

Mapping Timer Event from Business Process Model on Windows Workflow Foundation

Jian Deng^{*1,a}, Bo Chen^{2,b} and Jiazhi Zeng^{3,c}

¹School of Information & Software Engineering, University of Electronic Science & Technology of China, Chengdu 610054, Sichuan, China

²School of Information & Software Engineering, University of Electronic Science & Technology of China, Chengdu 610054, Sichuan, China

³School of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu 610054, Sichuan, China

^ajockeydj@gmail.com, ^bbochen@uestc.edu.cn, ^cjzzeng@uestc.edu.cn

Keywords: Timer Event, Flowchart Model, Business Process Modeling Notation, XML Process Definition Language, Windows Workflow Foundation

Abstract. A method to mapping timer event in business process model into Activity in Windows Workflow Foundation (WF) was proposed. This method can work with general business process because the model is designed in Business Process Modeling Notation (BPMN) 1.1 and it uses XML Process Definition Language (XPDL) 2.1 version as the file format. Experiment result showed that timer event could be directly mapped to flowchart model in Windows Workflow Foundation and the mapped flowchart model satisfied the functionality of timer event.

Introduction

Business Process Management (BPM) has been one of the hot research focuses in the domain of software engineering since 2003 [1]. Since different modeling tools often use similar but subtly different graphical symbols, this can hinder users from effectively understanding and communicating the process models.

A better way to model business process is adopting the Business Process Modeling Notation (BPMN) 1.1 version [2] published by Object Management Group (OMG) and saving it in the file format of XML Process Definition Language (XPDL) [3] proposed by Workflow Management Coalition (WfMC). TIBCO Business Studio [4] is a tool that supports designing in BPMN and XPDL. Microsoft's Windows Workflow Foundation (WF) [5] can be used to implement the business process model in the system implementation phase.

Automatic model conversion is one of the key technologies of MDA [6] to improve the development speed. The research works of reference [7, 8] show that the XPDL model can be mapped to WF model because there are certain workflow patterns supported by both XPDL 2.0 and Windows Workflow Foundation 3.5. Lin [9] developed a method to transform a subset of XPDL model into sequential model of WF 3.5. Due to the numerous types of elements in XPDL model, the scope of one paper does not permit a detailed discussion of all of these elements. For example, there are at least 30 types of events in XPDL specification, and each type of event has many attributes. Lin [9] did not discuss how to transform events in detail.

Timer event is an important type of event because it can act as a delay mechanism in the flow. As far as we know, there has been no published mapping method to map timer event from XPDL to WF. To solve the problem, the paper proposes a method to transform timer event from XPDL model to flowchart model in WF 4.0. WF 4.0 is our target platform because it has been totally redesigned and has many new features not existing in WF 3.5.

The rest of the paper is organized as the followings: Section 2 introduces the basic definitions in business process modeling. Section 3 presents the method to mapping timer event from XPDL to WF. Finally Section 4 draws the conclusion and previews future research content.

Business Process Model

The Business Process Modeling Notation (BPMN) and XML Process Definition Language (XPDL) are standards for illustrating and storing business process models in the requirement analysis phase of systems development. To facilitate the model mapping, the definition of process in our previous work [10] can be described as the following.

Definition 1. A process in a business process model is an eight-tuple $P = (E, T, S, G, O, D, C, \varphi)$, where

- E is a set of events which can be partitioned into disjoint sets of start events E_S , intermediate events E_I and end events E_E ,
- T is a set of tasks,
- S is a set of sub-processes that can be mapped to processes,
- G is a set of gateways used to model the convergence and divergence in a process,
- O is a set of pools,
- D is a set of data,
- C is a set of connections which can be partitioned into disjoint sets of sequence flow C_S , association flow C_A and message flow C_M ,
- φ is a set of function mappings from sub-processes to processes.

The Lanes, Text Annotations and Groups in XPDL are omitted because they do not have any direct effect on the flow of the process.

Windows Workflow Foundation (WF) 4.0 is a new paradigm for building business applications in the system implementation phase. Our previous work [11] shows that WF 4.0 has many advantages over WF 3.5. For example, the *Flowchart* activity is more attractive to programmers, and the function of *Sequence* activity is improved also.

Definition 2. A flowchart in Workflow Foundation is a directed graph $F = \{N, C\}$, where N is a set of nodes, C is a set of connections, $C \subseteq N \times N$.

FlowStep is one type of node used to perform a specified activity such as *WriteLine*, *Assign*, *Delay*, or a set of activities contained by *Sequence* or *Flowchart* activity, or any other type of class derived from *System.Activities.Activity*. The *Next* property of *FlowStep* is used to specify an unconditional successor node.

FlowDecision and *FlowSwitch* are the other two types of nodes. The *True* and *False* properties of *FlowDecision*, and *Default* and *Cases* properties of *FlowSwitch* are used to store their conditional successor nodes.

Mapping XPDL onto WF

This section discusses the methods to map timer event in XPDL to flowchart in WF in details.

Mapping *Id* of Event

In XPDL model, each element has a unique *Id* that distinguishes it from other elements within the model. The *Id* property of activity in WF is an identifier that is auto-generated and unique in the scope of the workflow model. Rule 1 shows how to establish the correlation between XPDL and WF.

Rule 1. Let e be an event in XPDL model, v_1 is the activity created according to e in WF model H , and the mapping *Id* of e can be formalized as following steps:

$$\left\{ \begin{array}{l} v = v_1 \\ H.Activites = H.Activites \cup \{v\} \end{array} \right\}, H.Type = Sequence$$

$$\left\{ \begin{array}{l} v = FlowStep \\ v.Action = v_1 \\ v.DisplayName = e.Name \\ H.Nodes = H.Nodes \cup \{v\} \end{array} \right\}, H.Type = Flowchart$$

$$wfobject = wfobject \cup \{ \langle e.Id, v \rangle \}$$
(1)


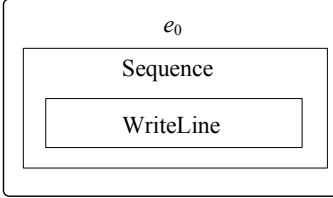
When H is an instance of *Sequence* activity, v equals to v_1 and it will be added to the *Activities* property of H directly, otherwise an instance of *FlowStep* is created to wrap v_1 so that it can be added into *Nodes* property of flowchart.

The *DisplayName* property of v is assigned so that it can be easily viewed in workflow designer. The *wfobject* is used to establish the correspondence of XPDL element and WF element. It is an instance of *Dictionary* so that the WF element can be found easily.

Mapping Name of Event

The *Name* attribute of XPDL element can be mapped to WF as the *DisplayName* property of the corresponding activity (Rule 1) and a *WriteLine* activity (Rule 2).

Table 1. Mapping the Name of event

XPDL model	WF model
	

Rule 2. Let e be an event in XPDL model, v is the *WriteLine* activity created according to e , and the mapping of *Name* can be formalized as following:


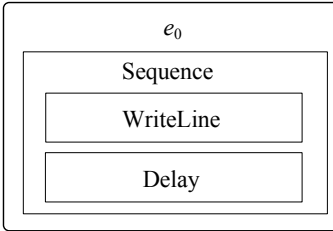
$$\begin{aligned} v &= \text{WriteLine} \\ v.\text{Text} &= e.\text{Name} \end{aligned} \quad (2)$$

The *Text* property of *WriteLine* is used to output some information so that we can trace the execution of each activity at runtime.

Mapping of Timer Event

In XPDL model, timer event will be triggered at a specific time-date or after a specific cycle. The timer event may be a start event or an intermediate event. When it acts as a delay mechanism in the flow, timer event can be implemented by *Delay* activity in WF. Table 2 shows the mapping method which can be implemented by Rule 3.

Table 2. Mapping of Timer Event

XPDL model	WF model
	

Rule 3. Let e be an event in XPDL model, v is the corresponding element in WF model H , and the mapping of e can be formalized as following steps:

$$\begin{aligned} v_1 &= \text{Sequence} \\ v_1.\text{DisplayName} &= e.\text{Name} \\ v_1.\text{Activity} &= \{v_2 \mid \text{map_element_name}(e)\} \cup \\ &\quad \{v_3 \mid v_3 = \text{Delay}, v_3.\text{Duration} = \text{TimeSpan.Parse}(e.\text{TriggerTimer.TimeCycle}), \\ &\quad \exists e.\text{Trigger} = \text{"Timer"}\} \\ v &= \text{add_to_model}(e, v_1, H) \end{aligned} \quad (3)$$

In Eq. 3, *map_element_name* is a function implemented according to Rule 2, and function *add_to_model* is realized by the steps shown in Eq. 1.

For timer event e , it is mapped to a sequence of *WriteLine* activity and *Delay* activity. For any other type of event, it is simply mapped to a sequence consisting of only *WriteLine* at present.

Summary

This paper has taken the first step to map timer event in XPDL model to flowchart model in Windows Workflow Foundation 4.0. It presents a model transformation method which has been implemented in a tool named XPDL2WF. By extending the workflow designer provided by reference [12], it allows the user to view and execute the result WF model. Future researches can include other types of events such as Message, Error, Cancel, Conditional, and Signal events.

Acknowledgements

This paper was supported by the project "Enterprise resource cooperation and research of key technologies" (No. 2007AA040801) and the project "International trade and regional economic cooperation platform and research of key technologies"(No.2009BAH46B03).

References

- [1] Hajo A. Reijers, Minseok Song, Heidi Romero, et al.:A Collaboration and Productiveness Analysis of the BPM Community, in: *7th International Conference on Business Process Management*, , edited by D. Umeshwar, E. Johann, p.1-14, Springer (2009).
- [2] Object Management Group: *DTC/2008-01-17 Business Process Modeling Notation* (OMG, USA, 2008).
- [3] Workflow Management Coalition: *WFMC-TC-1025 Process Definition Interface – XML Process Definition Language* (WfMC, USA, 2008).
- [4] TIBCO:"TIBCO Business Studio", http://developer.tibco.com/business_studio/
- [5] Microsoft:Windows Workflow Foundation,
<http://www.microsoft.com/visualstudio/en-us/products/2010-editions>
- [6] S. Sendall, W.Kozaczynski: Model transformation: The heart and soul of model-driven software development, *IEEE Software* vol.20 (2003), p.42-45.
- [7] Weqqing Lin, Jian Wang: Analysis of process control pattern of XPDL 2.0, *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems*, vol.13 (2007), p. 1839-1846.
- [8] M. Zapletal, W.M.P. Aalst, N. Russell, et al: An Analysis of Windows Workflow's Control-Flow Expressiveness, in: *Proceedings of the 7th IEEE European Conference on Web Services*, edited by R. Eshuis, P. Grefen, p. 200-209, IEEE Computer Society (2009).
- [9] Lin Miao, Tao Jin, Jianmin Wang: Transformation Technology from XPDL Model to WWF Model, *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, suppl.46 (2009), p.165-171.
- [10] Jian Deng, Zhi Chen, Jiazhi Zeng: Formal verification of business process models using Petri nets, *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems*, vol.17 (2011), p.1110-1119.
- [11] Jian Deng, Zhi Chen, Jiazhi Zeng: A method to check the structure of sequential business process models, in: *2011 International Conference on Automation and Robotic (ICAR 2011)*, edited by L. Gary, p.363-371, Springer (2011).
- [12] B. Bruce: *Pro WF:Windows Workflow in .NET 4* (Apress, USA, 2010)