

The Analysis on the Possibility of Business Process Modeling using UML Activity Diagram

Hyun-Min Ko, Myung-Geun Son, Youn-Ju Oh, Doo-Hwan Bae
Div. of CS, Dept. of EECS, Korea Advanced Institute of Science and Technology
{hmko, mgson, yjoh, bae}@salmosa.kaist.ac.kr

UML Activity Diagram 을 통한 비즈니스 프로세스 모델링 가능성 분석

고현민, 손명근, 오윤주, 배두환
한국과학기술원 전자전산학과 전산학전공

Abstract

In order to survive in today's competitive world, understanding the business process and how it works can be a key to success. Visualization improves understandability of business process for developers, and accompanies modeling. There are two modeling methods that are often used in the industry - standard BPMN (Business Process Modeling Notation) and UML activity diagram. This paper shows whether the activity graph of UML 2.0 draft can support business process modeling or not through comparing it with the BPMN features.

1. Introduction

In order to survive in today's competitive world, understanding the business process and how it works can be a key to success. Moreover, since business process is becoming more and more automated, the dependency of business process on business software application is increasing.

Inversely, to develop a successful business software application, understanding business process is needed. Visualization improves understandability of business process for developers.

Visualization accompanies modeling. There are two modeling methods that are often used in the industry. The first is the standard BPMN (Business Process Modeling Notation). It has several unique features, for example, end-to-end process modeling. The second is the UML activity diagram. UML activity diagram can provide process flow the best, compared with the other UML diagrams, which have more several limitations. [3]

This paper will analyze activity graph of UML 2.0 draft by comparing it with BPMN and will discuss the possibility for business process modeling. If the activity graph does not fully cover the features of BPMN, this paper will also suggest improvements of the UML 2.0 draft.

The remainder of this paper is organized as follows. In the next section, BPMN is described. Subsequently, in section 3, activity graphs of UML 2.0 draft is discussed. Section 4 describes

analysis of the activity graph of UML 2.0 draft for business process modeling. The paper is concluded in section 5.

2. BPMN (Business Process Modeling Notation)

The BPMN is a graphical notation for expressing business processes that was developed by the Business Process Management Initiative (BPMI). It will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. It also provides a formal mapping to execution languages of BPM systems. [2]

And unique features of BPMN are as follow

- End-to-End Process Modeling
- Control-Flow/Data-Flow Separation
- Produce/Consume Messaging
- Dynamic Control-Flow
- Transparent Persistence
- Embedded Business Rules
- Nested Processes
- Distributed Transactions
- Process-Oriented Exception Handling
- Underlying Mathematical Model

Each feature has the following meaning. First, End-to-End Process Modeling means BPMN can show the end-to-end process from supplier to customer. Control-Flow/Data-Flow Separation

means BPMN clearly separates Control-Flow and Data-Flow. Dynamic Control-Flow means BPMN supports dynamic branching based on message types or participants. Embedded Business Rules supports embedded business rules that drive the execution of dynamic business processes. Finally, Nested Processes means BPMN supports not only sub-processes but also nested processes that share the data of its parent process.

We consider the above features to be very useful concepts for business process modeling. Because the standard business modeling language support these features, they could help business analyst showing business processes visually.

The subset of the core element set is shown in Figure 1.








| Element | Notation |
|------------------------|--|
| Event (three types) | Start  Intermediate  End  |
| Task (atomic) |  |
| Sub-Process (Compound) |  |
| Decision |  |
| Sequence Flow |  |

Figure 1 The Subset of Core element set

3. Activity Graphs of UML 2.0 Draft

This paper is based on 'Unified Modeling Language: Superstructure version 2 beta R1 (draft)' submitted by one of most influencing groups, U2Partner, on 9 September 2002. [1]

3.1 Motivation

There are some difficulties in using UML 1.x Activity Diagrams. One of the fundamental sources of trouble is the interpretation of flows as state transitions, and activities as states. Similarly, the swimlanes used in activity diagrams are not well motivated. Moreover, the activities in an activity diagram are not reusable, since they are exclusively owned. There are also many limits in structuring the events and taking conditional paths in an activity diagram. Finally, activities are either performed by a single object, or are not associated with any participants.

3.2 Activity Graph Notation

The activity graph notation is based on 1.x Activity Diagrams. Activities are special kinds of Actions. The ends of an activity represent the pre/post states of that action. Various kinds of connections between these activities, including forks and joins, are relations between states. The Figure 2 shows a simple Activity Graph.

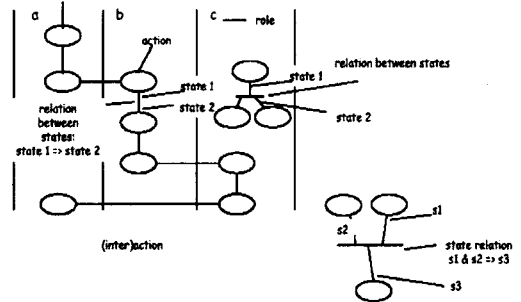


Figure 2 Activity Graph

UML 2.0 draft introduces new notations.

- Sub-Process : A compound activity that is included within a Process. It is compound in that it is broken down into a finer level of detail through a set of sub-activities
- Pool : A "swimlane" and a graphical container for partitioning a set of activities from other Pools

The Figure 3 displays the two notations.

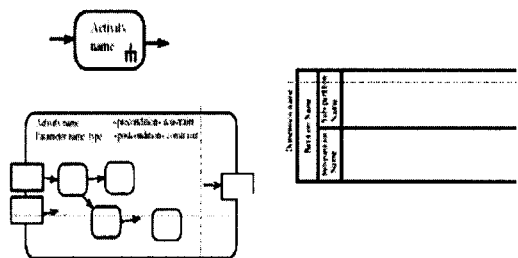


Figure 3 The Sub-Process and Pool notation

3.3 Possibility of Business Process Modeling

First, activity graphs describe behavioral aspects, in terms of business process modeling, and process flow. Moreover, new notations suggested in UML 2.0 draft also support better ways to show business processes than those of UML 1.x. Because of these reasons, UML 2.0 draft might make it possible for better business process modeling. In the following section,

analysis will be done by comparing UML 2.0 draft with the BPMN features about possibilities whether UML 2.0 draft fully cover business process modeling or not.

4. Analysis

4.1 Overview

In this section, it will be shown whether activity graph of UML 2.0 draft can support business process modeling or not through comparing with the BPMN features. The reason why BPMN is a basis for comparison is that BPMN includes all essential concepts for business process modeling.

4.2 Result

Table 1 shows the result of comparing UML 2.0 with the features of BPMN. As we can see, it is discovered that UML 2.0 draft cannot support the BPMN features such as Control-Flow/Data-Flow Separation, Produce/Consume Messaging, and Transparent Persistence.

Table 1 Comparison UML 2.0 with BPMN

| BPMN | UML 2.0 |
|-------------------------------------|---------|
| End-to-End process Modeling | O |
| Control-Flow/Data-Flow Separation | X |
| Produce/Consume Messaging | X |
| Dynamic Control-Flow | O |
| Transparent Persistence | X |
| Embedded Business Rules | O |
| Nested Processes | O |
| Distributed Transactions | O |
| Process-Oriented Exception Handling | O |

For Control-Flow/Data-Flow Separation, the activity graph of UML 2.0 draft only supports control flow without the separation of the two. Because of this, if control flow is different from data flow, data flow can be not expressed, and this leads to poor process maintainability and poor reuse.

For Produce/Consume Messaging, the activity graph of UML 2.0 makes Request/Reply in itself concept patterns. The patterns are dependent on the business domain in BPMN. However, the concept of Produce/Consume Messaging is more efficient.

For Transparent Persistence, activity graph of

UML 2.0 does not support this feature. However, it helps for process data to be made persistent for later reuse.

As the result of analysis, activity graph of UML 2.0 draft has some parts that do not support the BPMN unique features. However, these BPMN features are the elements of making business process modeling more efficient rather than the essential element in business process modeling. Consequently, we can say that with activity graph of UML2.0 draft, business process modeling is possible, but lacks more in efficiency than BPMN.

5. Conclusion and Future work

We surveyed on the features of BPMN, the graphical notation for expressing business processes developed by the BPMI, and did research on the characteristics of activity graphs of UML 2.0 draft.

After comparing UML 2.0 with the features of BPMN in terms of modeling concept of business process, we analyzed the result of comparison with BPMN.

This paper might be the first to compare activity graph of UML 2.0 draft with BPMN. Because of this reason, our paper has some limitations. One limitation is that there are no suggestions on the effective improvement for complementing deficiencies of activity graph of UML 2.0 draft. In the future, more research on this will be done.

Reference

- [1] Unified Modeling Language 2.0 Proposal version 0.671 (draft), *OMG, 2002*
- [2] "The final draft for the Business Process Modeling Language (BPML 1.0)", *The Business Process Management Initiative*
- [3] Sergio de Cesare : "Business Modeling with UML: Distilling Directions for Future Research"
- [4] Bryon Baker "Business Modeling with UML: The Light at the End of the Tunnel", *The Rational Edge, Dec. 2001*
- [5] Jim Heumann, "Introduction to Business Modeling Using the Unified Modeling Language (UML)", *The Rational Edge, Mar. 2001*
- [6] Pan-Wei Ng, "Business Process Modeling and Simulation with UML", *The Rational Edge May. 2002*
- [7] Axel Korthaus, "Using UML for Business Object Based Systems Modeling", *Workshop Einsatz, Bewertung und Stand der Unified Modeling Language (UML), 1997*