GENERATING IFC VIEWS AND CONFORMANCE CLASSES USING GTPPM

Ghang Lee[1], Charles Eastman[2], and Rafael Sacks[3]

ABSTRACT

This paper discusses the possibility of using the GTPPM (Georgia Tech Process to Product Modeling) method, which was initially developed as a systematic approach to product modeling, as a method to generate IFC views or CIS/2 conformance classes. A view is a valid subset of a product model. The view generation method is discussed in the context of the IFC Information Delivery Manual (IDM).

KEY WORDS

IFC view, conformance class, information delivery manual (IDM), information requirements, GTPPM, ISO STEP

INTRODUCTION

The structure and content of any data model are dependent on the information use cases within its Universe of Discourse (UoD). Any data modeling process (including product modeling) begins with specification of targeted information use cases and data model requirements (Eastman 1999; Elmasri and Navathe 2004).

The ISO-10303 STEP modeling process conforms to the general data modeling approach. It begins with Application Activity Model (AAM) development. In this step, product modelers collect and specify requirements for the product (data) model usually as a form of a process model. IDEF0 (NIST 1993a) is the most commonly used method for developing AAMs today. Based on the specified AAMs, a product model is developed and later integrated with other parts of a standard data model. However, current product modeling practice does not identify the intended information use cases, but rather develops a product model based on a single high-level process model without any link to specific information items required by the constituent processes. The modeling requirements and intentions remain unclear and it becomes impossible to validate the completeness of the resultant product model rigorously. Moreover, the lack of explicit definition regarding information use cases leaves product model implementers (e.g., translator developers, product lifecycle management system developers, etc.) with multiple possible ways of implementing a subset model for any specific information use case, which makes data exchange between different systems more difficult than necessary.

[1] Assistant Professor, Department of Architectural Engineering, Yonsei University, 134 Sinchon-Dong, Seodaemun-Gu, Seoul 120-749, Korea (Republic of), Phone: +82-2-2123-2780, Email: glee@yonsei.ac.kr
[2] Professor, College of Architecture / College of Computing, Georgia Institute of Technology
[3] Senior Lecturer, Civil and Environment Engineering, Technion Israel Institute of Technology

An ideal way to resolve this problem is to specify information requirements explicitly in the initial requirements collection phase when a product model is developed and reuse the requirement specifications as implementation guidelines. GTPPM, a product modeling method to capture information requirements from multiple use cases and derive a product model from them, was developed to support such a process (Lee 2002; Lee 2004; Lee et al. 2006a; Lee et al. 2006b; Lee et al. 2002). However, no standard product model that we are aware of has been developed using GTPPM itself or in a similar manner.

The lack of explicit description of information requirements motivated standard product modeling organizations such as ISO STEP, AISC, and IAI (AISC 2004; IAI 1994; ISO TC 184/SC 4 2004) to develop separate specifications of valid subsets of a product model for specific information use cases after product models were developed. ISO STEP and AISC, which respectively develop and maintain ISO STEP Parts and CIS/2, call them *conformance classes* (Crowley 2001). IAI, which develops and maintains IFC, calls them *IFC views* (or *exchange requirements*) similar to the view in relational database – a valid subset of a data schema. An Information Delivery Manual (IDM) specifies the definitions and the requirements of an IFC view (Wix 2005b).

This paper discusses the possibility of using GTPPM in a systematic way to generate IFC views using GTPPM. It first briefly describes GTPPM and the structure of an IDM , and then describes how IDM generation can be supported by GTPPM.

THE STRUCTRE OF AN IDM

An IDM defines *exchange requirements* and *functional parts*. An exchange requirement is a set of information "that must be passed from one business process to enable another business process to happen" (Wix 2005b). Functional parts are the actions that comprise an exchange requirement. A functional part is associated with a particular unit of information within an exchange requirement. A functional part is *reusable* (i.e., it can be associated with many exchange requirements and other functional parts) and can be broken down into other functional parts (Figure 1). Exchange requirements and functional parts are named as a composition of an action (usually a verb) and an object (usually a noun or a noun phrase): e.g. exchange_building_element in Figure 1.

An exchange requirement consists of four sections: *the header section, the overview section, the technical section,* and *the results section*. The header section describes the name, the author(s), the history of an exchange requirement, and relevant project stages. The overview section "states the aims and content of the exchange requirement in non technical terms (Wix 2005b)" and may include a process map (diagrammatical or textual), a summary, or a description. The technical section provides "the breakdown of information required by the exchange requirement (Wix 2005b)." It specifies whether the required information is mandatory or optional, actors that supply the information, and relevant functional parts. The result section defines "the form (e.g., a model, a document, or a drawing) in which information is most likely to be provided and what downstream processes most probably benefit (Wix 2005b)."

A functional part is composed of seven sections. In addition to the four sections of an exchange requirement, it has the list section, the schema section, and *the example section*. The contents of the four sections common to the functional part and the exchange requirement are different in detail, but since they are very similar, detailed descriptions of them are omitted. What is worth mentioning is that the technical section and the list section of functional parts specify the usage of individual IFC entities/attributes with examples and descriptions of the relationship between the IFC components and the functional parts. The IFC (sub-)schemas related to the specified functional parts are listed in the schema section. The schemas can be depicted in graphical form (e.g., an EXPRESS-G model) or in text form (in EXPRESS). The example section illustrates implementation. More details on the IDM are available on the IDM website (Wix 2005b).

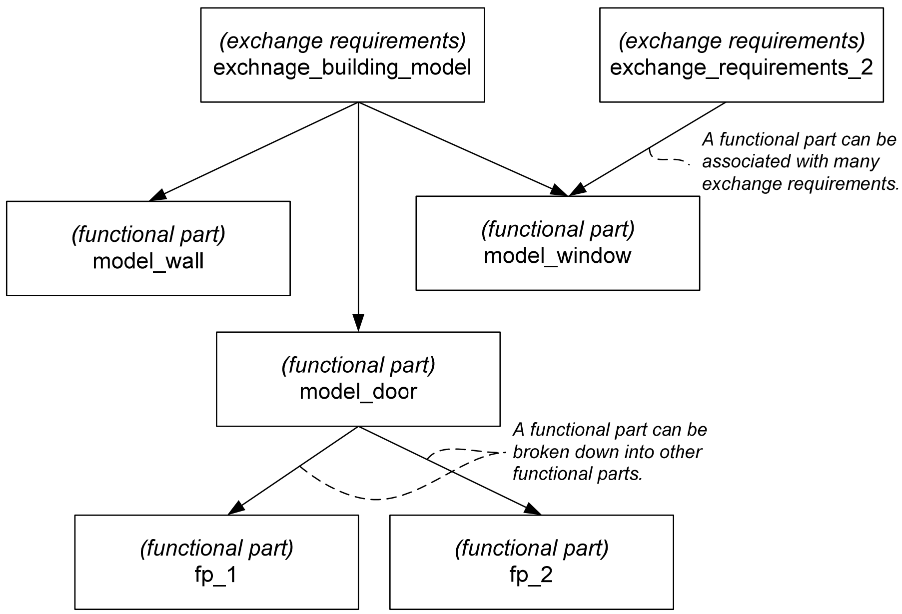


Figure 1: Exchange Requirements and Functional Parts (based on figures in (Wix 2005b))

In summary, the IDM provides a standardized format for predefining valid subsets of the IFC model required between processes or applications (i.e., exchange requirements), with examples and annotations.

THE STRUCTURE OF GTPPM

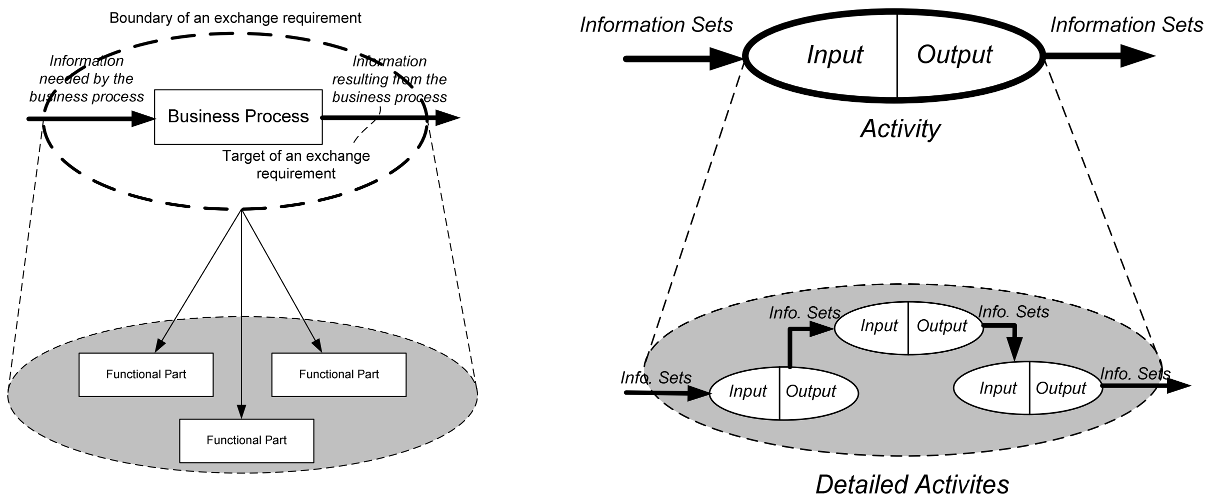
GTPPM is similar to the IDM in the way it categorizes information. It first categorizes information as input and output information. Input information is information needed by an activity or a series of activities (i.e., a process). Output information is information resulting from an activity or a process. The difference between the GTPPM information structure and that of the IDM is that GTPPM assumes that not all the input or output information is the information exchanged between two activities (or processes) or applications. GTPPM calls the information exchanged between two activities or applications *information sets*. Figure 2 illustrates the terms used in GTPPM and the IDM. In GTPPM, each information set represents a collection of information items that are commonly grouped together and passed between different activities or applications. Examples include a form, a tag, a drawing, or a CAD model. The relationships between information sets and input and output information are defined as follows in GTPPM:

1) Input information of an activity is a subset of information sets received from other activities. This implies that some information items in information sets available from other activities may not be needed or used by the target activity.

   Information Sets $\supseteq$ Input

2) Output information of an activity is a superset of information sets transferred to downstream activities. This implies that some information items in output information may not be needed or used by downstream activities.

   Information Sets $\subseteq$ Output



(a) A breakdown structure of an IDM exchange requirement (based on figures in (Wix 2005b))

(b) An information flow and activity breakdown structure of GTPPM

Figure 2: Information and Activity Breakdown Structures of the IDM and GTPPM

GTPPM subcategorizes the input and output information into five subtypes: used input information, unused (or remaining) input information, generated output information, modified output information, and unmodified (or passed-through) output information. GTPPM defines a set of logic rules between these information types called the dynamic information consistency checking rules. (The rules are defined in (Lee et al. 2002) in detail.) These rules do not prohibit product modelers from using GTPPM as an IFC view generation method. Rather, they can help product modelers to capture and specify high-quality information requirements by providing a mechanism to check the consistency of information flow.

GTPPM consists of two modeling phases. First, product modelers collect information requirements from a targeted domain to understand the terms and use cases and to define the scope of a model. This phase is generally called the *requirements collection and modeling phase*. Some common formal languages used in the requirements collection and modeling phase are Data Flow Diagrams (DFDs) (Osborne and Nakamura 2000), UML (Unified

Modeling Language) Uses Cases and Activity Diagrams (Booch et al. 1999; OMG 2003), and IDEF0 (NIST 1993a). Some of the major characteristics of GTPPM that distinguish it from other methods (such as UML, IDEF0) are:

1) It enables domain experts to specify a list of information items required for each activity within a project using their local terms.

2) It can automatically integrate and normalize information collected from multiple work processes into a single product model. Through this integration and normalization process, multiple lists of information are integrated into a structured data format and any conflicts or redundancies in the integrated model are resolved.

3) It maintains the relationship between the information list required for each process and the final product model. If a list of information required for processes is modified or updated, an updated product model can be derived automatically form the new list of information.

In the second phase, modelers interpret collected data requirements that are then mapped into a product data model. This phase is generally called *the logical (or conceptual) data modeling phase*. Examples of common graphical data modeling languages are EXPRES-G (ISO TC 184/SC 4 1994), ER Diagram (Ch. 3 of (Elmasri and Navathe 2004)), IDEF1x (NIST 1993b), NIAM (or ORM) (Halpin 2003; Nijssen and Halpin 1989), and UML Class Diagrams (Booch et al. 1999; OMG 2003). Some of standard text-based data modeling languages include SQL (ISO JTC 1/SC 32 2003), EXPRESS (ISO TC 184/SC 4 1994), and XSD (XML schema definition language) (OASIS 1999). Among these, GTPPM uses EXPRESS, the standard product modeling language, to represent a product model.

If GTPPM is to be deployed in a product model development effort, domain terms that are required to construct a product model must be captured and specified in the requirements collection and modeling phase. The collection of domain terms is called an *information menu*. There should not be any homonym or synonym in an information menu. Although in detail the information menu is different from the traditional *data dictionary*, the high-level concept of information menu is similar. If GTPPM is to be deployed in IFC view generation, an IFC model can be restructured and used as an information menu instead of creating a new one. Thus far, we have discussed the commonalities and differences between the GTPPM information structure and that of IDM. We have also shown an example of an IFC schema translated into an information menu of GTPPM. The next section describes a possible IFC view generation process using the GTPPM with a simple example. The example demonstrates the process of creating the IDM functional part "Model Door (Wix 2005a)" and compares the captured exchange requirements with those of "Model Door."

IFC VIEW GENERATION

As noted earlier, a view is a valid subset of a data model restructured for a specific use case. This means that entities in a view may or may not have the same structure as those of the original data model. For example, in a relational database, a view is typically created by joining several entities (or tables) to define a new one. Using GTPPM, different use cases of a certain set of information can be specified. Once this step is completed, GTPPM automatically collects and normalizes the information items required by the different use cases into a single view. The normalization rules are described in (Lee et al. 2006b) in detail. This process can be conducted in three steps. The first step is to define an information use cases as process models (Figure 3). A simple example process model is generated using a GTPPM tool built as a MS Visio add-on. In Figure 3, "Model Door" and "Install Door" are functional parts.

In the second step, an information set between two different activities is specified. In the example illustrated in Figure 3, "Door Exchange Requirement" is the information set that is passed from "Model Door" to "Install Door." To find and add IFC entities related to modeling doors to the information set "Door Exchange Requirement," the Search function in the GTPPM tool can be used. Figure 4 shows four possible IFC entities which are returned as search results for "door." The search function also searches through the Synonym field of the information menu shown in Figure 3 and returns IFC entities that have the word "door" in the Synonym field.

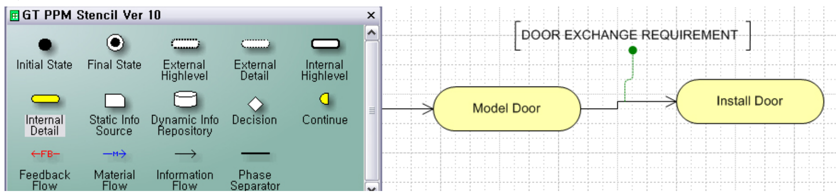


Figure 3: A GTPPM version of the IDM Model_Door Function Part

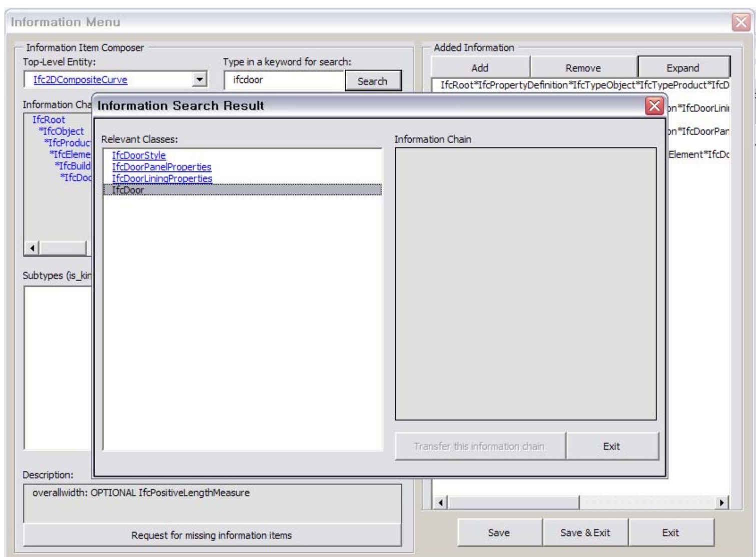


Figure 4: Information Search

If a specialization (supertype – subtype) tree structure of an entity is not fully captured in a subset model, some attributes may not be inherited properly and, as a result, the subset model will not be valid. The GTPPM tool supports a function to traverse back to the root of a specialization tree (Figure 5) so that the valid structure of a subset model can be captured.

The specified information can be saved as an information set using the Information Set Editor. Once an information set is defined, it can be used in defining other information flows or activities by selecting it from the Available Information Sets list on the right side of the Information Set Editor. By the same token, information required by each activity can be specified using the Activity Information Editor.

The third step is the logical modeling phase. A subset model is derived from the information specified from the previous two steps. If there is any conflict in the collected information, the conflict must be resolved. (Lee et al. 2006b) discusses twelve possible conflicts between specified information constructs and provides resolutions for each case implemented as design patterns. The GTPPM tool automatically collects the information specified in the processes and the design patterns integrate and normalize the collected data into a single view (or subset model) using the twelve conflict resolution rules. The EXPRESS-G model illustrated in Figure 6 below is an IFC view automatically extracted and derived from Figure 3 using the GTPPM method. The EXPRESS-G model was automatically generated from the EXPRESS model created from the GTPPM tool using EDMvisualExpress 4.01. The EXPRESS-G model in Figure 7 is an IFC view defined in the IDM Functional Part "Model Door." In the IFC view captured using the GTPPM method, the thirteen major IFC entities in the IDM IFC view are all captured. Also all the properties defined in the IDM IFC view are captured.

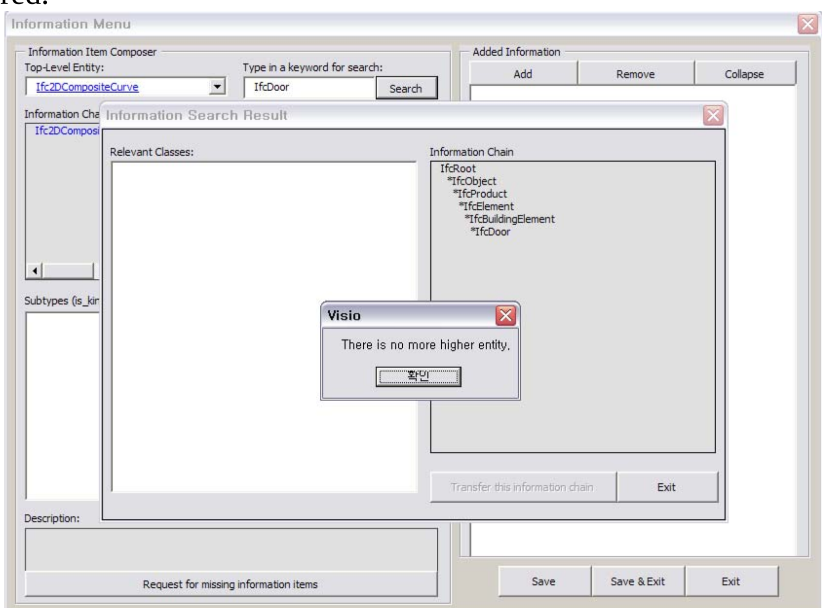


Figure 5: Information Tree

Note however that except for the four properties that are common to all the entities (namely, *OwnerHistory, Name, Description*, and *GlobalId*), the rest of the properties are captured as the properties of the entities directly relevant to "Model Door", whereas the IDM IFC view captures them as the properties of the supertypes as defined in the original IFC schema. For example, in the IFC schema, the 'representation' property is associated with ifcProduct and ifcDoor inherits the 'representation' property from ifcProduct. The IFC IDM view maintains this hierarchical structure and defines the 'representation' property as a property of ifcProduct. The IFC view generated using GTPPM, however, defines the 'representation' property as a property of ifcDoor. Since this is a view for the functional part "Model Door," the flattened structure generated by GTPPM seems closer to the original intention, as might typically be embodied in an Application Requirements Model (ARM) in the ISO-STEP process.

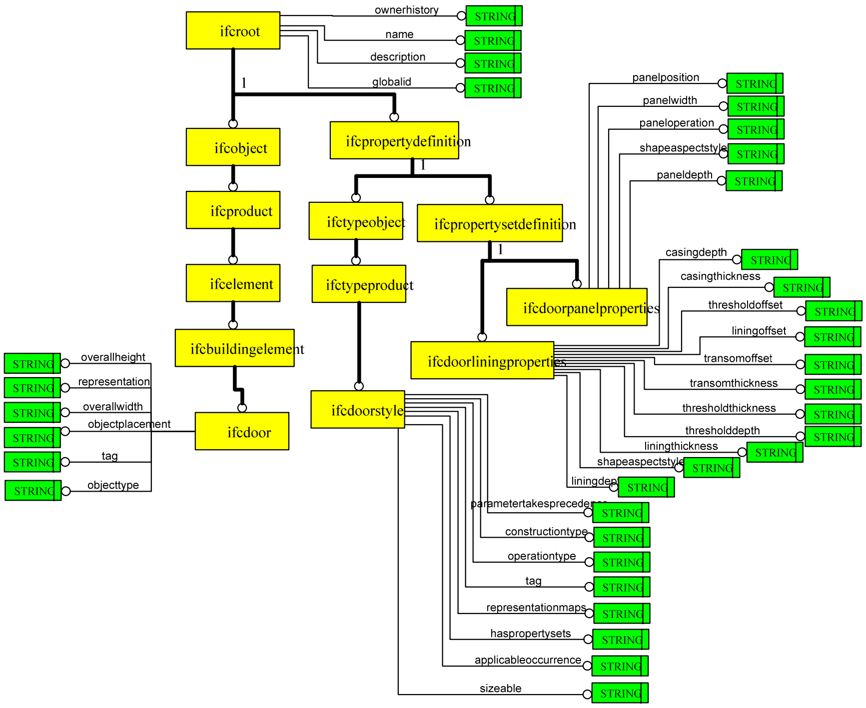


Figure 6: An IFC view generated using the GTPPM method for a functional model "Model_Door

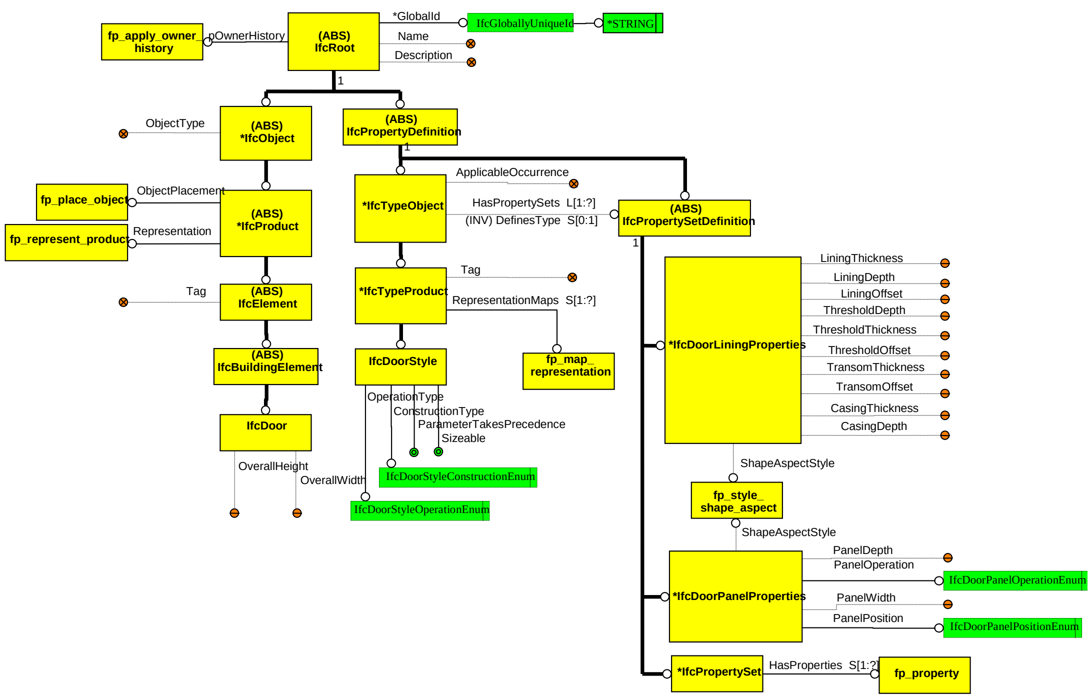


Figure 7: The IDM Functional Part "Model_Door" EXPRESS-G Model (Wix 2005a)

Due to the way the GTPPM tool is currently implemented, all the properties are defined as STRING type as illustrated in Figure 6. In order to obtain a more accurate model, this needs to be fixed in the near future. Overall, this example demonstrates a possibility of using GTPPM as a method to generate an IFC IDM view. Once a GTPPM model is generated, the GTPPM model can be easily updated or elaborated further as new requirements are recognized.

CONCLUSIONS

This paper presented and demonstrated the use of GTPPM as an IFC view or conformance class generation method. IDM defines a framework for manually defining an IFC View, based on a specified use case. GTPPM was originally developed to capture the work processes and information used within a UoD in order to generate a product model. GT PPM was developed to be used by domain experts without manual filtering and translation by modeling experts. Thus it has the benefit of moving toward a user-driven basis for defining use cases within IFC. However, since GTPPM does not support several IDM implementation details, it cannot automate the generation of an entire IDM. Benefits of using GTPPM as a method to create an IFC IDM view include its *traceability* and *reusability*. It leaves a clear record of how the view was generated and identifies the information in its source workflows. Also, if an IDM needs to be updated due to new demands, an IDM author can semi-automatically generate an updated IFC view by modifying previously defined GTPPM models. Through the simple test case, the needs for modifying the information menu structure and the GTPPM tool in order to use it as a more efficient and effective IFC view generation method have been identified.

REFERENCES

AISC. (2004). "CIS/2 Case Studies." http://www.aisc.org/Content/NavigationMenu/Steel_Solutions_Center/Time,_Money_and_Management/Electronic_Data_Interchange/CIS_2_Case_Studies/CIS_2_Case_Studies.htm (Last Accessed: 2005)

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*, Addison Wesley Longman, Inc., Reading, MA.

Crowley, A. (2001). "Questions on CIS/2 Conformance Classes ", http://www.cis2.org/faq/Implementation/ccs.htm (Last Accessed: 2006)

Eastman, C. M. (1999). "Ch 11. Modeling language Issues." Building Product Models: Computer Environments Supporting Design and Construction, CRC Press, Boca Raton, FL.

Elmasri, R., and Navathe, S. (2004). *Fundamentals of Database Systems*, Addison Wesley Longman, Inc., Reading, MA.

Halpin, T. (2003). "Object Role Modeling: the official Site for Conceptual Data Modeling." http://www.orm.net/ (Last Accessed: 2003)

IAI. (1994). "International Alliance for Interoperability." http://www.iai-international.org/iai_international/ (Last Accessed: 2005)

IAI. (2004). "Industry Foundation Classes IFC2x2 Addendum1." http://www.iai-international.org/Model/R2x2_add1/index.html (Last Accessed:

ISO JTC 1/SC 32. (2003). "ISO/IEC 9075-1:2003 Information technology - Database languages - SQL - Part 1: Framework (SQL/Framework)." *ISO/IEC 9075-1:2003*, ISO.

ISO TC 184/SC 4. (1994). "ISO 10303-11:1994 Industrial automation systems and integration - Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual." International Organization for Standardization.

ISO TC 184/SC 4. (2004). "ISO 10303 TC 184/SC 4." http://www.tc184-sc4.org/ (Last Accessed: 2005)

Lee, G. (2002). "GTPPM Official Website." http://dcom.arch.gatech.edu/glee/gtppm (Last Accessed: 2005)

Lee, G. (2004). "A New and Formal Process to Product Modeling (PPM) Method and its Application to the Precast Concrete Industry," Ph.D. dissertation, Georgia Institute of Technology, Atlanta, GA.

Lee, G., Eastman, C. M., and Sacks, R. (2006a). "Grammatical rules for specifying product information to support automated product data modeling (in press)." *Advanced Engineering Informatics*.

Lee, G., Eastman, C. M., and Sacks, R. (2006b). "Twelve Design Patterns for Integrating and Normalizing Product Model Schemas (in press)." *Computer-Aided Civil and Infrastructure Engineering (CACAIE)*.

Lee, G., Sacks, R., and Eastman, C. M. "Dynamic information consistency checking in the requirements analysis phase of data modeling (Keynote)." *eWork and eBusiness in Architecture, Engineering and Construction - European Conference for Process and Product Modeling (ECPPM)*, Slovenia, 285-291.

Nijssen, G. M., and Halpin, T. A. (1989). *Conceptual Schema and Relational Database Design: A Fact Oriented Approach*, New York: Prentice Hall.

NIST. (1993a). "FIPS Publication 183: Integration Definition of Function Modeling (IDEF0)." National Institute of Standards and Technology.

NIST. (1993b). "FIPS Publication 184: Integration Definition of Information Modeling (IDEF1X)." National Institute of Standards and Technology.

OASIS. (1999). "XML.org." http://www.xml.org (Last Accessed: 2005)

OMG. (2003). "Unified Modeling Language (UML) Specification V.1.5." Object Management Group.

Osborne, L. N., and Nakamura, M. (2000). *Systems Analysis for Librarians and Information Professionals*, Libraries Unlimited, Inc., Englewood, Colorado.

Wix, J. (2005a). "Functional Part: Model_Door." ftp://ftp.iai.no/pub/idm/ (Last Accessed:

Wix, J. (2005b). "Information Delivery Manual (IDM) Using IFC to Build SMART - The IDM Project Official Website ", http://www.iai.no/idm/learningpackage/idm_index.htm (Last Accessed: 2005)