Computer Languages, Systems & Structures 38 (2012) 279–299

Contents lists available at SciVerse ScienceDirect

Computer Languages, Systems & Structures

journal homepage: www.elsevier.com/locate/cl

Data-bound variables for WS-BPEL executable processes

Marcel Krizevnik*, Matjaz B. Juric

University of Ljubljana, Faculty of Computer and Information Science, Laboratory for Integration of Information Systems, Trzaska 25, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 30 January 2012
Received in revised form
8 May 2012
Accepted 18 June 2012
Available online 26 June 2012

Keywords:
WS-BPEL
Service oriented architecture
Service composition
Data synchronization

ABSTRACT

Standard BPEL (Business Process Execution Language) variables, if used to store the data from a data store, cannot be automatically synchronized with the data source in case other applications change the data during the BPEL process execution, which is a common occurrence particularly for long-running BPEL processes. BPEL also does not provide a mechanism for active monitoring of changes of data that would support automated detection and handling of such changes. This paper proposes a new type of BPEL variables, called data-bound variables. Data-bound variables are automatically synchronized with the data source and thus eliminate the need to implement data synchronization manually. To provide support for data-bound variables, we propose specific extensions to BPEL and the use of appropriate Data Access Services (DAS) that act as data providers. We introduce new BPEL activities to load, create and delete remote data. We also introduce observed properties, observed property groups and a variable handler. Using this mechanism, the BPEL process is able to automatically adapt to changes to data, made inside or outside the process scope, by following the Event, Condition, Action (ECA) paradigm. As a proof-of-concept, we have developed a prototype implementation of our proposed BPEL extensions and tested it by implementing three pilot projects. We have confirmed that our proposed solution decreases BPEL process size and complexity, increases readability and reduces semantic gap between BPMN process model and BPEL.

1. Introduction

In this article, we address the problem of using out-of-date data in Web Services Business Process Execution Language 2.0 (WS-BPEL 2.0, or simply BPEL) processes. BPEL is a commonly accepted standard for defining business processes with composition of services in service oriented architecture (SOA) [19]. Standard BPEL variables are a snapshot of data returned by a service and thus present a duplicate version of remote data at some particular time. However, in certain cases we want to ensure that the process always uses the latest version of important business data as other applications may change the data in the data source during the BPEL process execution. This problem arises particularly in case of long-running BPEL processes, which can take a few days, weeks or even months to complete.

The BPEL specification [17] does not provide any support for automated synchronization of variables. If we want to ensure that a BPEL process always uses the latest version of data, we have to implement data synchronization manually. This is usually done by adding additional service invocations as part of the business process flow. BPEL also does not provide a mechanism for active monitoring of variables that would allow us to define if-then rules that trigger

* Corresponding author. Tel.: +386 41319451; fax: +386 14264647.
E-mail addresses: marcel.krizevnik@fri.uni-lj.si, matjaz.juric@fri.uni-lj.si (M. Krizevnik).

1477-8424/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cl.2012.06.001

corresponding actions in case the data has changed and the specified conditions are satisfied. BPMN (Business Process Model and Notation) 2.0 [8], on the other hand, supports the definition of such rules through the conditional event construct (rule event in BPMN 1.0) [5]. To implement similar functionality in BPEL, we have to perform manual checking in form of additional process steps. Adding such data synchronization and data checking code in several places in the business process can lead to a bad process design with code duplication and the code size and complexity are increased. As the complexity directly impacts readability [29,30], the readability of the business process is also reduced.

To overcome these problems, we propose a new type of BPEL variables, called data-bound variables. Data-bound variables use Service Data Objects (SDO)-based form and are automatically synchronized with the data in the data source. They also support automated detection and handling of changes to data that were made inside or outside the BPEL process scope by following the Event, Condition, Action (ECA) paradigm [18], which is already supported in BPMN. In this way, we are able to remove most of the data synchronization and data checking code from the basic business process flow. Thus, we improve the BPEL process design. As we introduce support for a concept that is already supported in BPMN (data-based ECA rules), we also reduce semantic gap with BPMN and thus simplify the translation between BPMN and BPEL [11]. To provide support for data-bound variables, our solution extends BPEL and leverages the use of Data Access Services (DAS) that act as a data provider. We have designed the extensions for BPEL version 2.0 using the standard language extension mechanism.

The article is organized in nine sections and one appendix. In Section 2, we present motivation for providing automated synchronization between BPEL variables and the data source. Section 3 gives a brief overview of BPEL and SDO. In Section 4, we describe our proposed solution for data-bound variables from a high-level perspective. In Section 5, we describe the proposed solution for the DAS and our extensions to the BPEL partner link type (BPEL extensions to WSDL) that are used to standardize the communication between the BPEL engine and the DAS. Section 6 describes the proposed extensions to BPEL, such as new activities and extensions to existing activities. In Section 7, we present the proof-of-concept, where we describe a procurement process case study and evaluate positive effects of using our proposed solution using business process metrics. In Section 8, we present related work and discuss the results. In Section 9, we give conclusions. Appendix defines syntax specification for the proposed BPEL extensions.

2. Motivation

BPEL is widely used in different application domains, where special requirements and new challenges arise. One such important requirement is that a BPEL process should always use the latest version of important business data. We present two types of business processes where this requirement occurs:

- Long-running BPEL processes that have to follow strict laws and regulations. A typical example of such process is a procurement process in a large state-owned organization (for a complete motivating example please refer to Section 7), where some process instances can take a few weeks or even months to complete and some important business data (such as suppliers contact information, credit rating, list of references, status, etc.) has to be refreshed at various steps in the process. Other examples of such long-running processes are loan approval process, inventory control process, etc.
- BPEL processes that operate with important data that changes very often. For example, a business process for booking airline tickets should always use the latest air fare prices, as these can change very often.

BPEL currently supports three types of variables: WSDL message type, XML simple type and XML schema element [19]. WSDL message type variables are the most commonly used type of variables and are used to store the data that is exchanged between business partners. Other two types of variables hold data which is used in business logic and for composing messages sent to partners. BPEL process variables may contain important business data (sometimes called master data [1]) that presents the core of the business process. If the organization has already performed data integration, master data may reside inside a central MDM (Master Data Management) repository [1]. Master data may include information about (but not limited to) customers, products, employees, materials, suppliers, etc. If such key business data is stored using a standard BPEL variable, the variable presents a duplicate version of remote data at some particular time. This means that the process may use out-of-date version of the data, as other applications may change the data during the BPEL process execution. In certain cases this does not present a problem, however, in some applications, the use of out-of-date data may result in invalid execution of key business activities [1,33]. Therefore, this problem has to be addressed. If, for example, a flight ticket booking process operates with out-of-date flight ticket data, it may not be able to find the cheapest ticket or it may try to book a ticket that is no longer available. Similarly, if the long-running procurement process does not continuously synchronize the list of suppliers, it may not detect that certain suppliers that have sent offers may have gone out of business or have been suspended during the process execution. For the similar reason, the loan approval process should always use the customer's latest credit rating.

As the BPEL specification [17] does not provide a mechanism for automated synchronization of the data, a BPEL developer has to implement these steps manually. This can be done by adding data synchronization steps into the process flow. To perform one such data synchronization, we need to add at least one ⟨invoke⟩ and two ⟨assign⟩ activities and introduce two variables that represent input and output for the service call. Furthermore, sometimes it is not enough just