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

Ghang Lee, Ph.D. Associate Professor, glee@yonsei.ac.kr
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) as a means 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 in exchange of information 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 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 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, it becomes difficult to judge which information should and should not be included in the exchanged set of information. 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 it a “process map (PM)”. The second step is to define the data exchanged between activities in the PM. These are called “exchange requirements (ERs)” in 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 IFC. The view is called a “model view definition (MVD).”

However, there is not a direct connection between a process map (the first step) and 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 to adopt the Georgia Tech Process to Product Modeling (GTPPM), which provided the direct connection between a process model and a product model, in developing an IDM, but several limitations were identified:

- The Business Process Modeling Notation (BPMN) is generally used as a notation for specifying a process map of an IDM (ISO/TC59/SC13 2010). But GTPPM uses its own notation and does not support BPMN.
- The eventual goal of IDM is to define a model view definition (MVD) of IFC, but GTTPPM’s data structure requires modification to fully support an iterative IFC view development process.

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

2. IDM, GTTPPM, AND XPPM

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).” 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

GTTPPM is a product modeling method developed by 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 GTTPPM to the PM, ER, and FP in IDM are process model, information set, and information item. Similarly, Eastman et al. (2010) called ER, exchange model (EM), and FP, exchange object (EO). xPPM uses the same terms as GTTPPM because it extends from GTTPPM. Table 1 compares the terms used in IDM, GTTPPM, xPPM, and the exchange model approach.

<table>
<thead>
<tr>
<th>IDM</th>
<th>GTTPPM and XPPM</th>
<th>Exchange Model Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Map</td>
<td>Process Model</td>
<td>Process Map</td>
</tr>
<tr>
<td>Exchange Requirement</td>
<td>Information Set</td>
<td>Exchange Model</td>
</tr>
<tr>
<td>Functional Part</td>
<td>Information Item</td>
<td>Exchange Object</td>
</tr>
</tbody>
</table>

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

The major difference between GTTPPM and xPPM is that GTTPPM 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 GTTPPM uses its own process modeling notation. The next section describes the framework of xPPM in more detail.

3. PROCESS MODEL CONCEPTS OF GTTPPM AND BPMN IN XPPM

xPPM begins with process modeling. For process modeling, xPPM uses BPMN (OMG, 2011). 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.

<table>
<thead>
<tr>
<th>Concept</th>
<th>GTPPM</th>
<th>BPMN</th>
<th>Concept</th>
<th>GTPPM</th>
<th>BPMN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>Initial State</td>
<td>Start Event</td>
<td>Flow</td>
<td>Information Flow</td>
<td>Sequence Flow</td>
</tr>
<tr>
<td></td>
<td>Final State</td>
<td>End Event</td>
<td></td>
<td>Material Flow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Feedback Flow</td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>Internal Detail</td>
<td>Task</td>
<td>Connector</td>
<td>Continue</td>
<td>Off-page Connector</td>
</tr>
<tr>
<td></td>
<td>Internal Highlevel</td>
<td>Sub-Process</td>
<td>Association</td>
<td>Data Association</td>
<td>Data Association</td>
</tr>
<tr>
<td></td>
<td>External Detail</td>
<td></td>
<td>Data Object</td>
<td>[information set]</td>
<td>Data Object</td>
</tr>
<tr>
<td></td>
<td>External Highlevel</td>
<td></td>
<td>Data Store</td>
<td></td>
<td>Data Store</td>
</tr>
<tr>
<td>Gateway</td>
<td>Decision</td>
<td>Gateway</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Comparison of GTPPM and BPMN elements

Figure 1 compares GTPPM elements and matching BPMN elements used in xPPM. 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 were 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 object such as window frames, precast concrete pieces, pipes, ducts, elevators. Physical objects carry information. First, physical objects themselves are information. That is, their physical characteristics such as color, weight, shapes are information. Second, it 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 it distinguishes the input information from the returned information of an activity. Input information coming through the feedback flow is ignored in the consistency check.
Continue shape pauses and resumes flow. This is similar to the off-page connectors in BMPN. The only difference is that the continue shape is allowed to be used also 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 BMPN does not distinguish these two. Other concepts in GTPPM such as decision, information set, and data association are equivalent to gateway, data object, and data association in BMPN. 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, for efficient use of BMPN in IDM development, the xPPM development team is in a process of developing a recommended subset of BPMN 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). In order 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 “Entity project is a subtype of Entity object. Entity project has Entity site. The attributes of site are 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 to define the cardinality and other constraints in EXPRESS such as SET, BAG, LIST, ARRAY, OPTIONAL and UNIQUE between two entities during the data collection step. It is 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.

On the other hand, xPPM allows modelers to specify the constraints in EXPRESS during the data collection so that it can immediately produce an IFC view as soon as xPPM modeling is completed. xPPM assumes that IFC will be used as a predefined data dictionary when xPPM is used for IDM or IFC view development. 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) The first step is process modeling. A process map is defined in this step using BPMN.
2) Specify input and output information required by each activity. It is easier to
3) Check the consistency of 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 BMPN elements from the BPMN list in the right pane of the xPPM Process Modeling interface.

![Figure 4: the Process Modeling interface of xPPM based on BMPN](image)

After specifying a process map, modelers can specify 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 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 clicking the Add New button on the Activity Information interface.
Figure 5: Activity Information Interface for specifying input and output information of 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 framework of xPPM is based on GTPPM. GTPPM was a product modeling method developed to tightly 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 GTTPPM in IDM development was demonstrated (Lee, G., et al. 2006), GTTPPM had several limitations in fully supporting the IDM development process because GTTPPM was developed before IDM. xPPM was developed aiming at 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 the BPMN as the main process modeling notation following the general IDM development practice today. Also it incorporated more EXPRESS semantics in the information constructs to tightly link that the IDM and MVD development processes. An xPPM tool is being developed in Java. It is being tested and modified by several IDM development efforts in South Korea.

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).

REFERENCES


