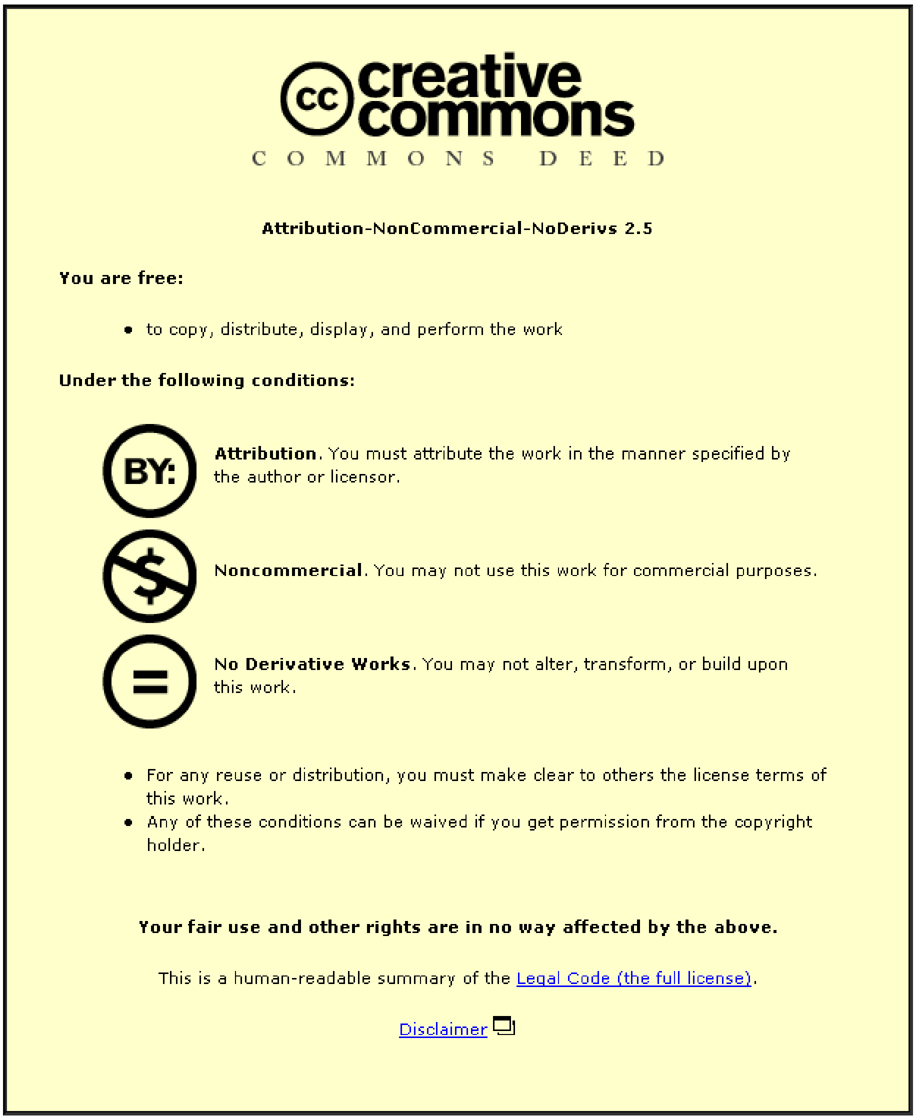

TOWARDS IMPLEMENTING INTEGRATED BUILDING PRODUCT LIBRARIES

A. A. Owolabi, C. J. Anumba, A. El-Hamalawi[1]

ABSTRACT: Electronic product catalogues and brochures are gaining popularity but there is little agreement on content, format and searching methods. This limits their usability and integration with existing construction software tools. This paper examines a product-modelling approach to delivering building product information and describes a proposed multi-tier client-server environment. ISO/STEP and IAI/IFC building product models are considered to facilitate representation, exchange and sharing of product information. The proposed architecture incorporates scalability with middleware components that would provide single or few points of entry to integrated product information. This paper is part of a research project, which builds on the results of related projects including ConstructIT Strategy, PROCAT-GEN, Active Catalog, COMBINE and ARROW, towards implementing the required software components.

Keywords: Building product modelling, Industry Foundation Classes, product libraries, data sharing, multi-tier client/server application.

INTRODUCTION

Construction product information is needed throughout the lifecycle of a construction project, from the feasibility study stage to demolition planning. Construction industry practitioners require complete and up-to-date information to assist in comparative evaluation and selection of products as well as for specification, installation, maintenance and disposal.

The major part of product information originates from product manufacturers, suppliers, fabricators and distributors. Research institutions occasionally add performance information. Users may also provide feedback, which influences future development or selection of the product.

Users’ expectation is that product information should be available, comprehensive, accessible and reusable in their software environments. Inadequate or unavailable information causes lost opportunity in sales to manufacturers whose products were not specified. On the other hand, repeated bad choices are costly to users. Poor accessibility results in wastage of time, and usability of the system hinges a great deal on the reusability of the information produced.

Product information is currently available from numerous information brokers who aggregate information about hundreds of products from several manufacturers and organise the information according to some classification system. In many cases, manufacturers also make their product information available directly. The information may be structured and presented in printed catalogues and brochures or in electronic formats on CD-ROMs, DVDs and online. There is little or no agreement on the contents, semantics, formats and search methods supported.

[1] Abidemi A. Owolabi, CICE, Loughborough University, Leicestershire LE11 3TU UK, A.A.Owolabi@lboro.ac.uk
Chimay J. Anumba, CICE, Loughborough University, Leicestershire LE11 3TU, UK, C.J.Anumba@lboro.ac.uk
Ashraf El-Hamalawi, Department of Civil and Building Engineering, Loughborough University, Leicestershire LE11 3TU, UK, A.El-Hamalawi@lboro.ac.uk

The characteristics of the construction industry and the limitations of the software technologies in use within the industry have limited the delivery and comprehensive use of building product information on construction projects. The prevalent contracting forms and fragmentation of the industry hampers sharing of product information at different phases of the building lifecycle. This is further complicated by the nature of buildings in that a building can be designed to incorporate products from different parts of the world and built with infinite combinations. In addition, the software tools used by project teams are incapable of exchanging or sharing data without some manual or electronic transcribing or translation.

Figure 1 is a simplified representation of the existing practices, highlighting the need for manual or automated translation or transcribing of product information between industry practitioners and their tools. Anumba et al. (2000) detail the complex communication requirements, which is also applicable to product information, within the virtual construction project team.

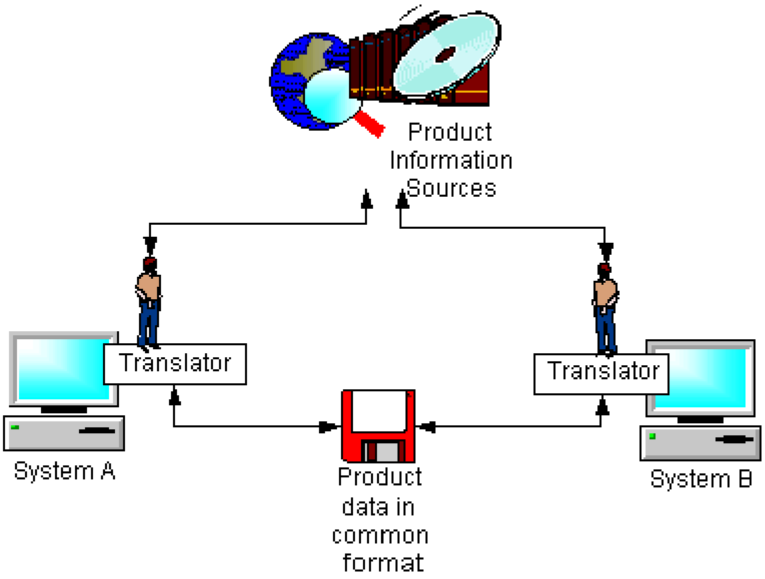

Figure 1 Existing method of obtaining and exchanging product information

In the UK, the construction industry is undergoing restructuring following the recommendations of Egan (1998). The dispersion of the industry is being addressed with partnering and team development. In addition, project teams are being encouraged to consider not only initial cost but also whole-life costs including opportunity costs and future costs; hence a “right first time” approach to eliminate waste and increase productivity. These changes require extensive sourcing and management of information, including product information.

Progressive international standardisation efforts defining industry-wide product models and specifications for exchange and sharing of product information have yielded implementable solutions addressing interoperability issues. The Industry Foundation Classes (IFC) being defined by the International Alliance for Interoperability (IAI), underpinned by International Standards Organisation’s Standard for the Exchange of Product Model Data (STEP), is most relevant. This is complemented by the IT standardisation efforts, including that of the World Wide Web Consortium (W3C) with eXtensible Markup Language (XML) specification, and Object Management Group with Common Object Request Broker Architecture (CORBA) specification.

This paper discusses a proposed architecture for the implementation and delivery of product libraries based on existing and emerging construction product models and open software technologies. It discusses current delivery methods, introduces the STEP and IFC standards, describes the proposed architecture for the implementation of integrated product libraries and highlights relevant past projects.

DEFINITIONS

The starting point to understanding what constitutes product information is to define what the product is. Generally, a product is something produced by human or mechanical effort or by a natural process. Construction products are items acquired, manufactured or processed, for incorporation into construction work as defined by International Standards Organisation (ISO, 1989) and British Standards Institute (BSI, 1991). This is similar to the International Alliance for Interoperability (IAI, 2000) definition:

> “A product is any object, manufactured, supplied or created for incorporation into an AEC/FM project.”

A project is understood here to mean some engineering activity leading to a product including objects that are created indirectly by other products. Hence an entire building is as much a construction product as a window frame.

Coyne et al. (1999) define *product information,* from the point of view of a designer, as

> “technical information about products, materials and finishes used in building and construction, including standardised components: doors, windows, taps, floor tiles, etc., which allow designers to make informed choices about alternative products, materials and finishes, and to know what attributes to specify, such as sizes, materials, finishes, colours, etc”.

The concept of product model captures all perspectives because product models represent the dynamic evolution of product information in the course of the design, construction and facility management processes.

> “A product model is a digital representation of a real world thing held to facilitate the unambiguous transfer of the information between computer systems and to support information sharing.” (Eastman, 1999)

Product information includes catalogue and technical information, specification, drawings, samples, costs and discounts, sources of supply and delivery, test reports, installation, operation and maintenance instructions, and disposal or recycling requirements. Performance information and user feedback could also be incorporated. Information required for a specific construction product would depend on the project phase, type of project and procurement route.

We define product library as a virtual or physical place, which contains systematically arranged digital information about physical, manufactured, supplied or created objects and provides access to users and owners of such information. It should support multiple classification systems, search methods and presentation formats.

RELATED PROJECTS

In order to support the highly distributed or dispersed construction product information sources, a standardised, flexible, and extensible solution with built-in scalability, robustness and platform neutrality is required (Zarli and Poyet, 1999). Standard interfaces, in the form of standardised data exchange, functional API to access shared data, common communication protocols and framework between applications, are also required to provide interoperability and an acceptable level of independence between the various subsystems. The reality is sources of product information are far from integrated and research efforts by different groups have attempted various aspects of the solution. Many of these projects achieved their set objective but an integrated solution still eludes us partly because all necessary standard interfaces and technologies were not available and partly because the developers assumed a utopia where all sources and users of product information would change or abandon their legacy systems, adopt a standard and invest in its implementation.

The architecture proposed here is designed as part of a research project towards developing a commercial strength integrated product library, building on findings of past feasibility studies and results of past research projects. This section describes some of these related projects including Construct IT Strategy, PROCAT-GEN, Active Catalog, RINET and ARROW.

Construct IT Strategy

Towards the implementation of the Latham report recommendations (Latham, 1994), the Construction Sponsorship Directorate of the Department of Environment, with industry representatives, carried out a study, which established the UK construction industry information requirements. Product information was recognised as principal to all industry participants and the requirements for a library of design objects that correspond to actual building products were laid out. The report (DETR, 1996), detailing the findings and recommendations, also considered the implementation of different search methods, classification of library objects, integration of existing tools and the applicability of standards including ISO/STEP, IAI/IFC, and CORBA.

PROCAT-GEN

Product Catalogue for Global Engineering Network Intelligent Access Library, (PROCAT-GEN) was part of the GEN Project, an European initiative with academic and industrial cooperation. It established the market potentials for electronic delivery of product catalogues and developed prototype catalogues within the integrated, platform-independent GEN infrastructure. The GEN infrastructure embraces the Application Service Provider (ASP) delivery model and specifies the implementation of search by keyword, classification codes, and parameters (Radeke, 1999). Java, XML and CORBA are some software development technologies utilised to support interoperability.

Active Catalog

The Active Catalog project, undertaken by Information Sciences Institute at University of Southern California, was aimed at improving the design environment by finding and using product information in building design applications within heterogeneous, internet-based, distributed computing environment (Cheng, 1997). The resulting prototype addressed the limitations of keyword search by implementing a knowledge-based search engine, which utilises domain knowledge in interpreting meanings of high-level user queries. Platform-independent data representation format, modular or component-based software design with well-defined interfaces and platform-independent implementation language are some of the attributes of the system used to facilitate reusability of information and interoperability across systems. CORBA and Java language were extensively used for their platform-independence and object-orientation.

ARROW

Advanced Reusable and Robust Object Warehouse (ARROW) project is a UK initiative carried out by Building Research Establishment and Visual Technology with support from Department of Environment (Newnham et al., 1997). It built on the COMBINE project (Augenbroe 1994) using an extended IFC model and purposed to implement a knowledge-based query handler with intelligent agents. The prototype application demonstrated integration with distributed manufacturer databases and CAD systems (Newnham and Amor, 1998; Amor and Newnham, 1999).

RINET

The RINET project is part of the VERA project being undertaken by VTT Building Technology, Finland (Hannus and Pirhonen, 2000). Like ARROW, it demonstrates the feasibility of an Internet-based, single point of entry to product information in distributed manufacturer databases. The prototype, implemented with C++, Java technologies and ODBMS, supported search by parameters, classification codes, and keywords. An authoring tool developed in C++ populates the product database from manufacturers' websites. A product data server along with an index database services user requests from Java applets running in Web browsers.

PRODUCT INFORMATION DELIVERY: CURRENT PRACTICES

This section explains current practices in electronic delivery of product information, particularly over the Internet. It highlights the possible search methods in Section 0 and provides a brief summary of the capabilities of existing systems in Section 0. The shortcomings of the systems were identified and solutions were proffered in Section 0. The rest of the paper addresses various parts of the solution.

Search Methods

Four search scenarios can be readily identified including search by keywords, classifications, parameters and navigation.

Search by Keyword

Keywords, or significant descriptive words can be associated with a product or group of products to initiate a database search, for example, a search for “window”. This method is widely used by Internet search engines for its ease of implementation and use. This method is however inadequate for construction products because keywords are language dependent and domain specific. Considerable user time can be wasted weeding out unwanted information from a search result. Virtually all the existing online product information sources implement this method.

Search by Classification

This method requires the use of some classification code corresponding to products of a specific grouping. A product can be mapped to multiple classification codes. Some examples of classification systems are:

- Construction Index/Samarbetskommitten for Byggnadsfragor (CI/SfB), a Scandinavian system of classification widely used in the UK;
- Construction Specification Institute (CSI) used in the US;
- Electronic Product Information Co-operation (EPIC) employed in Europe;
- Coordinated Classification System (CCS) in Australia; and
- BSAB 96 in Sweden.

Many online sources of product information provide at least one classification system depending on their target users. Others categorised products according to some arbitrary or in-house system.

Search by Navigation

This search method involves following hyperlinks from one product to related products, manufacturers or the industry knowledge base such as applicable standards, and performance information. It can provide fast access to all information relating to a product within user control due to avoiding information overload. Support for search by navigation is considerably limited in current implementations of online product libraries.

Search by Parameters

This method requires that specific functional characteristics, or requirements, be specified as attributes of the product being sought. Values can be restricted or relaxed to narrow or expand the search criteria. This is the most intuitive way to search for products because rarely would a designer begin a search for “door” without having some acceptance criteria or attributes to determine the suitability of the “door”. This method however is rarely supported because current implementations of online product libraries seek to replicate functionalities supported by the paper catalogues, which cannot support search by parameters. In addition, there is no agreed vocabulary among suppliers to facilitate search by parameters.

Product models provide definitions of properties for classes of product thereby providing common vocabulary for describing them and enabling the implementation of search by parameters. There is ongoing research in Artificial Intelligence towards utilising domain knowledge in improving search methods and using intelligent agents armed with appropriate ontology and conceptual vocabulary for the domain of interest (Clark et al., 2000; Obonyo et al., 2001).

Product Libraries on the Internet

Existing building product websites are built to service the local construction industries. Product information from suppliers within the country are hosted and categorised according to the classification system in use in the country. Even within the same country the information brokers maintaining the website do not utilise the same classification scheme or information model. The information provided is presented in varied formats as static HTML pages with text and pictures in GIF or JPG formats. For easy download, a PDF version is sometimes provided. A handful of websites provide CAD details in DXF or DWF formats for download and use in most CAD systems. VTT Construct IT research in Finland maintains a comprehensive link and brief description of building product libraries and related projects worldwide[2] and a review of search methods supported and classification system used is maintained at the Department of Architecture, University of Edinburgh[3].

Some websites have started delivering product information as program objects that could execute as Java applets or Visual Basic objects on the user's system. These objects attempt to encapsulate the geometry, cost

[2] http://cic.vtt.fi/links/prodlib.html

[3] http://www.caad.ed.ac.uk/~salih/prodlink.htm

information and some behaviour to enable them to reconfigure and re-render themselves in response to parameter changes (Emmerik, 2001).

Improving Existing Web-based Systems

There is a need to improve on existing web-based systems to facilitate the integration of the construction teams and provide product information support throughout the project life cycle. The shortcomings of the existing Web-based systems for delivering product information are identified as follows:

- There is no agreed content, structure and format for presentation of product information. This hampers exchange and sharing of product information for re-use between applications and project teams.
- They are specific to particular localities because of classification systems and languages employed. This is increasingly becoming an issue with international construction teams.
- The search methods implemented are inadequate because they are not tailored towards the way users obtain the information;
- There are too many unconnected sources, which prolongs the search process;
- They lack business continuity hence persistent access to product information because they are largely affected by changes in Internet technologies; and
- They are incapable of feedback and automatic updates because they are largely implemented with Internet “pull” technology where the users make all the requests for information without automatic updates or “push” technology where data is available once the user has registered interest.

All these limitations can be adequately addressed with existing standards for representation of building product data implemented in a multi-tier client/server environment with component-based, object-oriented technologies. Product data representation standards like STEP and IFC provide common definitions for building products. This can incorporate multiple classification system and facilitate advanced search methods. It can also facilitate integration of different sources of product information including supplier databases, manufacturer product data management systems (PDM), Enterprise Resource Planning (ERP) systems, and other product libraries, through the provision of open data representation format and Application Programmer Interface (API). A universal data format like XML can ensure business continuity when combined with open component-based software architectures, which facilitate pluggability, that is, the ability to build or scale a system by adding or removing components (Stevens and Pooley, 2000). Messaging components can be incorporated to support one-to-many push model communication of feedback and updates to subscribed clients. Subsequent sections describe the elements of this solution starting with representation standards.

REPRESENTATION STANDARDS FOR PRODUCT MODELS

Typically, buildings are operational for hundreds of years outliving the project team and software tools that may have been used in storing all information relating to them. There is also the need to harmonise or integrate product information from heterogeneous sources given the global nature of building products. Standards for the representation of product information address these needs by promoting interoperability across software platforms, hardware systems and providing support for sharing and exchange of information.

Data exchange is the process of extracting, transforming and delivering data, sometimes across disparate systems. It should be independent of data format, syntax, data source or target. Data sharing connotes common data sources serving multiple systems. Data exchange and sharing seeks to eliminate data re-entry, adoption or imposition of one system over another and direct translation. To achieve these, a standard for data sharing and exchange should incorporate metadata management, data transformation (or views), administration and implementations, and support for heterogeneous platforms and data sources.

There are industry standards that are widely adopted though developed by a single company or group (e.g. Rich Text Format (RTF) for documents), and there are *de facto* standards that originate from research and development projects (e.g. DXF for CAD). Open standards have wider reach and acceptability because they are products of large international communities of players with interests in the technologies and industries. They are the essence of interoperability, portability and reusability. Interoperability enables seamless integration of both internal and external systems, portability facilitates platform scalability and avoids proprietary limitations, and reusability prevents expensive translation, transcribing and redevelopment (Cumbers, 2001).

STEP and IFC are open standards resulting from the work of the International Standards Organisation and International Alliance for Interoperability respectively. They incorporate other open standards in their specification including XML specifications from the World Wide Web Consortium (W3C) and CORBA from Object Management Group (OMG). Some other industry standards have been developed to support file exchange for construction and other engineering disciplines. These include IGES, SET, EDIF, VDA-FS, and POSC (Fowler, 1995). All these standards have either been superseded by or integrated into the STEP standard. The Data eXchange Format (DXF) from AutoDesk remains the industry standard for construction engineering but with increasing software vendor support for IFC, an industry-wide acceptance can be expected in the nearest future.

Building Product Models

Product models provide formal and unambiguous (computer-sensible) representation of real world concepts like products and processes. They facilitate effective communication and seamless inter-working between disparate professionals by providing common terminologies, technologies, and ways of expressing and communicating information. They utilise an open data model, which provides common data representations to enable external programs to read and manipulate data. These models form the basis for automation, customisation, rich searching, and alternative interfaces (Myers, 1998). The need for a shared product model cannot be overemphasised given the fractured information systems prevalent in the construction industry, which has resulted in numerous problems for the industry.

ISO-STEP

The STandard for the Exchange of Product model data (STEP), otherwise known as ISO 10303 - Industrial Automation Systems - Product Data Representation and Exchange, is a family of specifications for the computer-interpretable representation of product data through the product's life cycle in a computing platform-independent manner (ISO, 1994). Product information including design, manufacture, use, maintenance and disposal can be described and stored in a neutral file exchange format to facilitate data sharing among dissimilar software applications.

The STEP architecture is layered with three logical levels namely; physical, logical, and application (Eastman, 1999). This separation distinguishes the logical structure of the information, or model, from the physical storage format and allows the definition of a subset of a complete model for a specific application. The standard utilises reusable constructs called classes, and it can be categorised into five structural groups as shown in Figure 2.

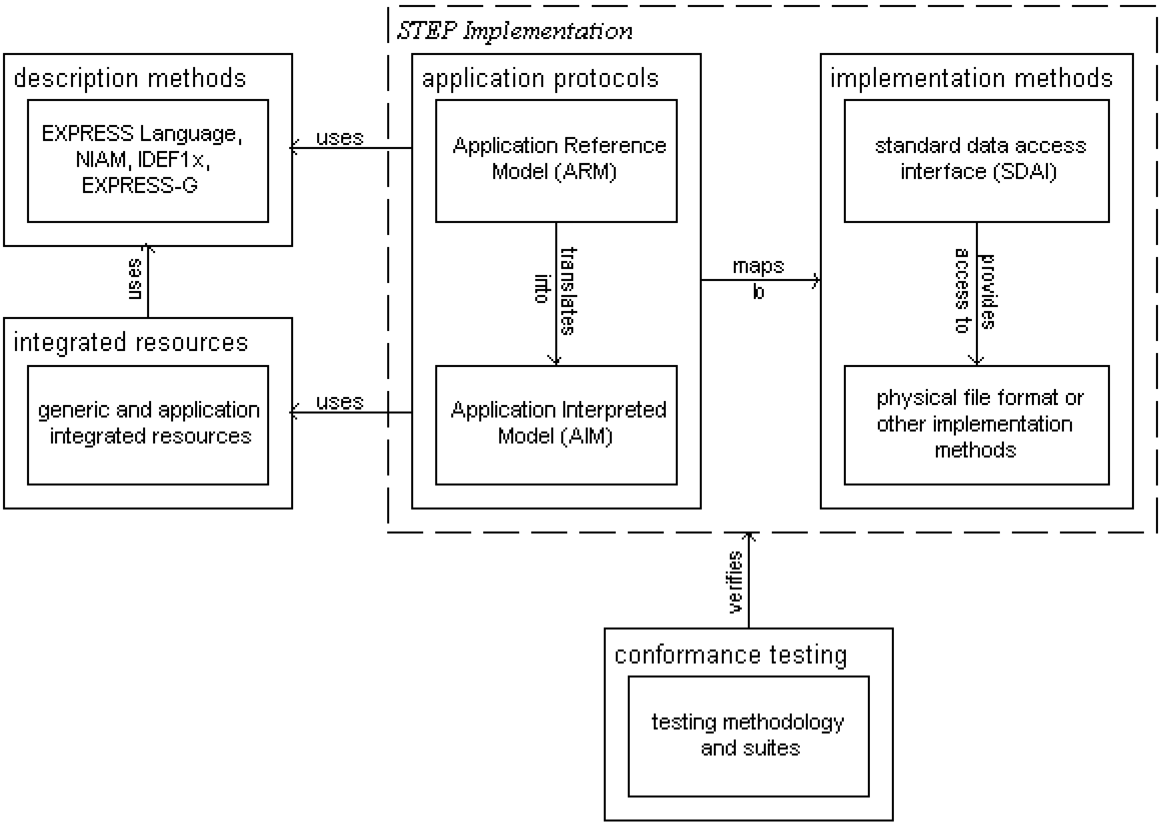
STEP Implementation
description methods
EXPRESS Language, NIAM, IDEF1x, EXPRESS-G
uses
application protocols
Application Reference Model (ARM)
translates into
Application Interpreted Model (AIM)
uses
integrated resources
generic and application integrated resources
uses
maps to
implementation methods
standard data access interface (SDAI)
provides access to
physical file format or other implementation methods
verifies
conformance testing
testing methodology and suites

Figure 2: Parts of STEP Architecture and their relationship

Description methods are information modelling languages used in specifying the other information models in the architecture. These include NIAM, IDEF1x, EXPRESS and EXPRESS-G. Integrated Resources are reusable conceptual model subsets that provide common requirements of product data for different application areas. Integrated Resources used in different domains are called Integrated Generic Resources, for example, geometry and material properties. Integrated Generic Resources can be extended into Integrated Application Resources to support the needs of a specific industry, for example, building, automotive, electronics, etc.

Application Protocols (APs) are implementable data specifications developed for particular application contexts using the Description Methods and Integrated Resources. It uses two types of models: Application Reference Model (ARM) and Application Interpreted Model (AIM). ARM specifies structures and constraints used to describe the information requirements of the application while the AIM is the interpreted version of the ARM. AIM contains a selection of integrated resources, constrained, specialised and completed to satisfy the information requirements of the ARM.

Application Protocols are combined with Implementation Methods, or standard implementation techniques, to form a STEP implementation. Implementation Methods include the STEP physical file exchange structure and Standard Data Access Interfaces, which provides an Application Programming Interface with language bindings for C, C++, Java and CORBA IDL. Conformance Testing provides conformance testing methodology, and framework and abstract test suites for testing implementations of various parts of STEP. This supports validation, interoperability, conformance, performance, robustness and acceptance tests.

STEP APs have been developed and applied successfully in many engineering disciplines including automotive, aerospace, shipbuilding, systems engineering and process plant particularly oil and gas. AP 225, Building elements using explicit shape representation, and AP 230, Building Structural Frame: Steelwork, were specifically targeted at the construction industry. The Building Construction Core Model (BCCM) was proposed as integrated resources for use by application protocols within building construction and for use in exchanging information between heterogeneous computer applications used by multiple disciplinary participants in the building construction domain but the project has been superseded by the IFC project.

IAI-IFC

The International Alliance for Interoperability (IAI) defines data structures, called Industry Foundation Classes (IFC) to support a shared project model for data sharing across applications specifically for the AEC+FM industry (IAI, 2001). IFCs represent building products (and abstract concepts like space, organisations, and processes) by describing their information requirements in a neutral computer language, the EXPRESS modelling language[4]. It therefore facilitates the exchange and sharing of information among the numerous homogenous and heterogeneous software applications employed in the multidisciplinary construction industry.

The IFC architecture provides a modular structure for the development of model components (IAI, 2000). Though it has a three-year release cycle like the STEP standard, it is more responsive to change due to its fewer architectural constraints and annual releases. The current release, IFC 2x, has four conceptual layers, including resource layer, core layer, interoperability layer and domain layer, representing the organisation of the model schema. Classes in each layer are organised to ensure modularity for reusability and ease of maintenance.

The resource layer consists of classes representing objects that are independent of application or domain need, for example, geometry, cost, and measure resources. The core layer classes represent concepts, relationships and roles that can be specialised for interoperability and specific domain requirements. The interoperability layer defines concepts and objects common across multiple applications or industry domains represented by the domain models. Finally, domain layer provides models tailored for a specific industry or application, for example, architecture, electrical, HVAC, or facility management.

ISO-PLIB

Basic to the management of product information is the management of information about and references to parts or components that constitute the product. The ISO 13584 – Parts LIBrary (PLib) standard, developed by the same ISO subcommittee for STEP, provides a series of specifications to facilitate unambiguous, application-independent representation and exchange of all technical data about parts. PLib reuses STEP capabilities for the representations of the parts defined in the PLib library and employs EXPRESS as the data specification language in addition to STEP implementation methods including STEP Physical File Format and SDAI. Pierra et al (1998) provides a detailed introduction to the PLib standard.

Implementing EXPRESS-Based Data Models

[4] IAI actually employ all the STEP description methods.

Common facilities for developing physical level implementations of EXPRESS models are provided in STEP SDAI and employed by the IFC specification. Both specifications have embraced XML as the physical format to replace the STEP physical format. Irrespective of physical formats or storage media, SDAI provides the same API for accessing the data repositories.

Four implementation levels, including passive file transfer, active file transfer, shared database access and integrated knowledge base, can be identified. Passive file transfer is the prevalent method and it involves the use of pre-processors for encoding and decoding the model data for the sending and receiving systems respectively. Active data transfer goes further by performing some analysis before sending. Shared database access extends the data exchange capabilities by storing the model data in an underlying database or repository for common access through a standard interface while the integrated knowledge base is envisioned to incorporate an intelligent design environment combining knowledge systems and artificial intelligence.

Implementing an EXPRESS data sharing system requires significant planning and software development effort. The system would need to provide functions for creating, destroying, accessing, updating and querying EXPRESS-data. It would model specific functions like translation, transformation, and rendering, in addition to low-level software functions like memory management and compilation. Tools for browsing model instances, validating constraints, visualising geometry and developing models would also be needed. Some tools already exist for implementing EXPRESS-data transfer and commercial strength EXPRESS-data repositories for shared database access are available from providers like EPM Technology, ProSTEP, STEP Tools Inc, and EuroSTEP Group.

IMPLEMENTATION OF INTEGRATED PRODUCT LIBRARIES

Product libraries must be implemented as an overlay to STEP or IFC standards because they support the product data modelling approach and are well suited to the needs of integration and management of the information resources over the whole life cycle (Wittenoom, 1998). Wittenoom (1998) predicted that technical and product information would proliferate and move from paper-based to on-line presentation. It was also predicted that there would be an increasing trend towards component software solutions in the “open” marketplace brought about by initiatives such as IAI/IFC.

It is worth reiterating at this point that an efficient implementation of product libraries must be built around the Internet as the communication infrastructure. It must also incorporate standardised product models and capabilities for organising product information effectively by different classification systems and representing the information in varied formats and languages. The system should be capable of interfacing with existing industry software systems, product data management systems and other sources of product information. In addition, it should provide an intuitive interface, be easy to set up and start up with minimal investment in hardware, software and training. The system should, as much as possible, be resilient to the ever-changing software technologies in order to secure users’ buy-in and justify its development. This section discusses the necessary framework and business technologies for implementing such a product library according to the logical components and delivery model. It proposes a multi-tier client/server architecture delivered as an application service.

Enterprise Client/Server Model for Product Libraries

In a logical client/server-computing environment, one or more machines act as server, providing shared resources, some application logic and presentation, and one or more machines act as clients providing some application logic and user interface. Since servers can connect to clients and other servers, and clients can similarly communicate with multiple servers and other clients, it is easier to describe the architecture by the basic elements of the application identified as presentation, logic and data; rather than by functions performed by machines. Shan and Earle (1998) identify five client/server topologies partitioned according to the three functional elements and the task division between the computers designated as server or client. The product libraries must incorporate all five topologies to cater for all usage scenarios as depicted in Figure 3.

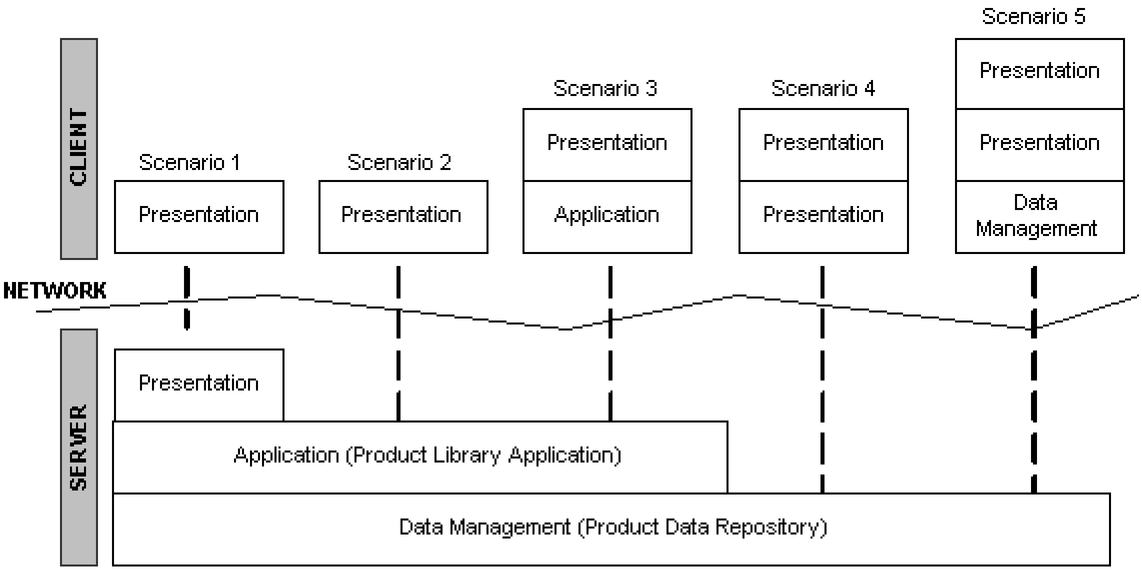

Figure 3 Client/Server Topologies for Product Libraries

Scenario 1: a client searches for product data from a basic Web browser with minimal rendering capability.

Scenario 2: a client using a dedicated GUI application, perhaps running in CAD software, browses the product library.

Scenario 3: a client accesses the product library from a Web browser capable of interacting with product objects or from the GUI application capable of rendering an entire building composed of product objects.

Scenario 4: a client (e.g. supplier) running an application capable of interacting directly with the product data repository to add or update restricted product information.

Scenario 5: an integrated project database interacting with the project data repository to obtain information about products applicable to a project or updates on products being incorporated into a project.

To support the above scenarios, a multi-tier client/server architecture is required. This architecture clearly demarcates the presentation layer, logic or middle layer and data layer by providing a number of middle layer components, or middleware, for accessing one or more data layer systems and presenting the data obtained through presentation layer components. It is based on an open system, which is a vendor-independent computing environment consisting of commonly available products that have been designed and implemented in accordance to standardised APIs and protocols. In addition, it leverages component-based and object-oriented software development to offer:

- Portability of application across operating systems and hardware platforms;
- Interoperability across networks and application and computer vendors;
- Flexibility through ability to take advantage of new technologies and changing technologies;

- Scalability to suit different needs and number of users with little or no change to underlying applications; and
- Business continuity for both application provider and users with continuous and easy rollout of upgrades as new business processes, standards and technologies emerge.

This system can adequately support multiple presentation formats and integration with manufacturers product data management systems and other product data repositories.

Figure 4 presents the proposed multi-tier architecture for delivering integrated product libraries. The data layer, Product Data Repository, captures all the data and translation components. The middle layer, Product Library Application, contains the core components of the application providing extensible functionalities while the presentation layer consists of components for interfacing with Product Library Clients. The complexity and dynamism of the system discourages the use of simpler, one-tier or two-tier, architectures in which two or more of the logical functions are closely coupled. Shan and Earle (1998) provide more detailed discussion of client/server architectures.

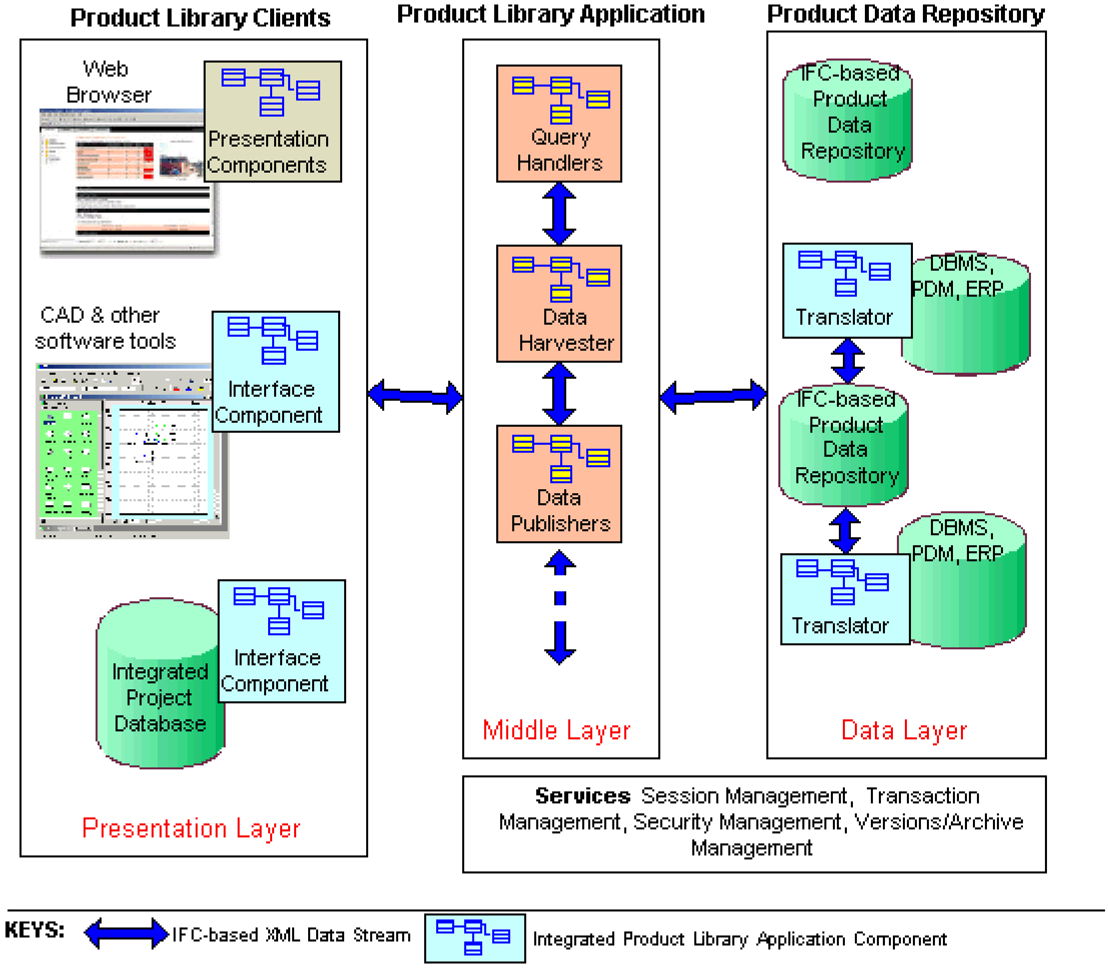
Product Library Clients
Product Library Application
Product Data Repository
Web Browser
Presentation Components
CAD & other software tools
Interface Component
Integrated Project Database
Interface Component
Presentation Layer
Query Handlers
Data Harvester
Data Publishers
Middle Layer
IFC-based Product Data Repository
Translator
DBMS, PDM, ERP
IFC-based Product Data Repository
DBMS, PDM, ERP
Translator
Data Layer
Services Session Management, Transaction Management, Security Management, Versions/Archive Management
KEYS:
IFC-based XML Data Stream
Integrated Product Library Application Component

Figure 4 Proposed Component-Based, Object-Oriented Multi-tier Client/Server Architecture for Implementation of Product Libraries

Data Layer

The data layer, also called the data tier, provides the information infrastructure. It would provide XML streams of IFC-based product information extracted from manufacturers’ and suppliers’ product data management (PDM) systems, database management systems (DBMS) and enterprise resource planning (ERP) systems. XML is preferred as the data-streaming format because it is largely technology independent and it supports on-the-fly content validation, transformation and presentation necessary to support all the identified application usage scenarios. Updates to the repository would be automated to run periodically, initiated by the middle layer Data Harvester or manually from the source. Where the data is not modelled around IFC, translator components would be deployed to map and transfer the data into XML for storage in the repository.

XML is a simple subclass of the Standard Generalised Markup Language (SGML, ISO 8879), which is fast becoming the standard for information interchange on the Internet and among application. It provides facilities for syntactic validation of documents against formal rules and defines a robust character syntax for representing structured data objects (Kimber, 1999). XML is more widely acceptable than the STEP physical file format, which has been adopted by IFC standards. The IAI defines ifcXML language bindings that would translate EXPRESS Schema to XML Schema and integrate with earlier efforts including the EU-funded Building Construction XML (bcXML) and aecXML initiated by Bentley in North America (Liebich, 2001).

Traditional data models and systems have used relational, network and hierarchical databases but object database management systems (ODBMS) are more suited to engineering applications because they offer flexibility that is otherwise not available. Though ODBMS are easy to use with IFC specifications and modern programming languages, which use the object-paradigm, XML data servers are preferred because they eliminate the need to reformat data for communication between components in the middle and presentation layers. An XML data server is a system usually built on a DBMS, that is capable of managing and delivering XML data. Examples of XML data servers include MS SQL Server 2000 from Microsoft and ObjectStore from Object Design. EXPRESS Data Manager from EPM Technology and ST-Repository from STEP Tools Inc. also provide XML support for querying and updating STEP databases.

Middle Layer

The middle layer consists of components and software tools, which facilitate building and running of multi-tier applications. An extensive middleware infrastructure is required to enable seamless integration and communication that encapsulates all user interactions with the product library as in the identified scenarios.

Some of the required middleware components can be identified as Query Handler, Data Indexer and Data Publisher. The Query Handlers would interpret user queries derived from different search methods and return the result to the appropriate presentation component.

The Data Indexer would subscribe to product information sources and capture published data through the Translator interface components. It would be implemented as a message-oriented component in a publish-subscribe model. In contrast to traditional remote procedure call (RPC), which may require periodic polling of sources of product information, this has the following advantages:

- It supports a failsafe architecture with loose coupling of remote subsystems like product information sources hence allowing for system crashes and scheduled downtimes;
- It provides a more flexible system architecture capable of supporting dynamic addition and removal of product information sources through subscriptions to the product information channel;
- It gives the suppliers of product information control over when to make their data, available for public viewing.
- It provides immediate access to product information while building on the information broker method of publishing product information.

Similarly, the Data Publisher would be a message-oriented component allowing client applications like other product libraries, project databases, GUI applications, etc. to register interest in specific items of product information through their Interface Components. It would publish notifications, which may include the required information, to all subscribers. Additional components may be added to model business logic and services in a business-to-business request/response interaction such as Request for Information (RFI) and Request for Quotation (RFQ).

The choice of middleware architecture that can support the development of the above architecture, with synchronous RPC and object-oriented methods invocation and asynchronous messaging in point-to-point and publish-subscribe model, is currently limited to Microsoft COM+, Enterprise JavaBeans (EJB) and CORBA Component Model (CCM). The actual choice may be limited to EJB and CCM, because they are open standards with many implementations, rather than COM+, which has only one implementation supported by the owner, though it runs on the pervasive Windows OS. Stal (2000) provides a detailed comparison of CCM, EJB and COM+ platforms.

Microsoft COM+ is a Windows-based, language-independent middleware technology for building reusable components based on the Distributed Component Object model (DCOM), which specifies the structure, API and communication protocols for the components. COM+ provides middleware services such as transaction, security, and messaging. The Microsoft .NET framework improves on this with a common language runtime, which supports unified programming classes to provide a multi-language environment for building, deploying and running Web applications. Morris (2001) highlights the features of .NET.

Enterprise JavaBeans (EJB) architecture is the cornerstone of Java 2 Platform, Enterprise Edition, which is an open standard for deploying cross-platform distributed application. It comprises APIs for accessing distributed computing and network services including database systems, enterprise name and directory services, enterprise messaging systems, email systems and CORBA services. Monson-Haefel (2000) provides in-depth tutorial on developing EJB components.

Common Object Request Broker Architecture (CORBA), from Object Management Group, is an open middleware specification for distributed applications. It is independent of programming languages, operating system platforms, communication protocols, and hardware. The CORBA Component Model (CCM) extends the CORBA object model by defining features and services that enable application developers to implement, manage, configure and deploy components that integrate CORBA services such as security, transaction, persistence, and event notification in a standard environment. An overview of CCM can be found in Heineman and Councill (2001). CCM is modelled closely to the EJB specification and provides standard mappings to both EJB and COM+.

Presentation Layer

The need to support many interfaces is obvious when product information usage scenarios are considered. Presentation components would be required to deliver textual and graphical product information in web pages. They would also attempt to model product characteristics as much as can be supported by web browsers using Java applets or ActiveX objects for example. Interfacing with heterogeneous construction software tools would leverage the increasing IFC supports in those tools. In addition, project databases would be interfaced with the system to improve productivity by facilitating automated two-way communication between the databases and the product libraries.

Application Service Model

Given the expected complexity of the application, it would be costly to implement by individual manufacturers. Not only would the hardware, software and IT staff requirement be prohibitive but it would also be a distraction from their core competence. Managing interaction between multiple sources would increase the complexity of the system and affect its productivity. Augenbroe (1998) recognised that a system that provides a single point of entry to all product information irrespective of location, yet supporting transparent awareness of the original sources and costs involved with the sources is the ideal. Radeke (1999) agrees and recommends the application service provider (ASP) model for delivering that solution.

An ASP is an organisation that deploys, hosts and manages software applications and data on its own server. Users access the applications over the Internet mostly using the Web browser. This enables manufacturers and suppliers to host their product information for a fee with the ASP and users to access the information from a single source. As mentioned earlier, existing manufacturers and suppliers do not have to change their systems as automatic translator components can be provided along with the service. The implementation of the system and the provision of the services can be adopted by an existing product information broker as an additional service or new product.

In defence of ASP computing for AEC industry, Unger (2001) identified lower total cost of ownership as the main reason why ASP will dominate the industry. This model does not only have the potential to lower cost and increase productivity but also to:

- Reduce time-to-benefits ratio for manufacturers and suppliers;
- Eliminate problems of upgrades to applications and hardware;
- Offer cost predictability; and
- Encourage global industry standardisation.

ASP models are gaining popularity with the evolution of e-business and the maturity of network infrastructures. The ASP Consortium is addressing security and other issues regarding this model.

SUMMARY AND DISCUSSION

The identified shortcomings of current methods of delivering product information and past projects can be classified according to their limitations in sourcing or acquiring product information, searching capabilities, level of integration with existing and emerging construction tools and mode of delivery. The proposed architecture, by employing product-modelling approach and multi-tier architecture including translation service for legacy data, component-based, message-oriented middleware, and multi-scenario (and possible multimedia) access, addresses these limitations and offers additional advantages as enumerated below:

1. It grants manufacturers, suppliers and other providers of product information the control over their information allowing them to make the information available when they desire thereby supporting current processes rather than imposing a drastic changeover procedure.
2. It supports enterprise application integration through interface/translation components provided as a service thereby preserving the benefits of past investments in hardware and software.
3. It improves efficiency and increases productivity in publishing product information.
4. It can support versioning and archiving of published product information thus ensuring persistent access to the rather volatile product information.
5. It incorporates feedback thereby supporting learning. Users can learn from others’ experience regarding a product and manufacturers can obtain necessary information for future improvements.

6. By separating business logic and application components from presentation components, it can support multiple usage scenarios and multimedia access as may be required in the future.
7. Longevity, reliability and maintainability of the product library are better assured through the adoption of open standards and a centralised approach to upgrades through the ASP model.

This paper has provided a high-level description of the proposed system from the product models to the architectural overview. It has highlighted the components that would comprise the system and how they would interoperate to achieve the improvements sought. There are however some barriers and technological constraints, which have to be overcome to make the architecture a reality. The industry comprises many small players to whom the cost of entry, usage and learning new systems, however small, is unsettling. Content providers may also be hindered by cost and the need to differentiate their products, which is undoubtedly limited by standardisation. Representing all required attributes of all products from all suppliers so they can be found through a single search that traverses local and distributed sources (mapping classifications systems as it progresses), and aggregates the result for multilingual, multimedia environments would be a formidable technological challenge. There is a parallel project looking into the barriers and enablers of the commerce aspect[5] while this project continues to address the technological constraints.

CONCLUSION

It is evident that for product libraries to be acceptable to suppliers and users of product information they should be globally accessible, affordable and offer business continuity. In addition, they should be flexible, integrate with existing and emerging construction industry tools. This paper has described the necessary infrastructure for implementing product libraries that meet users' expectations, including existing software architectures and supporting technologies, standards for representation of whole-life information of construction products, and application delivery model.

Some of the requirements for product libraries can be gathered from past projects and feasibility studies. However, additional information is required to cover all functional and non-functional requirements, which would ensure user satisfaction and take product libraries away from simply creation and acquisition of product information into sharing, exchange and modification of this information to reflect new knowledge and insight, hence intelligence. The next phase of this project would refine the requirements through involvement of typical end-users. Subsequently prototypes would be developed, tested, and refined towards developing commercial-strength integrated product libraries.

REFERENCES

Amor, R. and Newnham, L. (1999), CAD interfaces to the ARROW manufactured product server, in G. Augenbroe and C. Eastman, eds, Proceedings of the Eighth International Conference on Computer Aided Architectural Design Futures [ISBN 0-7923-8536-5] Atlanta, USA, Kluwer Academic Publishers, pp. 1-11.

[5] Project titled *Business Process Implications of eCommerce in Construction*, being undertaking in Centre for Innovative Construction Engineering, Dept. of Civil and Building Engineering, Loughborough University.

Anumba, C. J., Bouchlaghem, N. M., Whyte, J. and Duke, A. (2000), ‘Perspectives on an integrated construction project model’, International Journal of Cooperative Information Systems **9**(3), 283-313.

Augenbroe, G. (1994), An overview of the COMBINE project, in R. J. Scherer, ed., First European Conference on Product and Process Modelling in the Building Industry, Dresden, Germany. Copy available at http://erg.ucd.ie/combine/papers.html.

Augenbroe, G. (1998), Building product information technology, Executive white paper, Georgia Institute of Technology, Construction Research Center, Atlanta, GA 30332-0159.

BSI (1991), Glossary of building and civil engineering terms. Technical Report BS 6100-2.6:1991, British Standards Institute, London, UK.

Cheng, H. H., ed. (1997), Active Catalog: Searching and Using Catalog Information in Internet-Based Design, Sacramento, California. Copy available at http://www.isi.edu/active-catalog/asme-97.ps.

Clark, P., Thompson, J., Holmback, H. and Duncan, L. (2000), Exploiting a thesaurus-based semantic net for knowledge-based search, in 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI’2000) [ISBN 0-262-51112-6], Austin Texas, USA, AAAI Press (Distributed by MIT Press), California, USA, pp. 988-995.

Coyne, R., Lee, J., Duncan, D. and O S. (1999), Managing product information on the world wide web, Technical report, Department of Architecture, University of Edinburgh, UK. A report from an EPSRC - Multimedia and Networking Applications Programme, sponsored project, available at http://www.caad.ed.ac.uk/Coyne/ProductCatalogReport/.

Cumbers, S. (2001), ‘Open source for open government’, Visual Systems Journal October, 14-15. Part published as Letter to the Editor in The BCS Computer Bulletin, September 2001, pp. 5-6.

DETR (1996) Construct-IT: bridging the Gap, Scoping study for the Construction Industry Knowledge Base, Construction Sponsorship Directorate, Department of the Environment, HMSO, UK.

Eastman, C. M. (1999), Building Product Models : Computer Environments Supporting Design and Construction, CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431, USA. ISBN 0-8493-0295-5.

Egan, J. (1998), Rethinking construction, Technical report, Department of Environment, Transport and the Regions, HMSO, London. 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.

Emmerik, M. V. (2001), Product|Center: An integrated environment for selection, configuration and procurement of building materials via the internet, Technical report, Bricsnet, Portsmouth, USA. Copy available at http://corporate.bricsnet.com/about/documentation/ProductCenterWhitepaper.pdf.

Fowler, J. (1995), STEP for Data Management Exchange and Sharing, Technology Appraisals Ltd., Twickenham, UK. ISBN 1-871802-36-9.

Hannus, M. and Pirhonen, A. (2001), ‘RINET - building product database’, Project descriptions and demonstrations are available at the project website: http://cic.vtt./projects/rinet/index.html. Last visited: 12/12/01.

Heineman, G. T. and Councill, W. T. (2001), Component-Based Software Engineering, Addison-Wesley; ISBN 0201704854. Chapter 38 - An Overview of CORBA Component Model.

IAI (2000), IFC Technical Guide, International Alliance for Interoperability. Copy available at http://www.iai.org.uk/documentation/IFC 2x Technical Guide.pdf.

IAI (2001), ‘Introduction to International Alliance for Interoperability’, available at IAI UK Chapter website http://cig.bre.co.uk/iai uk/, last visited: 12/12/01.

ISO (1989), Building and civil engineering vocabulary - Part 1: General terms, Technical Report ISO 6707-1:1989, International Standards Organisation, Geneva, Switzerland.

ISO (1994), Industrial automation systems and integration - product data representation and exchange - part 1: Overview and fundamental principles, Technical Report ISO 10303-1, International Standards Organisation, Geneva, Switzerland.

Kimber, W. E. (1999), XML representation methods for EXPRESS-driven data, Technical Report GCR 99-781, Department of Commerce, Technology Administration, National Institute of Standards and Technology (NIST), USA. Available online at http://www.nist.gov/sc4/wg qc/wg11/n095/nistgcr99-781.pdf.

Latham, M. (1994), Constructing the team, Technical Report ISNB 0-11-752994-X, Department of the Environment, UK.

Liebich, T. (2001), XML schema language binding of EXPRESS for ifcXML, Technical Report MSG-01-001(Rev 4), International Alliance for Interoperability. Copy available at http://www.iai.org.uk/IFCXML.htm.

Lockley, S., Rombouts, W. and Plokker, W. (1994), The COMBINE data exchange system, in European Conference on Product and Process Modelling in the Building Industry, Dresden, Germany.

Monson-Haefel, R. (2000), Enterprise JavaBeans, 3rd edn, O’Reilly Publishers, UK. ISBN 0596002262.

Morris, P. J. (2001), ‘What’s good about .NET?’, Visual Systems Journal, December pp. 14-17.

Myers, B. A. (1998), The case for an open data model, Technical Report CMU-CS- 98-153, Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA. Copy available at http://reports- archive.adm.cs.cmu.edu/anon/1998/CMU-CS-98-153.pdf.

Newnham, L. and Amor, R. (1998), Translation of manufacturer’s product data for the ARROW product search system, in ECCPM 98 - The Second European Conference on Product and Process Modelling in the Building Industry, BRE, UK, Building Research Establishment (ISBN 1-86081-249-X), pp. 405-412.

Newnham, L., Parand, F., Amor, R. and Nisbet, N. (1997), The ARROW framework for a building object warehouse, in R. Drugemuller, ed., CIB W78

Workshop on Information Technology Support for Construction Process Re-Engineering, IT-CPR 97, International Council for Building Research Studies and Documentation, pp. 319-328.

Obonyo, E., Anumba, C. J., Thorpe, A. and Parkes, B. (2001), Agent-based support for electronic procurement in construction, in M. Mohammadian, ed., International Conference on Intelligent Agents, Web Technologies, and Internet Commerce (IAWTIC), Las Vegas, USA, University of Canberra, Australia [ISBN: 0858898470], pp. 268-279.

Pierra, G., Sardet, E., Potier, J. C., Battier, G., Derouet, J. C., Willmann N., and Mahir, A., Exchange of component data : the PLIB model, standard and tools, in CALS Europe '98 Conference, Paris, France, pp. 160-176.

Radeke, E. (1999), Final report of global engineering networking intelligent access libraries (GENIAL), ESPRIT Project 22.284, Technical report, The GENIAL Consortium.

RIBA (2001), ‘The 2001 survey on information services in the UK construction industry’, RIBA Companies Ltd., London. Copy available at http://www.ribac.co.uk/images/survey2001.pdf.

Schenck, D. and Wilson, P. (1997), Information Modeling : The Express Way, Oxford University Press, UK. ISBN: 0195087143.

Shan, Y.-P. and Earle, R. H. (1998), Enterprise Computing with Objects: From Client/Server Environments to the Internet, Object Technology Series, Edited by Grady Booch, Ivar Jacobson and James Rumbaugh, Addison-Wesley, USA.

Stal, M. (2000), Component technologies for the middle tier: CCM, EJB, COM+, in ACM Conference on Object-Oriented Programming, Systems, Languages and Applications, Minneapolis, Minnesota USA, OOPSLA 2000.

Stephens, J. (1999), Virtual enterprises using groupware tools and distributed architectures: Public nal report, Technical Report PFR-01, ESPRIT 20408 - VEGA.

Stevens, P. and Pooley, R. (2000), Using UML - Software Engineering with Objects and Components, Object Technology Series, updated edn, Addison Wesley Object Technology Series, ISBN 0201648607.

Unger, S. (2001), ‘Why ASPs will dominate AEC’, ConstrucTECH vol. **4**(No. 10). copy available at http://www.constructech.com/printresources/v4n10/v4n10034.htm.

Wittenoom, R. (1998), Automating realization of integrated project models, Technical report, AusSTEP Consultants, Dalkeith, Australia URL: www.ausstep.com.

Zarli, A. and Poyet, P. (1999), Distributed architectures and components for the virtual enterprises, Technical report, Centre Scientique et Technique du Batiment (CSTB), Sophia Antipolis, France (zarli/poyet@cstb.fr).