THE DYNAMIC DEFINITION OF DESIGN ELEMENT SPECIFICATIONS VIA A PRODUCT SUPPLIER DATABASE WEB-SITE

Dynamic specification definition via the web

J. UNDERWOOD, M.A. ALSHAWI, G. AOUAD, T. CHILD and I. FARAJ
Department of Surveying, University of Salford, UK

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier*. Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2629-2639.

Abstract

The AIC Research Group at the University of Salford have been involved in a government funded project that aimed to develop an integrated multi-user distributed construction project database - WISPER (**W**eb-based **I**ntegrated **S**hared **P**roject **E**nvi**R**onment). The objective of the project was to develop a working system capable of demonstrating the future direction of information integration with the project partners' businesses. This paper presents the development of the specification application that aims to demonstrate the potential for design elements to be specified directly from a product database Web site.

Keywords: Design element specifications, IFC, integrated distributed construction project database, three-tier client-server architecture, STEP Part21.

1 Introduction

The Internet with its open standards and accessibility is fast evolving into a powerful environment for supporting distributed group work. Originally the Web was a static two-tier client-server unidirectional environment for the publishing and broadcasting of electronic documents. However, demands for dynamic content is beginning to change that. The Web is being transformed from a publishing technology into a full-blown client-server medium, with the potential to run line-of-business applications and to deal with the complex requirements of multistep business-to-business and consumer-to-business transactions (Orfali et al. 1997).

This paper presents the specification application that has been developed as part of the implementation of a new generation of construction integrated environments. The application aims to demonstrate the potential for next-generation Web technology together with the IFCs to support the direct specification of design elements from a product database Web site. Such an

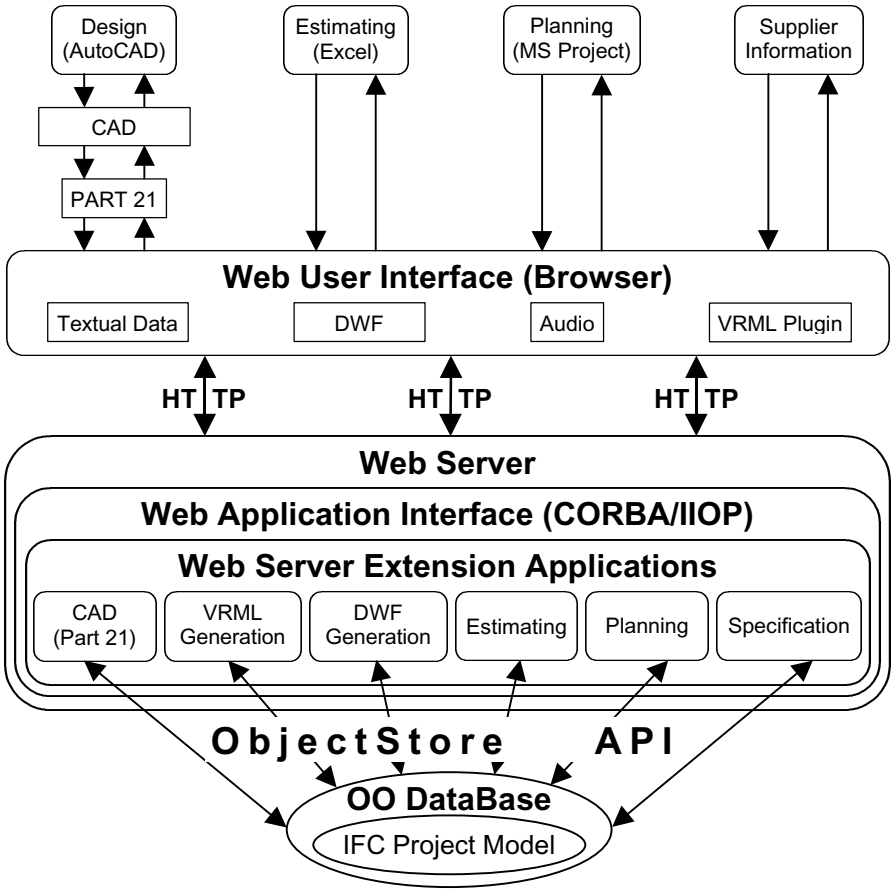

Fig. 1: Overview of WISPER

application enables the designer to retrieve their defined specifications (including technical information) in a standard format directly via the Web, and to upload this information into the project database to be accessed by the other applications, e.g. CAD, Estimating, etc. In addition, the paper provides a brief overview of the integrated distributed construction environment, the implemented supporting applications and the IFC object model architecture.

2 WISPER - Web-based integrated shared project environment

2.1 Three-tier client-server architecture

As shown in Fig. 1, the architecture of WISPER uses a three-tier client-server infrastructure to demonstrate the integration between detailed design, building element based cost estimating, and construction scheduling, in addition to a VRML viewer that allows the graphical querying of a project database (Alshawi et al. 1998; Faraj et al. 1998a, 1998b; WISPER 1998). Within a three-tier architecture each facet of the application, i.e. user interface, logic, and database, is isolated providing benefits such as maximum control, scalability, and flexibility (Campbell 1997; Orfali et al. 1997).

2.2 IFC object-oriented project database

At the lowest level of the system architecture (Bottom Tier), the main project database has been developed using the ObjectStore object-oriented database management system together with the implementation of Java classes derived

directly from the Final Version 1.5 of the International Alliance for Interoperability's (IAI) Industry Foundation Classes (IFCs). The IFCs are an initiative by leading construction software vendors to produce a standard data model for interoperability of construction related software applications (IAI 1999). The project model for the Final Version 1.5 of the IFCs has been published using EXPRESS. EXPRESS is a formal language developed by the STEP community for the description of data models (STEP Tools, Inc. 1999a; UKCIC 1999). To map this schema to ObjectStore, ST-Developer was used to convert the IFC schema from EXPRESS to a set of programming language classes (pure Java classes). ST-Developer is a Step Tools product that provides a comprehensive, stand-alone set of software development tools for handling the complexities of the STEP standard, which can be used to build software that works with STEP data in object-oriented databases, relational databases, and traditional files (STEP Tools, Inc. 1999b). ObjectStore utilities are then used to read the Java classes to create the IFC project database schema. This schema can then be used and instantiated by the various construction applications.

2.3 Application servers

Using Web client-server technology, i.e. Web Server Extensions (Java WAI applications), applications have been written in Java for the two-way exchange of information between the project database and existing software, i.e. AutoCAD, Excel, MS Project scheduling package, etc. (Middle Tier). WAI is a CORBA-based (Common Object Request Broker Architecture) programming interface that defines object interfaces to the HTTP request/response data and server information. Via WAI, the Web applications accept an HTTP request from a client (Web browser), process it, and return a response to the client.

2.4 User interface

Finally, the top tier of the system provides the interface to the overall system. The user interface has been implemented as a set of Web pages and other commercial applications to support the different construction experts that may need to interact with the environment. The user interface enables the user to submit, retrieve, process and manipulate data. Users can perform operations on the project data by selecting the appropriate menu item on a specific Web page. Each menu item references a URL that submits an http request to the server. The Web server assigns the responsibility for processing the HTTP request to the CORBA objects.

WISPER uses the concept of a 'project Web site' as a first point of access for all shared project information. Each new project has its own Web site, allowing project information to be accessed through an internal network (Intranet) or through the wider global Internet. Furthermore, existing security mechanisms such as; password protection, certificates, IP address filtering, etc., ensures the security of the project information by controlling access to the project Web pages.

2.5 Implemented applications

- *CAD (design)*: Due to the lack of availability of commercial IFC compatible CAD systems, the research team has implemented their own CAD system that uses the AutoCAD-14 geometry engine. The application supports both newly created and existing projects. The 'Newly Created Project' application

enables the user to create new building designs and to share information with other construction applications. Data in the application is saved in STEP Part 21 file format. The 'Existing Project' application allows the user to populate the database from a STEP Part 21 file. Part 21 files of existing projects can therefore be loaded to the database for amendment and saved as Part 21 file format.

- *VRML*: The application generates the VRML model of the design and displays the results within a Web browser. The user can interact with the design and retrieve other construction data associated with the object or project as a whole that are found in the database from the virtual reality model.
- *DWF*: DWF (Drawing Web Format) is a graphics format for the transfer of drawings over Intranets and the Internet. The application parses the database for the building elements that exist in a project and generates the DWF representation of each element.
- *Estimating*: Generates the elemental IFC cost group objects within the central project database from the design, before returning this information into Excel. The user add costs to the cost groups in Excel and uploads the CSV file into the project database, i.e. adding costs, date and time, etc. to the previously generated cost group objects. Finally, a cost summary of the project can be viewed in terms of both element type and storey.
- *Planning*: Generates the work group objects within the central project database from the design, before returning this information into MS Project scheduling package. Next, the user adds duration and links/dependencies and uploads the CSV file into the project database, i.e. adding the durations and links, to the previously generated work group objects.

3 IFC property type definitions & shared property sets

3.1 IFC object model architecture overview

The IFC Object Model architecture provides a modular structure for the development of model components. The architecture consists of four layers, which use a strict referencing hierarchy. Within each layer a number of grouped modules exist (IAI 1999).

- *Independent Resource Layer*: Independent Resources are ideas that do not rely on classes within the Kernel for their existence, e.g. geometry, cost, etc. Resources form the lowest layer in the IFC Model Architecture and can be used or referenced by all classes in other layers.
- *Core Layer*: The Core contains two sub-layers. The Kernel provides basic, shared concepts in AEC/FM projects and determines the model structure and decomposition. The Kernel includes fundamental concepts concerning the provision of objects, relationships, type definitions, attributes and roles. Kernel classes may only reference classes in the Independent Resources. A Core Extension is a specialisation of classes defined in the Kernel such as Product and Process. Primary relationships and roles are also defined within the Core Extensions.

- *Interoperability*: The Interoperability layer provides specialised, well-defined interfaces for one or more Domain Models. This provides for outsourcing of Domain Model development whilst retaining control over the key structuring and framework requirements of interoperability.
- *Domain Specifications*: Domain Models provide further model detail within the scope requirements for an AEC domain process or a type of application. Each is a separate model that may use or reference any class defined in the Core and Independent Resource layers. Examples of Domain Models are Architecture, HVAC, FM, Structural Engineering etc.

3.2 Type definitions

The IfcPropertyTypeResource defines the basic capabilities for Property definitions that can be attached to objects and relationships. This Resource module enables a "Type" of an Element to be defined, which in turn establishes a standard to be used many times in a project. A standard type is established through the definition of a set of Properties or Characteristics that will be held constant for all occurrences in the project model. Therefore, a single record of these attributes is associated with the Type Definition object rather than with each occurrence of the element.

For example, in the case of doors and windows, an IfcPropertyTypeDef object is created for each type of element and is referenced by those IfcObjects, e.g. IfcDoor, IfcWindow, etc., to which it is associated. The IfcPropertyTypeDef class provides for the capability to define a set of properties at runtime, since they do not need a separate static class definition, and share the values of the set of properties among multiple instances of the same class of objects. In addition, the IfcPropertyTypeDef class possesses a number of attributes. 'TypeDefName' represents the name of the specific Property Type Definition, 'TypedClass' refers to the name of the class for which the object defines a type, e.g. IfcDoor, IfcWindow, etc. and 'GenericType' specifies the IFC generic type designation, e.g. SingleSwing, FixedCasement, etc.

3.3 Shared property sets

Each Type Definition has a reference to an IfcSharedPropertySet. An IfcSharedPropertySet defines a set of type driven properties (and is therefore defined as part of the IFC standard) that are shared by multiple instances of a semantic object, i.e. each semantic object instance (having the same property values) shares by reference a single instance of the IfcSharedPropertySet with particular property values.

The type definition for a single swing door has a reference (in the Sharedproperties attribute) to a 'Pset_DoorSglSwing' shared property set, that in turn, has references to two further shared property sets, 'Pset_DoorType' and 'Pset_DoorPanel', that are held in the 'Hasproperties' attribute list. The 'Pset_DoorType' shared property set contains the common shared values/properties for the specific type of single swing door. These properties relate to information such as the reference ID for the door type within a particular project, specific description of the door type, manufacturer, nominal and rough heights and widths, fire, thermal and acoustic ratings, whether the door is designed for use in exterior walls, etc. Each property is created as IfcSimplyProperty object. IfcSimpleProperty defines an attribute class in which the unique description/name

of the property ('Descriptor') and a reference to the property's value ('ValueComponent') is held. The actual value of the property is held in an IfcMeasureValue object (part of the IfcMeasureResource module) based upon its type, e.g. Ifcdescriptivemeasure, Ifcnumericmeasure, etc.

The properties associated with the doorframe such as description, depth, thickness, etc., are also contained within a shared property set ('Pset_DoorWinFrameType'), which is referenced in the 'Hasproperties' attribute list of the 'Pset_DoorType' shared property set. Again, a simple property and associated measure value object represents each of the properties. An IfcMaterial object represents the material property of the doorframe. IfcMaterial is part of the Material model within the IfcPropertyResource module that provides the facility to specify the material from which an object is composed. The IfcMaterial property together with the simple properties is referenced in the 'Hasproperties' attribute of the 'Pset_DoorWinFrameType' shared property set.

Finally, the 'Pset_DoorPanel' shared property set contains the properties (or references to simple and material property objects) for the door panel/leaf associated with the single swing door type definition, e.g. height, width, thickness, material, etc.

In a similar manner, the type definition for a fixed casement window has a reference to a 'Pset_FixedCasement' shared property set, that in turn, has a reference to a further shared property set, 'Pset_WindowType'. The 'Pset_WindowType' shared property set contains the common shared values/properties (references to simple property objects) for the specific type of fixed casement window, e.g. ID reference, description, manufacturer, dimensions, etc. The 'Pset_DoorWinFrameType' shared property set for the window frame is also referenced in the 'Hasproperties' attribute of the 'Pset_WindowType' shared property set.

4 Product database

For the purpose of the prototype, a product relational database was developed using MS Access. The information relating to each particular type of component/element is held in a single table within the product database, i.e. Door Frames, Door Leaves, and Windows. The structure of each table is primarily based upon the properties specified in the IFC property sets, i.e. Pset_DoorPanel, Pset_DoorType, Pset_WindowType, etc. For example, the Door Leaves table comprises fields relating to information such as manufacturer, product name, height, width, thickness, material, fire, thermal and acoustic ratings, suitability for external or internal exposure, etc. In addition, the Door Leaves and Windows tables have a field to hold a reference to an image file of each specific product.

5 Specification web server extension applications

The Specification application has been implemented as two Netscape Web Application Interface (WAI) Web server extension applications written in Java. WAI is a CORBA-based programming interface that defines object interfaces to the HTTP request/response data and server information. Using WAI, Web applications can be written to accept HTTP requests from a client, process them, and return the response to the client. The end user does not need to know where the data resides or which server is processing it (Netscape 1999).

The user interacts with each application through a specific set of Web pages. The first of the applications enables the user to define the necessary door and window specifications from the suppliers database, which are returned to the user as IFC type definitions, shared property sets, etc., in the form of a STEP Part 21 file. Once the Part 21 file has been returned, the user can upload this file into the IFC object oriented project database, via the second of the WAI applications.

Each WAI application creates and registers CORBA objects (OMG 1999) with the respective Web server, i.e. suppliers database and IFC project database. The Web server associates each of these objects with a specific URL of the form: *http://server//iiop/object_name*. The *server* part of the URL identifies the name of the machine on which the Web server runs, while *iiop* (Internet Inter ORB Protocol) passes request or service replies through the internet, and the *object_name* is the name by which the application is registered and identified by the ORB (Object Request Broker). Once the URL is accessed the Web server delegates the responsibility for processing the HTTP request to the CORBA objects.

5.1 Specification definition WAI application

Fig. 2 shows an overview of the specification definition WAI application. The application receives an initial request from the hyperlink of a selected menu

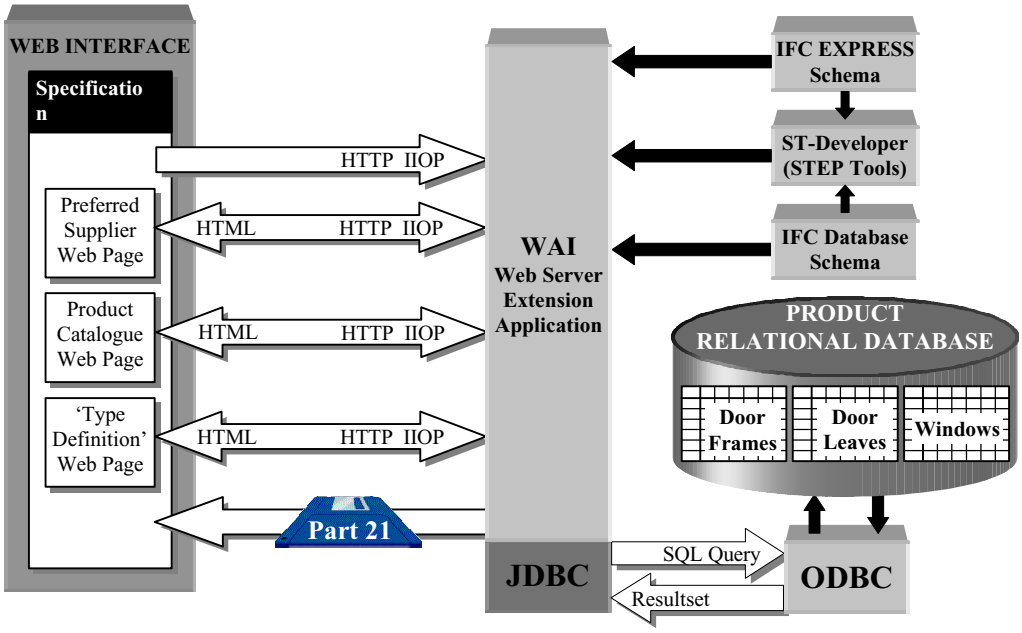

Fig. 2: Schematic overview of the specification definition WAI application

item ('Single Swing Door' or 'Fixed Casement Windows') on the specification Web page (Fig. 3). Using the value of a parameter received from the hyperlink, i.e. using the HTTP GET protocol, the application connects to the product database via JDBC and ODBC, and retrieves the names of suppliers of the type of element selected. JDBC is a Java API for executing SQL statements on relational databases. Via the ODBC API, JDBC enables Java programs to establish a connection with a relational database, send SQL statements, and process the results (Sun Microsystems, Inc. 1999). The supplier names received from the SQL query are placed as options within a selection list of a dynamic HTML page, which is then returned to the user. By submitting the selection of a particular supplier/manufacturer from the list, using the HTTP POST protocol, a further request is sent to the application. Upon receiving this request, the application connects to the products database, retrieves the necessary information such as element description, image file name, etc., and dynamically generates a catalogue HTML page of specific elements for the selected supplier, before returning the page to the user.

In the case of windows, a catalogue of window units is returned to the user. However, from the point of view of doors, the application initially returns a catalogue of door leaves (Fig. 3). Within the HTML catalogue page, each specific element's description is created as a hyperlink to a client-side JavaScript function. Client-side JavaScript statements embedded in an HTML page can respond to user events. When a user selects a particular element in the catalogue HTML page, the JavaScript function copies the element's description into the respective text input form element on the definition Web page, i.e. 'Window' or 'Door Leaf' input box. Following the selection of a door leaf, and the subsequent copying of its

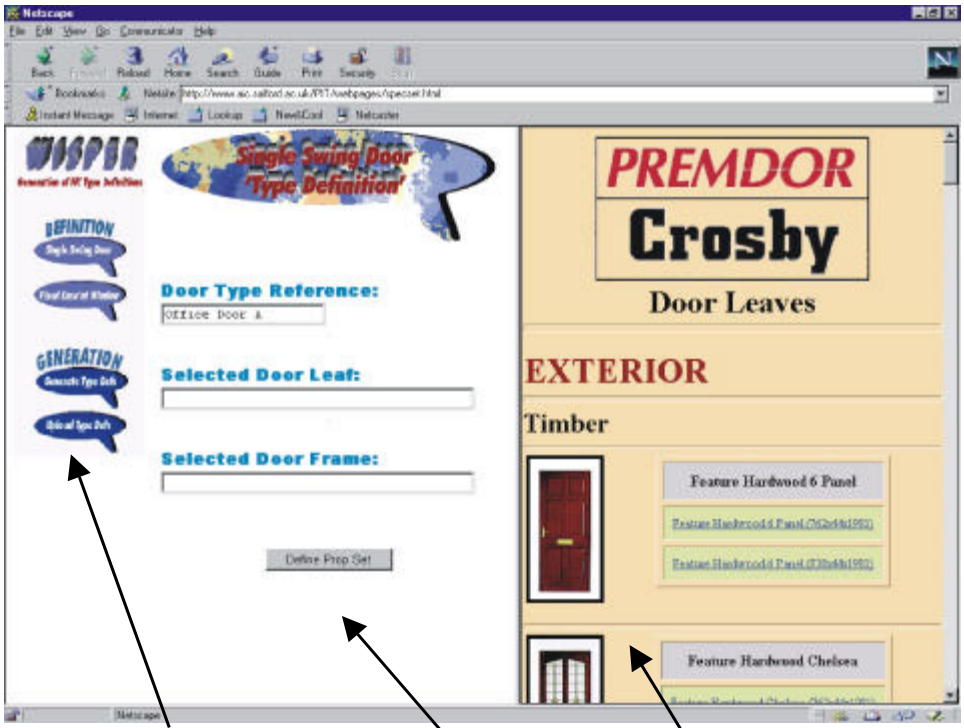

Fig. 3: Specification web pages

description into the 'Door Leaf' text box, the JavaScript function triggers a request to the application to return a catalogue HTML page of door frames suitable for the door leaf selected. Using the door description passed in the request, the application retrieves the descriptions of the suitable door frames from the database, dynamically generates the catalogue HTML page, including client-side JavaScript, etc. and returns the HTML page. When the user selects a door frame from the catalogue Web page, the description of the door frame is copied into the 'Door Frame' text input form element of the single swing door type definition Web page (Fig. 3). Within the type definition Web page, the user specifies the element type reference in the 'Type Reference' input box, i.e., the reference ID property of the 'Pset_DoorType'/'Pset_WindowType' shared property set. Once the user has completed defining a type definition/specification, i.e. specified the element type's reference and selected the element/components associated with the element type, a request is sent to the application via the 'Define Definition' submit button on the type definition Web page. The application responds to the request by holding the type reference and the associated element/component description(s) passed in the request, which provides the opportunity for additional type definitions to be defined.

Following the definition of the required door and window type definitions, a final request is passed to the application to generate and return the type definitions previously specified in Part 21 exchange file format, i.e. an EXPRESS-driven data exchange file specification (STEP Tools, Inc. 1999b; UKCIC 1999). The application acts upon this request by connecting to the product database and retrieving the various properties associated with the element(s)\component(s) specified for each type definition, e.g. dimensions, supplier, fire rating, material, etc. ST-Developer uses the EXPRESS IFC schema and the corresponding IFC Project Database schema (Java classes), together with the properties retrieved from the product database to generate the complete IFC Type Definitions for those specified. The complete IFC Type Definitions are written out to a Part 21 physical file, which is returned to the user.

5.2 Type definitions upload WAI application

The second WAI application resides on the project database Web server. By sending the Part 21 physical file of the previously defined type definitions, via a file input form element using the HTTP POST protocol, the application is delegated the responsibility of uploading the file to the IFC project database. ST-Developer uses the IFC schema and the corresponding IFC project database schema to create the Java objects that correspond to the project instances defined in the Part 21 file. Using ObjectStore utilities, the Java objects are instantiated in the IFC project database. Once uploaded to the IFC project database, the CAD application can download the Type Definitions and attach them to the relevant design elements as they are being designed.

6 Conclusions

This paper has presented a specification application that has been developed as part of the implementation of a new generation of computer integrated environments. The integrated multi-user distributed construction project database environment, WISPER, is based on a three-tier client-server architecture, in which the user interfaces, business logic and database are all kept separate. The development of the object-oriented project database has been achieved via the implementation of the IFC Final Version 1.5 project model. In addition, Web technology was used to develop the user interfaces and enable the communication of distributed applications, i.e. Web pages and Web server extensions, respectively.

The specification application aims to demonstrate the potential for design elements to be specified directly from a product database Web site. From the specification Web pages on the project Web site, the user can access the product catalogues of their preferred suppliers. The user is able to select products directly from the catalogue and define the required specifications. Finally, the user requests the product database WAI application to return a Part 21 file of the IFC Type Definitions for the specifications defined, which is subsequently uploaded into the IFC project database to be downloaded by the CAD application and attached to the respective design elements. Such an application would create a closer link between suppliers and the design process - bringing supplier information early into the design process. This would also potentially enable suppliers/manufacturers to influence the design.

7 References

Alshawi, M., Aouad, G., Faraj, I., Child, T. and Underwood, J. (1998) The Implementation Of The Industry Foundation Classes In Integrated Environments, *CIB W78 Conference*, Sweden, August.

Campbell, R. (1997) Middleware & Architecture: It All Ends in Tiers, *Application Development Advisor*, Vol. 1, No 2. pp. 56-8.

Faraj, I., Alshawi, M., Aouad, G., Child, T. and Underwood J. (1998a) Distributed Object Environment: Using International Standards For Data Exchange, *Computer-Aided Civil and Infrastructure Engineering*, accepted.

Faraj, I.Z., Alshawi, M.A., Aouad, G., Child and Underwood, J. (1998b) The Implementation of the IFC in a Distributed Computer Integrated Environment, *Second European Conference on Product and Process Modelling in the Building Industry*, Watford, UK, October 19-21.

IAI. (1999) *International Alliance for Interoperability*, http://www.interoperability.com/.

Netscape. (1999) *Writing Web Applications with WAI*, http://developer.netscape.com/docs/manuals/enterprise/wai/index.htm.

Orfali, R., Harkey, D. and Edwards, J. (1997) CORBA, Java, and the Object Web, *BYTE*, Vol. 2, No. 10. pp. 95-100.

OMG. (1999) *OMG: Object Management Group*, http://www.omg.org/.

STEP Tools, Inc. (1999a) *The ISO STEP Standards*, http://www.steptools.com /library/standard/.

STEP Tools, Inc. (1999b) *ST-Developer*, http://www.steptools.com/products/stdev/.
Sun Microsystems, Inc. (1999) *JDBC – Connecting Java and Databases*, http://www.javasoft.com/products/jdk/1.1/docs/guide/jdbc/index.html.
UKCIC. (1999) *STEP – ISO 10303*, http://www.ukcic.org/step/step.htm.
WISPER. (1999) *Welcome to WISPER: Web IFC Based Shared Project Environment*, http://www.aic.salford.ac.uk/pit/.