THE ARROW FRAMEWORK FOR A BUILDING OBJECT WAREHOUSE

L. Newnham, F. Parand, R. Amor and N. Nisbet

Building Research Establishment, Garston, Watford WD2 7JR, UK.

{newnhaml, parandf, trebor}@bre.co.uk

Abstract

This paper describes the DOE[1] supported ARROW (Advanced Reusable and Robust Objects Warehouse) project, the services it will provide, the research issues it is addressing and how it fits into the whole UK construction Industry Knowledge Base (IKB) system. The ARROW project will require the development of new techniques for selection and retrieval of data across distributed databases. The project is examining how to establish an interface to distributed databases containing product information, allowing fast and accurate retrieval of specified products, as well as delivering information in a form useable by CAD systems and other design tools (e.g., thermal simulation programs). Initial work is looking at appropriate product selection and techniques for the refinement of options to narrow and improve the selection process. Agent technologies are being investigated for retrieval processes, alongside standard methods for retrieval of information. Though developing standards (e.g., ISO-STEP, IAI) can be used for the representation of much information about products, the majority of manufacturers do not maintain their information in such a form, and do not have tools to manipulate their data in this form. This paper investigates what toolkits will have to be developed for manufacturers to be able to migrate their product data into the form that would be required to operate within the ARROW system. Prototypes of these toolkits will be developed in association with several large manufacturers who will be providing the initial population of product data for the project. This project is in line with the Construct-IT implementation strategy [DOE96a].

Introduction

The overarching theme of this project is concerned with developing methods to improve the quality of buildings as a manufactured product, having recognised the need to encourage designers to design buildings using components that contractors can actually buy and which fit together. We wish to provide designers, contractors and maintainers of buildings with access to on-line database services, which contain reusable and validated objects populated with manufacturers' product information, allowing a design based on generic components to be developed and tested with components from various manufacturers.

The DOE has recently completed a scoping study for an industry knowledge base (IKB) for the construction industry [DOE96b]. This knowledge base is envisioned as a single point of entry to all information required by the construction industry, from news, journals, standards, codes of practice, practitioners and through to actual product information. The DOE have previously commissioned a demonstrator of such a system [Par96a] and have now committed to establishing an IKB for the UK construction industry. The DOE has funded projects to examine various technical and commercial aspects of an IKB. The ARROW project [Par96b] is one of these projects.

[1] The Department of the Environment, UK.

The concept of object oriented databases can exploit the work done on standards for exchange of product data (ISO STEP) [ISO94]. The EC COMBINE project [Aug95a,Aug95b], has demonstrated how the STEP methodologies can be used to develop a single project database for sharing data between different design disciplines. A Building Components Database (BCD) [Par95] was developed, by BRE, within the COMBINE project and linked with two architectural CAD systems across a network. This linkage allowed the project database to be populated using objects within the BCD. Hence demonstrating the potential for widespread data storage independent from particular applications. The ARROW project extends that work to encompass a selection of products from key suppliers and will encourage applications developers to use this database instead of each vendor developing their own, probably very functionally limited, database.

This paper presents the concept of a virtual, distributed warehouse of building and construction objects, the technology on which such a warehouse could be built, the proposed components and architecture of such a system, and a possible usage scenario. We distinguish between distributed object libraries (or databases) created, populated, and maintained by information providers, e.g. manufacturers; and the Building Object Warehouse (BOW), as a virtual warehouse system with tools and services to allow the interaction between the user and the databases. The user or user's application, will communicate directly with the BOW server. The BOW server will in turn, and invisibly to the user, retrieve appropriate product information from the manufacturer's databases.

Intended Users of the Building Object Warehouse

It is envisaged that several groups will use the building object warehouse, namely: designers; specifiers; facility managers; information originators (manufacturers, research organisations, testing agencies, etc.); and information providers.

Designers

Designers will use it to select appropriate types of material and building components to assess the viability of their design options. They will be interested to know about the performance of a particular component, its appearance, its size, its availability, cost impact, its compatibility with other design aspects, maintenance and serviceability. Furthermore designers will want to import the detailed drawings, details of installation and linkage with other components, etc. They might also want to evaluate the performance of their design by importing relevant data into their simulation or calculation system. They will have better quality and more accessible information on building components. Research during the BRICC project [URL2] suggested that the time spent searching for information outweighed time spent applying it at the rate of 80% to 20%.

Specifiers

They will want to obtain information on the size, availability, previous testing, specification, cost and compliance with relevant standards and regulations. They will require this information to be imported into their specification system.

Facility Managers

The use of more standard, validated components will improve forecast completion dates and costing and reduce operating costs. Better records of exactly which materials and components were used will improve safety and allow alterations and maintenance to be more predictable.

Information Originators/Providers

Manufacturers and other information providers will provide most of the above information; they will own, maintain and update the data. They will require tools to enter data in an easy to use manner and make it available on the internet with little extra effort. They will increase their competitiveness, save costs and improve their response to customers. The use and re-use of standard components will be increased. Manufacturers will be encouraged to prepare product information in a standard, neutral form, and their clients will have a wider choice of applications to use with that data.

All the above will benefit from enhanced productivity at design and construction stages, reduced rates of error during the construction stage and more importantly at the hand-over stage.

Usage Scenario

To give a flavour of the system's intended functionality consider the following usage scenario. A user using a design tool wishes to find a suitable product, say a window, and include it in his drawing. The user may be unaware of; a) exactly what he wants, b) what existing products are available, c) the design implications of particular choices. In addition, when he has clarified (a) to (c) he may be unaware of where to find the product information.

In these circumstances the user can use the BOW system to initially roughly specify what he wants. The system, through the query handler, will then prompt for more information (using its built-in knowledge of data structures and generic levels) to make the query more specific. It will also point to any relevant, existing knowledge-based systems (KBSs) known to BOW. These KBSs may be able to assist the user to be more specific and draw attention to the design implications of particular choices.

Once the user feels he has been as specific as he either, can be or wants to be, the request is passed on to the BOW product retrieval mechanism. This system could reside on the user's machine or on a central server. Its function is to analyse and reformulate the query in such a way that the required information can be found on the network of the distributed manufacturer's databases through the use of an intelligent retrieval system.

The designer is then provided with visual pictures of suitable windows. He can then choose to investigate its other attributes. If satisfied, he will choose one of the windows that satisfied his criteria and will press a download button to import the object. Hidden from the user, the data will go into the user's project database, integrity checks will be performed and then the drawings and design tools updated.

The Need for the Building Object Warehouse (BOW)

The recent focus on improved process and information flow between disciplines is creating the market demand for software applications to hold more detailed information. Manufacturers are willing to provide component data electronically, but there seems little agreement on format. Traditional product catalogue information is in a format that is difficult to use in computer aided design systems, difficult to transfer to the contractors, and it rarely reaches the users or operators of the buildings. The new services which electronically publish catalogues are typically scanned text and pictures, with the actual data required for selection and design of building components missing or in some proprietary format which cannot be imported directly into applications.

These difficulties contribute to the delay in the selection of appropriate materials and components until the construction phase. This stifles innovation, as few design alternatives are considered, and can lead to conflict on site and defects in operation.

Almost every designer and draftsman has to re-draw the objects, fill in tables, and specification, which normally is produced by manufacturers, research organisations, professional bodies etc. This leads to inefficiency and inaccuracy. Conversely, the use of validated, re-usable, component objects, could lead to greater productivity, more innovation, improved quality and savings at all stages of building life cycle.

In principle, the product database should be capable of storing all the information normally required of a product, including images and manufacturer's product advertizement. Hence (again in principle) once information and data have been generated by their originators in an electronic format, they should not have to be keyed in again during the design process.

Research Issues

There are several main challenges. The project is still at a very early stage and the following are still very much open issues.

Handling Manufacturer's Data

Manufacturer's product data is currently stored in many different formats. We need to devise a common product data model that is capable of containing all the necessary and relevant information about a product. Then we need to consider a system to translate existing information into this product data model.

As regards the data model, in general, a lot of diverse information will be required to cover all building components and materials. In addition to standard fields such as dimensions, cost, expected lifetime, etc., there will need to be extra fields containing product images and free-text fields for additional information not covered by the standard product model and of course, the manufacturer's promotional information.

The issue of translating existing manufacturers' product information relies on establishing a mapping between the manufacturer's terms and the product data model fields names. It is our intention to develop a toolkit, allowing the user a simple visual and tabular way of transferring data from a manufacturer's current format (which may be html pages, relational database, image files, etc., or a combination of these) into an acceptable format. That is, into appropriate fields in one of a number of relational databases. The toolkit will rely on the manufacturer being able to enter a suitable mapping.

During the evaluation stage it will be important to obtain good performance data. We envisage this may be difficult as there is no such thing as a representative supplier. The variety will be great.

For the data model there is also the issue of level of detail. Initially we want a generic model that becomes more detailed at a later stage in the project. This will allow us to implement a basic system first and also to use the experience of working with this system to clarify the detail required in the final data model.

Indexing verses Parallel Queries

After a query is formed the system must locate the relevant information from a variety of databases across the internet. There are two alternatives here: either the system sends out parallel queries to all registered manufacturer's databases, or we periodically index all their databases and query this index. We deal with each option in turn, first parallel queries.

This is the fully distributed way of doing things. Once a query is formed it is sent to each registered database. The advantage is that it minimises centralized services, relying on manufacturers' machines to perform the search. However there are problems with compilation of the query results and waiting for timeout of the distant non-responding machines. Set this time too long and users will become frustrated waiting for results, too short and products will be missed over slow connections. In addition there is the problem of all remote databases needing to be able to respond to the same query. This would require further software at the manufacturer's end capable of rewriting a query into a form understandable to their preferred database system. For any free-text searches each manufacturer would have to be responsible for indexing and regularly updating this index of its own textual information to prevent search times becoming large.

Now we consider central indexing. This overcomes many of these problems by removing much responsibility from the manufacturer, thus making the system more attractive to the manufacturer. But it comes at the expense of the need for a potentially large central index database. Although large, it would not need to contain images, free-text, or non-standard fields. The non-standard and free-text fields would need to be indexed separately. Any image or promotional material would only need to be retrieved immediately prior to presentation of query results to the user.

Relational Database Queries verses Text Database Queries.

It is easier for the manufacturer to store product data in free-text form, such as html pages, but far less efficient for rapid searching. In free-text searches there is no easy way to asertain context of a search term, thus a search for "window" will not be able to differentiate between references to windows in a building and those on a computer screen. The use of a relational database means accepting the constraint of fixed queryable fields in exchange for more rapid searches and less ambiguity. However the constraint of fixed fields makes it difficult to store extra information about a product. There will always be some products that require more fields, no matter how good the data model. It is therefore proposed that extra textual product information is stored in an 'other information' field as free-text that can be indexed separately. The issue then arises of when, during the process of answering a query, to search the text index. It would be reasonable to take the default position of only querying the text index when a particular query term is not present in the standard data model.

Intelligent Product Retrieval

Essentially we want the system to avoid the usual problem in index searching, i.e. either the system will return many dozens of hits (indication the search was too general) or zero hits (indicating the search was too specific). To do this we need to be able to automatically generalize or specialize a query. For example, widening an exact parameter to a range of values or instantiating an empty field to commonly chosen or default values. It must, of course, remain in the control of the user as to when to override this automatic process.

Initial Work

In this section we present some initial thoughts for the system architecture. Figure 1 is the proposed architecture for the Building Object Warehouse (BOW) system.

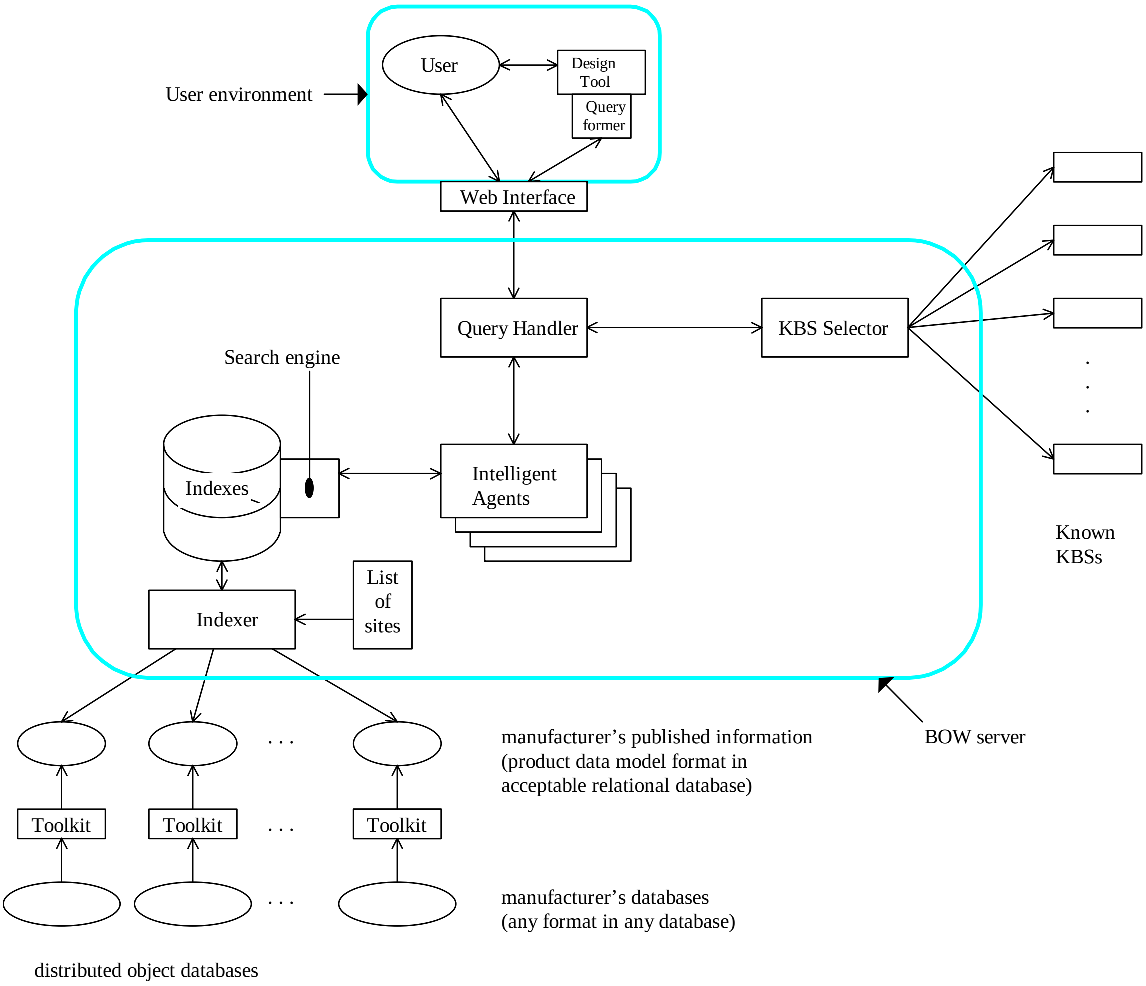

fig 1: BOW Architecture

The main parts of the system comprise the following. First, the user, who has an interface with the system directly through a web browser or indirectly through a design tool. Second, the main system or BOW server. Third, the distributed object databases at numerous sites across the Internet containing product data in a variety of formats. Finally, there is provision for interaction with existing KBS systems for product advice, design considerations, etc. Again these can be distributed across the Internet. All these sections are explained in more detail below.

User

It is intended that the user will be able to interact with the BOW system in one of two ways: either, directly through a web browser or alternately, through a design tool. In the former case the user will specify the initial query to the system, let us say "window: UPVC frame, double-glazed, width>3.20". In the later case the user may point to a wall and specify “window” with the design tool supplying the size parameters directly. In either case this initial query is then presented to the query handler.

Query Handler

After the initial query the user will interact directly with the query handler. The function of this is to help the user formulate exactly what he or she wants in terms of parameters used in the object model. The query handler contains knowledge of the product data model and so can prompt for parameters appropriate for a particular product type. There would be provision for unspecified parameters, parameters within a range and exact values only. In addition there can easily be provision for free-text searches of the manufacturer’s general product information.

As an example, say the user enters “door”. The product data model may subdivide doors into external and internal doors, which is then presented as a choice. If, say, internal is selected we may then be presented with a set of parameters to specify, let’s say; height, width, material, functional parameters, etc. Any combination of these parameters can be specified. (Parameters not specified at this stage may be instantiated with intelligent default values at a later stage by the intelligent agents if too many hits are retrieved.)

At any point in this process if the application specific KBS help is selected the current query is sent to the KBS selector, which will return a list of relevant KBS systems with which the user can interact directly. This may prompt the user to complete more parameter fields or even add extra parameters not suggested by the data model.

It should be noted that any word can be added as a search parameter. If a manufacturer has supplied an extra parameter not present in the formal object model then it can still be searched for. The function of the object data model is to provide a common language to make searching easier, but in reality it cannot cover all options a manufacturer wishes to specify and so there must be provision for manufacturers to add their own parameters and other general information. A problem that follows from the free addition of manufacturer’s parameters is one of several names for the same concept. An additional problem for the user is knowing what fields exist. The issue of suggestion of synonyms for search terms is dealt with later.

At any point the query formulation can be terminated and searching can commence. The “search” option sends the possibly semi-formed query to the intelligent agent module. The final act of the query handler is to expand all semi-formed queries according to the object product model hierarchy/network. That is, in the example, if the only search term was doors then this would be expanded to “internal doors OR external doors” and return all hits regardless of parameter values.

KBS Selector

The function of the KBS selector is to manage a list of known KBS systems relevant to the current query. These may be distant systems, systems on the BRE webserver such as Aggro (an aggrogates selection expert system [URL1]) or systems specifically built for BOW. It is intended that the KBS Selector will provide a gateway to these services that the user will then interact with directly. If useful information is gained from any such interaction then it will be up to the user to amend his currently semi-formed query before sending it off to the agent-based search mechanism. A simple algorithm for the KBS

Selector may be simply to match the words in the semi-formed query with a short description or set of keywords kept for every known accessible KBS.

Intelligent Agents

The function of the intelligent agent module is to retrieve a prioritized list of products. The agents take intelligent decisions to make the search either more general or more specific without the user's direct intervention, so the system always returns to the user a small positive number of hits, much as a human colleague would.

When presented with a problem where we are given a very wide range of choice, but with little time or information to differentiate products (a common situation in large supermarkets) we generally ask advice from someone we think we can trust. Depending on the degree of satisfaction we receive from their recommendation we then decide whether or not to consult that person again in the future. It is possible to use this technique within a machine where a community of users would each have his or her own agent. Associated with each agent is a list of previous successful retrievals. A clustering algorithm can then be used to form 'like minded' clusters based on the similarity of products they have previously retrieved. Let us imagine a user's query is presented to his or her agent. The agent queries the index and if the number of hits is a small positive number then the hits are immediately presented to the user. If, however, there are either a large number of hits, or zero hits, the agent can consult members of the like minded cluster for advice. Suggestions are sought for values of uninstantiated parameters, or parameter values to relax in order to achieve a list of hits within the specified range.

The use of user specific agents makes possible a synonym facility whereby users can enter synonyms to be recorded in their search agent when searches are not fruitful enough (especially useful for searching non-standard parameters). This then makes possible automatic suggestion of synonyms by the machine after consultation with fellow agents in its cluster.

The advantages of this system are:

1) Knowledge of current trends in product selection are stored specific to the type of product a particular user is choosing.
2) Ability to be able to deal intelligently with manufacturer supplied parameters extra to the strict product model.
3) The greater the number of users and the greater their use of the system, the better the referral mechanism will work.

Indexes, Indexers and the Search Engine

The indexer has an authoritative list of all product repository addresses. Periodically these are searched, all product parameters retrieved and indexed. It is envisaged that there are two indexes, one relational containing the minimum product information specified in the product model, the second for free text containing additional manufacturer information. The relational index itself can simply be stored in any large relational database. The free text index can be kept up to date with an indexer such as Harvest [HSW96].

Attached to the indexes is the search engine proper. This accepts the fully formed queries from the user's agent and presents the appropriate parts to the respective indexes.

Relationship to the IKB System

A recent project sponsored by the DOE developed a Construction Information Gateway (CIG) demonstrator [URL1], where the CIG is the public, implemented part of the IKB. The CIG demonstrator showed how the construction industry's knowledge base, in the form of product information, standards, research reports, test and certification information, cost, etc., can be accessed using the Internet's world wide web browser. The DOE is sponsoring a group to pave the way for commercial implementation or of the CIG proper.

It is our intention that the ARROW service is made available through the CIG. The CIG would effectively be just another user in figure 1, with its own web interface interacting with the query handler.

Summary

We have presented an overview of the issues concerned with efficient product data retrieval across a distributed network. The system is intended to be used by designers, specifiers and facility managers with the benefits of enhanced productivity and reduced rates of error at design and construction stages. We have also presented our initial thoughts on how to tackle the issues of a data product model and intelligent retrieval of products. It is our intention, using the outlined architecture, to build a demonstrator of the system that should be completed by March 1999.

References

[Aug95a] COMBINE-2 Final Report, CEC-DGXII, Brussels.

[Aug95b] An Overview of the COMBINE Project, The First European Conference on Product and Process Modelling in the Building Industry, Dresden, Germany, Oct 94, Scherer, R. (ed), Balkema, Rotterdam, pp. 547-554.

[DOE96a] Feasibility of a Library of Building Design Objects, Construction Sponsorship Directorate, Department of the Environment, UK.

[DOE96b] Construct-IT: bridging the Gap, Scoping study for the Construction Industry Knowledge Base, Construction Sponsorship Directorate, Department of the Environment, HMSO, UK.

[HSW96] Hardy, D. R., Schwartz, M. F. and Wessels, D., Harvest User's Manual, Technical report CU-CS-743-94, Dept of Computer Science, University of Colorado.

[ISO94] ISO/DIS 10303-1:1994. Industrial Automation Systems and Integration – Product Data Representation and Exchange Part 1: Overview and Fundamental Principles.

[Par95] Parand, F., Combine BCD Final report, CEC-DGXII, Brussels, Obtainable from BRE.

[Par96a] Parand, F., The Construction Information Gateway Demonstrator, CIBSE Journal, vol.

[Par96b] Parand, F., Advanced Reusable and Robust Objects Warehouse (ARROW), DOE PiT proposal, DOE, UK.

[URL1] Construction Information Gateway (CIG), http://cig.bre.co.uk/

[URL2] BRoadband Integrated Communication for Construction, (BRICC), http://www.hhdc.bicc.com/bricc.htm