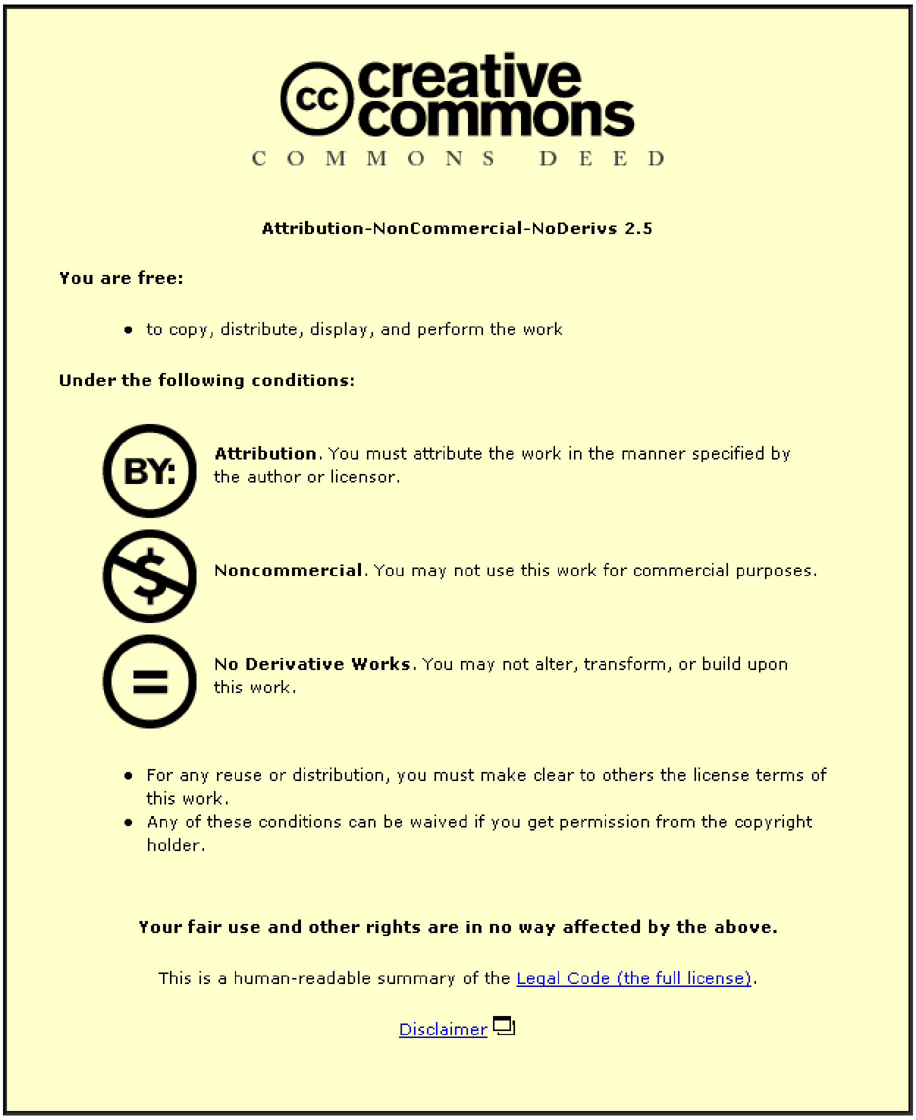


ARCHITECTURE FOR IMPLEMENTING IFC-BASED ONLINE CONSTRUCTION PRODUCT LIBRARIES

SUBMITTED: November 2002
REVISED: March 2003
PUBLISHED: July 2003 at http://www.itcon.org/2003/15
EDITOR: Väino Tarandi

Abidemi Owolabi, EngD Candidate
Centre for Innovative Construction Engineering, Loughborough University, UK
email: A.A.Owolabi@lboro.ac.uk

Chimay J Anumba, Professor
Dept. of Building and Civil Engineering, Loughborough University, UK
email: C.J.Anumba@lboro.ac.uk

Ashraf El-Hamalawi, PhD
Dept. of Building and Civil Engineering, Loughborough University, UK
email: A.El-Hamalawi@lboro.ac.uk

SUMMARY*: Construction product information providers have responded to the demand for electronic delivery by providing online access, CD-ROMs and DVDs but these solutions have limited usability and are generally incapable of supporting prevalent and emerging industry practices. The product library implementations attempt to replicate the functionalities of the paper versions, which serve for independent specification and procurement but gives little thought to teams and tools integration through support for automated information exchange and sharing. The IFC standard provides common terminologies, technologies, syntax and semantics necessary to address present and future compatibility and integration issues, hence IFC-based implementation of product libraries have good prospect for meeting the industry requirements. This paper reviews current product information delivery methods and examines the applicability of the IFC and other standards. The requirements for IFC-based construction product libraries are identified and an architecture for realising the requirements was presented.*

KEYWORDS*: IFC, product libraries, integration, data exchange and sharing, multi-tier architecture.*

1. INTRODUCTION

Printed catalogues and brochures are widely used within the construction industry as the primary source of product information to facilitate the specification and procurement of products. Useful product information including design, properties, costs, availabilities and suppliers detail can be garnered from these periodic publications and employed in the construction process. The industry practices, influenced by information and communication technologies (ICTs), have created demands for electronic versions of the information. Product information brokers, manufacturers and suppliers have responded by providing online access, CDs and DVDs.

Electronic product catalogues attempt to replicate the functionalities of the printed ancestors and have achieved that goal and slightly exceeded it but paper catalogues are still more popular with industry practitioners (RIBA, 2001). This is an aberration particularly when compared with the rate of uptake or success of other electronic tools employed in the industry. The reason for this could be the traditional slowness of the construction industry to embrace new practices or the limitations of the implementations.

This paper focuses on usability problems of the existing systems by identifying the shortcomings and examining possible solutions. It goes further to propose an application architecture and to examine implementation issues for a prospective solution.

Section 2 provides definition of terms. Section 3 summarises the limitations of existing system and section 4 highlights the part standardisation can play in addressing the identified problems. Section 5 justifies the product-modelling approach and the choice of IFC. It backs this with a cursory look at other local and international projects based on the product modelling approach. In order to highlight some implementation requirements, sections 6, 7 and 8 review the IFC architecture, classification systems and search methods respectively. The systems requirements are detailed in section 9 while section 10 describes two application architectures. The proposed implementation architecture is presented in section 11 and the conclusions in section 13.

2. DEFINITIONS

The following definitions have been adopted or derived for the purpose of this paper:

- A *product* is any object, manufactured, supplied or created, for incorporation into an AEC/FM project where a project is some engineering activity leading to a product including objects that are created indirectly by other products.
- *Product information* comprises of 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.
- A *product model* is a digital representation of a real world object held to facilitate the unambiguous transfer of the information between computer systems and to support information sharing (Eastman, 1999).
- A *product library* is any virtual or physical place or media, which contains systematically arranged digital information about physical, manufactured, supplied or created objects and provides access to users and owners of such information. This definition excludes the traditional methods of sourcing and maintaining product information **-** the printed catalogues.
- An *architecture* as defined by ANSI/IEEE (2000) is “the fundamental organisation of a system embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution”.

3. LIMITATIONS OF EXISTING SYSTEMS

The usability of existing product libraries is blighted by the lack of agreement in content and presentation format. This hampers the exchange and sharing of data for reuse in construction applications. Search methods implemented are limited and the multiplicity and lack of integration in the implementations have complicated the search process. Other problems including data (version) inconsistency and lack of persistent access can be associated with the existing process used in publishing updates. In all, the users’ experience has been that of considerable time wastage, frustration and disappointment.

Despite these demerits, the reachability and accessibility offered by the Web implementations continue to attract users as signalled by the proliferation of commercial online product libraries. The authors believe that if the current delivery methods can be improved, better implementations that adequately support prevalent and emerging industry practices can yield a better patronage of online product libraries.

4. STANDARDISATION FOR INTEROPERABILITY

Standardisation addresses the problem of incompatibility of data formats by providing industry-wide syntax and semantic necessary to support automated data exchange and sharing. The International Alliance for Interoperability (IAI) defines the Industry Foundation Classes (IFC) standard. The IFC is an effort to provide standard for digital representation of *things* to facilitate unambiguous transfer of information between computer systems and to support information sharing within the AEC/FM industry. Based on shared IFC schema definitions, aggregation of information from multiple sources can be implemented to provide an integrated access, and possibly single entry-point to product information.

The success of an IFC-based product library implementation hinges on the industry’s acceptance of the IFC standard. In addition to addressing the shortcomings of the existing systems, the popularity of the solution would depend on its ability to integrate with existing systems across platforms and provide future compatibility with the ever-changing software technologies.

5. JUSTIFICATION FOR PRODUCT MODELLING APPROACH

Construction products are international and long lasting. Products may be acquired from several countries and they may be operational for hundreds of years thus outliving the project team and ICT tools employed in creating and storing the information. In addition, the industry is fast migrating from exchange of CAD documents to sharing of both geometric and non-geometric information. This provides the motivation and the justification for adopting an industry-wide product-modelling standard.

Product modelling standards seek to 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).

Stouffs and Krishnamurti (2001) argue that standardisation is not the solution to data exchange, particularly in the design process, because it attempts to impose a common semantic model for all to adhere to with attendant restriction to possibly better solutions and impedance to creative new approaches to specific problems. The argument that if everyone adopts the same concepts, vocabulary and language, any data expressed within the language will be accessible to everyone was challenged on the basis of practicality and representational flexibility and extensibility. Stouffs and Krishnamurti (2001) therefore proposed a lexicon model based on syntactic similarity and data exchange by translations.

Howard (2001) likens the notion that standardisation limits design freedom to saying the alphabet limits literary expression. Adopting Stouffs and Krishnamurti (2001)'s propositions would do little to improve the existing complicated, highly-fragmented communication among the project team members with the accompanying omissions, repetitions, confusions, misunderstandings, errors, delays and litigations.

5.1 Why IFC?

The IFC platform specification, ISO/PAS 16739, provides data structures for the AEC/FM industry shared project model and seeks to enable data sharing across heterogeneous applications by representing building products (and abstract concepts like space, organisations, and processes) and their information requirements in a neutral computer language - the EXPRESS modelling language (IAI, 2000). The use of IFCs, for example, would enable a window manufacturer to provide its product data in a format that can simply be inserted into a CAD design program with embedded properties, such as dimensions, materials, strength, energy performance, fire rating, code compliance, applicability, cost, availability, and source. Appropriate property data about the window can then be exchanged with downstream applications such as cost estimating and energy analysis (IAI-NA, 2002).

IFC is consistent with and adopts a number of specifications from an earlier and ongoing standard, the STandard for Exchange of Product model data (STEP) (Daley et al., 1999), which has been successfully applied in other engineering disciplines including automotive, shipbuilding, aerospace, and process plant.

5.2 Related Projects

Product modelling has been recognised as a basis for computer integrated engineering since the late 1980s. Several EU projects including ATLAS, CIMSteel, COMBI, and COMBINE have developed prototype product model environments to prove the validity and effectiveness of the approach and stimulate further research, standardisation and implementation activities (Turk et al., 1997). The culmination of AEC/FM product modelling efforts in IAI/IFC has created a new breed of implementation attempts. This is exemplified by the Advanced Reusable and Robust Object Warehouse (ARROW) project carried out in the UK by Building Research Establishment and Visual Technology Ltd (Newnham et al., 1997). Online demonstration is available online at http://cig.bre.co.uk/arrow/OSDemo/devVer/mockup/index-frames.html.

Another key project which adopts the product modelling approach is Product Catalogue for Global Engineering Network (PROCAT-GEN) (Faux et al., 1998a), part of the EU-funded GEN project. PROCAT-GEN is exploring the requirements for and building multimedia online catalogue application to support acquisition, extraction and publishing of product information. It aims to support integration of suppliers' existing environments and users' software tools (Faux et al., 1998b). More details and project updates are available at http://www.procat-gen.org/.

Alternative approaches to integrated product libraries, based on proprietary technologies or *de-facto* standards and translators have been developed and are widely deployed in the industry. The most popular online product libraries are more interaction-oriented though some offer product drawings, usually in DXF or DWG format, which can be imported into CAD software and included in a design.

6. IFC ARCHITECTURE - AN OVERVIEW

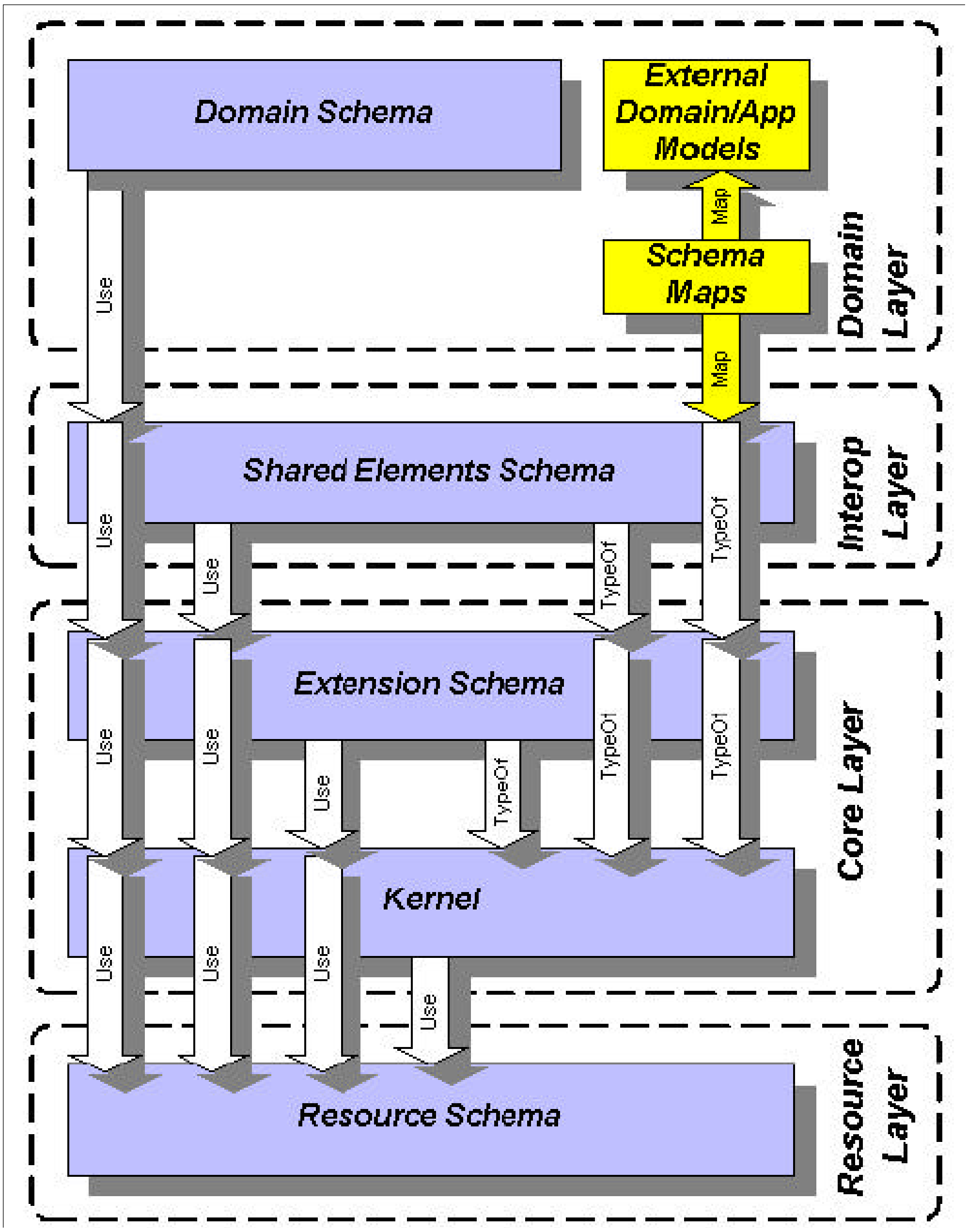


FIG. 1: IFC Model Architecture

IFC employs a layered architecture with four conceptual layers, namely resource layer, core layer, interoperability layer and domain layer. Each layer is comprised of modules containing reusable constructs called classes. 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. Fig. 1 depicts the architecture. A detailed explanation of the IFC architecture is available in IAI (2000).

IFC uses basic data units of the EXPRESS language, entities, attributes and relations, to define fundamental objects like *IfcObject*, *IfcRelationship* and *IfcPropertyDefinition*. These objects are further specialised into classes like mphIfcProduct, *IfcProcess*, and *IfcControl* to form the core classes. Core Layer Extensions specialises core classes into basic objects. For example, *IfcProductExtension* defines objects like *IfcElement*, which is further specialised into *IfcBuildingElement* and *IfcOpeningElement*. Interoperability classes, like *IfcSharedBldgElements*, inherits the core layer classes to provide constructs like *IfcWall*, *IfcSlab* and *IfcWindow*, which are common to different actors and disciplines. Specialised extensions such as *IfcLanding*, specific to disciplines, are derived from lower classes and further distinguished through property sets. Fig. 2 shows an annotated inheritance tree for *IfcLanding*.

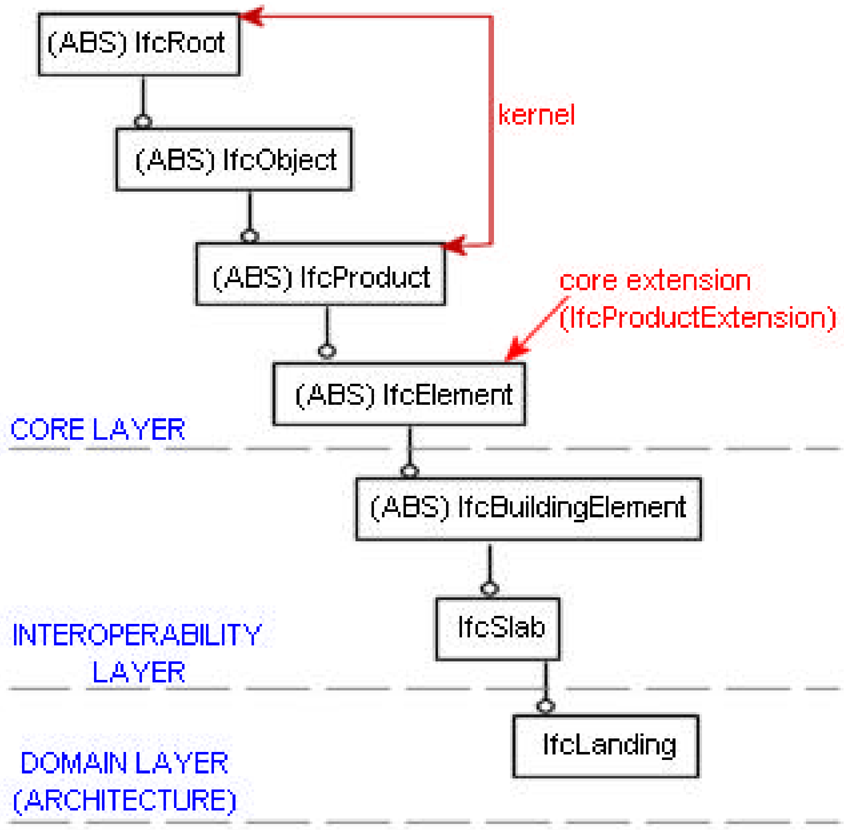


FIG. 2: Inheritance Tree for IfcLanding

6.1 Property Sets

Property Sets consists of property definitions, which may be defined generically to form part of the published IFC release or specifically to meet different application requirements. This is particularly important to the implementation of product libraries because they provide the means for differentiation of products without compromising the ability of the applications to exchange or share standard information. IAI provides guidelines for user-defined property sets and is putting considerable effort into maintaining consistency and coordinating these extensions and the explicitly defined property sets (Adachi, 2001; Ekholm, 2002).

6.2 Parts Library

The IFC standard is insufficient to implement product libraries. It is necessary to manage information about and references to parts or components that constitute products. The ISO 13584 - Parts Library (PLib) standard (Pierra et al., 1998) caters for this through a series of specification to facilitate unambiguous, application-independent representation and exchange of all technical data about parts. PLib, like IFC, reuses STEP capabilities including the EXPRESS modelling language, STEP Physical File format (Part 21), and Standard Data Access Interface (SDAI).

7. CLASSIFICATION SYSTEMS

The organisation of information is critical to its understanding and effective usage and there are a number of systems used to classify construction information today. National and international classification systems have been developed and employed to facilitate electronic exchange of construction information between project partners and suppliers. While search by keywords can be implemented to locate desired information, the existence of some agreed vocabularies, glossaries or thesauri is essential to data exchange and sharing.

The ISO/CD 12006-2 (ISO, 1997) classification system is a framework that identifies a set of classification tables based on views of construction works. The ISO PAS 12006-3 (ISO, 2000), introduces object-oriented information organisation by defining classes and their relationships and properties. Other national and international classification systems include MasterFormat™ (CSI, 1995), UniFormat™ (CSI, 1998), Uniclass (CPIC, 1997), EPIC (EPIC, 1999), OCCS (OCCS, 2001), SfB (Ray-Jones and Clegg, 1991), LexiCon (Woestenenk, 2000), and BSAB (Ekholm et al., 2000). Apart from, but not dissimilar to these classification systems are the product modelling classification systems like IFC and STEP. The harmonisation of the ISO12006-3 with IFC is an on-going task within the IAI-IFC R3 Project. Project information is available at http://cic.vtt.fi/niai/technical/ifc3/R3XM7.html. Reviews and comparisons of subsets of the available classification systems can be found in Ekholm (1999), Wright (1998) and Howard (2001).

It is impractical to map all the classification methods when implementing an integrated product library because of the diversity in their approach but each product can be associated with multiple classification codes. With the progressive convergence of national and international classification systems a single acceptable classification system may emerge in the future, but even then, there would be a need to integrate legacy systems.

8. SEARCH METHODS

Currently there are three approaches to finding products that meet users' requirements. These are text-based search by keyword, search by classification code and parametric search. Researchers are working on two additional search methods termed knowledge-based search and performance-based search.

8.1 Search by Keyword

This borrows from the traditional descriptive search, which involves the use of words that can be associated with actual products. It is usually language and domain specific and can benefit from a background implementation of domain thesauri, dictionaries, glossaries and classification systems when interpreting queries. For example, a search for *window* in English or *Fenster* in German should be interpreted similarly within Architecture domain. Kosovac et al. (2000) describes the development of thesauri for the roofing domain and software for eliminating irrelevant terms and extracting key phrases from textual data.

8.2 Search by Classification

Classification systems are traditionally hierarchical lists of standard elements sometimes with associated classification code. This can foster search by navigating the user through the tree hierarchy with the advantage that user errors, inherent in search by keyword, are eliminated. For example, 08 is the classification code for *Doors and Windows* in the Construction Specifications Institute (CSI) 16 Division format while 08310 is a more specific code for *Access Doors and Panels.* Furthermore, classification systems provide means of accessing data through different sets of views, for example, users can navigate to relevant information by the process model.

8.3 Parametric Search

In parametric search, or attribute-based search, specific functional requirements can be specified as attributes of the product sought. These requirements are evaluated and compared with product attributes to determine appropriate matches. Values can be relaxed or restricted to narrow or expand the search as desired. For example, a search for a door with specified finishing, glazing type, fire rating, light transmission, thickness, water permeability, etc.

The lack of common *language* amongst producers of product information has limited the implementation of parametric search until recently. The ARROW (Newnham et al., 1998), PROCAT-GEN (Faux et al., 1998b) and RINET (Hannus and Pirhonen, 2001) projects have implemented or plan to implement parametric search.

This search method is widely recognised as the most intuitive way to source product information because consumers are usually armed with more information for product discovery, comparison and evaluation than other search implementations allow them to input. This does not eliminate the need for consumers to evaluate the product, as it may be impossible to specify some design criteria in terms of product attributes. For this reason, Jain and Augenbroe (2002) argues that attribute-based search are more suitable for the construction phase of the building project when product specification are available than during the early stages of design when the product specifications are being developed.

8.4 Performance-based Search

The performance of a building object (or product) is defined by Woestenenk (1999) as a measure of the object's reaction to the environment in which it is placed. Performance-based search would extend parametric search by considering the role a product would play in a functional system and by attempting to match the product information with the functional requirements (Jain and Augenbroe, 2002). A method for quantifying, storing and evaluating performance requirements electronically would be required to successfully implement a performance-based search system. There is no known implementation of this search method for construction products but the Active Catalog project (Ling et al., 1997) provides a proof of the concept for electro-mechanical engineering parts library. The prototype, based on a *try-before-you-buy* paradigm, helps the designer to repetitively evaluate candidate components in the context of the system design until a satisfactory product is found.

8.5 Knowledge-based Search

Knowledge-based search exploits a conceptual vocabulary, domain knowledge and Artificial Intelligence (AI) search techniques in interpreting queries and returning best matches. It requires an unambiguous representation of user's requirements to apply domain knowledge to *reason* about the user's request (Clark et al., 2000). This search method can also benefit from categorisation or classification, context-based interpretation and management of user profiles. One emerging knowledge-based search technique is the use of agent technology (Denzinger et al., 2002). The application of this technique in the procurement of construction products is the subject of an ongoing research (Obonyo et al., 2001).

9. SYSTEM REQUIREMENTS

Extensive system analysis approached from the needs of users and drilled down to deployment issues is required before implementing any software. This section presents the functional and operational requirements for product libraries. The high-level functional requirements are discussed with the aid of a use case model.

The goal of the product library is to provide product information that is complete, accessible, and reusable. Completeness is the ability of the information to support all construction phases from feasibility to demolition as shown in Fig. 3. Accessibility implies the information is easily and speedily obtainable. Reusability is the degree to which the information can be readily utilised in consumer application environments. The system should be reliable, flexible and available. System reliability relates to the number of bugs in the software and flexibility is degree of responsiveness to changing user needs and technologies. Availability relates to the system's portability and maintainability. It is the ability of the software to run on readily available hardware, operating system, etc.

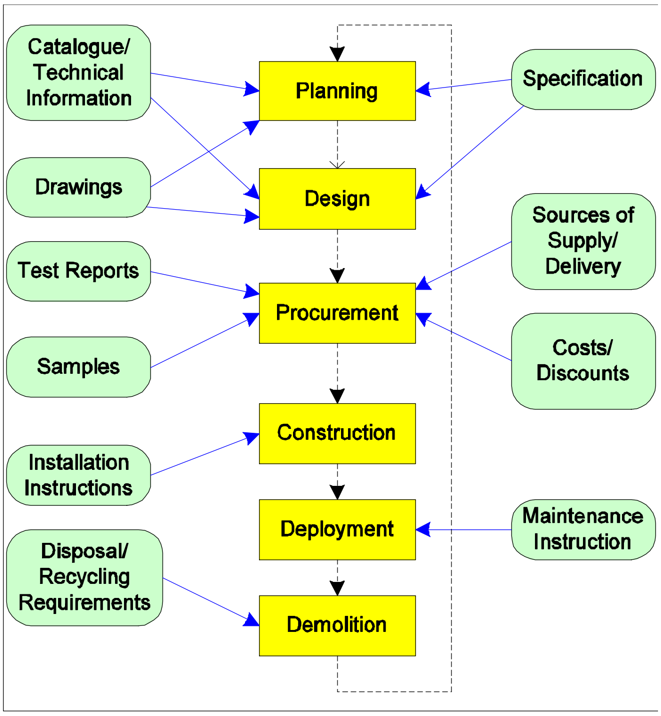


FIG. 3: Construction Process and Product Information Requirements

9.1 Functional Requirement

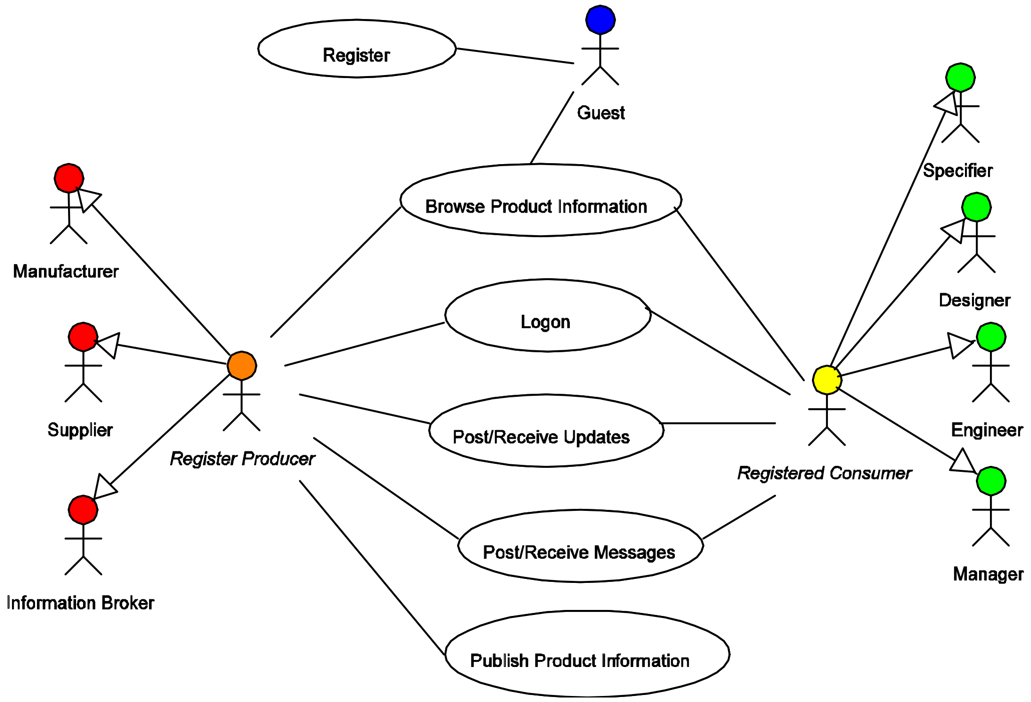


FIG. 4: Use Case Diagram for the Product Library

Use cases document the behaviour of a system from the users' point of view. Fig. 4 presents a use case model of the system. It details the high level needs and desired features of the system. *Actors*, represented by stick persons, are the roles that would be played by stakeholders and target users while *use cases* are the tasks they need to perform. This role-based approach would facilitate personalisation and customisation. For example, product information can be presented based on the construction project phase and the role being played by the actor. Specifiers, Designers, Engineers and Managers have been generalised as Registered Consumers. These are in turn generalisation of specific disciplines not explicitly represented. A single user or a group of users within an organisation or project team may play each actor's role.

All users should be able to interface with the system via standard Web browsers and their specific software applications. *Guest* (unregistered) users should be allowed to register for proper identification otherwise they should be limited to product browsing. *Registered Producer* should be able to publish their product information through Web interfaces or automatically by manual or scheduled running of an application on their local systems. *Registered Consumer* should be able to browse the product information by a variety of methods including from their specific applications. Facilities should also be provided for controlled exchanges between the producers and consumers in the form of updates and messages like feedbacks, request for information, etc. The system should be extensible to accommodation commercial exchanges and future requirements.

It is not feasible for a single implementation to cater for all producers and consumers. Given that this is a data-intensive application, a central implementation would constitute a bottleneck and create maintenance nightmare. The system should therefore provide the facility for transparent cooperation with other implementations and sources of relevant product information.

9.2 Operational Requirements

Traditionally, manufacturers, suppliers and information brokers have the responsibility of packaging and delivering product information. Manufacturers and suppliers should retain the ability to choose between implementing their own system and subscribing to information broker. A subscription-based implementation topology that enables implementations to cooperate to satisfy users' requests and eliminate duplication of data is required. This would also enable the formation of dynamic supply chains so that a *closed* group of producers and consumers can be formed around the project teams or businesses.

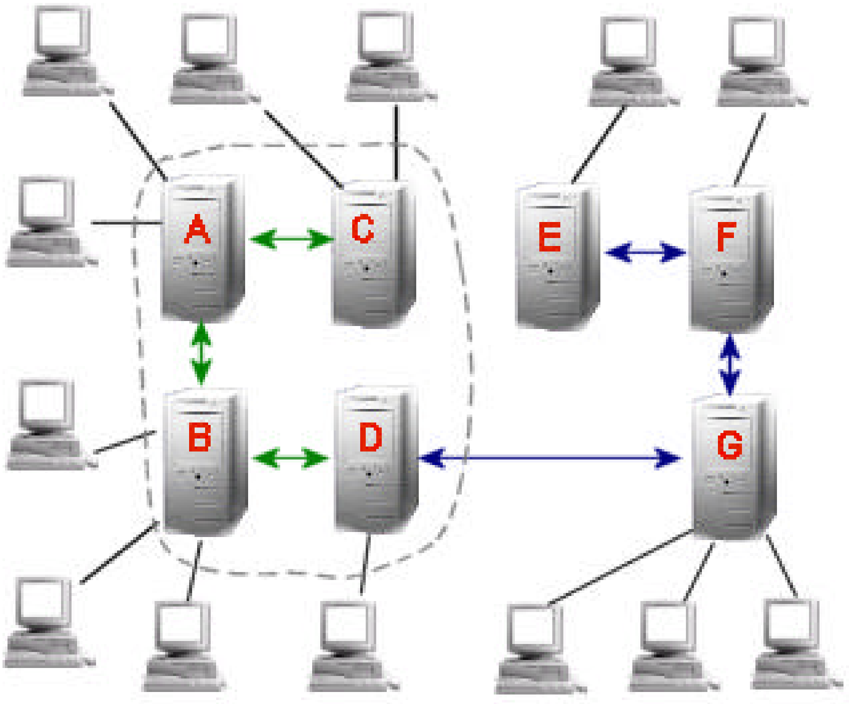


FIG. 5: Subscription-based implementation topology

Fig. 5 presents the required implementation topology with seven implementations labelled A through G. G can access E because it can access F but it cannot access A, B and C even though it can access D because A, B, and C are in a closed implementation group with D.

Each implementation should maintain a list of implementations it can reach including information on types of products and product information hosted. The list should be updated when subscribing to a new implementation to avoid duplicates. The system should be able to select relevant implementations to satisfy users' requests.

This topology would offer a truly dynamic, distributed network of product information sources that eliminates the need for multiple copies of the same information and is capable of supporting an integrated project process. This is a practical alternative that is more likely to succeed in the industry than the centralized approaches because it provides for gradual changeover and eliminates the access bottlenecks usually associated with single, centralized sources.

9.3 Deployment Context

This section examines the relationship between each implementation to other systems that it would interface with as shown in Fig. 6.

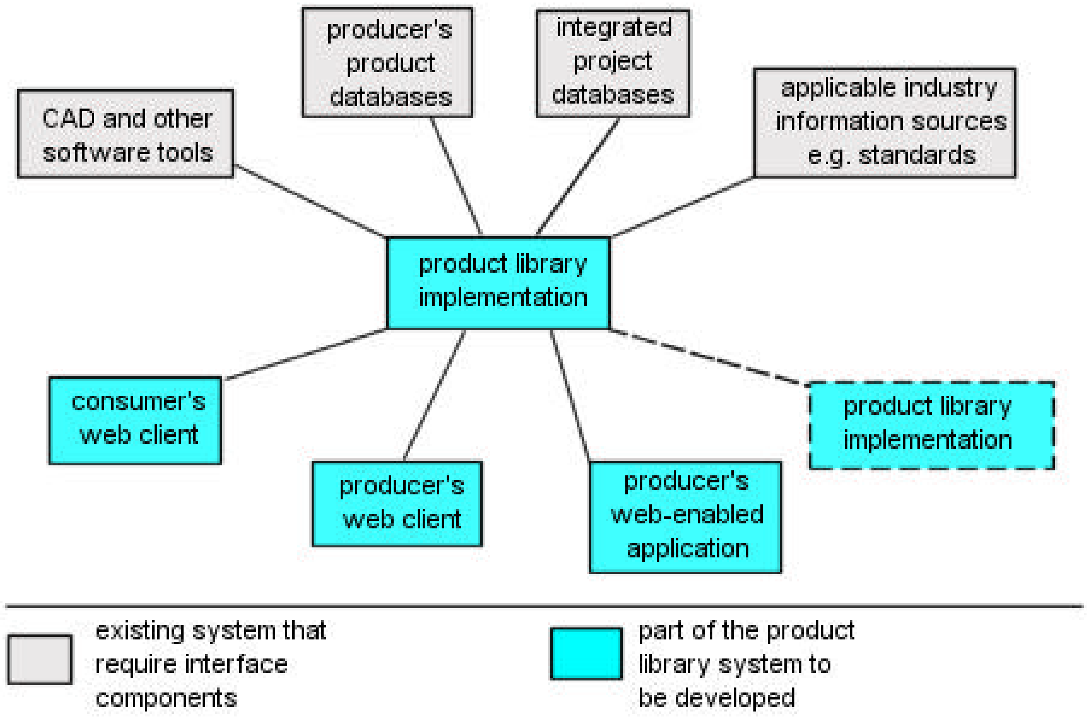

producer's product databases
integrated project databases
CAD and other software tools
applicable industry information sources e.g. standards
product library implementation
consumer's web client
product library implementation
producer's web client
producer's web-enabled application
existing system that require interface components
part of the product library system to be developed

FIG. 6: Context for the product library system

The implementation requires interface components for accessing other product library implementations. Interface components are also required for industry software tools, existing producer databases, integrated project databases and other sources of product-related information. The implementation will provide Web clients for consumers and producers in addition to Web-enabled applications for producers.

10. APPLICATION ARCHITECTURE

A detailed analysis of all the requirements suggests that a combination of application architectures is needed. This section introduces the multitier client-server architecture and the Web Services architecture.

10.1 Multitier Client-Server Architecture

The system requirements and deployment context described above suggest a client-server application scenario whereby a client (or user's machine) running a Web browser or dedicated application sends requests to a server (or product library implementation) over a communication network. The request triggers the execution of a set of programs on the server and a response is returned to the client.

The server architecture that interprets requests is usually component-based. These components are grouped into tiers (as shown in Fig. 7) for user interface, application and business logic and backend services such as file, database and communication services. The degree of coupling between these components determines the number of tiers in the server architecture. In one-tier, for example, all three components are closely coupled while they are separate in three-tier architecture. The multi-tier, or n-tier, extends the three-tier architecture to include scenarios where many middle layers cooperate with one another to respond to a request. This is the most applicable scenario to the product library implementation.

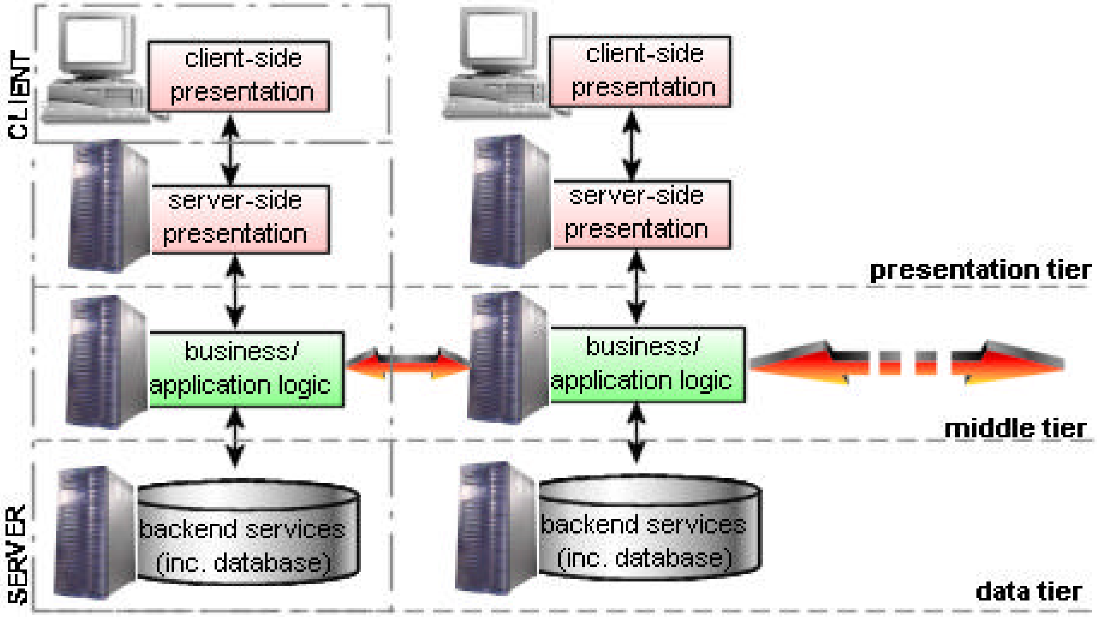


FIG. 7. Multitier Client/Server Architecture

Multitier architecture helps to decouple business logic from presentation and data access. It offers increased scalability, performance, network efficiency, and service reusability. It enables a great number of concurrent users and facilitates easier deployment and management of applications. Raduchel (1998) covers the benefits of multitiered architecture in more detail.

10.2 Web Services Architecture

The implementation is required to provide multiple interfaces to the same data from different platforms and devices. It should be easily extensible to cater for new devices that may emerge in the future. This requirement necessitates the use of open, cross-platform standards like TCP/IP for universal networking and XML (eXtensible Mark-up Language) for universal structured data. Web services offer this openness.

Web services are self-contained, modular applications capable of supporting dynamic collaboration or just-in-time integration using loosely coupled components. They are network applications that use Simple Object Access Protocol (SOAP) and Web Service Description Language (WSDL) to exchange information in the form of XML documents. SOAP is an XML grammar and an open application protocol for Remote Procedure Call (RPC) and asynchronous messaging. WSDL is also an XML-based Interface Definition Language (IDL), which can be used to describe Web services including the kind of message format expected, the Internet protocol used, and the Internet address of the Web service. A detailed introduction to Web services and the benefits of the architecture can be found in Tidwell (2000), Gottschalk (2000) and Monson-Haefel (2002).

The Web services architecture contains three components and three operations as shown in Fig. 8. The Service Provider encapsulates implementation details and publishes an Application Program Interface (API) for use by other services on the network. Service Requestors can discover available services through Service Brokers and then invoke (or request) those services from the Service Provider.

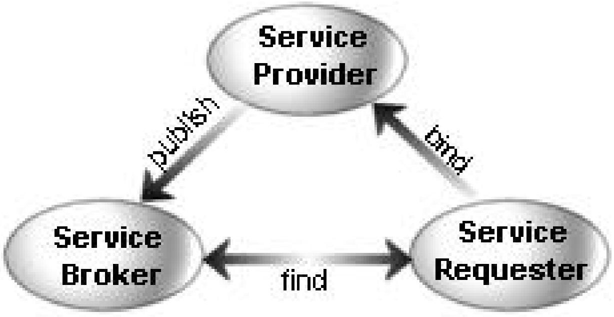


FIG. 8: Web Services Architecture

11. IMPLEMENTATION ARCHITECTURE

The implementation is envisioned as a multitiered, service-based, message-oriented, client-server application. This is a hybrid solution, which combines the elements of Web services and multitier applications by implementing the middle tier of the latter as the Service Provider and Service Broker of the former. The backend services are still encapsulated in the middle tier and presentation components access the business and application logic as services. Fig. 9 shows the proposed architecture.

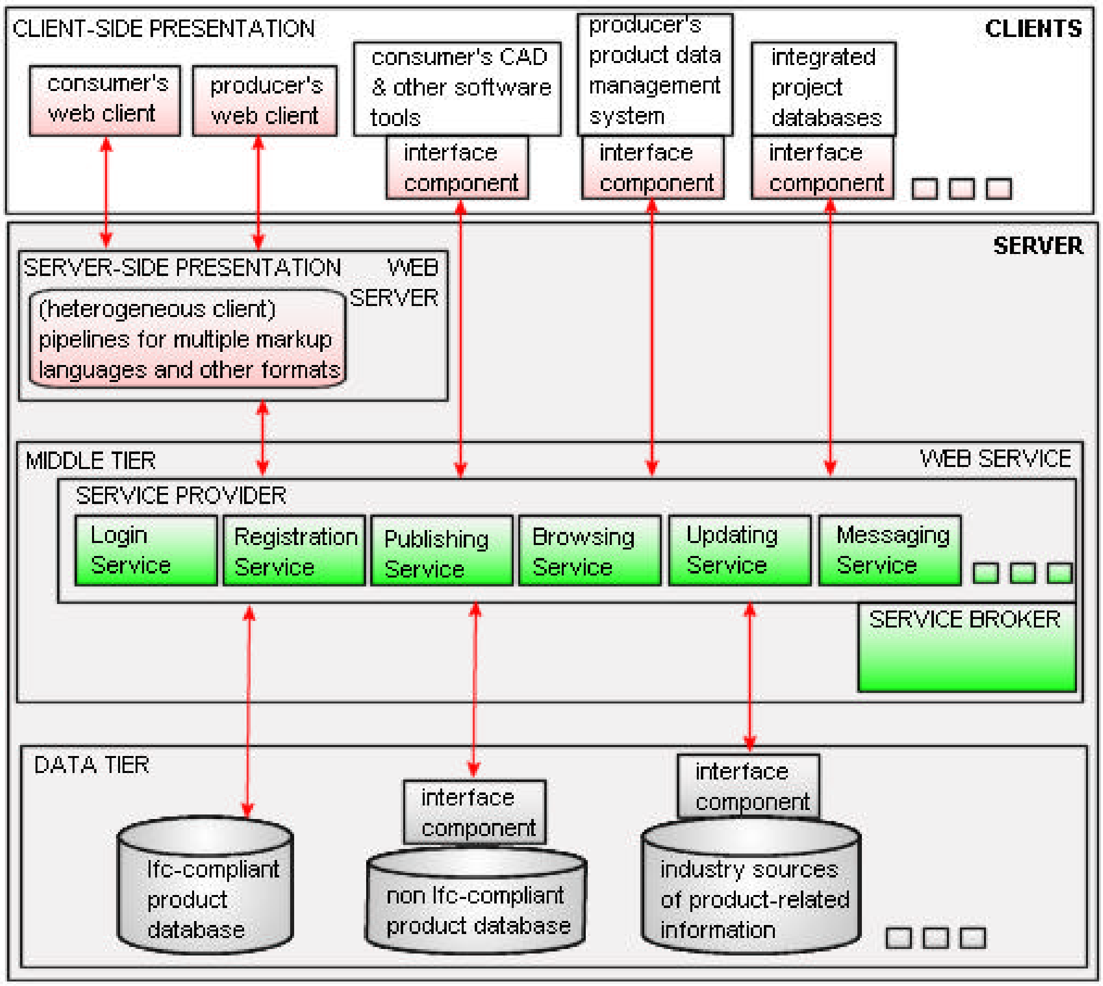


FIG. 9: Multitiered, service-based, message-oriented client-server application architecture

In the past, many researchers have proposed and employed various aspects of the above architecture for IFC related implementations. Adachi (2002) adopted a Web Service approach to the development of IFC Model

Server. Halfawy et al. (2002) recommended IFC-based messaging for systems interoperability and efficient data exchange. Faraj et al. (2000) developed an IFC-based distributed computer integrated environment using the three-tier architecture. With recent developments in software development platforms it is possible to harness the benefits of all these approaches in a single, yet cohesive, solution.

11.1 Data Tier

The data tier consists of an implementation of IFC-complaint database and legacy or existing product data sources. Where possible, interface components that transform data from the non-compliant sources like manufacturers' existing product databases and other sources of product-related information, would be developed or implemented.

To ensure persistent access, data in this tier would grow incrementally. This implies, data, once published cannot be deleted but new versions can be released.

11.2 Middleware

The middleware is a SOAP-based Web Service, which implements the application features including login, registration, publishing, and browsing components. Updating service and Messaging service would be asynchronous message-based services. The Updating Service would provide update information to consumers and allow consumers to register interest in specific parts of a particular product's information. It would implement a publish-subscribe communication model whereby notification of updates would be broadcasted to subscribed consumers. The Messaging Service would support one-to-one communication between producers and consumers. Additional services could be plugged in as required.

11.3 Presentation

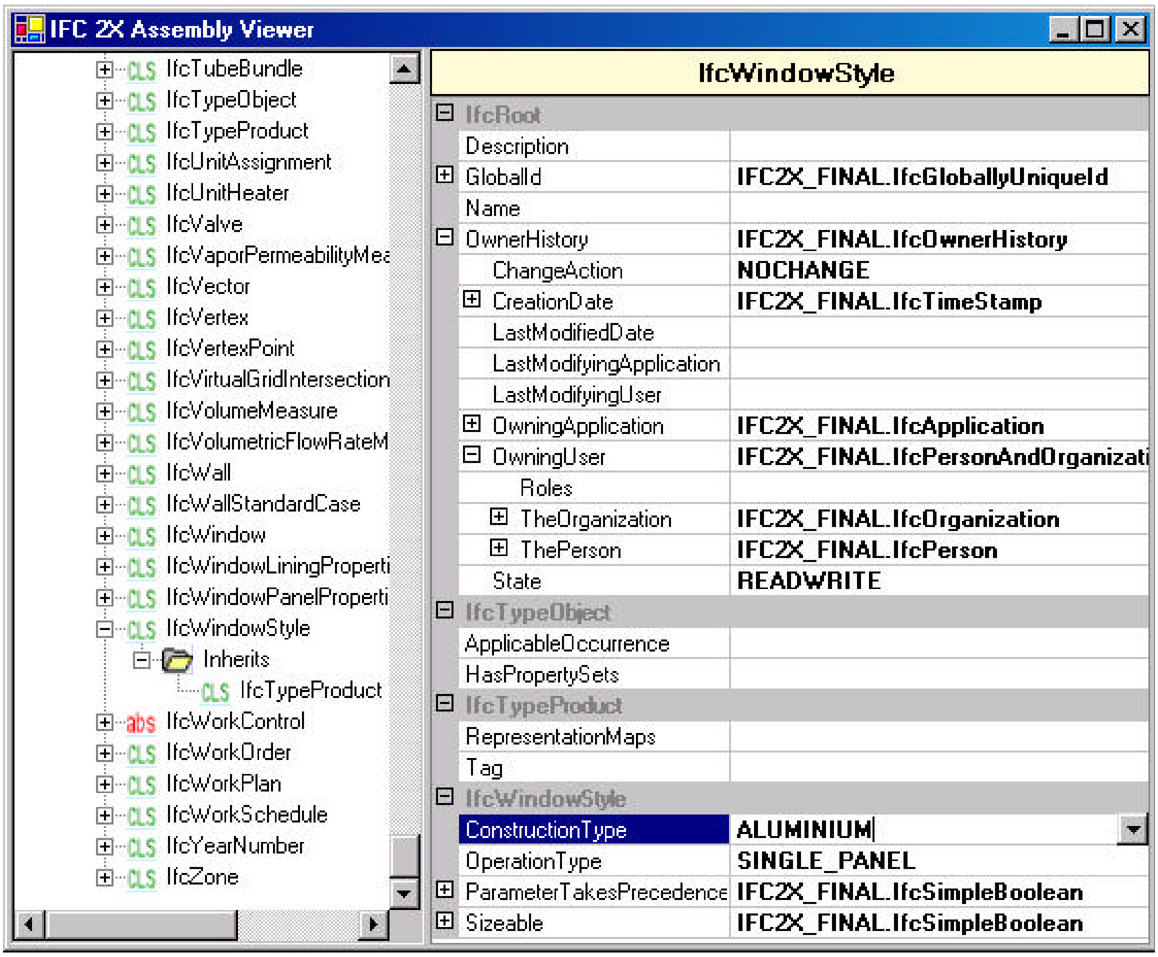
The Web Server would provide a transformation engine, which receives XML messages from the Web Service and translates it into required data formats and device or browser specific mark-up language such as HyperText Mark-up Language (HTML) and Wireless Mark-up Language (WML). This application would provide Web access to consumers and producers. Interface components would be developed to interface with other applications that would access the system over the Internet. These components would receive XML data directly from the Web service and transform or process it based on the specific deployment environment.

12. IMPLEMENTATION STATUS

The proposed solution is being implemented as part of an ongoing research project being undertaken by Loughborough University and Corus (UK) Ltd. Detailed specifications for the prototype applications are being produced as outstanding implementation issues are being clarified. Microsoft .NET platform was chosen for the delivery after a comparative study of applicable implementation platforms on cost, ease and efficiency of implementation. Available software tools ranging from integrated systems for storing and managing EXPRESS model data to basic tools like translators and libraries are being investigated.

An EXPRESS to C# code generator is being developed based on the Java EXPRESS parser developed by National Institute of Standards and Technology (NIST), US (Lardet, 2001). The code generator is being tested with IFC2x. It reads the IFC specification file, parses it, builds an abstract syntax tree and traverses the tree creating the C# source code file. The output code is then compiled into a .NET class assembly using Visual Studio .NET. The parser is being further developed to include methods for generating STEP P21 and IfcXML representation in each class.

Fig. 10 is a screenshot of an IFC Assembly viewer designed for the generated .NET class Assembly. It uses the .NET object reflection to provide a navigable tree representation of all the class attributes, including inherited attributes. The representation for IfcWindowStyle is shown in the figure.


IFC 2X Assembly Viewer

```
⊞ CLS IfcTubeBundle
⊞ CLS IfcTypeObject
⊞ CLS IfcTypeProduct
⊞ CLS IfcUnitAssignment
⊞ CLS IfcUnitHeater
⊞ CLS IfcValve
⊞ CLS IfcVaporPermeabilityMea
⊞ CLS IfcVector
⊞ CLS IfcVertex
⊞ CLS IfcVertexPoint
⊞ CLS IfcVirtualGridIntersection
⊞ CLS IfcVolumeMeasure
⊞ CLS IfcVolumetricFlowRateM
⊞ CLS IfcWall
⊞ CLS IfcWallStandardCase
⊞ CLS IfcWindow
⊞ CLS IfcWindowLiningProperti
⊞ CLS IfcWindowPanelProperti
⊟ CLS IfcWindowStyle
    ⊟ Inherits
        CLS IfcTypeProduct
⊞ abs IfcWorkControl
⊞ CLS IfcWorkOrder
⊞ CLS IfcWorkPlan
⊞ CLS IfcWorkSchedule
⊞ CLS IfcYearNumber
⊞ CLS IfcZone
```

IfcWindowStyle

⊟ IfcRoot	
Description	
⊞ GlobalId	IFC2X_FINAL.IfcGloballyUniqueId
Name	
⊟ OwnerHistory	IFC2X_FINAL.IfcOwnerHistory
ChangeAction	NOCHANGE
⊞ CreationDate	IFC2X_FINAL.IfcTimeStamp
LastModifiedDate	
LastModifyingApplication	
LastModifyingUser	
⊞ OwningApplication	IFC2X_FINAL.IfcApplication
⊟ OwningUser	IFC2X_FINAL.IfcPersonAndOrganizati
Roles	
⊞ TheOrganization	IFC2X_FINAL.IfcOrganization
⊞ ThePerson	IFC2X_FINAL.IfcPerson
State	READWRITE
⊟ IfcTypeObject	
ApplicableOccurrence	
HasPropertySets	
⊟ IfcTypeProduct	
RepresentationMaps	
Tag	
⊟ IfcWindowStyle	
ConstructionType	ALUMINIUM
OperationType	SINGLE_PANEL
⊞ ParameterTakesPrecedence	IFC2X_FINAL.IfcSimpleBoolean
⊞ Sizeable	IFC2X_FINAL.IfcSimpleBoolean

FIG. 10: Ifc Assembly Viewer

This tool has proved an invaluable tool in fostering better understanding of the IFC model amongst the development team. The development project is expected to complete in Spring, 2004 with the demonstration of the prototype application.

13. DISCUSSION AND CONCLUSION

The IFC model is the foundation for interoperability in the AEC/FM industry. A well-engineered application architecture and implementation based on the model can meet the industry's present and future requirements for reliable, flexible and available product information, which is complete and reusable in software applications employed in the industry. This paper identifies key requirements for online product libraries and presents an architecture, which has been derived for the implementation.

The derived architecture is aimed at supporting present and emerging industry practices in the production and consumption of product information. The multi-tiered, service-based, client/server architecture presented can support role-based, multimedia presentation and subscription-based cooperation among implementations. This would enable the systems to support the project team and the supply chain. In addition, it makes provision for gradual buy-in and staged system changeovers as each system is capable of existing independently or within a group.

Loose coupling of components and asynchronous messaging are employed to further decouple the system and increase flexibility of integration. The development of interface components is proposed as a mean for integrating existing systems and tools. Versioning and persistent access problems are addressed through a write-once publishing approach.

The major hindrances to utilising the IFC model for the implementation of a prototype system are the maturity of the model and the lack, or limited availability, of software tools that can facilitate rapid application prototyping. The IFC model is fast evolving and has good prospects for industry acceptance hence the decision to adopt the standard. The EXPRESS modelling language, used to define the model, poses some challenges to developers. It is designed to be human-readable but as the number of, and inter-relationships between, its elements grow its human-readability diminishes. The graphical representation, in EXPRESS-G, does not improve the situation much. The Ifc Assembly Viewer (presented in Section 12) addresses this problem.

There are still a number of questions and issues that must be addressed in implementing the architecture proposed here. Some of these include:

- Is the industry ready for standardisation considering the cost of translation?
- How should the solution be packaged and delivered to the industry?
- Who should provide the translators?
- What happens if the IFC changes in its core?
- How to manage versions, persist links and eliminate data relating to dead products.
- How should multiple classification systems be mapped to objects?

These and many other issues have been identified and are being researched as the implementation progresses.

14. REFERENCES

Adachi, Y. (2001). IFC 2x property set development guide. Technical Report MSG-01- Draft 2, International Alliance for Interoperability.

Adachi, Y. (2002). Overview of IFC model server framework. In *European Conference of Product and Process Modelling (ECPPM) eWork and eBusiness in AEC* (Turk, Z. and Scherer, R., editors).

ANSI/IEEE (2000). *1471-2000 IEEE Recommended Practice for Architectural Description for Software-Intensive Systems*. ISBN 0-7381-2518-0.

Clark, P., Thompson, J., Holmback, H., and Duncan, L. (2000). Exploiting a thesaurus-based semantic net for knowledge-based search. In *Proceedings of 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}, pages 988-995, California, USA. AAAI Press (Distributed by MIT Press).

CPIC (1997).*UNICLASS: Unified Classification for the Construction Industry. RIBA* Publications. Construction Project Information Committee.

CSI (1995). *MasterFormat 1995.*Construction Specification Institute, 99 Canal Center Plaza, Suite 300, Alexandria, VA 22314.

CSI (1998).*UniFormat 1998 Edition.* Construction Specification Institute, 99 Canal Center Plaza, Suite 300, Alexandria, VA 22314.

Daley, W. M., Bachula, G. R., and Kammer, R. G. (1999). *STEP: The Grand Experience. US* Department of Commerce, Technology Administration, National Institute of Standards and Technology. Edited by Sharon J. Kemmerer.

Denzinger, J., Sinz, C., Avenhaus, J., and Uchlin, W. K. (2002). Teamwork-PaReDuX: Knowledge-based search with multiple parallel agents. In *Proceedings of International Conference on Massively Parallel Computing Systems, Ischia, Italy.*

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.

Ekholm, A. (1999). Co-ordination of classifications for product modelling and established building classifications. In *CIB W78 Workshop on Durability of Building Materials and Components 8, Vol. 4 Information Technology in Construction* (Lacasse, M. A. and Vanier, D. J., editors), Ottawa, Canada. NRC Research Press.

Ekholm, A. (2002). Principles for classification of properties of construction objects. In *Distributing Knowledge in Building* (Agger, K., Christiansson, P., and Howard, R., editors), Aarhus School of Architecture, Denmark. CIB w78 Conference. ISBN 87-90078-36-5.

Ekholm, A., Haggstrom, L., and Karlsson, H. (2000). *BSAB 96: The Swedish Construction Industry Classification System.* The Swedish Building Centre Systematics, S-113 87 Stockholm, Sweden.

EPIC (1999). *EPIC Master Version 1999.*EPIC General Secretariat, WTCB/CSTC, Lozenberg I Nr 7, B-1932 Zaventem, Belgium, version 2.0 edition.

Faraj, I., Alshawi, M., Aouad, G., Child, T., and Underwood, J. (2000). An industry foundation classes web-based collaborative construction computer environment: WISPER. *Journal of Automation in Construction,* 10:79-99.

Faux, Kesteloot, Stewing, and Pusch (1998a). Product catalogues in global engineering networking. In *Proceedings of ACM.* Also presented at GEN '97, Antwerp, Belgium, 1997.

Faux, I., Radeke, E., Stewing, F.-J., Broek, G. V. D., Kesteloot, P., and Sabin, A. (1998b). Intelligent access, publishing and collaboration in global engineering networking. *Computer Networks,* 30(13):1249-1262.

Gottschalk, K. (2000). Web services architecture overview: The next stage of evolution for e-business. *IBM DeveloperWorks.* Available online at ftp://www6.software.ibm.com/software/developer/library/w-ovr.pdf.

Halfawy, M. R., Pouria, A., and Froese, T. (2002). Developing message-based interoperability protocols for distributed AEC/FM systems. In *Distributing Knowledge in Building* (Agger, K., Christiansson, P., and Howard, R., editors), Aarhus School of Architecture, Denmark. CIB w78 Conference. ISBN 87-90078-36-5.

Hannus, M. and Pirhonen, A. (2001). RINET - building product database. http://cic.vtt.fi/projects/rinet/index.html.

Howard, R. (2001). Classification of building information - European and IT systems. In *Construction Information Technology, International Conference : IT in Construction in Africa,* pages 9-1 to 9-14. CSIR, Building and Construction Technology.

IAI (2000). *IFC Technical Guide.* International Alliance for Interoperability. Copy available at http://www.iai.org.uk/documentation/IFC_2x_Technical_Guide.pdf.

IAI-NA (2002). Interoperability in building industry moves forward with ISO ballot. News release published online at http://www.iai-na.org/news/121802.php. International Alliance For Interoperability, North America. Last visited: 28 March, 2003.

ISO (1997). Organisation of information about construction works - part 2: Framework for classification of information. Technical Report ISO/TC 59/SC 13 N 86, International Organisation for Standardisation, Geneva, Switzerland.

ISO (2000). Building construction - organization of information about construction works - part 3: Framework for object-oriented information exchange. Technical Report ISO/PAS 12006-3, International Standards Organisation, Geneva, Switzerland.

Jain, S. and Augenbroe, G. (2002). Performance-based electronic product catalogues for the building industry. In *Proceedings of e-2002, eBusiness and eWork Conference in Prague.*

Kosovac, B., Vanier, D. J., and Froese, T. M. (2000). Use of keyphrase extraction software for creation of an AEC/FM thesaurus. *Electronic Journal of Information Technology in Construction, Edited by Bo-Christer Björk,* 5:25-36.Available online at http://itcon.org/2000/2.

Lardet, S. (2001). *Java Express Parser User Documentation.* National Institute of Standards and Technology, US.

Ling, S. R., Will, P., Kim, J., and Luo, P. (1997). Active Catalog: Searching and using catalog information in internet-based design. In *Proceedings of DETC '97 - 1997 ASME Design Engineering Technical Conferences* (Cheng, H. H., editor), Sacramento, California. Copy available at http://www.isi.edu/active-catalog/asme-97.ps.

Monson-Haefel, R. (2002). EJB 2.1 web services. *theserverside.com.* Available at www.theserverside.com.

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., Amor, R., and Parand, F. (1998). Gaining quality manufactured product information through ARROW.*PDT Days, BRE, UK,* pages 39-46.

Newnham, L., Parand, F., Amor, R., and Nisbet, N. (1997). The ARROW framework for a building object warehouse. In *CIB W78 Workshop on Information Technology Support for Construction Process Re-Engineering, IT-CPR 97* (Drugemuller, R., editor), pages 319-328. International Council for Building Research Studies and Documentation.

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

OCCS (2001). *Overall Construction Classification System.* The OCCS Development Committee. www.occsnet.org.

Pierra, G., Sardet, E., Potier, J. C., Battier, G., Derouet, J. C., Willmann, N., and Mahir, A. (1998). Exchange of component data : The PLIB model, standard and tools. In *CALS Europe '98,* pages 160-176, Paris, France.

Raduchel, W. (1998). Implementing a multitier, services-based architecture on the Java platform at Sun - a case study. Technical report, Sun Microsystems. Copy available at http://www.sun.com/980602/wp/index.html.

Ray-Jones, A. and Clegg, D. (1991). *CI/SfB Construction Indexing Manual.* RIBA Publications, London, 2001]

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.

Stouffs, R. and Krishnamurti, R. (2001). Standardization: A critical review. In *International Conference on IT in Construction in Africa,* pages 42-1 to 42-10. CSIR, Building and Construction Technology. Conference website and papers at http://buildnet.csir.co.za/constructitafrica/.

Tidwell, D. (2000). Web services - the web's next revolution. *IBM DeveloperWorks.* Available online at http://www-106.ibm.com/developerworks/.

Turk, Z., Wasserfuhr, R., Katranuschkov, P., Amor, R., Hannus, M., and Scherer, R. J. (1997). Conceptual modelling of a concurrent engineering environment. In *Proceedings of First International Conference on Concurrent Engineering in Construction.*

Woestenenk, K. (1999). LexiCon - a common vocabulary for the construction industry. In *Proceedings of 2nd International Conference on Concurrent Engineering and Construction {(CEC99),* Espoo, Finland}.

Woestenenk, K. (2000). The LexiCon: An update. In *Proceedings of Product and Process Modelling in Building and Construction* (Gonçalves, R., Steiger-Garção, A., and Scherer, R., editors). ECPPM 2000, Balkema, Rotterdam.

Wright, T. (1998). Classifying building information: An historical perspective. Technical report, Construction Information Systems Australia Pty Ltd.