Journal of Computational Science 10 (2015) 126–136

Contents lists available at ScienceDirect

Journal of Computational Science

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

Time-based orchestration of workflow, interoperability with G-Devs/Hla

Judicaël Ribault*, Gregory Zacharewicz

Univ. Bordeaux, IMS, UMR 5251, F-33400 Talence, France

ARTICLE INFO

Article history:
Received 24 February 2014
Received in revised form 11 June 2014
Accepted 23 July 2014
Available online 5 August 2014

Keywords:
Workflow
Taverna
Interoperability
Discrete event simulation
G-DEVS
HLA

ABSTRACT

Enterprises information systems (EIS) take benefits of latest advanced of web services and Internet of things to improve information retrieving and gathering for decision making. Furthermore, EIS should permit a more comprehensive information routing in the company within an electronic workflow in order to save time, cost and to reduce production impact on the environment. Such software has to interact frequently with real world data acquired by different sensors. Nevertheless this combination of software and hardware devices frequently faces interoperability problems. Also, testing and validating the EIS is not trivial without testing in real condition that can lead to deploy the large system. Authors assumed that testing and validating part of the system behaviour can be anticipated progressively by simulation, permitting then more progressive and confident system integration. This paper proposes to introduce a new workflow demonstration platform to combine simulation world with real world interacting with sensor, human interfacing and web service calls. In detail, this paper proposes to combine the Taverna Workflow tool, which handles and triggers web services call proposed by a platform server, to other software components. This combination has revealed one drawback of major workflows orchestrators; they do not provide time management facilities to handle synchronization during parallel execution of interdependent workflows. To overcome that limitation a clock ordering solution has been added by reusing G-DEVS/HLA to synchronize workflows running in parallel. The imbrications of G-DEVS M&S with Taverna workflow is now operating thanks to HLA. This work is validated by demonstrating the interoperability and the complementarity of these approaches on a logistic platform study case.

1. Introduction

The effectiveness of enterprise information system (EIS) depends not only on the internal interconnectivity of its inner software components, but more and more on its ability to exchange data, so to collaborate, with every day new tools developed and updated in the environing digital world. This requirement led to the development of the concept called interoperability that intends to improve collaborations between EIS companies. No doubt, in such context where more and more networked enterprises are developed; enterprise interoperability is seen as one of the most wanted feature in the development of an EIS. Also, data treatment calls actions of both human processing and automatic treatments. The sequencing of these actions should be controlled or orchestrated by a high level application that can decide the human resource and/or component to solicit. The sequence of actions is commonly entitled Workflow (WF) and its administration Workflow management. This field is studied and standardized by the Workflow Management Coalition (WfMC) [1,2].

Several research-works have been launched since the 1990s in the field of WF. Workflow was first designed to formalize and improve enterprise business process. A production workflow is a set of linked steps required for developing a product until it is put on the market [3]. The workflow steps are based on observing a number of steps that were originally manually enchained then formalizing them to be computer assisted. The research on the WF initiated by the Workflow Management Coalition [1,2] and used for instance in [4] was a premise to current WF modelling (e.g., with Business Process Model and Notation (BPMN) [5]). It has permitted for instance the development of Build Time models used for setting Enterprise Resource Planning (ERP) systems.

Deploying such WF is a critical task for the companies that continue to rely on their EIS during the setting. Moreover, the proper functioning is difficult to achieve because the installing team does not have vision or access to the whole system (EIS environment)

* Corresponding author. Tel.: +33 626907958.
E-mail addresses: judicael.ribault@u-bordeaux1.fr, jribault@gmail.com (J. Ribault), gregory.zacharewicz@u-bordeaux1.fr (G. Zacharewicz).

http://dx.doi.org/10.1016/j.jocs.2014.07.002
1877-7503/© 2014 Elsevier B.V. All rights reserved.

during settings, so the final global behaviour is difficult to predict. Executing the WF on part of the real system, while simulating some critical parts that will then be deployed, can be a good option to test the WF behaviour and reduce risk and cost. However, most WF tools and service orchestration are limited in the handling of time management. But without time consideration, executing a parallel simulation with disjunction and junction gateway between tasks is difficult. Distributed simulation has a long time experience in this field and can be an answer for this problem. Few approaches combine efficiently Modelling and Simulation (M&S) and real executions in the WF domain. Main reasons are: slowing-down due to synchronization of the simulation engine, that is usually constrained by pessimistic causality [6] between real and simulated time, and interoperability barriers that are faced between different hardware and software [7].

Recent improvements in web-based development propose new facilities to connect applications in a more convenient way. For instance, web services can solve part of the interoperability question. WF can be used as an interoperability layer between services, and especially Web Services (WS). This paper proposes to use web services and WF for interoperability between simulation and real-world application. Web services enable the integration of applications or data from heterogeneous sources (i.e., Mash-up). Nevertheless the synchronization is not solved; this work proposes an additional component from the High Level Architecture standard named Run Time Infrastructure (RTI) reused from the G-DEVS/HLA works. In the end, this paper proposes to apply the use of WF Web services and simulation to the transport domain application through the PRODIGE project to validate the approach.

Section 2 describes the necessary background needed to understand how WFs of services and simulation can drive real application. Section 3 presents the scientific contribution while Section 4 put it into practice in a real application framework.

2. Background

In this section, we first present the enterprise interoperability concept. Then we briefly present the HLA standard for interoperability of simulation and how WF can be used for experimentation. We recall the DEVS formalism and G-DEVS. Finally we introduce the Taverna WF management system to orchestrate the experimentation.

2.1. Enterprise interoperability

Enterprise interoperability [7] refers to the interaction ability between enterprise systems. The interoperability is considered as significant if the interactions can take place at least at the three different levels: data, services and process, with a semantic defined in a given business context.

Interoperability extends beyond the boundaries of any single system, and involves at least two entities. Consequently establishing interoperability implies relating two systems together and removing incompatibilities. Concepts related to enterprise interoperability are classified into three main dimensions as described in [7]:

- The integrated approach demands all partners to have the same description of information.
- The unified approach asks partners to prepare the data to exchange in order for it to be compliant with a Meta model but local description can be kept.
- The third approach is federated. Here, interoperability must be accommodated on-the-fly between partners without considering a pre-existing Meta model.

The goal of these approaches is to tackle interoperability problems through the identification of barriers (incompatibilities) that prevent interoperability. The first kind of barrier concerns the non-existence of commonly recognized paradigms and data structure, for that, clarification is required to propose a sound paradigm. The second requirement is the synchronization of data. The order of exchanged data is important, ignoring this can lead to misunderstanding and malfunction of the model. Finally the enterprise modelling must take into account the confidential management of data. In this privacy context, concurrent enterprises must define data sharing strategies. The interoperability can be considered between concurrent enterprises in that context, a strategy of data sharing/not sharing between these must be defined. In the presented work the interoperability is focused between WF simulation and service calls. In the simulation domain, the HLA is established as the interoperability reference.

2.2. Simulation interoperability with HLA

The High Level Architecture (HLA) [8,9] is a software architecture specification that defines how to create a global software execution composed of distributed simulations and software applications. This standard was originally introduced by the Defense Modelling and Simulation Office (DMSO) of the US Department Of Defence (DOD). The original goal was the reuse and interoperability of military applications, simulations and sensors.

2.2.1. HLA concepts

In HLA, every participating application is called federate. A federate interacts with other federates within a HLA federation, which is in fact a group of federates. The HLA set of definitions brought about the creation of the standard 1.3 in 1996, which then evolved to HLA 1516 in 2000 [8] and finally to 1516 evolved [10].

The interface specification of HLA describes how to communicate within the federation through the implementation of HLA specification: the Run Time Infrastructure (RTI). Federates interact using the proposed services by the RTI. They can notably “Publish” to inform on the intention to send information to the federation and “Subscribe” to reflect information created and updated by other federates. The information exchanged in HLA is represented in the form of classical object-oriented programming. The two kinds of object exchanged in HLA are Object Class and Interaction Class. The first kind is persistent during run time, the other one is just transmitted between two federates. These objects are implemented with XML format. More details on RTI services and information distributed in HLA are presented in [8,10]. In order to respect the temporal causality relations in the execution of distributed computerized applications; HLA proposes to use classical conservative or optimistic synchronization mechanisms [11]. In HLA 1516 Evolved [10] the service approach is demanded as core feature. Nevertheless no software addresses completely that goal at the moment [26].

2.2.2. HLA implementation components

An HLA federation is composed of federates and a Run time Infrastructure (RTI) [8].

A federate is a HLA-compliant program, the code of that federate keeps its original features but must be extended by other functions to communicate with other members of the federation. These functions, contained in the HLA-specified class code *FederateAmbassador*, make the information received from the federation interpretable by a local process. Therefore, the federate program code must inherit the *FederateAmbassador* class code, complete the abstract methods defined in this class, to be able to receive information from the RTI.