External Product Library System – An Implementation of the Industry Foundation Classes Release 2.0 Library Model

James Nyambayo, Robert Amor, Ihsan Faraj
Building Research Establishment
Bucknalls Lane, Garston, Watford, WD2 7JR, UK
Phone +44-1923-664168. Fax +44-1923-664689
E-mail: nyambayoj@bre.co.uk, trebor@bre.co.uk, faraji@bre.co.uk

and Jeffrey Wix
Jeff Wix Consulting/AEC3 Ltd
30 Northfield Road, Thatcham, Berkshire RG18 3ES, England
Tel. +44-1635 864590 Fax. +44-1635 864590
Email: jeffrey.wix@dial.pipex.com

1 Introduction

This paper describes a generic standards-based product library access system developed as part of a project funded by DETR UK. The aim of the project, developed in conjunction with the Facilities Management domain of the International Alliance for Interoperability (IAI UK Chapter), has been to provide input into the development of the library model for the Industry Foundation Classes (IFC) release 2.0 and to develop a demonstrator product library system to illustrate the use of the model.

Product information is currently available through a number of sources, which include manufacturers, suppliers, and information providers such as product catalogue companies.. There are a number of classification systems both in-house and national that are currently being used by large information providers to organize product information. However this information is often not structured in formats usable by applications, and there are no standard methods in place for accessing this information electronically. The challenge is that of exchanging the product information between the providers and the project environment. For example how can an architect/engineer integrate manufactured product information into their design.

This paper reviews the IFC property set and library models and an implementation of the models to provide access to externally defined manufactured product information. The system described in this paper demonstrates how product information could be structured. It also describes a generic mechanism for searching for product information structured as property sets.

2 Product Information Representation in IFC's

The IFC model is a data representation of the physical and conceptual objects used in the whole life cycle of a building. Most objects are defined as explicit classes, but the architects of the IFC model recognized that it is not possible, or at least not desirable' to try and define all the possible objects in this way. The model allows the classes to be extended using a meta-model known as property set definition. In this model, most of the manufactured product information is defined as property sets that can be related to predefined classes in the model at runtime. From a product library point of view, this flexibility is a powerful feature in that it enables new products to be defined at runtime. However this "soft" information structure poses its own challenges that include:

- How to search for products against the values of their attributes from within the product libraries
- How do applications that use the information do so effectively if it is not explicitly structured.
- How to validate the information. Meta-model based information could comply with the model but not necessarily mean anything.

- How is the information used in the applications that receive it, e.g. CAD applications.

2.1 Property Set and Library Models in IFC Release 2.0

The property set model is the meta-model used to extend the explicit classes in the IFC model. The property set (*IfcPropertySet*) is defined in the *IfcKernel* schema.. Property Sets can be related to predefined classes through the *IfcRelationship* class. A partial model is shown in Figure 2.1. The property set itself does not define any properties, but can contain a list of properties defined in the *IfcPropertyResource* schema. A number of properties are defined in this schema. These are shown in Figures 2.2 and 2.3, and described in detail in [IAI, 1999a] and [IAI, 1999b]. Of particular interest to this paper are the *IfcLibraryReference*, *IfcObjectReference* and *IfcLibrary*.

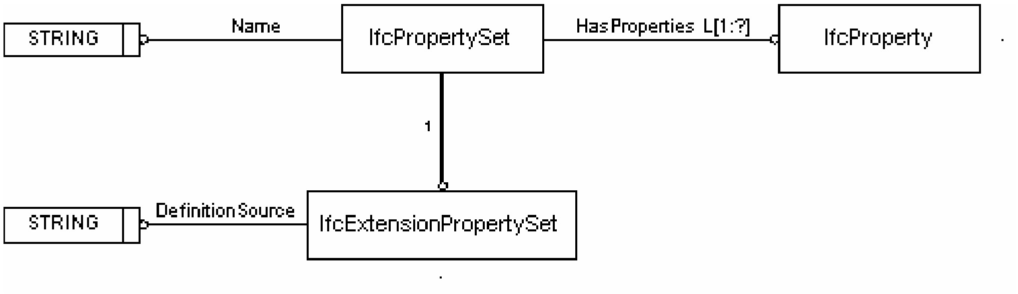


Figure 2.1. Property Set Model in IFC Release 2.0

The main classes in Figures 2.2 and 2.3 are as follows:

- *IfcSimpleProperty* is a property that could have a Real, Integer or String value. This property does not have units.
- *IfcSimplePropertyWithUnit* is a property whose value has a unit assigned to it.
- The *IfcLibrary* is a representation of an external product library. It describes information required to identify a particular external source of product information.
- The *IfcLibraryReference* is a property (subclass of *IfcProperty*), used to provide a reference to an external library (*IfcLibrary*).
- *IfcObjectReference* is a property used to reference other classes defined in the resource layer of the IFC model.
- *IfcEnumeratedProperty* represents properties that are elements of enumerations.
- *IfcPropertyList* is a list of properties.

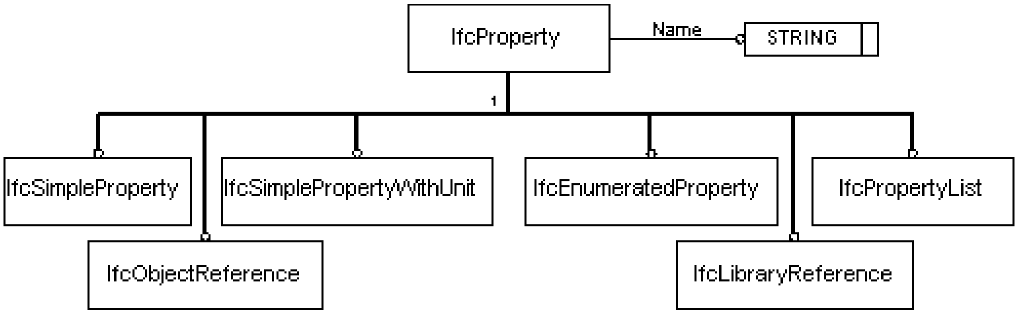


Figure 2.2: Part of Property Resource Model in IFC Release 2.0

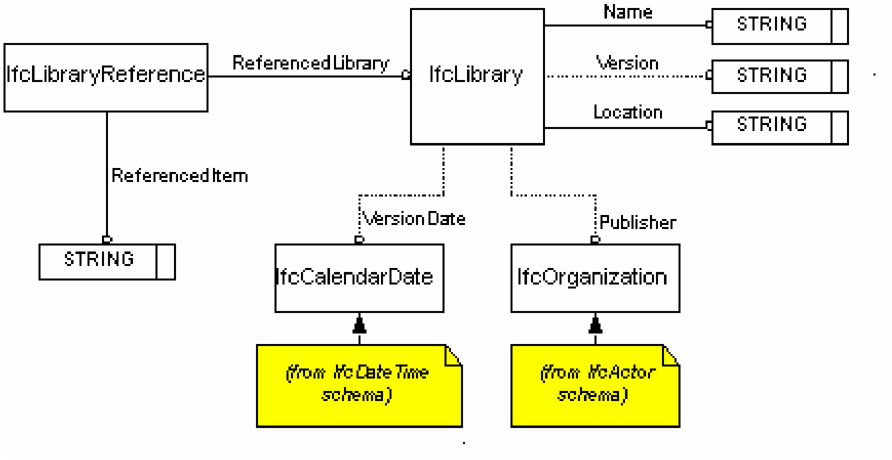


Figure 2.3: Library Model in IFC release 2.0

An illustration of how the model can be used to structure external library product information is shown in Figure 2.4. In this example a building stone (set_Cotswold_Hill_Honey_Bed3_Limestone) is structured as an instance of *IfcPropertySet* (from Figure 2.1) and its properties as instances of the subclasses of *IfcProperty* (from Figure 2.2). The reference to the source library is through *IfcLibraryReference* property. The source library (Cotswold Hill Stone Ltd) is an instance of *IfcLibrary* (from Figure 2.3).

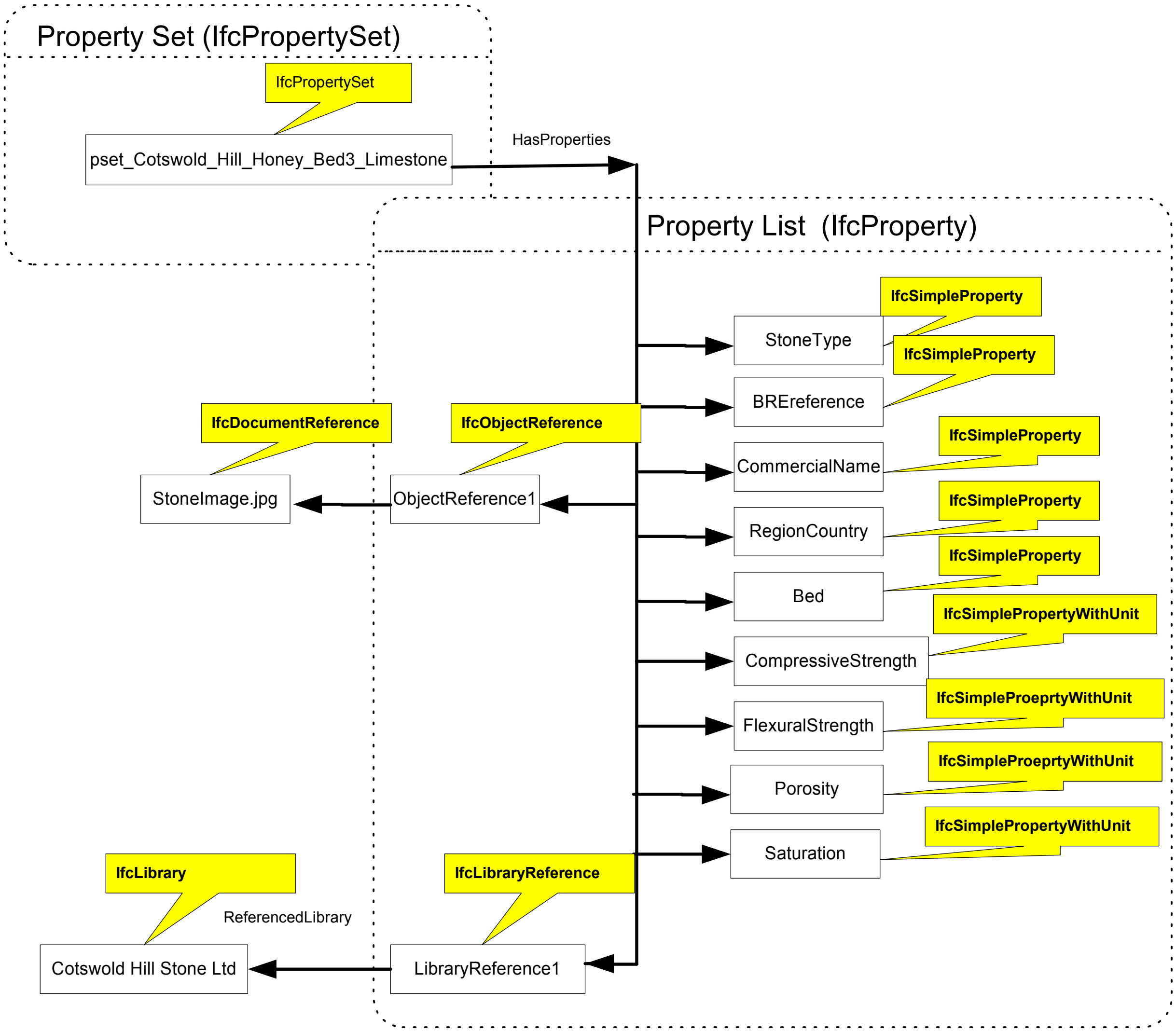


Figure 2.4: An example of a stone product structured as an instance of IfcPropertySet

3 Architecture of the product library demonstrator

The External Product Library demonstrator is an implementation of the property set and library models to demonstrate a possible use of the model to access manufactured products defined as property sets. The product library demonstrator is a web-based system comprising an object-oriented database (ObjectStore), a product server and a web interface. The objective of this system has been to model external libraries and their products (using IFC's) within an ObjectStore database and develop methods for searching for products defined as property sets against values of their attributes.

The structure of the system was inspired by the need to meet the above usage scenarios i.e. enabling users to search for products existing in external libraries and enabling the use of the external library products with design tools. The structure has been influenced in part by the ARROW system [Newnham et al. 1997] and CONNET [Turk and Amor 2000]. Lessons have also been drawn from the WISPER project [Faraj et al 1998]. ARROW is based on explicit class extensions of the IFC model. For example an *ArrowDoor* class is an explicit class that extends the *IfcDoor* class. While parametric searches for products using the ARROW system is efficient, the system is rigid. It cannot easily be extended to cater for new products or variations in attributes. CONNET uses a combination of catalogs and property sets (not IFC-based) to define product information. This approach resembles the real world more closely than the ARROW approach. However it does not address the exchange of product information between applications.

The system developed in this project is meta-model based, enables any product information to be defined, searched for and provides for the exchange of information between applications.

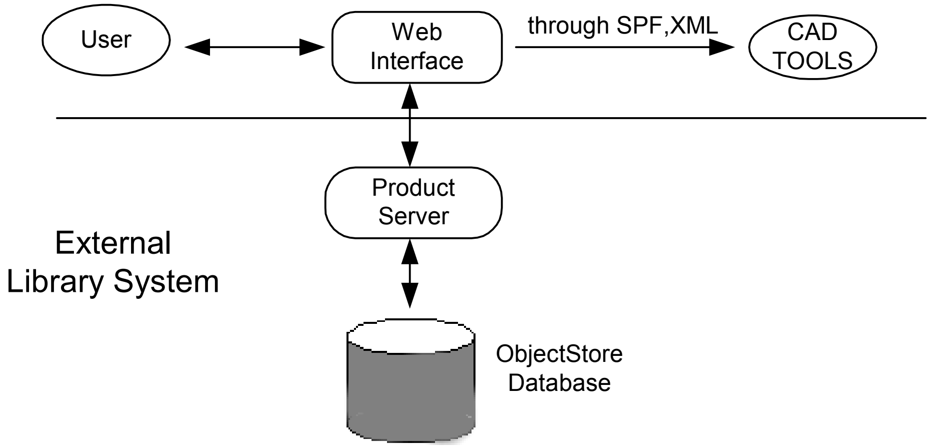


Figure 3.1: Library Access System Architecture

The architecture recognizes that product information existing in external libraries are potentially non-uniform in structure and therefore using the property set model represents a flexible way of accommodating these variations.

3.1 Database

The release 2.0 model was loaded into an object-oriented database (ObjectStore) as Java classes. A number of external libraries including their products were loaded into the database as *IfcLibrary* and *IfcPropertySet* (Figures 2.1 and 2.3) instances respectively. Building stones from the British Stone collection [URL1] and BRE stone listing [URL2] were used as test data. In this case building stones were defined as property sets (*IfcPropertySet*) and the stone quarries as external product libraries (*IfcLibrary*). The properties such as compressive strength, flexural strength, were defined as *IfcSimpleProperty* or *IfcSimplepropertyWithUnits.* The values of the properties were defined as selection of Integers (*IfcInteger type*), Real values *(IfcReal type*) or String values (*IfcString type*). The library of annotated Java classes was used to access the data in the ObjectStore database. The access to the data is discussed in more detail the next section.

The data was indexed in collections for easy access. Working with multiple libraries within the ObjectStore database helped us to simulate distributed product libraries.

3.2 Product Server

The product server is a set of Java APIs that use ObjecStore API's, library of annotated Java classes of the IFC release 2.0 model to provide an interface to the data stored in the IFC based data stored in the database. It has the following functionality's:

- Building queries from the parameters specified in the user interface and handles a request for data from the interface and builds appropriate ObjectStore query, pass the query to the database, and returns the product data satisfying the search criteria.
- Formatting output to specified formats such as HTML for display purposes, SPF and XML for exchange purposes.
- Adding and editing data within the database. Adds data by instantiating the appropriate *IfcPropertySet*, *IfcLibrary*, *IfcProperty* classes to define an instance of a property set.
- Providing standard API's for access from other applications or systems.

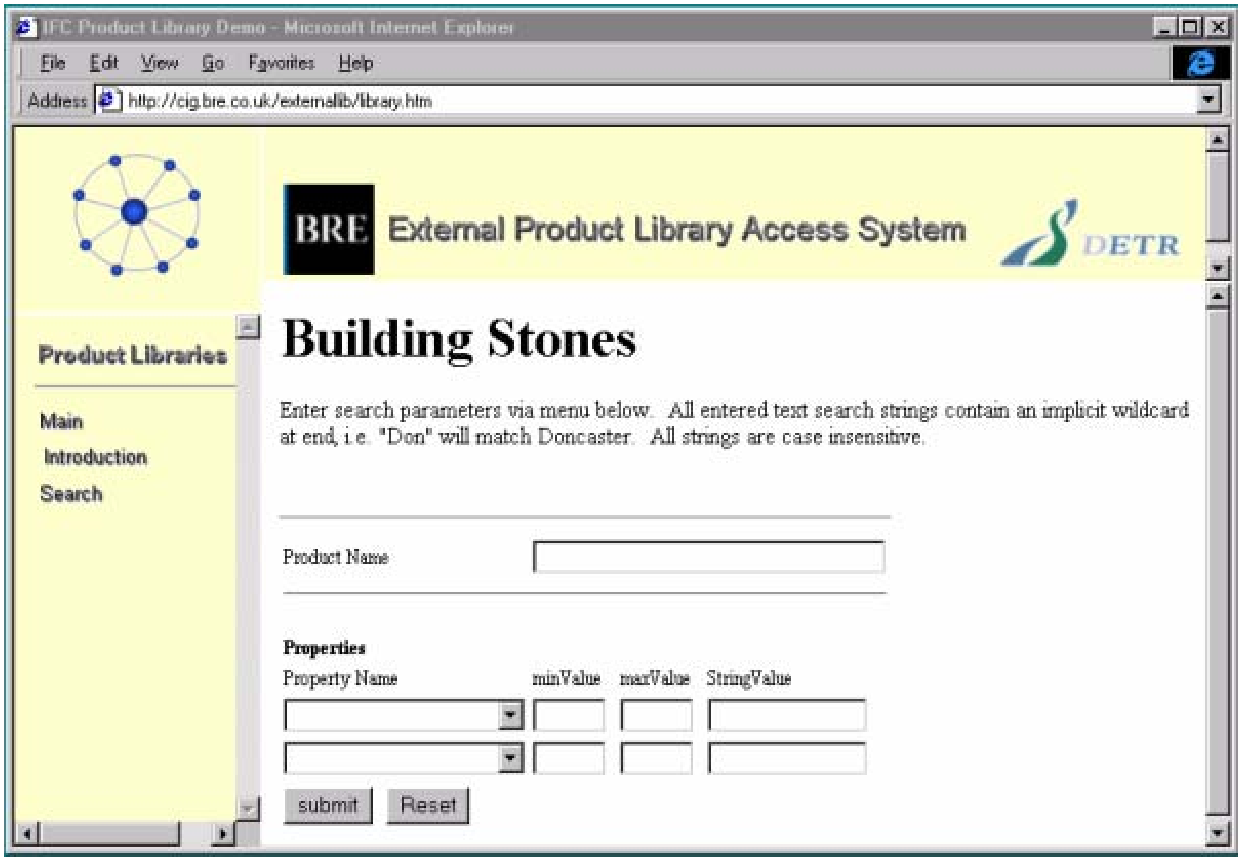
The main feature of this server is the ability to search property sets against the values of their properties regardless of the type of properties or the data types of the attribute values. It enables products defined at runtime to be identified. The search returns the products satisfying the search criteria as well as the attributes and attribute values of the product, and all the other information such as documents related to the product

3.3 Web Interface

The web browser provides the interface for users to search for products across the libraries. It provides an interface for users to specify search criteria, such as the product name and attribute values it must satisfy. The attributes could be anything ranging from dimensions, materials to the library to which the products belong. More than one criterion can be specified. It displays the search results. A typical search screen is shown in Figure 3.3.1.

3.3.1 Searching for product Data

The system was design to provide a generic access to products, without the need for prior knowledge of the attribute, or the need to fix attributes. The search page could be automatically generated for each product type to provide a current listing of attributes. This is intended to cater for the fact that attributes are not predefined, or for any changes that might occur during the use of the database.



http://cig.bre.co.uk/externallib/library.htm

External Product Library Access System

Product Libraries

Main
Introduction
Search

Building Stones

Enter search parameters via menu below. All entered text search strings contain an implicit wildcard at end, i.e. "Don" will match Doncaster. All strings are case insensitive.

Product Name

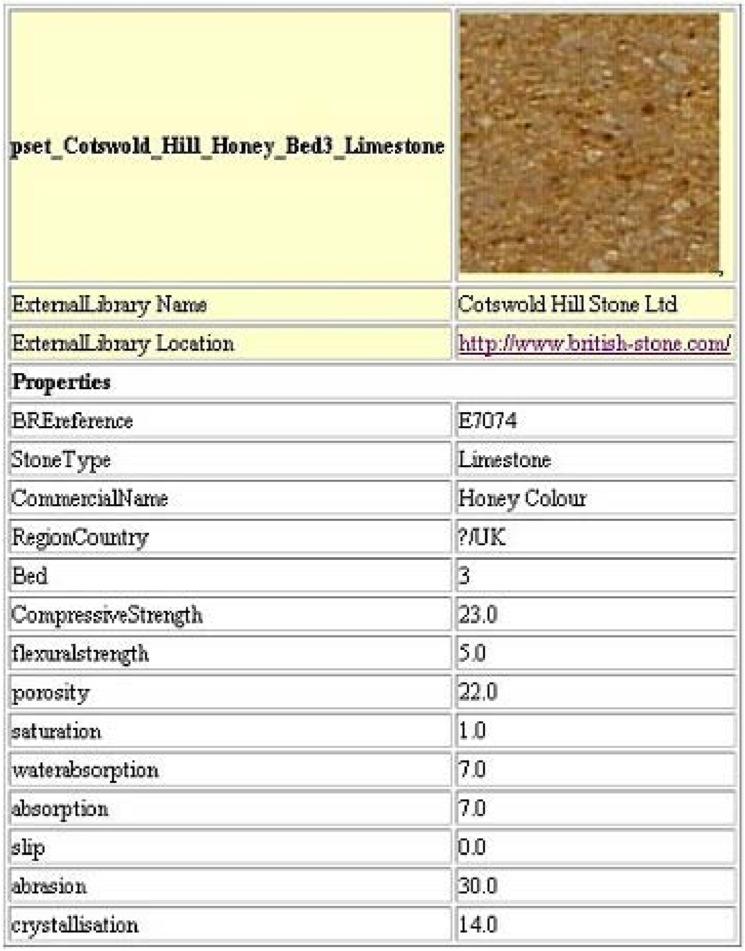
Properties

Property Name minValue maxValue StringValue

Figure 3.3.1: Search Screen

In this project Building stones from Building Stone collection were used as test data. Building stones have a number of attributes such as strength, porosity that can be searched against. The search returns all the stone properties including images of the stone's finish. These images could be used to render building components within a CAD system

The interface for adding information into the database is generic. It enables any product information to be added to the system.



pset_Cotswold_Hill_Honey_Bed3_Limestone	
ExternalLibrary Name	Cotswold Hill Stone Ltd
ExternalLibrary Location	http://www.british-stone.com/
Properties	
BREreference	E7074
StoneType	Limestone
CommercialName	Honey Colour
RegionCountry	?/UK
Bed	3
CompressiveStrength	23.0
flexuralstrength	5.0
porosity	22.0
saturation	1.0
waterabsorption	7.0
absorption	7.0
slip	0.0
abrasion	30.0
crystallisation	14.0

Figure 3.3.2: Typical Product Information held in the system

4 Future Links to Manufacturer Libraries and Product Information Catalogs

The system could be used in conjunction with manufacturer databases and product information catalogue systems to provide access to distributed product libraries. A typical manufacturer database would have a structure shown in Figure 4.1. This generic database structure enables any product data to be structured.

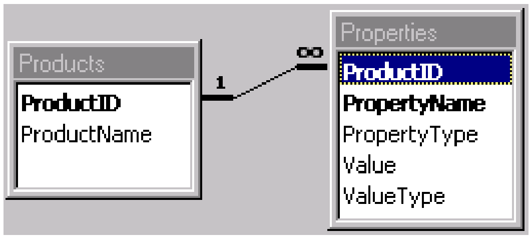


Figure 4.1 A typical Manufacturer Database Structure

The role of this system in this usage scenario is to provide a generic search mechanism across the distributed product libraries is illustrated in Figure 5.1. In this diagram “Product Library 1” to “Product Library 3” represent the distributed product libraries while the “External Library Access System” is the system described in this paper. XML and the Internet could be used to provide a low cost link between manufacturer databases and the system.

5 Future Links to Design Tools and IPDB Systems

The system could be used with design tools and Integrated Project Database (IPDB) Systems to provide access to manufactured product information from within the project environment. This usage scenario is also illustrated in Figure 5.1.

Product information could be exported to design tools and IPDB systems in STEP physical file format, XML or Standard Data Access Interface (SDAI). This would provide seamless access to product information defined as property sets.

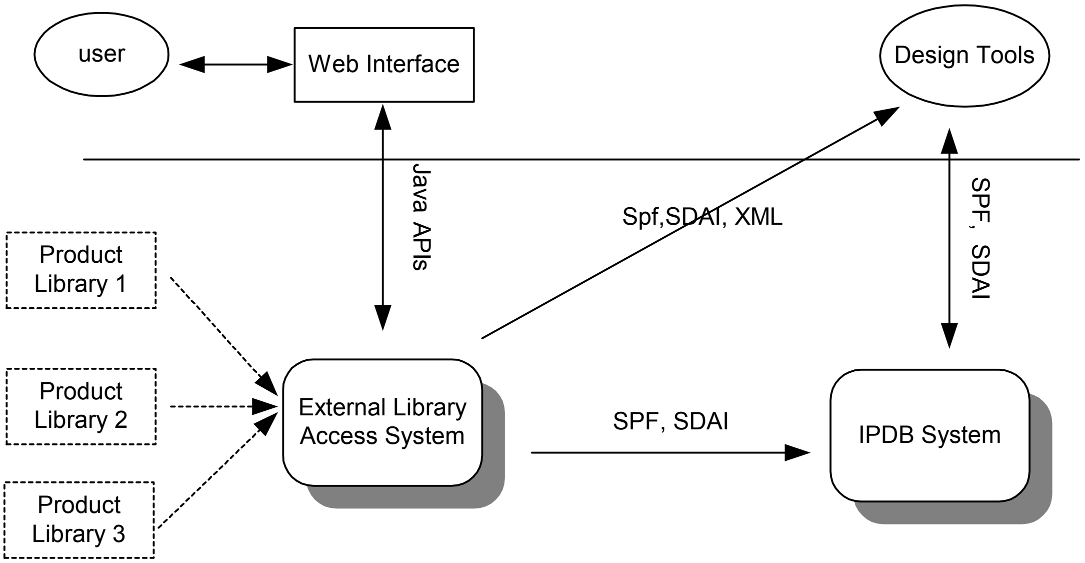


Figure 5.1: Integrating External Product Libraries into the Project Environment

6 Lessons Learned from the implementation

The library and property set model is a first step towards providing access to externally defined products. The system described in this paper demonstrates generic access to information defined as property sets. Lessons learned during the implementation include:

- The model seems to satisfy the basic requirement of identifying the source of product information defined as a property set, by using the *IfcLibraryReference* property.
- The property set itself does not have sufficient attributes to adequately identify it. For example the only the “Name” attribute exists.
- Searching for information defined as property sets against the values of attributes (properties) is computationally inefficient
- The level of abstraction within the model often too high, as a result effectively associating the manufactured product information defined as property sets to the predefined part of the model is not easy.
- There is currently no way of aggregating property sets belonging to library. A catalogue could be defined in the library model.
- There is a need to uniquely identify a library, using names and string labels does not seem to be adequate.

8 Conclusions

This system demonstrates the use of the IFC release 2.0 model in providing access to external product libraries. In particular it illustrates the ability to define manufactured products as property sets and a generic mechanism for searching for products against the values of their attributes.

The Library model in IFC release 2.0 represents a first step towards integrating external product libraries into the project environment. This model is still evolving, and it is expected that more comprehensive features will be incorporated in future releases of the IFC model. The issues arising from the development of this system have provided valuable feedback to the development of the library model within the IFC's and, in particular, to the extension of capabilities within the proposed IFC Release 2.X model.

7 Further Developments

The system, on completion, will be able to export product data to XML and STEP physical file formats. It is expected that such a system would eventually form part of an integrated project environment supporting the project life cycle.

9 REFERENCES

1. Faraj, I., Alshawi, M., Aouad, G., Child, T. and Underwood, J. (1998), The Implementation of the IFC in a Distributed Computer Integrated environment in R. Amor (ed) ECPPM '98

2. IAI (1999a) IFC Object Model Architecture Guide, Vol3: IFC Object Model Reference, IFC Release 2.0 Documentation 1999

3. IAI (1999b)I FC Specifications Development Guide: Appendix F, IFC Properties and Property, IFC Release 2.0 Documentation 1999

4. Newnham,L.; Amor R. and Parand, F., (1997): Gaining Quality Manufactured Product information through Arrow.; IT-CPR-97, Cairns, Australia, 9-11 July 1997.

5. Newnham, L. and Amor, R. (1998), Translation of Manufacturer's Product Data for the Arrow Product Search System, in Amor, R. (ed) ECPPM '98.

6. Nyambayo J. (1999), Client Report CR 104/99, development and adoption of standards for interoperability: external library Access Model, interim Report 1.

7. Turk, Z. and Amor, R. (2000) Architectural foundations of a construction information network, International Journal of Construction Information Technology, 7(2). Pp. 85-97.

8. URL1 http://www.british-stone.com/

9. URL2 http://projects.bre.co.uk/stonelist/