FRAMEWORK OF THE EXTENDED PROCESS TO PRODUCT MODELING (XPPM) FOR EFFICIENT IDM DEVELOPMENT

Ghang Lee, Ph.D. Associate Professor, glee@yonsei.ac.kr, Corresponding Author
Sungil Ham, Ph.D. / Postdoctoral Researcher, archispace@yonsei.ac.kr
Young Hyun Park / Ph.D. Student, nuyhyun@yonsei.ac.kr
Building Informatics Group, Department of Architectural Engineering, Yonsei University, South Korea

ABSTRACT

This paper introduces a new "extended Process to Product Modeling (xPPM)" for efficient Information Delivery Manual (IDM) development. The current IDM development typically uses Business Process Modeling Notation (BPMN) to represent a Process Map (PM). However, the resultant Process Map is isolated from the development of Exchange Requirements (ERs) and Functional Parts (FPs). ERs and FPs specify the information required when information is exchanged between different activities. The extended Process to Product Modeling (xPPM) method is proposed to provide a tight connection between PMs, ERs, and FPs. The theoretical framework is based on the Georgia Tech Process to Product Modeling. An xPPM tool is being developed in Java to support several IDM development efforts in South Korea.

Keywords: xPPM, IDM, product modeling, process modeling

1. INTRODUCTION

The Industry Foundation Classes (IFC) is an international standard data model for exchanging facility information (ISO/TC184/SC4 2005). Because the IFC includes the information created and used throughout the lifecycle of various types of facilities such as buildings, civil infrastructures, and potentially plants in the near future, judging which information should be included and excluded in the exchanged set of information becomes difficult. This problem initiated the Information Delivery Manual (IDM) efforts (ISO/TC59/SC13 2010).

The IDM follows the typical data-model specification process. The first step is to define the target process in which a data model is used. The IDM standard calls this process model a "process map (PM)." The second step is to define the data exchanged between activities in the PM. These data are called "exchange requirements (ERs)" in the IDM. A unit set of specific information items, which forms an ER, is called a "functional part (FP)." Later, FPs can be mapped to a view (subset) of the IFC. The view is called a "model view definition (MVD)."

However, there is no direct connection between a process map (the first step) and the exchange requirements (the second step). This disconnection was pointed out as a problem in judging and validating which information should be used to satisfy the specified process map (Eastman et al. 2002, Lee et al. 2006, Eastman et al. 2010). To overcome this problem, Lee et al. (2006) proposed adopting Georgia Tech Process to Product Modeling (GTPPM), which provides a direct connection between a process model and a product model, in developing an IDM but identified several limitations:

- Business Process Modeling Notation (BPMN) is generally used as the notation for specifying an IDM process map (ISO/TC59/SC13 2010). However, GTPPM uses its own notation and does not support BPMN.

- The eventual goal of the IDM is to define a model view definition (MVD) of IFC, but GTPPM's data structure requires modification to fully support an iterative IFC view development process.

The authors extended GTPPM to support the IDM development process. The extended GTPPM is called extended Process to Product Modeling (xPPM). The extended Process to Product Modeling (xPPM) method is proposed to provide a close connection between PMs, ERs, and FPs. This paper introduces the xPPM framework and describes how xPPM can support the IDM development process.

2. IDM, GTPPM, AND XPPM

The IDM is the international standard for specifying "information to be exchanged about a particular topic or business requirement in the construction process" (ISO/TC 59/SC 13 2010). An IDM is composed of a process map, exchange requirements, and functional parts. The IDM standard defines them as follows:

- Process map (PM): a representation of the relevant characteristics of a process for a defined purpose
- Exchange requirement (ER): the set of information that needs to be exchanged to support a particular process stage or stages
- Functional part (FP): a unit of information within an exchange requirement that may be fully specified in its own right.

GTPPM is a product modeling method developed at Georgia Tech (Eastman et al. 2002, Lee 2004, Lee et al. 2007) as a method to tightly link a process model and a product model. The equivalent concepts in GTPPM to the PM, ER, and FP in the IDM are the process model, information set, and information item. Similarly, Eastman et al. (2010) called ER, an exchange model (EM), and FP, an exchange object (EO). xPPM uses the same terms as GTPPM because the proposed method is an extension of GTPPM. Table 1 compares the terms used in the IDM, GTPPM, xPPM, and the exchange model approach.

Table 1: Comparison of Terms in the IDM, GTPPM, xPPM, and Exchange Model Approach

IDM	GTPPM and XPPM	Exchange Model Approach
Process Map	Process Model	Process Map
Exchange Requirement	Information Set	Exchange Model
Functional Part	Information Item	Exchange Object

The major difference between the IDM and xPPM is that the IDM aims to provide ERs and FPs as neutral specifications independent of a specific data model such as IFC. The IDM, thus, requires additional steps to map ERs and FPs to a specific data model to define an MVD of a certain data model required to support the target PM. In contrast, xPPM allows modelers to directly use a specific data model such as IFC to define the information sets and information items so that additional steps for mapping ERs and FPs to a specific data model can be eliminated in defining an MVD. Another difference is that xPPM provides a close connection between a process model and sets of information used by the process by defining the information sets required by each activity.

The major difference between GTPPM and xPPM is that GTPPM cannot specify the "aggregation" relation such as many-to-one or many-to-many relations whereas xPPM can specify and maintain such relations. Another difference is that xPPM uses BPMN to define a process model whereas GTPPM uses its own process modeling notation. The next section describes the xPPM framework in more detail.

3. GTPPM AND BPMN PROCESS MODEL CONCEPTS IN XPPM

xPPM begins with process modeling. xPPM uses BPMN for process modeling (OMG 2011). Figure 1 shows a comparison of the GTPPM elements and the matching BPMN elements used in xPPM.

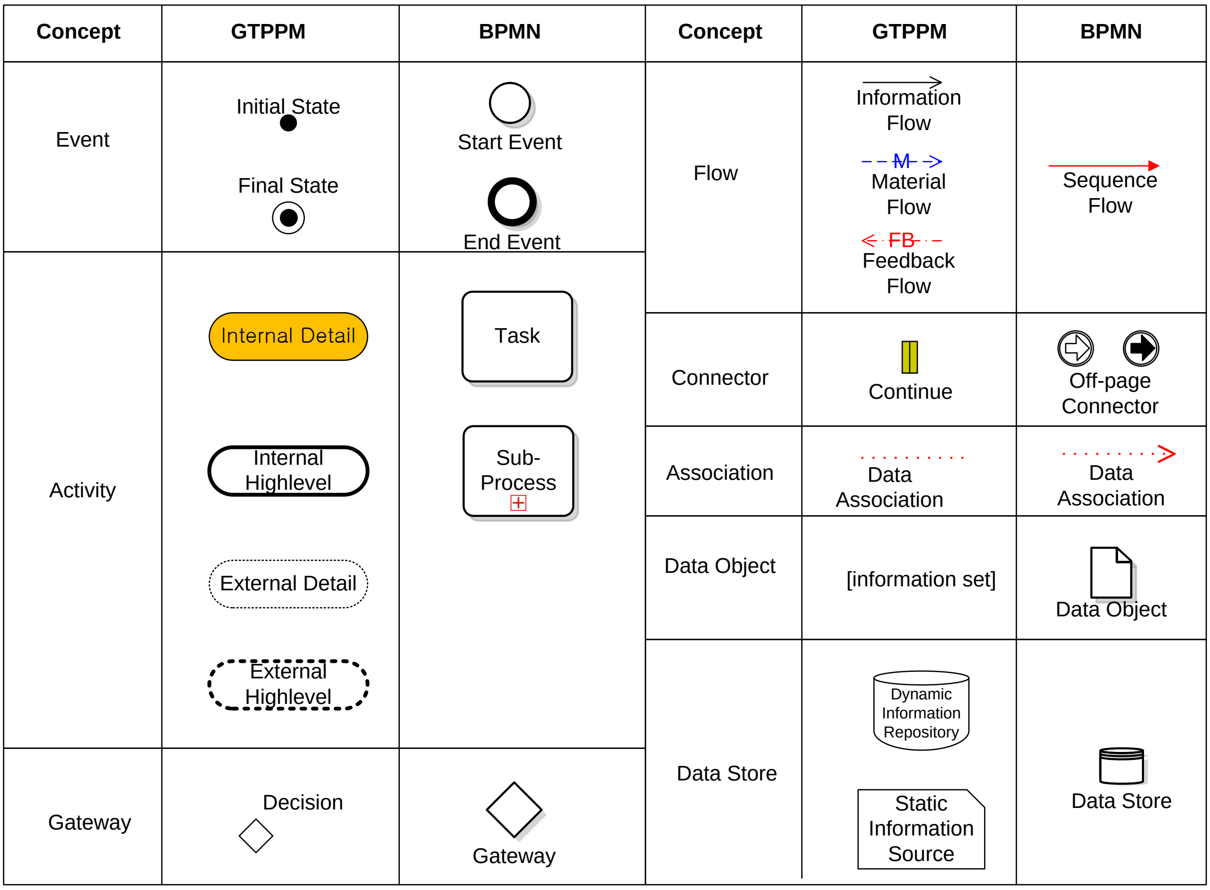

Figure 1: Comparison of GTPPM and BPMN elements

GTPPM has two types of events, the initial state and the final state. They are equivalent to the start event and the end event in BPMN. GTPPM has four types of activities, the internal detail activity, the internal high-level activity, the external detail activity, and the external high-level activity. The distinction between the internal and external activities was made to allow data exchange between activities that are included in the scope of product modeling (thus called internal activities) and activities that are outside the scope of product modeling (thus called external activities). The distinction is needed because GTPPM includes logic to check the consistency of information flow (Lee et al. 2002) and ignores the external activities in the consistency check. The high-level activity indicates that a sub-process is defined in the process model. The high-level activity is equivalent to the collapsed sub-process concept in BPMN.

There are four types of flow in GTPPM. Information flow is the normal sequence flow in BPMN. Material flow is the flow of physical objects such as window frames, precast concrete pieces, pipes, ducts, and elevators. Physical objects carry information. First, the physical objects themselves are information. That is, their physical characteristics such as color, weight, and shapes are information. Second, GTPPM carries additional information such as producer and invoice information. The distinction between information flow and material flow is made because it is possible to continue building a building without a certain set of information, but it is not possible to carry on construction activities without materials delivered to a construction site. Feedback flow distinguishes the flow going forward from the flow returned in a cyclic process. This distinction is important in checking the consistency of information flow because the input information is separate from the returned information of an activity. Input information coming through the feedback flow is ignored in the consistency check.

The Continue shape pauses and resumes flow. This is similar to the off-page connectors in BPMN. The only difference is that the Continue shape can be used on the same page to avoid complex overlaps between flow shapes.

GTPPM distinguishes the data stored and updated within the specified process from the data stored and updated outside the specified process. The former is called the dynamic information repository, and the latter is called the static information source. The data store concept in BPMN does not distinguish between these two types of information.

Other concepts in GTPPM such as decision, information set, and data association are equivalent to gateway, data object, and data association in BPMN. Currently, xPPM allows modelers to use all BPMN concepts in addition to the ones that are equivalent to the GTPPM concepts introduced in this section. However, because BPMN was developed to cover various and detailed business cases and rules; it includes too many concepts (e.g., 48 type dimensions in BPMN 2.0) for IDM development and more generally for information flow modeling. For efficient use of BPMN in developing an IDM, the xPPM development team is in the process of developing a recommended BPMN subset and discussing whether to expand the BPMN to support some of the concepts in GTPPM, which do not exist in BPMN.

4. BASIC INFORMATION REPRESENTATION IN XPPM

Both GTPPM and xPPM are based on the structure and concepts of EXPRESS, the standard data modeling language for defining product models (ISO/TC 184/SC4 1994). To quickly and simply represent information in GTPPM, information is represented based on fundamental conceptualization mechanisms: generalization, specialization, assembly, decomposition, and association. The generalization/specialization relations are represented using the asterisk (*). The assembly/decomposition and association relations are represented using the plus (+) sign. In addition, attributes are represented using brackets. Figure 2 shows an example. It reads "The entity Project is a subtype of the entity Object. The entity Project has the entity Site. The attributes of the Site are the Name and Address."

Object*project+site{name;address}

*: inheritance, generalization/specialization
+: assembly/decomposition, association
{ }: attributes

Figure 2: Information representation in GTPPM

This structure is efficient in quickly defining information required by a certain work process. However, GTPPM does not provide a mechanism for defining the cardinality and other constraints in EXPRESS such as SET, BAG, LIST, ARRAY, OPTIONAL, and UNIQUE between two entities during the data collection step, because GTPPM assumes that the constraints between entities will be elaborated during the refinement step after collecting all the information required for the target process in order to minimize conflicts in the definitions of the cardinality between the same two entities.

In contrast, xPPM allows modelers to specify the constraints in EXPRESS during the data collection process so that it can immediately produce an IFC view as soon as xPPM has been completed. xPPM assumes that IFC will be used as a predefined data dictionary when xPPM is used for developing an IDM or IFC view. Figure 3 is an example of specifying the above site information example in Figure 2 in terms of the xPPM data format.

IfcRoot*IfcObjectDefinition*IfcObject*IfcProduct*IfcSpatialStructureEl
ement*IfcSite{Name: OPTIONAL IfcLabel;
SiteAddress: OPTIONAL IfcPostalAddress;}

Figure 3: Information representation in xPPM

5. XPPM MODELING PROCESS

The xPPM modeling process consists of four steps:

1) Define a process map using BPMN (process modeling).
2) Specify the input and output information required by each activity.
3) Check the consistency of the information flow using the dynamic information consistency checking logic specified in Lee et al. (2002).
4) Collect all the information specified in the xPPM model and generate an IFC view (MVD) based on the integration and normalization rules specified in Lee et al. (2007).

6. SYSTEM DEVELOPMENT

The xPPM tool is under development using Java based on the above framework. The xPPM tool is mainly composed of three interfaces: the Process Modeling interface (Figure 4), the Activity Information interface (Figure 5), and the IFC Menu interface (Figure 6).

Modelers define a process map by dragging and dropping BPMN elements from the BPMN list in the right pane of the xPPM Process Modeling interface.

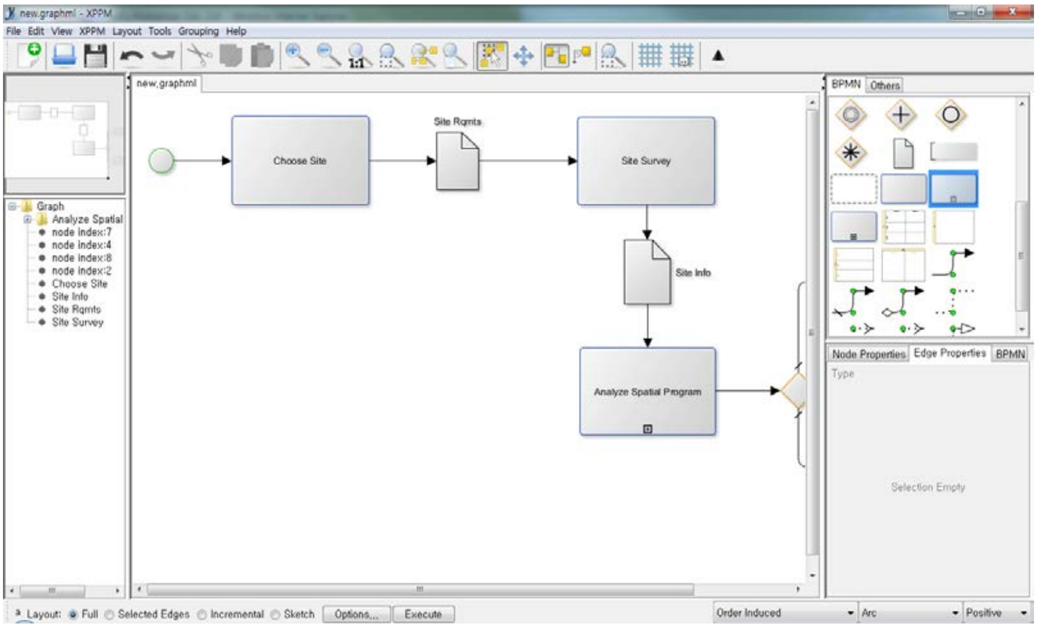

Figure 4: The xPPM Process Modeling interface based on BPMN

After specifying a process map, modelers can specify the input and output information required by each activity and sets of information (i.e., data objects) transferred from one activity to another using the Activity Information interface shown in Figure 5. When an activity or a data object is double-clicked, the Activity Information interface pops up.

Modelers can specify information by choosing IFC entities and their attributes from the IFC Menu interface in Figure 6 or passing input information to the output information list. The IFC menu opens up when the Add New button on the Activity Information interface is clicked.

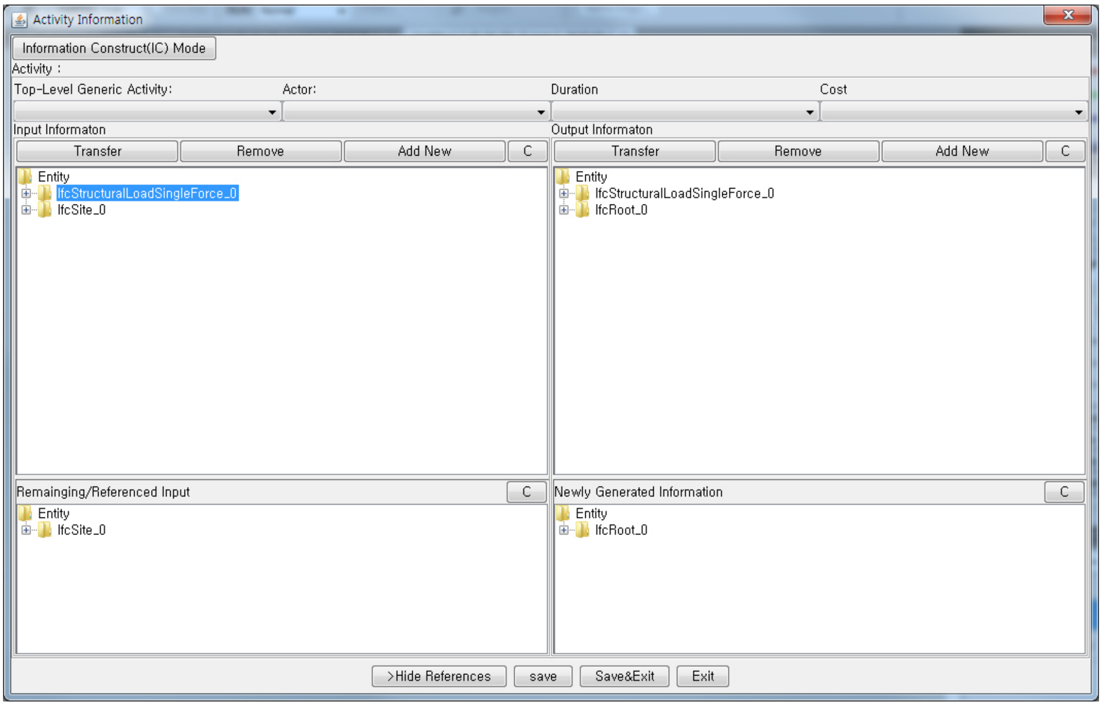

Figure 5: The Activity Information Interface for specifying the input and output information for each activity

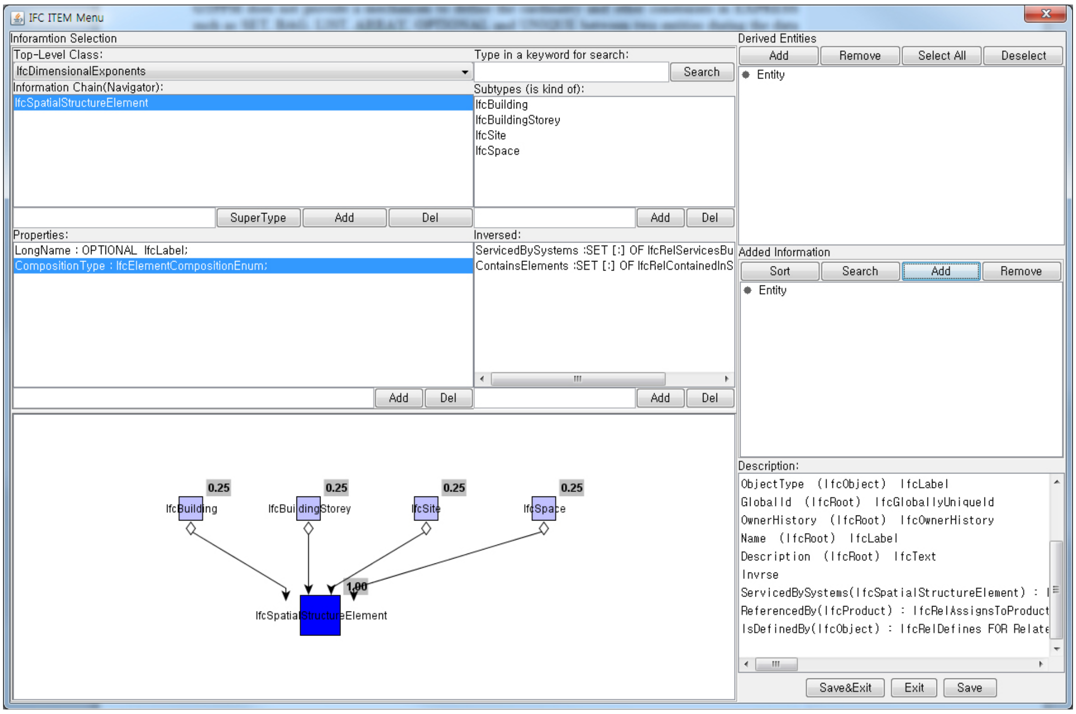

Figure 6: IFC Menu Interface. This menu pops up when the “Add new” button is clicked from the Activity Information interface.

7. SUMMARY AND FUTURE WORK

This paper introduces the extended process to product modeling (xPPM) method. The theoretical xPPM framework is based on GTPPM. GTPPM was a product modeling method developed to closely bind the process modeling (or use case specification) phase and the product modeling phase so that the target processes of product models could be easily traceable. Although the possibility of adopting GTPPM in developing an IDM has been demonstrated (Lee, G., et al. 2006), GTPPM had several limitations in fully supporting the IDM development process because this method was developed before the IDM. xPPM was developed with two goals:

1) xPPM should be able to fully support the IDM development process.
2) It should be possible to produce an IFC view (MVD) when IDM development based on the xPPM method is completed.

To fulfill these goals, xPPM adopted BPMN as the main process modeling notation following the general IDM development practice today. In addition, xPPM incorporated more EXPRESS semantics in the information constructs to closely link the IDM and MVD development processes. An xPPM tool is being developed in Java. The tool is being tested and modified with several IDM development efforts in South Korea. We expect that the xPPM can make the IDM and MVD development process more efficient by providing the modelers with the traceability and reusability of MVDs and the close link between the MVD and IDM development when its development is completed.

ACKNOWLEDGMENTS

This research was supported by grant "06-Unified and Advanced Construction Technology Program-E01" from the Korean Institute of Construction & Transportation Technology Evaluation and Planning (KICTEP) and by the MKE(The Ministry of Knowledge Economy), Korea, under the Convergence-ITRC(Convergence Information Technology Research Center) support program (NIPA-2011-C6150-1101-0003) supervised by the NIPA(National IT Industry Promotion Agency)".

REFERENCES

Eastman, C. M., et al. (2002) "A new formal and analytical approach to modeling engineering project information processes." *CIB W78*, Aarhus, Denmark, 125-132.

Eastman, C. M., et al. (2010). "Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards." *Journal of Computing in Civil Engineering*, 24(1), 25-34.

Elmasri, R., et al. (2004). “Ch.2 Database system concepts and architecture.” FundamentalsofDatabaseSystems.Reading,MA,AddisonWesleyLongman,Inc.

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." ISO, Geneva, Switzerland.

ISO/TC 59/SC 13 (2010). ISO 29481-1:2010 Building information models — Information delivery manual — Part 1: Methodology and format. Geneva, Switzerland, ISO**:** 34.

ISO/TC184/SC4 (2005). ISO/PAS (Publicly Available Specification) 16739, IAI Industry Foundation Classes 2x Platform Specification, ISO.

Lee, G., et al. (2002) "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.

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., et al. (2006) "Generating IFC views and Conformance Classes using GTPPM." *Joint International Conference on Computing and Decision Making in Civil and Building Engineering (ICCCBE)*, Montreal, Canada, 1715-1724.

Lee, G., et al. (2007). "Eliciting Information for Product Modeling using Process Modeling." *Data & Knowledge Engineering*, 62(2), 292-307.

Lee, G., et al. (2007). "Twelve Design Patterns for Integrating and Normalizing Product Model Schemas." *Computer-AidedCivilandInfrastructureEngineering(CACAIE)* **22**(3): 163-181.

OMG. (2011). "Business Process Model and Notation (BPMN) Version 2.0." OMG, Needham, MA.