. Folks, you have taken the world one step closer to sanity in the arena of document mangement. Finally, something that takes a hodge podge collection of faxes, spreadsheets, scanned tiff files, etc...... and assembles them in a usable fashion. I have scoured the web for months searching for such a program. Thank you very much.
Ps: all that and 'mapi compliant' also........yippee"
Paul Clark, Forum Posting

"I had the fortunate experience of being BSI audited a few weeks ago with half an hours notice. Not only did Columbus save my bacon (and possibly my job) but also the auditor himself went away 'drooling' at what he'd just seen. His parting comment was that is was the best Document control system that he had seen and will all the office be running Columbus on his next visit. There were no non-conformities found and I think Juan and Alec will be fully justified to carry on as they are. Keep up the good work."
Ian Humphreys
Arup

"Congratulations on your software Columbus. Having spent some time assessing your software I found it to be very robust and simple to use and it compares favourably with many entry-level (commercial) document management systems that I have previously seen. I was particularly impressed with your navigator-style front end and in particular its ability to group files under a common heading even though stored in different locations. I found the 'CDS' data structure easy to use and was able to create my own templates quickly and easily. I was able to populate the drawing history using our attributed title blocks via your projects setting file."
Hoare Lea & Partners

"Your software will be featured in one of our upcoming Lockergnome e-mail newsletters. We have over 250,000 subscribers. You may find your Web logs spinning soon; this is a nice "side effect" from being featured in Lockergnome.
Lockergnome only chooses the best of the best... you're definitely noteworthy!
Please keep us abreast of any new developments. Congratulations on providing fine resources to the Internet community at large! If you have any other questions, please let me know."
Chris Pirillo
Lockergonome Newsletter

"I did a presentation for a group representing all our offices, Columbus was received very well and all the CAD people are quite excited. I believe that Columbus is a great product and an excellent solution for document/drawing management for the small to mid sized AEC firm."
Jim Brinkmeyer, OTAK
USA

"I've just downloaded and installed Columbus, and I'm stunned. I can really appreciate a) the scale of the problem you guys set out to tackle and b) how well you've tackled it - I'm currently working for a web design company and we were considering getting a DMS, I remembered something about Columbus being released as freeware and looked it up, and I'm incredibly impressed. Many, many congratulations on a job well done."
Dylan Beattie, Software Developer
X-RM Ltd.

"I am very impressed...
We have been looking at document management systems for a while now and this one almost rivals Bentley's Project Wise, which is a couple of tens of thousands of US dollar investment with a hefty maintenance fee each year on top of that."
Gary Mensch, CAD Manager
The Lauck Group

"We first became interested in Columbus because of its flexibility and cost. We have customers all over the place and we can set up project ftp sites and deal with it all within Columbus without resorting to an expensive and cumbersome "collaboration" system. It's fairly easy to convince a customer to use software that is freely available. It's worked fairly well thus far. We also use suppliers that have ftp sites with electronic parts files and can set these up for easy access as well.
What we are really looking for is a PDM system without the constraints of a typical PDM system. We like the localized structure of the Columbus system. We do all our data generation and storage in SolidWorks files, so it makes complete sense to do the same with the data management information."
Christopher Dubea, Vice President of Engineering
Moving Parts L.L.C., New Orleans, USA

"Just as some background, I downloaded Columbus 2-02 a couple of years ago and reviewed it for use in our organization. At the time the overhead of the software was just too much for our AutoCAD installation and the rather ancient hardware we had at the time. I have just recently downloaded and installed the latest 2-4-b version and I am quite impressed! It looks to be the solution to our document management problems! The overhead of the software during an AutoCAD session is imperceptable and after configuration of the projectsettings.txt files, I am able to extract "most" of the information I need from my drawings."
Robert Carroll
Forum Posting

"You work for an excellent company if they support this kind of sharing. I guess that is the point. An atmosphere of sharing builds a strong company in so many ways."
The Glave Firm, Architects
Virginia, USA

Appendix C

Institution of Civil Engineers Award

Over the lifetime of Columbus, the product has received a number of awards from the software industry. However, what is most cherished above all, is winning a special award from the Institution of Civil Engineers. The award, shown in Figure C.1, was presented in July 2000 at a ceremony held at the Natural History museum in London. In presenting the award, Professor Nethercot of Imperial College, head of the awards committee, made the following statement:

"Six projects were short-listed - only one can win - yet by common consent this year's group contained several entries of award standard.

One of these was not a construction project in the accepted sense. Columbus is a piece of software. Created by the Ove Arup organisation and made available to the industry free of charge through the Internet, it represents an imaginative contribution to the Latham and Egan imperatives of facilitating working in traditional inefficiencies and artificial boundaries. As an easy to use document management, viewing, filing and communication vehicle, I challenge any organisation represented here this evening not to find uses for it. Indeed, as a result of last week's demonstration, I mentioned it to a colleague in my own university department; the next day I received an e-mail listing six uses to which we could immediately put it.

Therefore in recognition of both its innovative concept and, in particular the innovative way in which it is being used to facilitate cross-industry improvement the judges wish to make a special award to Columbus."

Professor David Nethercot of Imperial College London,
presenting the ICE Special Award in July 2000.

Appendix D

Chiswick Park Procedural Notes

As described in Chapter 7, the Columbus metadata format was used with great success to exchange information between participants on the Chiswick Park project. Though it was not a requirement to use the Columbus application, it was compulsory to exchange information it that format. As part of the project, the following "Procedural Notes For Issuing Drawings Using Columbus" were issued to participants:

General

Project parities will issue drawings and documents in a file type best suited for the use the information will be put to. The following file types are recommended:

- Drawings which need to used by or added to by another party = native format i.e. dwg
- Drawings for comment or construction = image format i.e. plt files
- Documents which need to used by or added to by another party = native format i.e. .doc
- Documents for comment or construction = image format i.e. pdf

Trade Contractors

This is what happens to a package of drawings when it is issued for approval by a trade contractor:

1. The package of drawings to be issued should be given a unique reference number by the Trade Contractor. For example Hewetson Ltd. would give the first package of drawings a reference number RFA-HEW-001. This

unique reference number is important because it can be used to track comments and status of the issue.

RFA = Type of issue (Request For Approval)

HEW = Trade Contractors ID (Hewetson Ltd)

001 = Unique number

2. The package of drawings are issued by the Trade Contractor to the FTP site in their named area in the folder 'For Approval' in native format i.e. AutoCAD 14 together with any associated/reference files.
   e.g. Phase 2/Contractors/A4410 Hewetson Access Floor Tech/For Approval/RFA-HEW-001
   Bovis is notified of this by email

3. Bovis download the package of drawings onto to their network, and decide who it should be sent to and identify who is going to be the lead coordinator of the comments (The other designers and trade contractors who will need to comment are decided by Bovis and/or the lead coordinator).

Once it has been decided who will review the package of drawings, Bovis will upload the drawings into the lead coordinator's 'TC Drgs For Approval' folder keeping the reference number associated with the package of drawings.
e.g. Phase 2/Consultants/RRPartnership/TC For Approval/RFA-HEW-001
The lead coordinator, other interested designers and trade contractors are notified of this by email.

4. The lead coordinator and/or other designers and trade contractors download this package of drawings from the 'TC Drgs For Approval' folder, and

review the drawings and provide comments as required either in native format, PDF or PLT format (HPGL2).

5. Other designers and trade contractors should upload their comments to their own 'TC Drgs With Comments' folder making sure to retain the original TC's reference number /RFA-HEW-001.
   e.g. Phase 2/Consultants/Ove Arup & Partners Services/TC Drgs With Comments/RFA-HEW-001
   The lead coordinator is notified of this by email.

6. The lead coordinator then downloads the other designers and trade contractors comments and produces a coordinated set of comments either in native format or PDF, PLT format (HPGL2). This coordinated set of comments is then issued to their own 'TC Drgs With Comments/RFA-HEW-001 folder and notified to Bovis and other designers.
   e.g. Phase 2/Consultants/Richard Rogers Partnership/TC Drgs With Comments/RFA-HEW-001

7. Bovis download the coordinated set of comments, review and add further comments if necessary.
   If approved: Bovis issues drawings to the 'For Construction' folder under the relevant building no. and package and notifies all parties.
   e.g. Phase 2/For Construction/Building 1/Contractors/A4410 Hewetson Access Floor Tech
   If not approved: Bovis issues comments back to the Trade Contractor in the 'With Comments' folder
   e.g. Phase 2/Contractor/A4410 Hewetson Access Floor Tech/With Comments/RFA-HEW-001
   The Trade Contractor notified of this by email

Trade Contractors & Consultants

This is what happens to a package of drawings when it is issued for construction by a trade contractor, consultant or Architect:

1. Using the same package number as defined for the approvals procedure or with a new reference number for packages which has not gone through the approvals procedure, the Trade Contractor, Consultant or Architect will place the drawings in their For Construction area and notify Bovis of this issue.

2. Bovis will then check this issue and move the drawings/documents within this package to the For Construction area where they will be placed in folders with reference to each specific building. Bovis will then notify the required project parties that this information is available for construction.

Consultant / Architect

This is what happens to a package of drawings when it is issued between Consultant and Architect:

1. Consultants and architect are free to issue drawings to their own areas as required to facilitate the flow of information between each party.
2. File types and redlining procedures can be defined by the parties as required.