Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 221–229
Expert Systems with Applications
www.elsevier.com/locate/eswa

# Composition of executable business process models by combining business rules and process flows

Sunjae Lee [a], Tae-Young Kim [a], Dongwoo Kang [a], Kwangsoo Kim [a,*], Jae Yeol Lee [b,1]

[a] *Department of Industrial and Management Engineering, Pohang University of Science and Technology, San 31, Hyoja-dong, Namgu, Pohang 790-784, Republic of Korea*

[b] *Department of Industrial Engineering, Chonnam National University, Gwangju, Republic of Korea*

**Abstract**

Emergency of Web services has promoted a new paradigm of a business process which is called the *Service-Oriented Business Process* (SOBP). The SOBP uses Web services as an implementation platform for activities that belong to a business process, and is modeled with a *Business Process Definition Language* (BPDL). A de facto standard of BPDLs is the Business Process Execution Language for Web Services (BPEL4WS), but the BPEL4WS requires extra mechanisms to explicitly specify process states in a process model and to separate business rules from process flows. By specifying the states explicitly and separating the business rules, a SOBP gets abilities to rapidly monitor a process in execution time, to efficiently define state-dependent process behavior, to freely change the business rules without modification of the flow, and to firmly guarantee business security by hiding the rules. In order to explicitly specify the process states and to separate the business rules from the process flow, an approach to a state-driven specification of a business process is suggested in this paper. The suggested state-driven approach inserts process states into a process model using process units, and separates the business rules from the process flow by regarding the states as the milestones which indicate where the rules are separated. Because the suggested approach supports a composition of a BPEL4WS executable process by combining the business rules and the process flow, the suggested approach can be used as a complementary design method of a BPEL4WS process model, and not as a substitute method of the BPEL4WS model.


*Keywords:* Service-oriented business process; Process states; Separation of business rules; Web services; Business process management

## 1. Introduction

Emergency of Web services has promoted a new paradigm of a business process which is called the *Service-Oriented Business Process* (SOBP). The SOBP uses Web services as an implementation platform for activities that belong to a business process, and the SOBP can be externalized as Web services (Leymann, Roller, & Schmidt, 2002). The SOBP is modeled with a *Business Process Definition Language* (BPDL) that composes Web services and defines complex interactions among the Web services. A de facto standard of BPDLs is the Business Process Execution Language for Web Services (BPEL4WS) which defines complex interactions among Web services focusing on the flow of message exchanges.

However, the BPEL4WS has two controversial points. First, the BPEL4WS model requires extra mechanisms to explicitly specify process states in a process model because the model is biased toward process flows. Since a process state is process condition at a particular execution time, and if states are explicitly specified in a process model, the progress of a process in execution time can be rapidly monitored based on the states. In addition, process states support to efficiently define state-dependent process behavior

* Corresponding author. Tel.: +82 54 279 2195; fax: +82 54 279 5998.
*E-mail addresses:* supersun@postech.ac.kr (S. Lee), west@postech.ac.kr (T.-Y. Kim), hyunil@postech.ac.kr (D. Kang), kskim@postech.ac.kr (K. Kim), jaeyeol@chonnam.ac.kr (J.Y. Lee).
[1] Tel.: +82 62 530 1782.

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.04.024

such as compensation/fault handling, negotiation, etc. Second, the BPEL4WS model needs additional mechanisms to separate the business rules from the process flow, because the flow and the rules are mixed up in the model. The separation of the business rules from the process flow provides the freedom of changing the rules without modification of the flow and guarantees business security by hiding the rules.

In order to explicitly specify the process states and to separate the business rules from the process flow, an approach to a state-driven specification of a business process is suggested in this paper. The suggested state-driven approach inserts the process states into a process model using process units, and separates the business rules from the process flow by regarding the states as the milestones that indicate where the rules are separated.

In the state-driven approach, it is suggested that process states should be explicitly specified in a process flow. The process states are explicitly specified using process units composed of an activity and two states that are process conditions before and after execution of the activity, respectively. The process units are connected by assigning a same identity to two process states that each belong to two different process units, and this set of connected process units defines the *Abstract Process Flow* (APF). Through the process units, therefore, the process states are explicitly specified between activities in the APF.

It is also suggested that the business rules should be separated from the process flow. If the suggested definition of the APF is conformed, it is natural that any business rule is not included in the APF. As a result, it can be considered that the business rules are successfully separated from the process flow. But the rules should be combined with the APF to create an execution model, so that the separated business rules are required to recognize the locations where the rules are applied in the APF. In order to recognize the location, the separated business rules generally in the format of ⟨***if–then–else–***⟩ should be extended to the format of ⟨***when–if–then–else–***⟩. This *Extended Business Rule* (EBR) includes state information at the ⟨***when–***⟩ element which indicates where the business rule is applied. Based on the state information, the EBR provides a foundation of combining the business rules with the APF when an execution model of a SOBP is created.

This paper is organized as follows: Section 2 provides a description of related works and Section 3 explains state-driven specification of a SOBP by separating business rules. Section 3.1 overviews the state-driven executable process modeling framework. An explicit specification of process states using process units is discussed in Section 3.2. Section 3.3 explains a separation of the business rules from the APF by extending the rules. Section 3.4 presents a specification of state-dependent process behavior, and a composition of an executable process model by combining the rules, and the flow is shown in Section 3.5. Section 4 discusses some issues on this research and concludes this paper.

## 2. Related works

The explicit specification of process states is not supported by most BPDLs such as the BPEL4WS (RosettaNet.org) and the Web Service Choreography Description Language (WSCDL) (Kavantzas, Burdett, Ritzinger, Fletcher, & Lafon, 2004). Both the BPEL4WS and the WSCDL do not provide any standard element for the purpose of explicit specification of the process states. Instead of explicit specification, they suggested an ad hoc mechanism of implicit state specification. The ad hoc mechanism of the implicit specification manages a process state by assigning an arbitrary variable to keep information on the process state, and then updating the variable whenever the process state changes. However, because the ad hoc mechanism does not provide any standard way of naming the state-related variable, it causes considerable ambiguity about which variable keeps the process state. In order to overcome this ambiguity about the state-related variable, systematic management of the process state is required. As the systematic management, it is suggested in this paper that process states at significant process times should be explicitly specified across a process flow using a state-related standard element.

The importance of separating the business rules from process flows is mentioned in the BPEL4WS specification document. However, the BPEL4WS specification document does not contain any mechanism as to how the business rules are separated from the process flow in the BPEL4WS model. The separation mechanism included in the BPEL4WS document is that a common model is extended either to an executable process which should not be opened to the public or to a business protocol which can be opened to the public (Thatte, 2003). According to the BPEL4WS document, the common model contains essential elements of process flow, business rules, and data manipulations (Thatte, 2003). This common model can be extended to the executable process by adding execution-related elements to the common model, and the common model also can be extended to the business protocol by hiding only data manipulations (Thatte, 2003). However, because this extension to the business protocol does not provide any mechanism for hiding the business rules, the business rules are in danger of being revealed. Therefore, the BPEL4WS requires an extra mechanism for the separation of the business rules from the process flow. In conclusion, though the BPEL4WS specification document mentions importance of the separation, the document does not provide any mechanism for the separation of the business rules from the process flow, so that an extra mechanism for the separation is required.

The separation of the business rules from the process flow seems to be considered in the WSCDL specification document, since any business rule is not included in the WSCDL process model. Actually, the WSCDL process model ignores the business rules, but does not separates the business rules from the process flow (Kavantzas