SIMULATE ENTERPRISE CONCEPTUAL MODELS EXECUTION BY COUPLING ENTERPRISE SERVICES AND DEVS MODELS USING WORKFLOW

Hassan Bazoun[a,* **], **Judicael Ribault**[b, *], **Gregory Zacharewicz**[c, *], **Yves Ducq**[d, *], **Hadrien Boyé**[e, **]

(*) University of Bordeaux, Lab. IMS, UMR CNRS 5218, 33405 Talence, France.
(**) Hardis / Hardis Conseil, 3 rue Guglielmo Marconi, 44800 Saint Herblain, France

[a]hassan.bazoun@hardis.fr, [b]judicael.ribault@ims-bordeaux.fr, [c]gregory.zacharewicz@ims-bordeaux.fr, [d]yves.ducq@ims-bordeaux.fr, [e]hadrien.boye@hardis.fr

ABSTRACT

The service process modeling is emerging in the enterprise. Nevertheless, business modelling can be very complex, lying at the heart of many business decisions and demanding a lot of time and effort. A well-designed, well-built business model can lower risk and make enterprises more successful in their objectives. The SLMToolBox is an Eclipse RCP that proposes to transform conceptual models of service processes coming from business level to BPMN models (OMG 2011) and then from BPMN into DEVS models in order to simulate the behavior of the entire conceptual model. For a better integration and deployment of service models in the enterprise, we propose in this paper to test first - thanks to simulation - services freshly modeled (yet non-existing) coupled with existing enterprise services. This paper is a work in progress that recalls the MDSEA methodology and presents the key concept of the transformation of BPMN concepts into executable workflows within the SLMToolBox software, where unavailable enterprise services are simulated using DEVS. The interoperability between real and simulated services will be handled by the tool Taverna workflow and HLA RTI. This step is one step further in the MDSEA development loop.

1. INTRODUCTION

Enterprises develop Business Models to ensure they can produce services in the most effective and efficient manner possible. Business oriented people need a common and explicit graphical notation for business process modelling. They need to describe and communicate high level business processes involving enterprises resources with the help of a simple and explicit formalism. In (Bazoun et al 2013) authors proposed the Extended Actigram Star (EA*) language as a business process modelling language that facilitates the modeling of business process in enterprise offering a dynamic view of the process being modeled.

Then the authors have integrated a Model Driven approach: MDSEA to support the model transformation from conceptual level into more technical oriented models. The software developed is entitled SLMToolBox. This tool is an Eclipse RCP that proposes to transform EA* business level models to BPMN models (Bazoun et al 2013) and then from BPMN into DEVS models (Bazoun et al 2014). This sequence of transformation favors the simulation of the entire conceptual model's behavior. Although this solution allows testing and verifying of the conceptual model's behavior before its implementation and development, but it lacks continuous integration and interoperability with the enterprise services.

This position paper recalls the MDSEA methodology and the existing software bricks to be reused from the SLMToolBox which already proposes a set of transformation. Nevertheless this tool is missing one last step to embrace the full MDSEA methodology. This last step will consist in generating concrete web service calls from the SLMToolBox models thanks to a Workflow Engine. The missing interoperability and time management interoperability will completed by HLA Run Time Infrastructure (RTI) In that objective, we propose a connector between SLMToolBox-DEVS-simulator and the workflow engine Taverna.

2. BACKGROUND

This section presents briefly the EA* language and BPMN, then we describe the SLMToolBox and the Taverna Workflow engine. Finally we introduce DEVS and HLA-RTI.

2.1. Process Modeling Languages: EA*, BPMN

In (Bazoun et al 2013) the Extended Actigram Star (EA*) language was described as a business process modelling language addressed to business users responsible of the creation of the first model, business people responsible of the management, and to technical developers responsible of the development of business process modeling tools. As a graphical modeling language, EA* provides to business users and analysts a standard to visualize business processes in an enterprise, and thus with a comprehensible and easy way to handle these processes. EA*relies on a reduce set of graphical objects and focus on the "business" aspects of enterprise processes. By its simple and accessible syntax, EA* intends to reduce the gap between the ideation and the design of business process.

EA* models need to be enriched with IT elements so that models can be interpreted by developers and technical teams. The choice of modeling languages is BPMN (OMG 2012). It permits to represent the process with activity sequence flow, message, events and resources. It prepares the model to integrate the implementation architecture. Then these models are transformed to DEVS simulation models (Zeigler 2000) and validated thanks to simulation transformations between these models should be framed by a well specified methodology. For this purpose, section 2.2 detailed the model driven service engineering architecture (MDSEA) as a proposed a methodology.

2.2. Methodology MDSEA

The Model Driven Service Engineering Architecture (MDSEA) is inspired from MDA (OMG 2005)/MDI (Bourey et al, 2007). This methodology is proposed in the frame of the MSEE project (FP7 2012) that defines its first Grand Challenge as making SSME (Service Science, Management and Engineering) evolving towards Manufacturing Systems and Factories of the Future. MDSEA provides an integrated methodology dealing with modeling languages at various abstraction levels to support Service models and Service System design and implementation. The relationship between the MDSEA modeling levels (Business Specific Model, Technology independent Model, and Technological Specific Model) and the Service System lifecycle phases (user-requirements, design and implementation) is established. One of the important innovations in MDSEA is to define the integration between domain components (IT, Organization/Human and Physical Means) at the BSM level in order to ensure that these integration aspects will be spread out at other levels. In this sense, this is therefore considered as an adaptation and an extension of MDA/MDI approaches to the engineering context of product related services in virtual enterprise environment. On the basis of MDA/MDI, the proposed MDSEA defines a framework for service system modeling around three abstraction levels: BSM (Business Service Model), TIM (Technology Independent Model) and TSM (Technology Specific Model).

2.2.1. Business Service Model (BSM)

BSM specifies models at a global level, describing the service running inside a single enterprise or inside a set of enterprises as well as the links between these enterprises. The models at the BSM level must be independent from future technologies that will be used for the various resources and must reflect the business perspective of the service system. In this sense, it's useful not only as an aid to understand a problem, but also it plays an important role in bridging the gap between domain experts and development experts. The BSM level allows also defining the link between Products' production and Services' production.

2.2.2. Technology Independent Model (TIM)

TIM delivers models at a second level of abstraction independent from the technology used to implement the system. It provides detailed specifications of the structure and functionality of the service system without including technological details. More concretely, it focuses on the operational details while hiding specific details of particular technology in order to stay technologically independent. At TIM level, the detailed specification of a service system's components are elaborated with respect to IT, Organization/Human and Physical means involved within the production of the service. This is important to mention that in comparison to MDA or MDI or SOMA (Service Oriented Modeling and Architecture) (Bazoun et al. 2014), the objective of MDSEA is not only IT oriented and this requires enabling the representation of human and physical resources from the BSM level. At TIM level, these representations must add some information in comparison to BSM models.

2.2.3. Technology Specific Model (TSM)

TSM enhances the specifications of the TIM model with details that specify how the implementation of the system uses a particular type of technology (such as, for example IT applications, Machine technology or a specific person). At TSM level, the models must provide sufficient details to allow developing or buying suitable software applications, hardware components, recruiting human operators / managers or establishing internal training plans, buying and realizing machine devices. For instance for IT applications, a TSM model enhance a TIM model with technological details and implementation constructs that are available in a specific implementation platform including middleware, operating systems and programming languages (e.g. Java, C++, EJB, CORBA, XML, Web Services, etc.). Based on the technical specifications given at TSM level, the next step consists of implementing the designed service system in terms of IT components (Applications and Services), Physical Means (machine or device components or material handling), and human resources and organization.

2.2.4. Proposed Modeling Languages

Based on the described modeling levels, MDSEA proposes to associate relevant modeling languages at each level in order to represent confidently the existing system, future service product and future service system. For choosing modeling languages, the required abstraction level is important.

It is obvious to say that the first specification step of a service to be established between two partners is crucial. At the BSM level, the modeling language must be simple to use, powerful and understandable by business oriented users. Moreover, this (or these) language(s) must cover process and decision with coherent models. The choice is affected by the capacity of the language to propose a hierarchical decomposition (global view to detailed ones). Indeed, business decision-makers often have a global view of the running system and need languages allowing this global representation with few high level activities (process or decisions). This global view must be completed by more detailed activities models elaborated by enterprise sector responsible. These models are connected to top level models in a hierarchical and inclusive way. These are the principles of systemic and system theory which must be taken into account in the choice of the languages. But it is also obvious that the choice of modeling languages is subjective, depending on the experience of the languages' practitioners and on the dissemination of these languages within enterprises.

As for process modeling at business level, several languages exist. Extended Actigrams Star (EA*), extended from GRAI extended Actigram (Grangel et al 2008), that was itself de-rived from IDEF0 (NIST 1993), was chosen to model processes at BSM level due to its independence regarding IT consideration, its hierarchical decomposition and the fact it can model three supported resources: material, human and IT. It has been developed as an answer to previous issues encountered with GRAI extended actigram language regarding its interoperability. It intends to capture business process models at a high semantic level, independently from any technological or detailed specifications. Service Oriented Modeling and Architecture principles (Bell M, 2008) developed by IBM were also considered, but these languages are more IT oriented and thus were far away from our requirements. Moreover, GRAI Grid (Doumeingts et al 1998) was selected for modeling governance in a service system. GRAI Grid aims at proposing a cartography of company's decisions which controls business processes, as proposed for instance in the ISO 9000-2008 standard (Goult R, 2008). The interest of GRAI Grid is to represent all decisions and their coordination, from the strategic to the operational levels. This representation is very important for business users because the results of decision making are also at the origin of performance evolution and achievement.

At the TIM level, BPMN 2.0 (OMG 2012) was chosen in particular because this language offers a large set of detailed modeling construct, including IT aspects and benefits from the interoperability of many BPM IT platforms allowing the deployment and automated transformation to execution of BPMN processes. Moreover, BPMN enables also to represent human and technical resources which are required in the MDSEA principles of representation. BPMN has also the advantage to provide a meta-model developed by OMG which facilitates the implementation of the language. Other process modeling languages coexist, for instance GRAI nets are proposed in order to detail the decision processes in coherence with the decisions identified in the GRAI Grid but with adding technical and organization information as the decision rules, the decision makers, and the decision support modules.

2.3. SLMToolBox

SLMToolBox is a software tool developed by Hardis (Hardis). The SLMToolBox is an implementation of the BSM and TIM levels of MDSEA. It will be used by enterprises willing to develop a new service or improve an existing one, within a single enterprise or a virtual manufacturing enterprise (Thoben et al, 2001).

2.3.1. Modeling

MDSEA Meta models and Languages

MDSEA defines a set of constructs and relationships (described with "templates") which are specific to the domain of service system modeling, at 3 modeling levels: BSM/TIM/TSM. For each abstraction level, MDSEA suggest a set of references to standard or former graphical modeling languages (which are domain agnostic), in order to extend and complete the representation of the system to be modeled, under different perspectives (e.g.: decision structure; process; use cases; …). This type of modeling architecture is based on a "view model" pattern (or "viewpoints framework") (ISO 2011) as it defines a coherent set of views to be used, in the construction of a manufacturing service. The purpose of views and viewpoints is to enable humans to comprehend very complex systems, to organize the elements of the problem and the solution around domains of expertise and to separate concerns. In the engineering of physically intensive systems, viewpoints often correspond to capabilities and responsibilities within the engineering organization.

Both BSM (Business Service Models) and TIM (Technology Independent Models) are structured in the same manner. A "core" model gathers a set of generic (meta-) data in order to qualify the service to be modeled (specified / designed) ; this "core" model refers to external graphical modeling languages (e.g. : UML (OMG 2011) so that certain aspects of the service model can be elaborated in more details with the help of graphical languages.

This structure allows to map "view specific" modeling languages (e.g.: GraiGrid, UML Class Diagram) with "domain specific" constructs (i.e.: MDSEA BSM) without introducing modifications or restrictions to the MDSEA metamodel. From the user point of view, it allows the possibility to edit core information, independent from any specific modeling language, and to retrieve and reuse this data under different views, accomplished with the help of several graphical diagrams. With this approach, MDSEA Core Constructs remain agnostic from any representation formalism. Their implementation is realized by a core model, which acts as domain specific (Service System Modeling) "glue" between several modeling languages. Thus, we can reuse standard modeling languages without introducing modifications to their metamodels (e.g.: BPMN, UML…). Graphical languages such as "Extended Actigram Star" or "GRAIGrid" can continue to evolve, with (almost) no impact on MDSEA Core metamodels (i.e.: BSM and TIM).

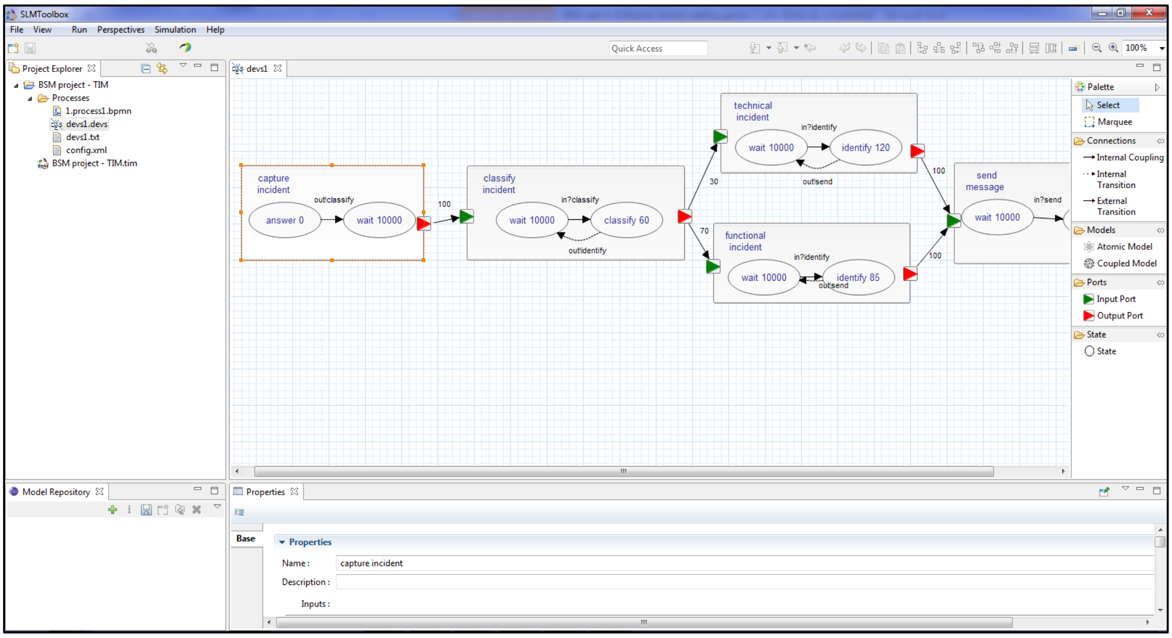

Figure 1 SLMToolBox

Modeling editors
The modeling environment will support service system modeling activities by providing editors for domain specific models (BSM, TIM) and related modeling languages to enhance the description of the BSM and TIM models. A set of language specific modeling editors is provided for each modeling language. These editors are either the result of a Hardis's specific development (BSM templates, EA* and TIM templates editors) or open source plugins integrated within the environment (UML and BPMN editors, DEVS editor).

Model transformation
SLMToolBox supports specific model transformations, mostly to support the continuity between the service concepts & requirements phase to the service design phase. In addition model transformation aims to save effort and reduce errors by automating the development of models when possible. Information and requirements collected at BSM level are reused at TIM level. SLMToolBox supports the transformation of BSM data models into TIM data models, and the transformation of EA* model into BPMN process and collaboration diagrams (Bazoun et al 2013). In addition, it supports the transformation from BPM models to DEVS (Bazoun et al 2014) for simulation purposes.

2.3.2. Integration with external tools

Up to now, the SLMToolBox is preparing the models to be ready for use at TIM level; the models can be simulated. Nevertheless these models miss technical details and so cannot be connected directly and immediately with other software or material components to support a concrete service implementation. The idea proposed in this paper is to reuse the experience acquired when connecting DEVS models with service calls. The interoperability can gain from the service architecture and distributed interoperability simulation architecture. The missing element in the workflow chain is the component that links the model with the service call. DEVS, HLA and Taverna will help for that.

2.4. Workflow Engine Orchestration with HLA

In (Ribault 2014), authors describe how to facilitate interoperability between real services and DEVS simulation using workflow of services and HLA/RTI.

2.4.1. Workflow of services

Workflow was first designed to improve the business process. A production workflow is a set of steps required for developing a product until it is put on the market (Weske, 2012). The workflow steps are based on observing a number of steps that are usually repeated manually and formalizing them. The Workflow Management Coalition (WMC) standard group (WMC 2009) proposes a WF reference model in which the WF is in the center and interacts with other surrounding applications or WF components.
Several surveys have compared different workflow management systems. In (Deelman et al., 2009), the authors analyzed and classified the functionality of workflow system based on the needs of scientists who use them. In (Yu and Buyya, 2006), the authors focused on the features to access distributed resources. In (Curcin and Ghanem, 2008), four of the most popular scientific systems were reviewed. In (Tan et al., 2009), the authors compare the service discovery, service composition, workflow execution, and workflow result analysis between BPEL and a workflow management system (Taverna) in the use of scientific workflows. Taverna was chosen to demonstrate the feasibility of the methodology because Taverna eases the interoperability with other services and the data flow modelling compare to other workflow management system we studied.

2.4.2. Taverna

Taverna (Hull et al. 2006) is an application that facilitates the use and integration of a number of tools and databases available on the web, in particular Web services. It allows users who are not necessarily programmers to design, execute, and share workflows. These workflows can integrate many different resources in a single experiment.

A Taverna workflow can contain several services including: Java code, Remote application via the REST protocol, SOAP/WSDL protocol, Workflow nested within another hierarchically and the use of local tools within a workflow.

In Taverna, a service can take input and produce output. The workflow input can be part of the workflow or can be given prior to the execution of the workflow. For example, the Taverna RESTful service takes in input various data, and it returns a status code and a response. A WSDL Taverna service will find automatically the number and type of input and output thanks to the WSDL file. Taverna offers the possibility to automatically format the input and output based on the type of parameters required by the Web service.

Workflows are particularly suited to automate experiments, but all necessary parameters cannot always be specified in advance. In these cases, it is desirable to interact with users for decision making. Taverna offers several graphical interfaces for interacting with the user. Taverna offers several user interfaces with this purpose:

- Ask: opens a box so the user can enter text.
- Choose, Select: lets the user select a value among a list of values.
- Select File: lets the user select a file in the system.
- Tell, Warn: gives a message to the user.

A Taverna workflow can contain nested workflows. Thus, it is possible to create a parent workflow that contains several workflows. Several workflows can be combined together to obtain more complex workflows that do not need the external inputs and are fully automated.

2.5. DEVS & G-DEVS

Discrete EVent Specification (DEVS) was introduced by (Zeigler et al., 2000). This Moore based language describes a dynamic system with a discrete event approach using some typical concepts. In particular, it represents a state lifetime. When a lifetime is elapsed an internal transition occurs that changes the state of the model. The model also takes into account the elapsed time while firing an external state transition triggered by an event received from outside the considered model.

The behavioral models are encapsulated in atomic models that are completed with input and output ports. Then, these models can be composed with others by connecting inputs and outputs. The composed models are called coupled models.

Generalized DEVS (G-DEVS) emerged with the drawback that most classical discrete event abstraction formalisms (e.g. DEVS) face: they approximate observed input–output signals as piecewise constant trajectories. G-DEVS defines abstractions of signals with piecewise polynomial trajectories (Giambiasi et al., 2000]. Thus, G-DEVS defines the coefficient-event as a list of values representing the polynomial coefficients that approximate the input–output trajectory. Therefore, an initial DEVS model is a zero order G-DEVS model (the input–output trajectories are piecewise constants). In fact G-DEVS was the pioneer DEVS extension proposing a multi value event.

On the simulation side, G-DEVS models employ an abstract simulator (Zeigler et al., 2000) that defines the simulation semantics of the formalism. The architecture of the simulator is derived from the hierarchical model structure. Processors involved in a hierarchical simulation are: Simulators, which implement the simulation of atomic models; Coordinators, which implement the routing of messages between coupled models; and the Root Coordinator, which implement global simulation management. The simulation runs by sending different kind of messages between components. The specificity of G-DEVS model simulation is that the definition of an event is a list of coefficient values instead of a unique value in DEVS.

Zacharewicz et al. proposed in (Zacharewicz et al., 2008), an environment, named DEVS Model Editor (LSIS_DME), to create G-DEVS models that are HLA compliant and simulating them in a distributed fashion. In LSIS_DME, a G-DEVS model structure can be split into federate component models in order to build a HLA federation (i.e. a distributed G-DEVS coupled model). The environment maps DEVS Local Coordinator and Simulators into HLA federates and it maps Root Coordinator into RTI. Thus, the “global distributed” model (i.e. the federation) is composed of federates intercommunicating.

2.6. HLA

The High Level Architecture (HLA) (IEEE, 2000) (IEEE, 2003) 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 Defense (DOD). The original goal was the reuse and interoperability of military applications, simulations and sensors.

In HLA, every participating application is called federate. A federate interacts with other federates within a federation (i.e. 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 (IEEE, 2000) and finally to 1516 Evolved (IEEE, 2010).

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 (IEEE, 2000) and (IEEE, 2010). 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 (Fujimoto, 2000). In HLA 1516 Evolved (IEEE, 2010) the service approach is demanded as core feature. Nevertheless no software addresses completely that goal at the moment (Tu et al., 2013).

2.7. Time Management

In (Ribault 2014), authors addressed the problem of time synchronization management between real world and the simulated part.

The time is not progressing with the same dimension in the simulated part and was not taken into account in the workflow of service approach. The HLA can be an issue; authors in (Zacharewicz et al., 2008) have proposed to handle time related message exchange between the workflow components and others thanks to RTI. A specific service calls have been specified and adapted for HLA in (Ribault, 2014) making service able to be the bridge between in the real world and the simulated components.

3. CONTRIBUTION

To improve the integration and the validation of the conceptual model, we propose to compose a test that reuses existing enterprise services while simulating non-existing or unavailable enterprise services. This approach should support the progressive involvement of new components to be added to the existing system by adopting the System of Systems (SoS) paradigm. The test, thanks to simulation, is confronting the future components to their future environment. This proposition should anticipate problems that can be faced at final implementation. In particular the causality relations of events and calls to services that are planned to be chained are here tested within the time constraints of the real future system.

More concretely, the first problem is the matching between the concepts announced in the enterprise BPMN models of services with technical services. This issue can be addressed by transforming BPMN concepts into executable workflow of services as described in the figure 2 with the dashed link going from BPMN 2.0 Diagram to Workflow Engine Orchestration. The second problem is to deal with non-existing or unavailable services in the enterprise. This issue can be resolved using DEVS models to mimics the behavior of enterprise services completed with HLA envelop to make them interoperable with other distributed and heterogeneous components. This is represented on the figure 2 with dashed link from DEVS Model to RTI.

Currently, the SLMToolBox exclusively proposes to transform all BPMN components to DEVS models. The idea proposed in this paper is to adapt the tool in order to propose users to select, on one side, the part of the BPMN model that will be transformed into workflow of service in order to call the existing enterprise web services. For this part of the model the tool will prepare the service calls by configuring the service query and locating the server to be called. On the other side, the other part of the model will be automatically transformed into DEVS models and simulate the behavior of the part of the system including the time to respond and the state that memorizes the process status. Some previous work has already put the first stones in this domain. This step will go further in the MDSEA lifecycle by starting to generate real calls to services and external systems.

3.1. From BPMN to executable workflow

The first problem is the matching between the concepts announced in the enterprise models of services with technical services. The SLMToolBox is preparing the model at TIM but the model is not yet extended with primitives to connect with existing systems and in particular servers that provide services. This step is supposed to be assumed only at TSM. Nevertheless it is interesting to test by simulation the system in its future real environment.

This new step in MDSEA approach will start from the BPMN model produced by the SLMToolbox, it will gain in interoperability by coding, from the BPMN communication actions, the primitives for service calls. This will be facilitated by the model structure already saved with the XML format in the Eclipse standard. Concretely a resulting executable WF will be generated from the BPMN model. This model could be played with Taverna to sequence the service calls and answers. It will facilitate the interoperability with existing services and will close the loop of service verification and validation.

After the edition of the BPMN model, the user will be able to annotate the model and select the part that will be transformed to simulation models or Workflow of services. Concerning the transformation of BPMN parts to executable workflow, the Meta model approach will be preferred. For instance, Taverna saves the workflows in XML, so we could transform BPMN message and data flow part that link lanes or pools into Taverna workflow abstract service calls. The user will fulfill the query detail for the service call.

On the other side, unavailable services will be transformed to DEVS models to be simulated. This transformation is highlighted in the next section (3.2).

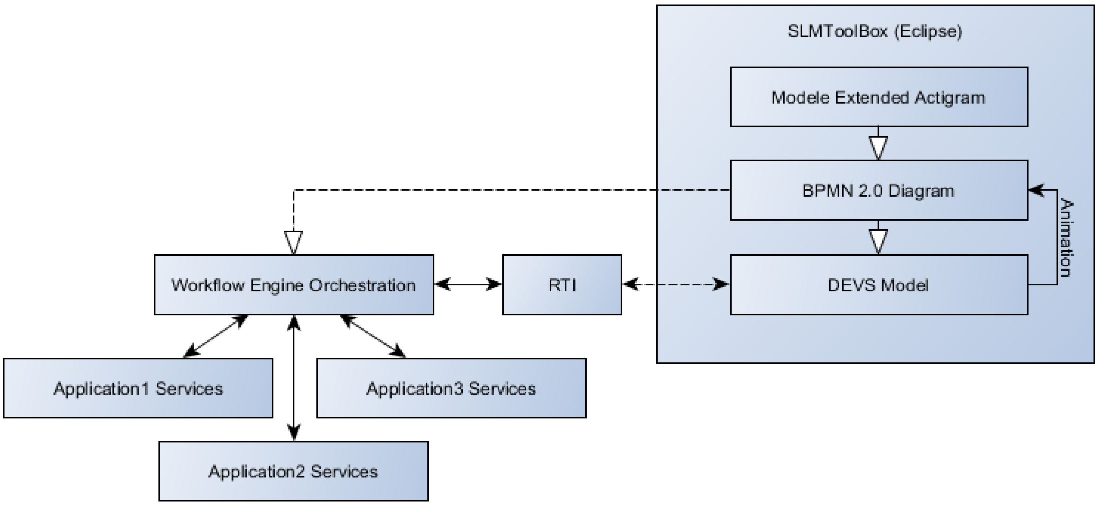

Figure 2 From conceptual model to enterprise execution.

3.2. From BPMN to DEVS Simulated Enterprise Services

The second problem is to deal with, at the same time, on one side existing and/or legacy system and on the other side non-existing or unavailable services in the enterprise. The idea is to mix simulated parts with real environment of web services.

In that objective, the authors of (Ribault 2014 proposed an architecture that is dedicated to compose DEVS models with workflow of service tools. In detail, the workflow is orchestrated with a tool able to call and orchestrate the answers of web services and for services that are not defined already and when the behavior of external actors is required, the system is calling a DEVS component that is timed with a local behavior.

SLMToolBox generates DEVS Models out of BPMN models. This mechanism is based on an implemented transformation from BPMN concepts to DEVS concepts. In (Bazoun 2014) authors explained this transformation and how certain BPMN concepts are mapped to DEVS concept. After the transformation to DEVS Models the process can be simulated depending on cost and time. A link should be established to use the obtained simulation results by Taverna. A possible solution would be to develop web services that can return this simulation results so that it can be used by Taverna.

DEVS models resulting should be extended to be able to communicate with the Taverna Workflow. In that objective the works proposed in (Ribault 2014) and (Zacharewicz 2008) can be adapted and reused. They proposed to embed DEVS models into HLA “federates”. These “federates” gain interoperability properties to communicate within a distributed simulation, thanks to HLA. The “federate” can publish messages and subscribe to information within a time synchronized environment. Nevertheless, these models should, even synchronized, still need to be enriched to have the primitives for “real world” services calls and communication including how to form the query and how to reach the URL of the service.

4. CONCLUSION

This paper has proposed the principle of a mechanism to generate from the conceptual model a continuous integration facility and a testing and validation platform within the existing enterprise services system. This contribution faces two major problems: (1) conceptual models are just blueprint and they contain components with behaviors that need to be simulated in order to test and verify their correctness in the global behavior in the enterprise system, and (2) simulated conceptual models must be able to interact with existing enterprise services. The first problem has been previously tackled up to TIM level thanks to model transformation proposed in the SLMToolBox architecture. The second problem has also been addressed in composing workflow and DEVS. Nevertheless no works were proposed to compose these two questions.

In this paper we proposed a solution to go one step further in the direction of a - as much as possible - fully automated MDSEA approach. We described the basement for this new approach. In particular we have proposed a method for decomposing models, originally based on a BPMN description, into on one side simulation and, on the other, real world interaction. These two sides are then coupled again in distributed testing system that composes simulation models with concrete service calls. This work will be extended in order to generate the technical workflow of services calls.

REFERENCES

Bazoun, H., Zacharewicz, G., Ducq, Y., Boye, H., 2013. Transformation of Extended Actigram Star to BPMN 2.0 and Simulation Model in the Frame of MDSEA. *Proceedings of the Symposium on Theory of Modeling & Simulation 2013*. April 7-10, (San Diego, USA).

Bazoun, H, Zacharewicz, G, Ducq, Y, Boye, H., 2014. Business Process Simulation: Transformation of BPMN 2.0 to DEVS Models. *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium*, April 13-16, Tampa. (Tampa, USA)

Bell, M., 2008. *Service-Oriented Modeling: Service Analysis, Design, and Architecture*. Wiley publishing.

Bourey, JP, Grangel, R, Doumeingts, G, Berre, A. 2007. Deliverable DTG2.3 from the INTEROP project. *Report on Model Driven Interoperability*. Available from: http://interop-vlab.eu [accessed 15 May 2014]

Curcin, V. and Ghanem, M., 2008. Scientific workflow systems can one size fit all? *Biomedical Engineering Conference*, pp. 1-9. December 18-20, Cairo (Cairo, Egypt).

Deelman, E., Gannon, D., Shields, M., and Taylor, I., 2009. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25 (5) 528–540.

Doumeingts, G., Vallespir, B., Chen, D., 1998. GRAI grid, decisional modeling. *In International Handbook on Architecture of Information System*. Springer Verlag, 313-337.

IEEE std 1516TM-2010, 2010. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) -- Framework and Rules*, New York: Institute of Electrical and Electronic Engineers

IEEE std 1516.2-2000, 2000. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification*, New York: Institute of Electrical and Electronic Engineers.

FP7 – FoF-ICT-2011.7.3. 2012. *Manufacturing SErvice Ecosystem Project- Annex 1 description of work*; July 29, Available from: http://msee-ip.eu/ [accessed May 15 2014]

Goult, R. *Introduction to ISO 9001.2008 Handbook*. Available from : http://fr.scribd.com/doc/171647046/ISO-9001-2008-Handbook [accessed 18 October 2013]

Grangel Seguer, R., Cutting-Decelle, A.F., and Bourey, JP., A UML profile for transforming GRAI Extended Actigrams into UML. *In Proceedings of the 17th IFAC World Congress, Session: Architectures and Software Tools for Enterprise Integration and Networking in Manufacturing*, Paper ThB24.3, Séoul, Korea; 2008

Giambiasi, N., Escude B., Ghosh, S., 2000. GDEVS A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems. *SCS Transaction of the Computer Society*, 17 (3), 120-134.

Hardis Company Website Available from: http://www.hardis.fr/eng/jsp/site/Portal.jsp [accessed 18 October 2013]

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., & Oinn, T. 2006. Taverna: a tool for building and running workflows of services. *Nucleic Acids Research*, 34 (suppl. 2), W729-W732.

ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description.

National Institute of Standards and Technology (NIST). *IDEF, Integrated DEFinition Methods.* Available from http://www.idef.com/IDEF0.htm [accessed 18 May 2014].

OMG 2011. *OMG Unified Modeling Language (OMG UML), Infrastructure Version 2.4.1*. Available from: http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/. (accessed 18 October 2013)

OMG. Business Process Model and Notation (BPMN) Version 2.0. http://www.omg.org/spec/BPMN/2.0/PDF/ (accessed 18 October 2013).

OMG. MDA Guide Version 1.0. 2005 http://www.omg.org/cgi-bin/doc?omg/03-06-01 (accessed 18 October 2013).

Ribault, J., & Zacharewicz, G. (2014) Time-Based Orchestration of Workflow, Interoperability with G-DEVS/HLA, *Journal of Computational Science*, available online august 2014, DOI: 10.1016/j.jocs.2014.07.002

Missier Tan, W.,, P., Madduri, R., & Foster, I., 2009. Building scientific workflow with taverna and BPEL: A comparative study in cagrid. *In Service-Oriented Computing–ICSOC 2008 Workshops* (pp. 118-129). Springer Berlin/Heidelberg.

Tu Z., Zacharewicz G., Chen D., 2014. Building an HLA Federated Interoperable Framework from Legacy Information Systems, *In International Journal of Computer Integrated Manufacturing*, 27 (4), 313-332.

Weske, M. 2012. *Business Process Management Architectures*. Business Process Management, pp 333-371.

Yu, J., and Buyya, R., 2006. A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing*, 3 (3-4), 171–200

Zacharewicz, G., Frydman, C., Giambiasi, N., 2008 G-DEVS/HLA Environment for Distributed Simulations of Workflows, *Simulation*, 84 (5), 197–213

Zeigler, B. P., Praehofer, H., & Kim, T. G., 2000. *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems.* Academic Press.

ACKNOWLEDGEMENT

This work has been partially supported by the FP7 Research Project MSEE "Manufacturing Service Ecosystem" ID 284860. http://www.msee-ip.eu/

BIOGRAPHY

Hassan BAZOUN is a Software Engineer at Hardis and PhD Student at IMS Lab. His research domain includes business process modeling, model transformation, and Discrete Event Simulation. He is involved in the FP7 European project (MSEE) and has participated in the development of the modeling tool SLMToolBox.

Judicael RIBAULT is Postdoctoral fellow at the University of Bordeaux (IUT MP) Lab. IMS. He received his Ph.D. at INRIA Sophia Antipolis, France. His research interests include Component-based software engineering and simulation, Discrete Event Modelling (e.g. DEVS, G-DEVS), Distributed Simulation, Large-Scale Simulation, Simulation Interoperability and Workflow. He has published several papers in Conferences and is frequent Reviewer in Conferences (SpringSim, TMS/DEVS, etc.). He is involved in European projects.

Gregory ZACHAREWICZ is Associate Professor at University of Bordeaux (IUT MP) with both competences in enterprise engineering and computer sciences. His research interests include Discrete Event modelling (e.g. DEVS), Distributed Simulation, HLA, and Workflow. He has been involved in FP7 European projects (e.g. MSEE and FITMAN). He is recently focused on Enterprise Modelling and Semantic Interoperability. He has published more than 50 papers in international journals and conferences.

Yves DUCQ is Full Professor at University of Bordeaux from September 2008. He is working on Performance Measurement, Enterprise Modelling and Production Management since fifteen years and has published more than 20 papers in books and international journals and more than 70 papers in international conferences. He is responsible of the research group dedicated to production engineering sciences (GRAI research Group) at IMS laboratory. He has been involved in many European and French projects for twenty years and has also act as research engineer on several contracts with industry on performance improvement. He is also quality manager for University of Bordeaux.

Hadrien BOYE is a Project Manager at HARDIS. He is in charge of the e-solution skills center in the western offices of the company. His main fields of interest include web technologies, software architecture and enterprise information systems.