Event-based Business Process Editor and Simulator

Vatcharaphun Rajsiri, Nicholas Fleury, Graham Crosmarie and Jean-Pierre Lorré

EBM WebSourcing, 4 rue Amélie, 31000 Toulouse, France
{netty.rajsiri, nicholas.fleury, graham.grosmarie, jean-pierre.lorre}@petalslink.com

Abstract. The growing of business market dictates new requirements of agility to the business process environment. An event-driven approach can deal with this issue since an event can be defined as a significant change in the state of a system or an environment. This paper is focused on the combination of the event-driven approach and the business process modeling one by developing a cloud-enabled event-based business process editor and simulator. BPMN2.0 is the relevant business process formalism used since it can represent graphically various kind of operating activities and events.

Keywords: BPMN, business process modeling, simulation, EDA, SOA

1 Introduction

This paper aims at presenting the value of the event-based business process modeling approach illustrated thanks to software components which are part of a complete event based open-source platform.

The paper starts by discussing the importance of business process modeling and event-driven approaches and synergies with the Service-Oriented Architecture (SOA) principles. Then, it presents the goals of EBM WebSourcing in terms of open-source cloud enable platform based on such approaches. Finally, the software components designed to support these perspectives are described.

1.1 Event and Business Process Management (BPM)

Enterprises are now operating in a complex economical environment where markets are more competitive and dynamic. They require the agility to be able to operate under such pressures and thus ensure their survival. The way for an enterprise to do business is captured in its business process which is seen today as the most valuable corporate asset. A business process according to [18] is an artifact made of a set of activities executed in order to achieve at least one objective. The agility of enterprise business processes to change in a timely manner and rapidly adapt themselves to new conditions is though very important.

SOA promises to provide a decentralized and loosely coupled environment that enables flexible, reliable and coordinated integration of dynamic applications belonging to different organizations. More and more enterprises start organizing their

business processes by means of service aggregation. We assume that business process is based on the SOA principles, with both aiming to empower the organization to more quickly respond to changing business requirements that result from events [3]. This is where an event-driven approach comes into play.

The fundamental concept of the event-driven approach is, obviously, an event. An event is a notable thing that happens inside or outside the enterprise [11]. It can be defined as a significant change in the state of a system or an environment [10]. The event-driven approach concerns the production, detection, consumption of, and reaction to events. This event-driven approach may be applied to the design and implementation of applications and systems which transmit events among loosely coupled software components and services.

Business process and event-driven approach complement service-oriented architecture because services (within business process) can be started by triggering events which can influence the execution of a process or a service.

Moreover, nowadays the modern enterprise operates in a cloud of business events emanating from sources all over the world. Events can be monitored, managed and processed by a complex event processing (CEP) engine using event patterns [17]. CEP allows real-time respond and better control of business processes.

[2] confirms the requirements for handling events in process models and discusses the notion of events for triggering process instantiation or steps within a process instance. Events provide content for analyzing and acting during the business process. They are a way to loosely interconnect different process instances: Events produced in one process instance are consumed by one or several other process instances. Furthermore, composite events, i.e. the combination of different interrelated events, must also be handled in process models thanks to Complex Event Processing (CEP).

Besides of describing the architecture and tools, [4] outlines a methodology by which each participant can define, detect, and respond to events. It proposes an Event-driven Architecture (EDA) which aims to address the engineering challenge of incorporating events in a web process in a reusable manner - that is, without hard coding events into a process model. From [17], an EDA provides a way of organizing systems that sense, analyze, and respond to events. For business processes, sensing involves receiving events from multiple sources (sensors, software applications, and such), analyzing involves deciding a response (perhaps by aggregating such events), and responding involves updating expectations and modifying executions.

According to [7], the trend of BPM is shifting from the design of business process towards improving and analyzing business processes. Therefore, we have devoted our interests to a platform allowing collaborative design and verification of event driven business processes through the development of an editor and a simulator. The editor is dedicated to the process design, whereas the simulator is focused on the process verification.

1.2 Technologies positioning

An event-driven business process modeling approach is the basis of the development of the Business process editor and simulator which aim to offer users the ability to collaboratively create, modify, and simulate processes made of events and services.

The positioning of the editor and simulator in the existing open-source SOA products portfolio of EBM WebSourcing is shown in Fig.1 below:

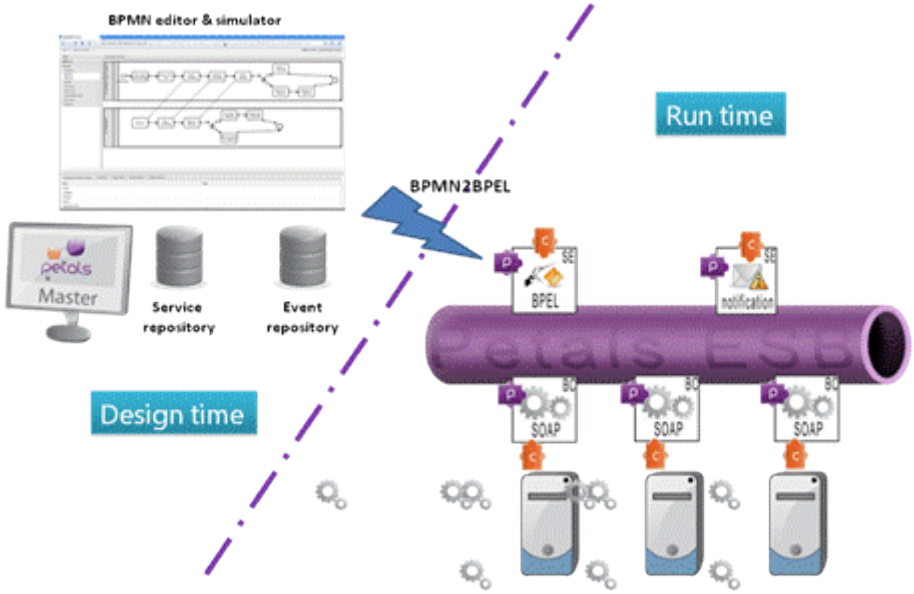

Fig. 1. Positioning of the Business Process Editor and Simulator

The editor and simulator are used during design time to build Event Driven Applications deployed into the service cloud managed by the Petals ESB (Enterprise Service Bus) event enabled service infrastructure. Users define processes using available event types and services from the event and service repositories. The service repository is the Petals Master open-source registry. Event repository contains an event taxonomy that describes event types.

The process designed with the editor is then translated to be executed by the BPEL [9] (Business Process Execution Language) engine provided by the Petals ESB open-source JBI[1] compliant ESB. This ESB may be distributed either physically or logically on many nodes that manage their own services and events according to an OASIS/WS-Notification[2] compliant engine. Communication between service providers and consumers use the event paradigm as explain below:

An event can happen at any moment produced by any kind of services: a new observation, a file created or deleted, a change on a meta-data, etc. Once events have been produced, consumers have to be notified. However, in order to ensure loose coupling between services, event producers should not know event consumers, neither the means to send them notifications. The classical way to deal with such a schema is to route the notifications via a broker who will be in charge of maintaining a list of consumers with their specific protocol of exchange, as well as their subjects of

[1] Java Business Integration: http://jcp.org/en/jsr/detail?id=208
[2] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

interest (also known as topics of publish/subscribe design pattern). Such kind of communication is covered by the OASIS WS-Notification standard.

The BPMN2BPEL translator converts BPMN 2.0 compliant business processes to a set of executable artifacts: BPEL code, WSDL service interfaces, JBI configuration files, XSD data models (service parameters and events). The BPEL engine orchestrates services in order to implement the specified process. It takes in charge service invocation triggered by events thanks to features provided by the underlying bus infrastructure and notification engine.

The business process editor and simulator are detailed in Sections 2 and 3 respectively.

2 Business Process Editor

The Business Process Editor aims at providing an environment for users to be able to model their business processes through a Web client interface. It provides a set of features allowing the creation of standard elements of BPMN 2.0 (Business Process Modeling Notation).

In this section, we will survey the main reasons behind the choice of the BPMN syntax, the functionalities of the editor that have already been implemented in the current version, and finally detail the collaborative feature.

2.1 Business Process Modeling Notation (BPMN)

The Business Process Editor has been developed to support the process representation based on the BPMN 2.0 specification [15]. BPMN 2.0 introduces additional constructs and a standardized serialization format to the previous one (BPMN 1.2).

BPMN is a semi-formal language for graphically representing processes defined initially by the BPMI[3] (Business Process management Initiative). BPMI has been established in order to promote and develop the use of Business Process Management (BPM) through the use of standards for process design, deployment, execution, maintenance, and optimization of processes. It merged with OMG (Object Management Group) in 2005.

BPMN is compatible with XML-based workflow languages like BPEL. The version 2.0 provides a complete mapping between BPMN models to BPEL. This facilitates the transformation of a BPMN model into an executable process one.

According to [13], BPMN can cover mainly the functional view and partially the organizational, informational, and resource views of enterprise modeling through pools, lanes, activities, flows, and data notations. BPMN integrates the event notation which is the fundamental concept of the event-driven approach.

In the first section, we discussed about the necessary of taking the event-driven approach into account when modeling a business process. An event gives the dynamic behavior to business process because it can deal with changes that happen in any moment of times and affect the sequence of activities of a process.

[3] www.bpmi.org

BPMN takes into account event by specifying three types of events: start, intermediate, and end events. A BPMN process is triggered by an event to start. Events can also appear anywhere within the process (as intermediate events) and it is always placed at the end of process (as end events), for instance, the process is stopped immediately when an error occurs within the process.

Thus, BPMN language is the relevant formalism for us since it allows combining BPM and event approaches.

2.2 Technologies

The main target user group of the Business Process Editor and Simulator are business users, not technical ones; consequently tools should be easy to install, update, and use. Moreover business processes instances will run into the cloud, meaning services and events managed by the bus are available through Internet. Such requirements imply to develop our software as lightweight Web applications.

Rich Internet Application (RIA) [6] aims to build lightweight applications that can be accessible from anywhere and they are upgraded instantly while providing the same rich full features as the classical software. The HTML standard and Javascript language are commonly used to develop RIA as they are supported on many devices and are quiet easy to manipulate. However, over time, it becomes difficult to maintain applications developed with such technologies. On one hand, the supports on which they are based are evolving rapidly and are also quiet numerous. The problem is that all these technologies have no common standards, thus requiring specific development for the application to develop. On the other hand, developing rich applications involve the integration of many features that require a development language that can lead to a robust application. A language such as Javascript does not allow this because it is weakly typed making the developed application hard to maintain.

We selected Google Web Toolkit (GWT) [5] since it is a framework that aims to develop RIAs using Java as a language of development which, unlike JavaScript, is a strongly typed language and whose maintenance can be easier. Moreover GWT offers the advantage of not having to worry about the supports on which the application will run, because once the application has been developed, it is compiled to be compatible with a maximum of support. GWT also allows developing libraries that can easily be integrated within other applications, helping by the way the reuse of components.

Besides, Scalable Vector Graphic (SVG) [15] is used to deal with the graphical representation of the main BPMN widgets. It allows rendering graphical elements within an HTML page. SVG is an XML-based file format for describing two dimensional vector graphics. The advantage of SVG, in addition of being easily manipulated, is that it is readable by many applications as this is a well established standard.

2.3 Features

The aim of the Business Process Editor is to be a design tool for business process in BPMN representation. The Fig. 2 shows the user interface of the editor:

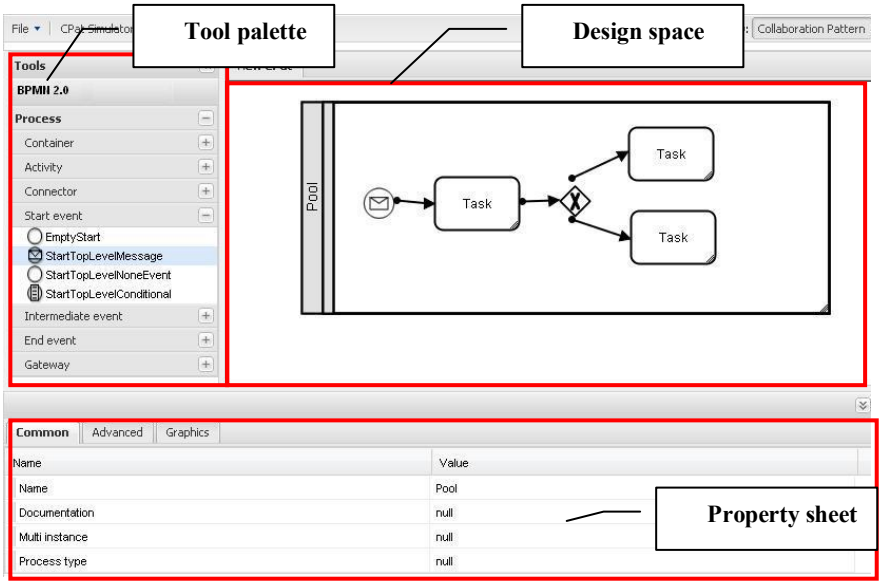

Fig. 2. User interface of the Business Process Editor

The editor provides a tool palette, a design space, and a property sheet. The tool palette contains the widgets for creating a BPMN diagram. Here below are the BPMN elements that are already implemented:

- Containers: pool, and lane
- Activity: task
- Start event: empty, message, and conditional events
- Intermediate event: message event
- End event: empty, and message events
- Gateway: exclusive, inclusive, and parallel gateways
- Connector: sequence, and message flows

The design space is an empty space for drawing a BPMN process by drag and drop from the palette. The graphics displayed on this space is compliant with the BPMN 2.0 specifications. The property sheet is a set of attributes that correspond to the selected BPMN element.

The above functionalities concern the client side of the editor. The server side of the editor hosts the database that stores BPMN processes for reusability. The processes stored can then be used by the simulator (Section 3) or by the editor itself to modify it and save as a new process.

The editor can generate two kinds of artifacts:

- An XML file compliant with the BPMN 2.0 metamodel. This is the primary objective of the version 2.0 that allows enabling the exchange of business process

models and their diagram layout among process modeling tools to preserve semantic integrity.

– • An XPDL (XML Process Definition Language) file compliant to the XPDL 2.1 specification [19] exchange format. It allows representing process description and graphical information (e.g. coordinates, width, and height) of process's objects.

There is already a rich set of available tools allowing to design BPMN compliant business processes, however none of them combine the following benefits: to be based on open-standards, to be cloud ready and to be provided as a component of a complete open-source SOA stack. Moreover our tools provide a very advance collaborative feature that allows a group of business process designers to collaborate in real-time on processes definition.

Most of the market tools are provided as client / server applications. Some are proprietary (ARIS Express, Tibco Business studio, Intalio, BizAgi), some are open-source (jBPM, Bonitasoft). Among the cloud based solutions (Lombardi Blueprint, Appian Anywhere, Cordys Process Boardroom, IBM Blueworks, etc.) only Oryx Editor is provided under open-source license, however it is written in Javascript and doesn't provide any collaborative features.

2.4 Collaborative edition

The BPMN editor allows an unlimited number of users to collaborate in real-time on the definition of business processes. The central concept is the collaboration session where members of a group of work collaborate on the same process by alternatively bringing their own contributions on the designed process.

The collaboration session is instantiated by a member of a virtual organization; this member is therefore considered as the founder of the concerned collaboration. The founder can then invite other members (of the same group of work or not) to join the session, but external members can also ask to join the session as they can see all the ongoing sessions (of the same group of work to which they belongs) on the main dashboard.

When a new member arrives in an ongoing session he is notified of all the previous transactions that were made and can start to contribute to the actual process. The other members are also automatically notified of the arrival of a new participant as a transactional history system allows seeing all the participants of the collaboration and their respective contributions since the beginning of the session. This system also allows to roll-back on the contributions that are not accepted by the majority of the participants. The collaborative process can be exported to BPMN standard format or saved at any time to be completed later on.

The main difficulty in collaborative tools is to deal with distributed transactions. In a tool such as the BPMN editor multiple clients interact on the same process that is stored on the same server.

In a web environment the classical paradigm is the client pull where a client request is responded to by a server whenever there is a full/effective response or not. Using such an approach for a collaborative tool can lead to a loss of efficiency as all clients would have to constantly check for new updates even if there is none and therefore increasing the server calls.

Rather than using this classical approach the collaborative editor has adopted the Comet [14] approach along with an event-based communication where each client subscribe and publishes events notified through a server push mechanism. Thus, all clients are notified of any update only when there is an effective one.

The collaborative BPMN editor uses the GWTEventService[4] open-source library to achieve this goal as it perfectly integrates the Comet approach within the GWT framework.

3 Event-driven Simulator

The main aim of the simulator is to let users to testing processes behavior at design time from the event point of view. Processes may be very complex by involving a combination of sub-processes triggered by many kinds of events. Due to such events combinatory the global behavior of such a system is hard to address for the end user. Moreover some process representation languages, like BPMN, include many elements, some are simple to understand and others require a little more expertise before they can be mastered. Thus, being able to simulate a process allows the designers to compare their expectations with the actual outcome, eliminating by this way any doubt about the process behavior.

According to [8], simulation is mentioned as one of the techniques suitable for the support of redesign of business process. The simulation of business processes helps in understanding, analyzing, and designing processes. With the use of simulation the (re)designed processes can be evaluated and compared.

The simulation is mostly useful if a report summarizes, for a specific scenario, the process flow that is produced. The result of a process is determined by the semantic of all of the involved elements (activities, events, flows etc.). Our focus is to study how gateways and events may affect the flow of a process, while activities represent the work performed within a Business Process.

Simulating a maximum of scenarios can reduce the risks and contingencies that may arise in a real environment. It can also be interesting to see the effects of introducing new processes within an organization as the new process can affect the old ones. As mentioned in [12], simulation is largely used to assess the impact that changes to business processes may have on the organization and to explore different business process scenarios.

3.1 Simulation concepts – an event-based approach

According to [15], research on discrete event-simulation mostly dates back to the eighties of the last century and no major progress has been made lately. However, the current emergence of event-driven architectures for business applications continues these efforts.

[16] summarizes different approaches of simulation found in the literatures: the event-scheduling approach uses an ordered set of timed (so-called determined events),

[4] http://code.google.com/p/gwteventservice/

on the other hand, it includes condition checks for other (so-called contingent events). As discussed in [1], a simulated system can be described in terms of objects (entities), attributes defining these objects, events causing changes in object states, activities that transform an object's state over time and processes that are a sequence of activities or events ordered by time.

Our simulator works on the basis of an event-driven business process approach. It can be considered as discrete event simulation because the operation of the system is represented as a sequence of events and activities. BPMN is used as the business process representation language in our simulator. BPMN is a formalism that uses event as a first order element to describe process. A BPMN process is always started and ended with an event, and deals with intermediate events during the course of the process, the simulation takes into account all these events during the execution.

The basic principle is that a triggering event of the process will create a token and instantiate the process. These triggering events are most of the time starting events. A token is a "theoretical" object used to create a descriptive simulation of the behavior of BPMN elements (it is not currently a formal part of the BPMN specification). When an item receives a token it can then go through several states. Thus, some items may be put on hold while others continue to be executed. As long as an activity does not have all the needed resources it cannot be performed. Once the element is considered as complete, the token may "pass through" the next element by the connecting sequence flow.

The specific behavior of each BPMN elements is also taken into account because it influences the process flow. The gateways are useful to control how the process diverges or converges; they represent points of control for the paths within the process. Gateways can either split or merge the flow of a process through sequence flow. Events can start, delay, interrupt or end the flow of the process.

3.2 Functional description

The primary intention of the simulator is to provide users the possibility to verify their business processes. It is also aimed at providing the ability to interact with the BPMN process, so that users can play with the process elements and their behaviors. The users can make any changes on the process scenario during execution in order to observe the consequences of changes.

The Simulator integrates an engine which manages the token(s) within a process through all the various elements while "executing" their behavior and considering the scenario that the designer specified for each element. Once all the tokens have been consumed, the process can be considered as complete and a report on the simulation of the process is generated.

The Fig. 3 shows the graphical interactive web interface of the simulator:

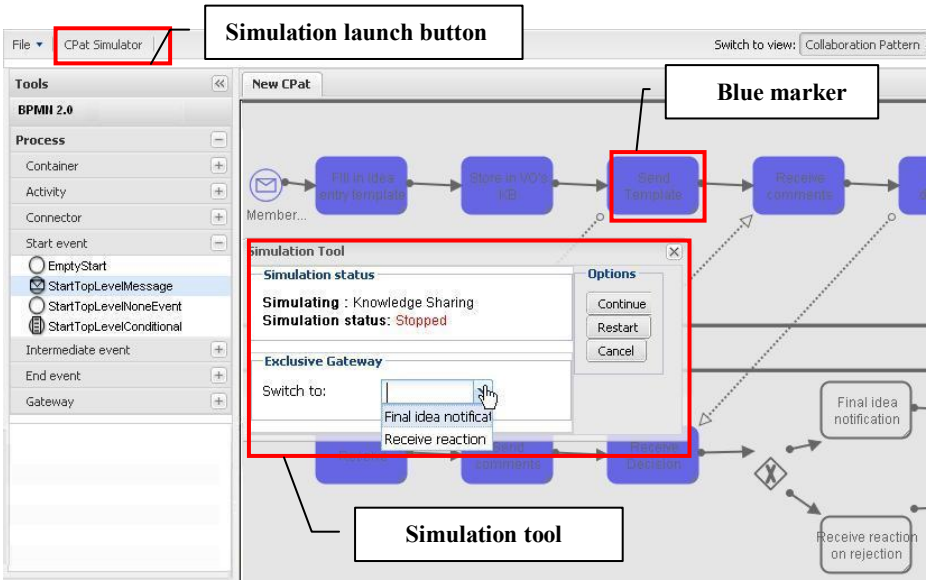

Fig. 3. Graphical interactive interface of the Simulator

The graphical simulation allows the user to see the progress of the simulation by the presence of markers (in the blue and yellow colors) for distinguishing the state of each elements of the process. The maker represents the token of the simulation engine. The blue marker stresses on the already executed elements, while the yellow one shows elements in waiting status for receiving a message.

The simulation tool is an interactive window shown during the simulation. It describes the current status of the process and asks the user for inputs when necessary, for example, when the simulator arrives at a gateway that requires selecting an appropriate route to continue, or an intermediate event where user must attach a document. The simulation tool can also be used to pause, restart or cancel the simulation.

Once a simulation is completed or if the user decides to cancel the simulation, a result window is automatically shown. This result window summarizes the choices that were made during the simulation (ex. gateways), messages exchanged between the activities, and the activities that have been executed. The Fig. 4 shows the result window:

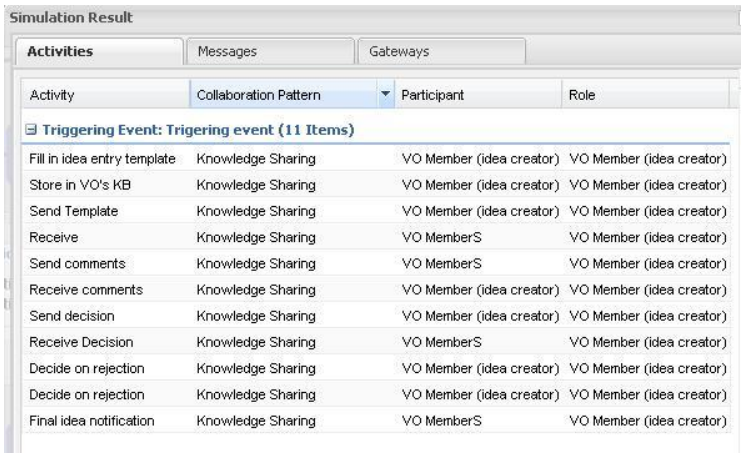

Fig. 4. Three-panels simulation result window

4 Conclusions and future works

The objective of this paper is to present the Business Process Editor and Simulator developed on the basis of an event-driven business process modelling approach. Our motivations to develop such tools originally come from the interests in providing cloud enabled event infrastructure that fully support the collaborative business process management from the design to the execution phases. The business process management is nowadays an important element of every organization, according to the SOA principles. It is required to support the real time changes of events in the current business environment. The integration of an event-driven approach into a business process modelling can empower the organization to more quickly respond to changing business requirements (business agility).

Future work deals mainly, for the design part, with the implementation of the new features provided by BPMN 2.0 to support choreographies and for the runtime part to address large scale distributed event driven architectures.

Integration of the event run-time infrastructure with a Complex Event Processing (CEP) engine is foreseen in order to handle complex events. This requires interfacing the Petals notification engine and the CEP in order to be able to detect event patterns.

The first open-source version of the Business Process Editor and the Simulator will be released under the GNU Affero General Public Licence. They support the BPMN 2.0 specification and provide as outputs a BPMN XML file compliant with BPMN 2.0 meta-model and a XPDL 2.1 file, as well as the necessary artefacts to deploy applications into a private or public cloud supported by the Petals ESB with notification and BPEL engines.

Acknowledgments. This work has been partially funded by the European Commission in the frame of the STREP research project SYNERGY (Supporting

highlY-adaptive Network Enterprise collaboration thRouGh semanticallY-enabled knowledge services), ICT No 63637. The authors would like to thank the project partners for their advices and comments regarding this work.

References

1. Balci O. The implementation of four conceptual frameworks for simulation modeling in high-level languages. Proceedings of the 20th conference on Winter simulation, 287-295, SanDiego, CA, (1988).
2. Barros, A., Decker, G. and Grosskopf, A. Complex Events in Business Processes. Lecture notes in computer science, (2007).
3. Carter S. Management in SOA, Information Management Magazine, May (2007).
4. Chakravarty, P. and Singh, M. P. Incorporating Events into Cross-Organizational Business Processes. IEEE INTERNET COMPUTING**,** 46-53, (2008)
5. Dwyer J. Pro Web 2.0 Application Development with GWT. USA: Apress, ISBN 978-1590599853 (2008)
6. Jaber S. Programmation GWT 2 : Développer des applications RIA et Ajax avec Google Web Toolkit . France: Eyrolles, ISBN 978-2-212-12569-6, (2010)
7. Jansen-Vullers M. and Netjes M. Business Process simulation – a tool survey, paper presented at 7th Workshop and Tutorial on the Practical Use of Coloured Petri Nets and the CPN Tools (CPN' 06), Denmark, (2006)
8. Kettinger W.J., Teng J.T.C., and Guha S. Business Process Change: A Study of Methodologies, Techniques, and Tools. MIS Quarterly, 21(1):55–80, (1997)
9. Lorré J.P., Verginadis Y., Papageorgiou N. and Salatge N. Ad-hoc execution of collaboration patterns using dynamic orchestration, Proceedings of the I-ESA conference, Coventry UK, (2010)
10. Mani Chandy K. Event-Driven Applications: Costs, Benefits and Design Approaches, California Institute of Technology, Gartner Application Integration and Web Services, (2006).
11. Michelson BM. Event-Driven Architecture Overview, Patricia Seybold Group, (2006).
12. Paul R.J. and Serrano A. Collaborative Information Systems and Business Process Design Using Simulation, Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 1, p.10009.1, January 05-08, (2004)
13. Rajsiri, V. Knowledge-based system for collaborative process definition, Ph.D Thesis, Institute Nationale Polytechnique de Toulouse, (2009)
14. Russell A. Comet: Low Latency Data for the Browser. Alex Russell's blog, (2006).
15. Scalable Vector Graphics (SVG). W3C (2010) http://www.w3.org/Graphics/SVG/
16. Schiefer J., Roth H., Suntinger M. and Schatten A. Simulating business process scenarios for event-based systems, Vortrag: 15th European Conference on Information Systems (ECIS 2007), in: Proceedings of the 15th European Conference on Information Systems (ECIS), University of St. Gallen, (2007)
17. SYNERGY project. Deliverable D4.1 Collaboration patterns model and ontology, (2008).
18. Vernadat F. Techniques de modelisation en entreprise : Applications aux processus Operationnels, economica, (1999).
19. White, S.,Miers, D. BPMN Modeling and Reference Guide. Florida: Future Strategies Inc, ISBN 978-0-9777527-2-0, (2008)
20. Workflow standard process definition interface – XML Process Definition Language, version 2.1a, Workflow Management Coalition, WFMC-TC-1025, October (2008).