Interoperable and Extensible Design Information Modelling

Qizhen YANG and Lu CUI
Singapore Institute of Manufacturing Technology, Singapore

Keywords: IFC, information modelling, interoperability

Abstract: Modelling of interoperable and extensible design information is one of the key issues in computer-aided architectural design. IFC technology provides standardised mechanisms for the development of such information models. By using the IFC dynamic definition extension mechanism this paper presents a method for IFC-compliant design information modelling for architectural CAD objects. The method has been implemented as an add-on toolkit to the Architectural Desktop CAD package. The use scenarios of the toolkit are discussed in the paper for CAD property modelling, property database management, and interoperable design information modelling with property set extensions.

1 INTRODUCTION

The development of a building project usually involves multi-disciplinary participants for building design, analysis, procurement, construction, facility management, etc. Each participant requires creation, consuming, exchange and sharing of design information with other participants, ideally in a neutral format independent from disciplinary application tools. The design information is usually packed within the computer generated geometric models – CAD models (Autodesk 2001; Graphisoft 2002). However, a CAD model with only geometric representations and design parameters can hardly fulfil the information needs from other life-cycle processes, apart from CAD process, involved in the project development. In order to satisfy different user requirements, CAD models should be developed with interoperable and extensible design information.

An interoperable and extensible design information model consists of not only CAD geometric representations and parameters of architectural design objects, but also a wide variety of non-graphical and project-related information, with standardised definitions for common semantics of design concept. The Industry Foundation Class (IFC) technology (IAI 2000a) provides a way to develop such a model with interoperability and extensibility. IFC specifies the standardised definitions for the uniform representation and exchange of project information to facilitate the software interoperability. It also provides approaches for extending the IFC object model to
Digital Design

cater for the application-specific information needs from each project participant. Specifically, the IFC Property Set (Pset) is defined as a mechanism for dynamic definition extension of the standard IFC model (IAI 2001).

The research and development of IFC-compliant information models in various AEC application domains have been conducted for a while. And the efforts have generated critical impacts on the ways of representing and exchanging application-specific information between AEC software systems. One of such efforts was reported (Froese et al. 1999) for the development of IFC-based information models for cost estimation and scheduling in project management. A trial implementation was conducted to test and evaluate these models for representing and integrating product, work process, estimating, and scheduling information. The benefit and use of IFC product models to boost the efficiency of the construction process was described in (YIT 2002). The IFC modelling approaches were also employed in the research (Hassanain et al. 2001) for the development of a maintenance management model. From this study, several extensions to the IFC object model have been proposed. Product modelling by both STEP (ISO 1994) and IFC technologies was discussed in (Debras et al. 1998), which also presented a case study illustrating the CAD systems interoperability based on the STEP and IFC product models. Another development for achieving interoperability among software tools was found in (Karola and Lahtela 2000). Its software implementation allows exchange of IFC-compliant information for building services. To support the building code checking applications, IFC-compliant information models have been developed and implemented in software systems (Yang and Li 2001). The IFC information models were also used for performance-based energy code compliance checking in another research (IAI 2000b).

Although many other success projects and researches on IFC-based interoperable information modelling and on other IFC development can be found in literature, the IFC technology has not achieved widespread adoption on a general level in the industry (Stouffs and Krishnamurti 2001). For example, the methods and software implementations of the IFC Pset extension mechanism on commercial CAD platforms are required in the real life projects for generating the extended IFC models with both the IFC-defined and user-defined design properties to satisfy the specific information needs from different disciplinary software applications. But currently, such implementations in the commercial CAD packages, such as in the Architectural Desktop (ADT) system, are not available.

This paper describes a method to develop IFC-compliant design information models with interoperability and extensibility for 3D architectural CAD objects by the use of IFC Pset extension mechanism. It also discusses the software implementation of the method to support the generation of the extended IFC information models from ADT architectural designs. The paper is divided into five sections. The method for the extended IFC information modelling on the ADT CAD platform is described in Section 2. It is followed by the software implementation of the method in Section 3. Applications of the software prototype are illustrated by a series of use scenarios in Section 4. The conclusion and further development of the method are sketched in Section 5.
2 DESIGN INFORMATION MODELLING

2.1 Modelling Design Properties

As discussed in the previous section, the CAD objects embedded with supplementary design information are required in many real-world projects. The supplementary design information may include any additional data sets structured for the better description of a design, such as material, cost, quality and performance criterion data, etc. In this research, the supplementary design information of CAD objects generated from ADT is modelled by ADT entity properties. An ADT entity property is defined by attributes and quantitative constraints of attributes. An attribute has a name and a data type. All attributes of an entity property together describe the semantics of the design property being defined. A quantitative constraint contains a set of methods to control the validity of the attribute types and values that ADT CAD can support.

An ADT property object is instantiated using an entity property as a template. In general, quantitative constraints of an entity property are not presented in the property objects of this entity property. Instead, they are implemented as constraint algorithms in the software add-on tool of ADT and associated with the respective attributes in this entity property. Whenever a property object is instantiated by assigning attribute values to the attribute names of this entity property, these quantitative constraint algorithms will be invoked to ensure the auto-extracted or user-entered attribute types and values compliant with the data type and data scope defined in the entity property for these attributes. Table 1 gives the definitions of an ADT entity property for work surface area of office furniture. Table 2 shows some of the work surface area property objects instantiated from Table 1.

<table>
<thead>
<tr>
<th>Table 1 Definitions of WorkSurfaceArea Entity Property</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attributes</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>PropertyName</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>Unit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2 Property Objects for WorkSurfaceArea</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Name</strong></td>
</tr>
<tr>
<td>WorkSurfaceArea 1</td>
</tr>
<tr>
<td>WorkSurfaceArea 2</td>
</tr>
<tr>
<td>WorkSurfaceArea 3</td>
</tr>
</tbody>
</table>
2.2 Linking Properties with CAD Models

Similar to the concept of IFC Pset, the ADT entity properties are grouped as ADT Property Sets (ADT Pset). An ADT Pset can be associated with one or more ADT CAD objects. This association is established in the ADT modelling environment by direct linkage relations between them. A linkage has two nodes: a Pset node and an ADT design object node. Each Pset node points to an IFC-predefined or project-defined Pset object. Each design object node points to a 3D CAD object in the current ADT design space. When a linkage is instantiated between a Pset and a CAD model, this relation will be stored in a data structure recognisable and processible by the ADT system. Figure 1 shows a linkage between an ADT Pset and CAD model of an air conditioner.

![Figure 1 ADT Pset Modelling and Linking with a CAD Model](image)

2.3 Mapping Property Definitions

In order to make the ADT property information compatible with the IFC model, property mapping strategies are developed, so that the ADT properties can be translated, then directly exported into IFC files by using the IFC translation facility from ADT system. The property mapping from ADT to IFC definitions consists of three steps:

- Mapping of the attributes for an entity property;
- Mapping of the linkages between properties and ADT models; and
- Mapping of the quantitative constrains for data type and scope control.

All the mapping strategies defined have been implemented in the ADT add-on toolkit and verified by the ADT-exported IFC files. Figure 2 shows examples of the mapped IFC objects in UML (Rational 2000) for the properties and linkage of the air conditioner illustrated in Figure 1.
2.4 Managing Property Database

A property database is modelled to store and manage the pre-defined and newly created property and Pset definitions in the XML (Ahmed et al. 2001) format for ADT. The property database is composed of XML documents. Each XML file defines one Pset with a collection of properties. A rich set of pre-defined properties has been specified, stored and managed by the property database. The properties in the database are extensible and customisable through the GUIs provided with the ADT add-on toolkit.

The management capability of the database includes:

- To store and manage pre-defined standard vocabulary in an XML document for use in property and Pset creation;
- To store and manage default Pssets with their inclusion properties in XML documents;
- To validate the properties and Pssets to be stored for compliance with the vocabulary;
- To search and compare properties and Pssets with the existing ones in the database; and
- To provide selection, opening, retrieval, and deletion operations to manipulate the property XML elements in the database.
SOFTWARE DESIGN AND IMPLEMENTATION

3.1 Requirement Analysis

The main function requirements of the toolkit are identified as:

- Defining the semantics of entity properties and Psets;
- Instantiating these definitions with design parameters; and
- Structuring the property information for export/import of extended IFC files.

These functional requirements are captured and analysed by the use case diagrams of UML. Figure 3 shows one of these use cases for Pset definition.

![Figure 3 UML Use Case Diagram for Pset Definition](image)

3.2 Toolkit Design

The toolkit is designed for use with ADT system to provide the following services:

- To define the semantics of entity properties and Psets for ADT models;
- To instantiate the property definitions with design parameters extracted automatically from ADT models or entered by users;
- To validate the quantitative constraints of property values;
- To provide facilities for linking Psets and ADT objects or styles;
- To map property definitions between ADT and IFC formats; and
- To structure property information for export/import of extended IFC files.

Three functional modules are developed to implement these services.
**Interoperable and Extensible Design Information Modelling**

**Property Definition Module.** This module is used to define the semantics of ADT properties and to specify the constituents of ADT Psets for CAD models. It provides customised ADT assistance interfaces to define/modify the entity properties using a standardised vocabulary library provided. One of the UML sequence diagrams for Pset definition is shown in Figure 4. The module also supports selection of properties for inclusion within a Pset. The quantitative constraints of the entity properties identified have been implemented in the module.

![Figure 4 Sequence of Operations for Defining a Pset](image)

**Property Instantiation Module.** It provides the functions for automatic extraction of property attributes from design parameters in ADT models. If the design parameters are not available from a CAD model, the customised interfaces in this
Digital Design

module can be used for user-keyed-in property attributes. As mentioned early, the keyed-in attributes are validated by the implemented quantitative constraint algorithms during instantiation. Only valid property types and values are accepted and embedded, together with the Pset and property definitions, into the CAD model by using the linkage relation. One of the property instantiation interfaces of the module is shown in Figure 5.

<table>
<thead>
<tr>
<th>Pset_Basin</th>
<th>color</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>double tab opening</td>
<td>False</td>
</tr>
<tr>
<td></td>
<td>manufacturer</td>
<td>MuthootCeramic</td>
</tr>
</tbody>
</table>

Figure 5 Property Instantiation

**IFC Import/ Export Enhancement Module.** This module improves functionality of the existing IFC translator in the ADT package by structuring special Psets to hold the drafting property objects and other ADT meta data (configuration, scale, etc) for ADT models. The mapping algorithms are implemented in this module as well. Another major function of the module is to make all the Pset extensions of both ADT standard and user-defined CAD models exportable to the extended IFC files. It also re-uses the property information in the extended IFC files to restore the drafting and configuration characteristics of these CAD models when they are imported from the IFC files to the native ADT workspace.

3.3 Software Development

The toolkit is implemented using ADT APIs on the ADT platform. It is developed to support the user access from the ADT design environment. A vocabulary library in XML is incorporated with the toolkit to provide standard terms for property and Pset naming. Functions and procedures are developed to implement the tool services mentioned in 3.2 and in Figure 4. All the supplementary design information, including property names, descriptions, values, units, Pset names and Pset constituents are defined through two forms of the toolkit. Both forms are implemented with a set of dialog boxes, combo boxes, text fields, list boxes, and buttons to enable user’s effective modelling of supplementary design information through properties and Pssets defined.

4 APPLICATION SCENARIOS

4.1 CAD Property and Pset Modelling Scenario

This scenario demonstrates the ADT property and Pset modelling by use of the toolkit. The modelling of a non-standard ADT CAD object, such as a basin, is taken
as an example in this scenario.

The 3D CAD geometry of the basin is modelled by the ADT system and packed with multi-view blocks to fully describe the graphical representations of the CAD object. By right-mouse click on the CAD model, the add-on tool is invoked from within the ADT environment to define the entity properties and Psets for the basin object. A snapshot of the Pset definition interface is given in Figure 6. If the user cannot find a property he needs from the existing property definitions retrieved from the database and listed in the left column of the interface, he can use the Add/Modify button in Figure 6 to create a new one. Otherwise he selects those needed and adds to the Pset listed in the right column of the interface for the design object he is working on. Figure 7 shows the definition interface for property modelling.

![Figure 6 Creation of Pset](image1)

![Figure 7 Creation of Entity Property](image2)

4.2 Property Database Management Scenario

In this scenario, a property database containing a collection of property sets and a vocabulary XML document is used to illustrate the property management functions of the toolkit.

**Pset storage and management.** An XML document for a Pset is created or modified by a user. Whenever the user creates a Pset for a CAD model, the toolkit will search the database for a match. If found, all properties from the matched Pset will be retrieved from the property database and re-used for the CAD model. Otherwise, facilities will be invoked by the user to assemble his own Pset with properties stored in the database. Once the user completes the definitions for a new
Digital Design

or modified Pset, the tool retrieves the definitions of all the specified properties for this Pset and populates them to an XML instance file. The file is then stored in the database.

**Property validation.** The pre-defined standard vocabulary is supplied for naming properties and Psets. The vocabulary is extensible and editable. Suppose a user is creating a new property for the basin CAD object using Figure 7. If he cannot find a suitable one from the existing list in the upper selection box in Figure 7, he may create a new one by defining a property name, description, data type, unit, and default value using Figure 7. The property name has to be selected from the vocabulary, as it will be validated later when the property is stored in the database. To do so, functions are provided with the database management facility for searching and comparing the property names with those defined in the vocabulary.

**Database manipulation.** All XML elements in the property database are maintained by the tool for retrieval, modification, deletion, and adding manipulations. The tool also provide facilities for searching and loading the user-selected Psets in database to an ADT CAD drawing database once a linkage between the Pset and the CAD model having been established.

4.3 **IFC-Compliant Design Information Modelling Scenario**

This scenario describes how to build an IFC-compliant design information model with Pset extensions for the basin CAD object.

The basin CAD model is generated by ADT system. The properties and Psets are then specified for the CAD model following the procedures and function invocations from the toolkit (refer to 4.1 for details). The design characteristics for the specified properties are extracted from the CAD model or defined by its designer using the toolkit interfaces shown in Figure 8.

![Figure 8 Design Information Modelling with Pset Extensions](image-url)

Figure 8 Design Information Modelling with Pset Extensions
Once the property attribute instantiations are completed and the CAD file saved, a linkage between the CAD object and the Pset object is recorded and stored, with all defined properties, into the basin drawing database. The property attributes can be modified in the ADT environment. Besides these user-defined Pset objects, some default Psets for CAD drafting and ADT meta data may also be linked with the CAD model. The mapping algorithms are then invoked to make all the user-defined and system-default Pset extensions exportable to an IFC file for the basin design. If the basin model in IFC is later imported back to CAD, the embedded properties will be re-used to set the basin CAD model by clicking on the Model Regenerate button in Figure 8.

If only the ADT IFC facility is used, when this non-standard basin CAD model is imported from an IFC file it will loss some of its original drafting features, such as the colour, layer name, etc. The model regeneration facility of the add-on tool enhances the IFC handling capability of ADT.

5 CONCLUSION

This paper presents a new approach for modelling the IFC-compliant design information models with interoperability and extensibility for architectural CAD objects. These models can provide effective information support for various product life-cycle software applications, as the information required can be encapsulated and extracted from the models with unambiguous semantic definitions. The development of an ADT add-on toolkit, as a software implementation of this method, has also been discussed. Three application scenarios demonstrate the capabilities of the toolkit in providing the services for modelling the ADT properties and Pssets, associating Pssets and CAD models, mapping of properties from ADT to IFC, managing the property database, and generating the interoperable design information models with IFC Pset extensions for ADT objects.

Future development is needed to enhance the IFC-compliant information modelling method and toolkit to encapsulate CAD object behaviours and external references to other digital product information resources.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Rajesh Babu for his co-operation and support in this research.

REFERENCES


