Simulation Modelling Practice and Theory 18 (2010) 650–662

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Mediator approach to direct workflow simulation

Duckwoong Lee, Hayong Shin, Byoung K. Choi *

VMS Lab., Department of Industrial and Systems Engineering, KAIST, 335 Gwahak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea

ARTICLE INFO

Article history:
Received 26 November 2008
Received in revised form 29 July 2009
Accepted 12 January 2010
Available online 18 January 2010

Keywords:
Direct workflow simulation
Mediator
DEVS model
Participant simulator

ABSTRACT

This paper presents a *direct workflow simulation* method with which the future *enactment service processes* of a BPM system can be simulated directly (i.e., without a model conversion). The proposed method may easily be implemented on a commercial BPM system by plugging in a couple of software modules (no internal modification of the BPM system is required). Previous researches on workflow simulation relied mostly on *conversion methods* in which *process definition models* (PDMs) are converted to simulation models and the simulation is performed by a separate simulator. More recently, a direct workflow simulation method based on the concept of *listener* was proposed. However, with the *listener approach*, (1) some internal modification of the BPM system is required, (2) PDMs have to be modified slightly, and (3) reliable simulation is not guaranteed. The direct workflow simulation approach proposed in this paper, which we call the *mediator approach*, is free of these shortcomings. Moreover, the mediator approach is suitable for workflow simulation involving multiple BPM systems. In a 'direct' workflow simulation, (1) the work-list handler of each *participant* is replaced by a *participant simulator*, (2) simulation is carried out by the workflow engine of the BPM system, and (3) a software module called *synchronization manager* (mediator or listener) handles time synchronization during simulation. In this paper, the architecture and detailed logic of the mediator are described as DEVS models. The behaviors of participant simulators are also described as DEVS models. The proposed workflow simulation method has been implemented on a commercial BPM system as well as on an academic BPM system, and an illustrative workflow simulation example is provided.

1. Introduction

According to the Workflow Management Coalition, the *workflow management system* (WfMS) is a software system that completely defines and automatically executes workflows in order to manage the actual flow of work so that the right work is done at the right time with the right information by the right person in the organization [10]. Recently, a WfMS used for business process management (BPM) is often called a ***BPM system***, with an emphasis on orchestrating operational business processes that are driven by explicit process designs [2,9]. Along with this development is an increased awareness of need for workflow simulation in BPR (business process reengineering) [9], and the simulation is regarded as a key function of a BPM system [2,3].

The software module of a BPM system in charge of managing the actual flow of work is called a *workflow engine* or ***enactment server*** (we use the latter term throughout the rest of the paper), the service provided by the enactment server to automatically execute workflows is called *enactment service*, and the people involved with enactment service are called ***participants***. An explicit model of business processes to be managed is called **PDM** (process definition model), and an instance of PDM being executed is called ***process instance***. A process instance is a directed graph with each node representing an *activity* to be performed by a participant. Activities are sent to the participants via ***work-list handlers***.

* Corresponding author. Tel.: +82 42 350 3115; fax: +82 42 350 3110.
E-mail address: bkchoi@kaist.ac.kr (B.K. Choi).

1569-190X/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2010.01.009

The process life cycle of BPM consists of a series of phases [9]: *Discovery* (clarification of how business processes work), *design* (building PDMs for business processes), *execution* (providing enactment services), *operation* (interacting with participants while monitoring the business processes), *optimization* (improving PDM), and *analysis* phases. There may be three levels of feedback in the BPM life cycle: (1) feedback from the operation phase to the execution phase at run time, (2) feedback from the optimization phase to the design phase at build time, and (3) feedback from the analysis phase to the discovery phase. Workflow simulation corresponding to each level of feedback is referred to as *run-time workflow simulation* (for dynamic rescheduling of process instances), *build-time workflow simulation* (for improving PDM), and *BPR workflow simulation*, respectively. A BPM system equipped with these feedback mechanisms (i.e., workflow simulations) is called a *closed-loop BPM system* [34]. Often the terms ***workflow simulation*** and *BPM simulation* are used interchangeably [18].

Early researches on workflow simulation focused mostly on BPR [17,19–21,23,24,33], where general business processes or workflows were simulated. These researches had little to do with BPM systems. Another line of workflow simulation researches focused on validating and/or optimizing PDMs of a BPM system, where the PDM is converted into a formal model such as Petri-net and DEVS or to a proprietary simulation language and then simulation is carried out with the converted model [1,14,15,26,28,30–32]. The build-time simulation concept was also employed in animation-based debugging of software systems [6]. With this conversion approach, there may be some information loss during the conversion, and the participants' behaviors may not be easily converted.

Recently, the authors' group presented a *listener approach* to *direct workflow simulation* with which process instances, together with the behaviors of participants, can be simulated directly (i.e., without a model conversion) using the enactment service mechanism of a BPM system [22,29]. In a direct workflow simulation, (1) the work-list handler of each *participant* is replaced by a *participant simulator*, (2) simulation is carried out by the enactment server of the BPM system, and (3) a software module called *synchronization manager* handles time synchronization during simulation. However, with the *listener approach*, (1) some internal modification of the BPM system is required, (2) PDMs have to be modified slightly, and (3) reliable simulation is not guaranteed. More details of the listener approach will be given in the next section.

This paper proposes a different method of *direct workflow simulation* in which the synchronization manager 'mediates' the communications between the enactment server and participant simulators. The direct workflow simulation approach proposed in this paper, which we call a *mediator approach*, is free of all the shortcomings of the listener approach. Another advantage of the mediator approach is that it is suitable for workflow simulation involving multiple BPM systems. In this paper, details of the mediator approach are described employing the DEVS formalism [13], and an illustrative implementation using a commercial BPM system [1] is presented.

The rest of the paper is organized as follows. In order to make the paper self-contained, basics of enactment service mechanism and a review of previous works are presented in the next section. DEVS models of the mediator approach are presented in Section 3 and an implementation example is given in the section that follows. Conclusion and discussions are provided in the final section.

2. Background and previous works

2.1. Enactment service mechanism of a BPM system

As the proposed workflow simulation method makes use of the enactment service mechanism of the BPM system, basics of enactment service are briefly described using Fig. 1. For each instance of workflow, a process instance (PI) is created from its process definition model (PDM). Depicted in Fig. 1 is a PI consisting of seven activities including start and end activities

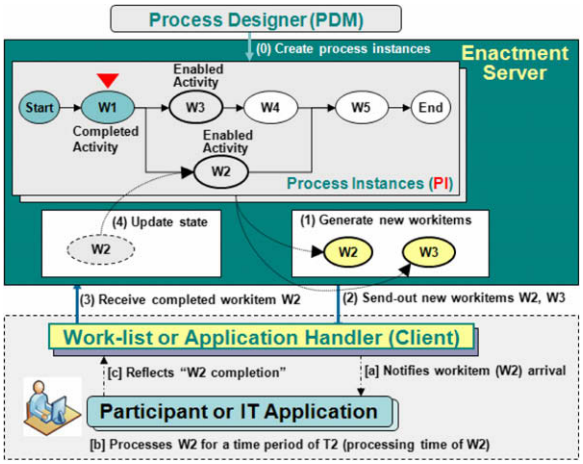

Fig. 1. Enactment service mechanism.