



# SLMTOOLBOX: Enterprise Service Process Modeling And Simulation By Coupling DEVS And Services Workflow

Hassan Bazoun, Judicael Ribault, Grégory Zacharewicz, Yves Ducq, Hadrien  
Boyé

## ► To cite this version:

Hassan Bazoun, Judicael Ribault, Grégory Zacharewicz, Yves Ducq, Hadrien Boyé. SLMTOOLBOX: Enterprise Service Process Modeling And Simulation By Coupling DEVS And Services Workflow. International Journal of Simulation and Process Modelling, 2016, 11 (6), pp.453-467. 10.1504/IJSPM.2016.082907 . hal-01313707

**HAL Id: hal-01313707**

**<https://hal.science/hal-01313707>**

Submitted on 10 May 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## **SLMTOOLBOX: ENTERPRISE SERVICE PROCESS MODELING AND SIMULATION BY COUPLING DEVS AND SERVICES WORKFLOW**

**Hassan Bazoun<sup>(a,\* \*\*)</sup>, Judicael Ribault<sup>(b,\* )</sup>, Gregory Zacharewicz<sup>(c,\* )</sup>, Yves Ducq<sup>(d,\* )</sup>, Hadrien  
Boyé<sup>(e,\* \*)</sup>**

(\*) University of Bordeaux, Lab. IMS, UMR CNRS 5218, 33405 Talence, France.

(\*\*) Hardis / Hardis Conseil, 3 rue Guglielmo Marconi, 44800 Saint Herblain, France

<sup>(a)</sup>[hassan.bazoun@hardis.fr](mailto:hassan.bazoun@hardis.fr), <sup>(b)</sup>[judicael.ribault@ims-bordeaux.fr](mailto:judicael.ribault@ims-bordeaux.fr), <sup>(c)</sup>[gregory.zacharewicz@ims-bordeaux.fr](mailto:gregory.zacharewicz@ims-bordeaux.fr), <sup>(d)</sup>[yves.ducq@ims-bordeaux.fr](mailto:yves.ducq@ims-bordeaux.fr), <sup>(e)</sup>[hadrien.boyé@hardis.fr](mailto:hadrien.boyé@hardis.fr)

## ABSTRACT

Market competition is pushing companies to differentiate themselves from competitors by developing customized services in addition to their original production (either physical or digital). It drives the emergence of service process modelling to describe more precisely the composition of services. Nevertheless, business initiatives modelling can be very complex to set, laying at the heart of many business decisions and demanding a lot of time and effort to handle and operate unambiguously. A well-designed and well-built business model can lower the risk of operating a service process and in consequence making enterprises more successful in their objectives. To this end, the paper recalls the MDSEA methodology and presents the key concept of the transformation of EA\* and BPMN concepts into simulation Workflows. Then it introduces the implementation done with the SLMToolBox that is an Eclipse RCP service graphical modeler, model transformer, and simulation engine. In more details, it runs transformation from service processes models designed by business users to BPMN models. Then the BPMN models can be transformed to DEVS models to simulate the behavior of the entire process model. In addition, enterprises are facing situations where future (undeveloped yet) enterprise services need to be integrated with existing ones. To go further and for a better integration and deployment of service models in the enterprise, we propose to combine service process M&S with service calls execution Workflow. To achieve that goal, we are mashing up simulation of services modelled with existing enterprise web services calls. The interoperability between real and simulated services is handled by the tool Taverna Workflow and HLA RTI. This step is pushing one step further the expertise in the MDSEA methodology, attempting to pave the way from service design to IT development.

## 1. INTRODUCTION

---

In a standardized production world, a recurrent challenge for enterprises is to keep distinguishing their offer from other players. One way consists in elaborating services to complete and customize better their offer regarding customers' needs. Prior to place on the market new services, enterprises could judiciously elaborate business models to confirm, at least theoretically, that they can produce services effectively and efficiently. Nevertheless, to elaborate business process models, business oriented people, that are not IT specialist, are looking for an explicit graphical notation to easily describe and share their models. They need to describe and communicate a high level description of business processes activities 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 inside the enterprise and in the context of Virtual Manufacturing Enterprises (VME) offering a sequential view of the processes being modeled. They also introduced the Model Driven Service Engineering Approach (MDSEA) to support the service model transformation from concepts to service development.

This paper is using EA\* as the starting point and higher abstraction modeling language. The first contribution of the authors consists in following MDSEA to support the model transformation from conceptual level into more technical oriented models. In detail, the EA\* models are transformed into BPMN models (Bazoun et al., 2013). This level is integrating some of the information technology constraints. These models give a view in terms of actions sequence and resources involvement regarding the service process. Nevertheless the time and the synchronization are not detailed in this view. To perform a run of the process in order to observe variable evolution regarding time, the approach proposes a transformation from BPMN into DEVS models (Bazoun et al., 2014). The DEVS formalism has been elected for its capacity to describe process behavioral models with discrete events, explicit states and time life variables. At the end, to validate the approach and to provide users a tool, the

software entitled SLMToolBox that is an Eclipse RCP is proposed. This sequence of transformation permits the simulation of the entire conceptual model's behavior. In consequence it permits testing, tracking KPI and verifying of conceptual models behavior before their implementation and development.

The SLMToolBox follows and implements the MDSEA methodology, it supports models design and their reuse at different abstraction steps thanks to a set of linked transformation from concepts to implementation. Nevertheless, in the enterprise real life, the upcoming service process will not be stand alone, it will have to collaborate and communicate with other existing process, e.g. at least to be supplied with upstream data, to send downstream data and communicate during its execution. So at simulation or execution time, the model lacks integration and interoperability with the existing and environing enterprise services; as a consequence, it fails to represent a fully realistic situation. As a final contribution, this paper describes a proposition for mixing real services and simulation. It will consist in generating concrete web service calls from the SLMToolBox models thanks to a Workflow Engine (Ribault et al., 2014). The missing data and time management interoperability will be completed by using the HLA Run Time Infrastructure (RTI) (Zacharewicz et al., 2008). In this objective, we propose an HLA connection between the SLMToolBox-DEVS-simulator and the Workflow engine Taverna.

## **2. BACKGROUND**

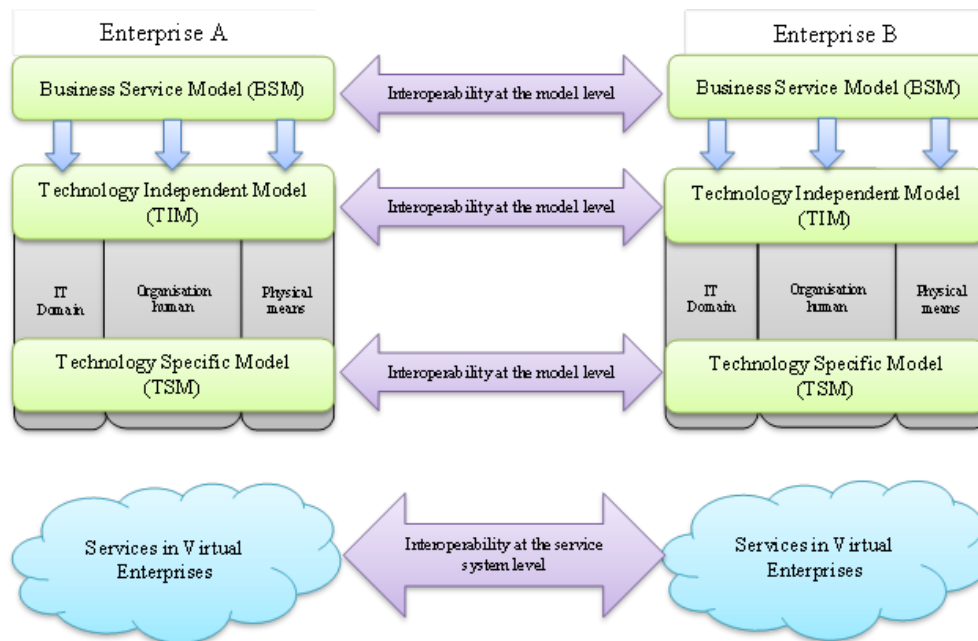
---

This section describes first the MDSEA methodology to situate the levels of description and to drive model transformation. Then it briefly recalls the EA\*, BPMN and DEVS languages. The notion of Workflow of services and the Taverna engine is added in order to couple Service Simulation and connecting with real service calls. Finally, it introduces HLA as a distributed simulation relevant standard.

### **2.1 Model Driven Service Engineering Architecture (MDSEA) Methodology**

The Model Driven Service Engineering Architecture (MDSEA) is inspired from MDA (OMG 2005) and 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 to 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). Then the final IT solution is developed from the source of the previous levels. This framework is illustrated in the Figure 1 in the context of a simple Virtual Manufacturing Enterprise (VME) built of two enterprises (here A & B).



**Figure 1 MDSEA Framework**

### 2.1.1 Business Service Model (BSM)

BSM level is intended to specify models at a global level, describing the service running inside a single enterprise or inside a set of enterprises (VME) as well as the links between these enterprises. The models at the BSM level must be independent from any 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 and difference between the production of products and services.

### 2.1.2 Technology Independent Model (TIM)

TIM delivers models with a lower level of abstraction regarding BSM but these models are still independent from any technology to be 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, 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 and/or detail some information in comparison to BSM models.

### 2.1.3 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 in 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 Service Process Modeling Languages: EA\*, BPMN & DEVS**

The Extended Actigram Star (EA\*) language has been introduced in (Bazoun et al., 2013) as a high abstraction level business process modelling language. It is intended to business users in charge of creating the initial model and business people responsible of the management. As well, technical developers can use it for preparing 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 reduced and accessible syntax, EA\* intends to reduce the gap between the ideation and the design of business process at BSM level.

In addition, EA\* models were required to be enriched with IT elements so that models can be interpreted by developers and technical teams at TIM level. In that objective, the choice of modeling languages felled on the Business Process modeling and Notation (BPMN) (OMG, 2011b). It permits to represent the process with activity, sequence and message flows, events and resources. It prepares the model to integrate the implementation architecture. In particular, the pools and lanes representation in BPMN permit to rationalize the responsibility of different resources (Human, IT or Physical).

Nevertheless, BPMN does not tackle in the model the dynamic aspect of the service and it is not associated to any simulation language. The Discrete Event Specification (DEVS), introduced by (Zeigler et al., 2000), has been used, at TIM level, to add the behavioral vision to the BPMN model. This Moore based language describes dynamic systems 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 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. In conclusion DEVS fits accurately the need to add the temporal dimension in the process model.

## **2.3 Service Workflow Orchestration**

Mixing at execution time simulated service process models and calls to real web services is another challenge of this research. So, the service simulation tool will have to call a Workflow that orchestrates web services during run time.

### **2.3.1 Workflow of services**

Workflow was first introduced, previous to recent process modeling language such as BPMN, to formalize, secure, and orchestrate the business process. According to (Weske, 2012), a

production workflow is a set of steps required for developing a product until being put on the market. The workflow steps are based on observing a number of steps that are usually repeated manually and formalizing them. The Workflow Management Coalition (WfMC) standardization group (WfMC, 2005) has proposed a WF reference model in which they describe a Workflow Management System (WFMS) with a central Workflow engine that orchestrates and interacts with other surrounding applications or Wf components.

Several surveys have compared different WFMS for orchestrating service calls. 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 frequently selected to demonstrate the feasibility of a methodology because it eases the orchestration and interoperability with other services and the data flow modelling compare to other Workflow management system we studied.

### **2.3.2 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. It proposes several actions for the user including: Ask, Choose, Select, Select File, Tell and Warn. A Taverna workflow can contain embedded 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.

## **2.4 Simulation**

On the simulation side, 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. Nevertheless, DEVS simulations that will be involved in a distributed and heterogeneous simulation needs in consequence to be extended.

### **2.4.1 Distributed Simulation based on the High Level Architecture (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 one core feature. Nevertheless no software addresses completely that goal at the moment (Tu et al., 2013).

Zacharewicz et al. proposed in (Zacharewicz et al., 2008), a Java based environment, named LSIS DEVS Model Editor (LSIS\_DME), to create a HLA 1516 compliant DEVS or G-DEVS models (Giambiasi et al., 2000) and simulating them in a distributed environment. In LSIS\_DME, a DEVS model structure can be split into federate component models in order to build a HLA federation (i.e. a distributed 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 DEVS coupled model federation) is composed of DEVS (atomic or coupled) federates intercommunicating.

#### **2.4.2 Time Management for Mixing Web Service Calls and Simulation**

More recently, (Ribault et al., 2014), authors have pushed further the DEVS/HLA framework by describing how to facilitate interoperability between web services and DEVS simulation using Taverna as a Workflow of services orchestration and a HLA/RTI. In this approach, authors proposed a solution to address the problem of time synchronization management between Workflow of real Web Service calls and simulated service process models. Indeed, the time is not progressing in the same dimension in the simulated part and was generally not taken into account in the Workflow of service approach. The HLA was an issue, reusing results from (Zacharewicz et al., 2008), extended to handle time related message exchange between the Workflow components and DEVS models thanks to RTI. At the end, a specific service call have been specified and adapted to merge with HLA publish/subscribe mechanism making simulated components bridged with “real world” service calls.

### **3. CONTRIBUTION**

---

The contribution can be divided into three parts. The first two ones consist in presenting the MDSEA models matching and the implementation in the SLMToolBox. Then we present the extension of the SLMToolBox to support the mixing between simulation and web services



calls.

### **3.1 Matching MDSEA Levels with 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 derived 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, 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.

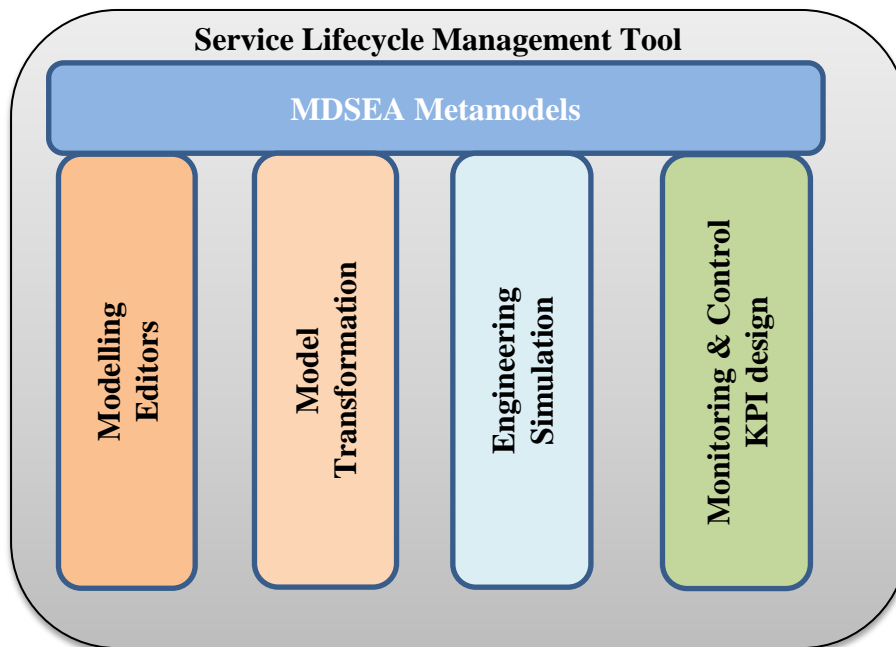
At the TIM level, BPMN 2.0 (OMG 2012) was chosen in particular because of the large set of detailed modeling construct this language offers, including IT aspects and benefits from the interoperability of many BPM IT platforms which allow deployment, automated transformation, and execution of BPMN processes. 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 maintained by OMG which facilitates unambiguous implementation of the language. Other process modeling languages coexist. In addition a need was identified to simulate the TIM models to observe the behavior, track KPI and verify the matching with objectives before the implementation. For that purpose, DEVS has been chosen for its capacity to represent the behavior of the EA\* and BPMN activities. Also it permits to represent explicitly the time life of the process activities and resources allocations.

## 3.2 SLMToolBox

The SLMToolBox (Service Life Management Tool Box) is a software tool developed in the frame of the EU FP7 MSEE project. It is an implementation of the BSM and TIM levels of MDSEA (Bazoun et al., 2014). It is intended to 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).

### 3.2.1 Logical architecture

As introduced previously, the foundation of the SLMToolBox is based on the MDSEA modelling architecture. This model centric approach provides the appropriate structure for elaborating service requirement and design thanks to a set of specific metamodels – dedicated to the domain of manufacturing services. The Figure 2 is representing the four pillars feature of the tool. It is commented just after.



**Figure 2 Service Life Cycle Management Tool**

The first pillar of the architecture brings a set of modelling editors, enabling the user to elaborate structured and graphical descriptions of the service and its aspects (IT, Human, and Physical Means) – at the business level (BSM : Business Service Models) and the design level (TIM: Technology Independent Models). As a complement, model transformation facilities will leverage interoperability of the models and enforce consistence between the Business requirements of the service and its design at TIM level.

The second pillar aims at sustaining the modelling activities thanks to a methodological support. Guidance will be provided to the user through the modelling activities of the service via an appropriate service engineering methodology. Besides, some support will be provided to assess the overall quality of the service at high level – at design time, thanks to appropriate tools.

The third pillar is responsible for the simulation of business processes providing animation and simulation reports.

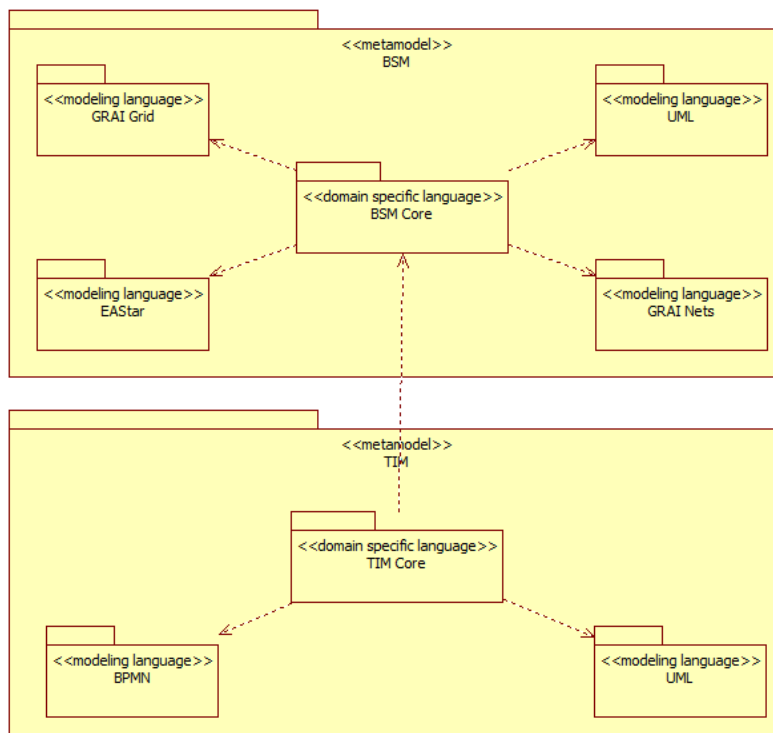
The fourth pillar will support the definition of the service system's governance, which will be then implemented by the organization to continuously assess the performance of the service

according to the three decision levels of the organization (Strategic, Tactical, and Operational), its functions and its detailed objectives.

### 3.2.2 Modelling architecture overview

MDSEA defines a set of constructs and relationships (described with “templates”) which are specific to the domain of service system modelling at three modelling levels: BSM, TIM, and TSM. For each abstraction level, MDSEA suggest a set of graphical modelling languages (which are domain agnostic) in order to extend and complete the representation of the system to be modelled under different perspectives (e.g. decision structure, process, use cases...).

This type of modelling architecture is based on a “view model” pattern (or “viewpoints framework”) (ISO/IEC/IEEE 42010 2011), Systems and software engineering — Architecture description) 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 better very complex systems, to organize the elements of the problem and the solution around domains of expertise by separating and decomposing the concerns. In the engineering of physically intensive systems, viewpoints often correspond to capabilities and responsibilities within the engineering organization. Both BSM and TIM are structured in the same manner. A “core” model gathers a set of generic (meta-) data in order to qualify the service to be modelled (specified / designed). This “core” model refers to external graphical modelling languages (e.g. UML (OMG 2011a) (section 3.2.3.3)) so that certain aspects of the service model can be elaborated in more details. This structure allows mapping “view specific” modelling languages (e.g. GRAI Grid (section 3.2.3.1), EA\* (section 3.2.3.2)) with “domain specific” constructs (e.g. at BSM level) without introducing modifications or restrictions to the MDSEA metamodel (Figure 3). From the user point of view, it allows the possibility to edit core information, independently from any specific modelling language, and to retrieve and reuse this data under different views, accomplished with the help of several graphical diagrams.



**Figure 3. SLMToolBox - MetaModel Architecture Overview**

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 Modelling) “glue” between several modelling languages. Thus, we can reuse standard modelling languages without introducing modifications to their metamodel (e.g. BPMN, UML...). Graphical languages such as “ExtendedActigramStar” or “GraiGrid” can continue to evolve, with (almost) no impact on MDSEA Core metamodels (i.e. BSM).

### 3.2.3 Service Modelling features

The modelling environment supports the service system modelling activities by providing editors for domain specific models (BSM, TIM) and related modelling languages to enhance the description of the BSM and TIM models. In this research work, the SLMToolBox is providing a set of language specific modelling editors. The Table 1 gives an overview of the modelling editors to be included in the SLMToolBox for each modelling level (BSM and TIM). These modelling editors are integrated within the same environment and technical platform (Eclipse Juno, 2015) in order to maintain data interoperability; coherence between models and improve the usability of the tool, from the user perspective.

Modelling Level	Goal	Modelling Language	Editor
BSM	Describe service at high level	BSM Templates	Specific Development
BSM	Describe simple business processes	Extended Actigram Star	Specific Development
BSM	Describe decisional structures of the organization	GRAI Grid	Specific Development
BSM	Describe Information Structures	UML (Use Case, Class Diagrams...)	Open Source Plugin ( <a href="#">PAPYRUS</a> )
TIM	Describe service at high level	TIM Templates	Specific Development
TIM	Describe detailed business processes	BPMN 2.0	Open Source Plugin ( <a href="#">BPMN2.0 Modeler</a> )
TIM	Specify the IT artefacts	UML (Use Case, Class Diagrams ...)	Open Source Plugin ( <a href="#">PAPYRUS</a> )
TIM	Simulate the business processes	DEVS (Atomic and Coupled Models)	Specific Development

**Table 1** - SLMToolBox - Modelling Editors & Languages Mapping Overview

#### 3.2.3.1 GRAI Grid Editor

The SLMToolBox proposes a GRAI Grid (Doumeingts et al., 1998) editor modelling language for modelling the decisional structure of specific enterprise. The GRAI Grid concept relies on the fact that any decision management that needs to be taken will always be made with reference to a specific time horizon. Managers typically define strategic, tactical, operational and real-time management levels. These levels implicitly involve a hierarchy of decision functions structured according to decision horizons or periods. The user of the SLMToolBox GRAI Grid at BSM level can represent the decision structures that belong to the VME partners and the collaboration dependencies between them.

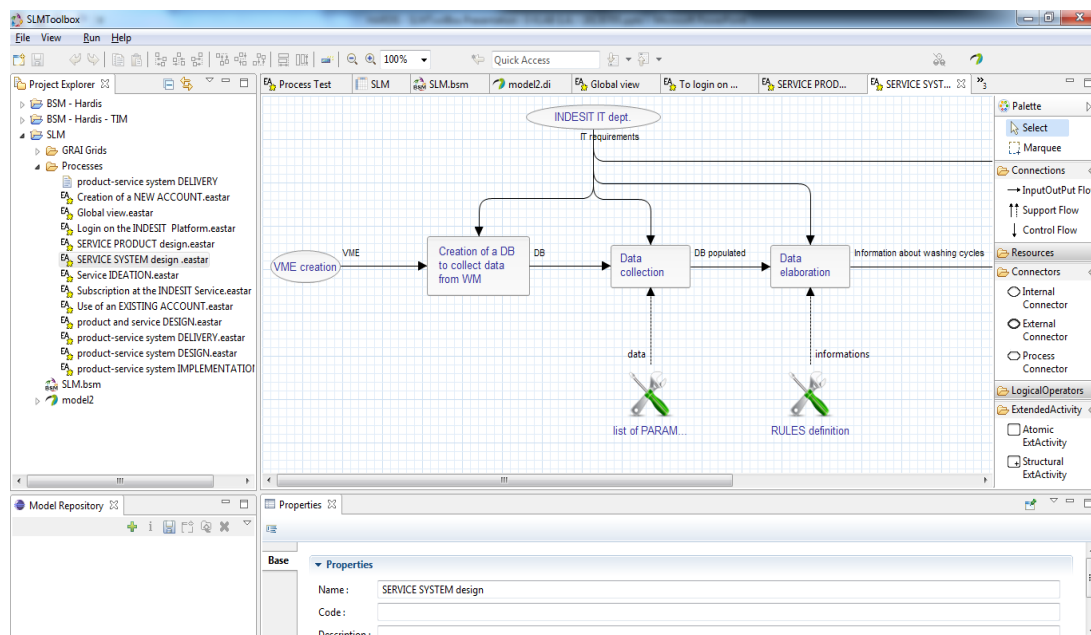
In detail, the decisions formalized on the basis of the GRAI Grid modeling editor enrich the BSM models with the data related to the governance model of the service system. The user can select appropriate indicators from a reference list according to a set of search criteria's, and propose facilities to check the coherence (links and weights) of the triplets {objective, drivers – decision variables, and primary indicators} for each decision center. This modeling language is not considered in the model transformation detailed in this paper.

### 3.2.3.2 Extended Actigram Star Editor (BSM Level)

A VME is an organizational form that marshals more resources than it currently possesses on its own, using collaborations both inside and outside of its boundaries, presenting itself to the customer as one unit. It is a set of (legally) independent enterprises that share resources and skills to achieve a mission/goal. In order to model these relations and collaborations between partners, collaboration diagrams should be developed (were necessary) at the various abstraction levels of the MDSEA (BSM-TIM-TSM). The Extended Actigram Star language models business processes at the business level (BSM), it offers the concept of connectors (InternalConnectors, ExternalConnectors, and ProcessConnectors) which represents collaboration between entities within the same organization (single enterprise) or between different organizations (partners in a VME). In certain cases (collaboration between partners) users need a more presentable and readable presentation to demonstrate the collaboration.

The user of the SLMToolBox at BSM level is usually aware of the different processes that belong to the VME partners and the dependencies between them. As a result he is able to model this collaboration in a one detailed EA\* diagram. Entities belonging to different organizations are differentiated using the organization concept introduced in EA\* and implemented in the EA\* editor (Figure 4).

In the same collaboration context the user of the SLMToolBox is able to connect to the Assets Repository, to browse and search for relevant assets to include in its service models, so that he is able to “compose” a new service, on the basis of existing assets exposed by the members of the VME.



**Figure 4 SLMToolBox - User Interface for EA\* Modeling at BSM Level**

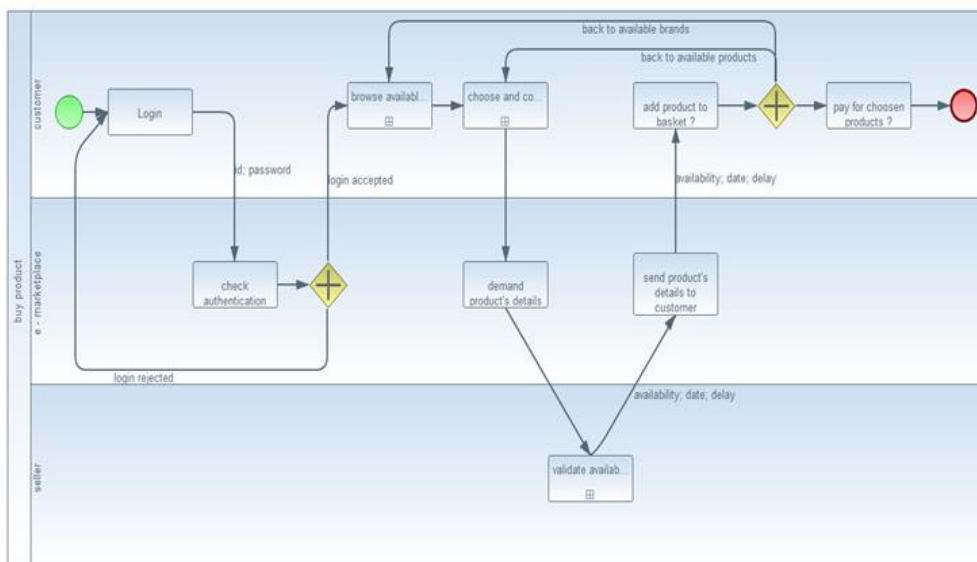
### 3.2.3.3 UML Editor (BSM Level)

The SLMToolBox is integrating a UML (OMG, 2011a) editor that is required in order to capture the “domain model” at the BSM level and elaborate TIM models. The UML modeler must satisfy several constraints. First it has to be integrated with the technical platform of the SLMToolBox (Eclipse Platform). It has to comply with UML 2 standard XMI representation format. It needs to support the following UML diagram types: Use Case diagrams, Class Diagrams, Component Diagrams, Sequence Diagrams, and Activity Diagrams.

Authors have concluded that Papyrus (Papyrus, 2015) matches the requirements. It is a dedicated tool for modelling within UML 2, open source and based on the Eclipse environment. The key feature of Papyrus can be summarized as follows. It is Eclipse UML2 compliant. It fully respects of the UML 2 standard as defined by the OMG and the DI2 (Diagram Interchange) (OMG, 2012) standard. The architecture of Papyrus is extendable; it allows users to add new diagrams, new code generators, etc. At the end, the profile development supports facilities for UML 2 profiles. In this paper, these models are not considered in the model transformation process.

### 3.2.3.4 BPMN Editor (TIM Level)

In MDSEA, the Business Process Management Notation (BPMN) (OMG 2011b) is used for Business process modelling at the TIM level. The BPMN editor researched to be integrated in the SLMToolBox was supposed to be integrated with the eclipse platform, to fully conform the BPMN specifications, and support BPMN process and collaboration diagrams. BPMN 2.0 Modeler (BPMN2-modeler, 2015) was matching these criteria. It provides an intuitive modeling tool for the business analyst, which conforms to well-established Eclipse user interface design practices. It also provides visual, graphical edition and creation of BPMN 2.0 compliant files with support for the BPMN domain. The Figure 5 illustrates a BPMN model that results from an edition in the SLMToolBox.



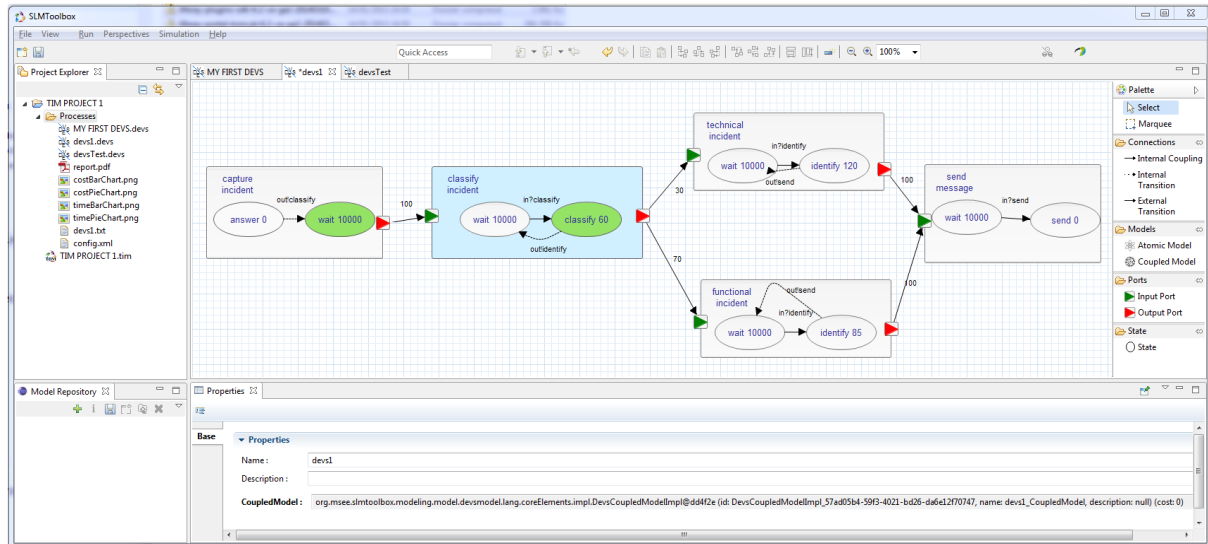
**Figure 5 SLMToolBox - User Interface for BPMN Modeling at TIM Level**

### 3.2.3.5 DEVS Editor (TIM Level)

To observe and verify the desired dynamic behavior of EA\* and BPMN models, a simulation model has been introduced. The SLMToolBox has been extended with a DEVS model editor

developed by conforming (Zeigler et al., 2000) specification. It generates a DEVS coupled model thanks to the template instantiation of DEVS atomic models and their coupling.

The tool is capable of running simulation to observe performance indicators evolution such as time to achieve a service process. Also for pedagogic objective, it shows a step by step animation, starting from the first active state in models till reaching the last active ones. Animation of DEVS diagrams is based on the results obtained from the simulation. The animation indicates active states and models that represent BPMN activities with associated resources. Step by step animation can be visualized by color timed change such as indicated in Figure 6.



**Figure 6 SLMToolBox - User Interface for DEVS M & S at TIM Level**

### 3.2.3.6 Model transformation features

The mapping of languages and concepts proposed in previous sections is implemented using ATL (Atlas Transformation Language) (Jouault et al., 2008). Then XSLT (eXtensible Stylesheet Language Transformations) is used to create the graphical objects in order to open the transformed diagrams in their corresponding graphical editors (i.e. EA\*, BPMN, DEVS). Using this combination of ATL and XSLT helps in separating the model concepts from graphical ones.

The SLMToolBox possesses a wizard for the creation of new diagrams. User is able to create BPMN diagrams in two ways: either to start from scratch and to create a new BPMN diagram by the standard way, or to create a new diagram from an existing EA\*. The second choice requires a set of implementations in order to make it possible. After the user has chosen the EA\* diagram, an ATL transformation is applied which transforms the EA\* model contained in the diagram into a BPMN model. Now that the BPMN model is available, it is important to generate its corresponding graphical objects. XSLT is used for such purpose and generate the graphical diagram view of the model. The result of The XSLT transformation will be a BPMN diagram that can be opened using the BPMN modeler of the SLMToolBox.

BPMN to DEVS transformation has been implemented for simulation purposes. DEVS is the formalism used to study if the objectives identified by the user could be accomplished by business processes developed. The transformation from BPMN to DEVS is implemented and developed using same implementation strategy used for EA\* to BPMN transformation. Again, as BPMN diagrams, DEVS diagrams can be created in two ways either from scratch or from an existing BPMN diagram. ATL and XSLT are also used to obtain a final DEVS

diagram that can be viewed and simulated by the DEVS editor. The mapping has used the metamodel proposed in SimStudio (Touraille et al., 2011).

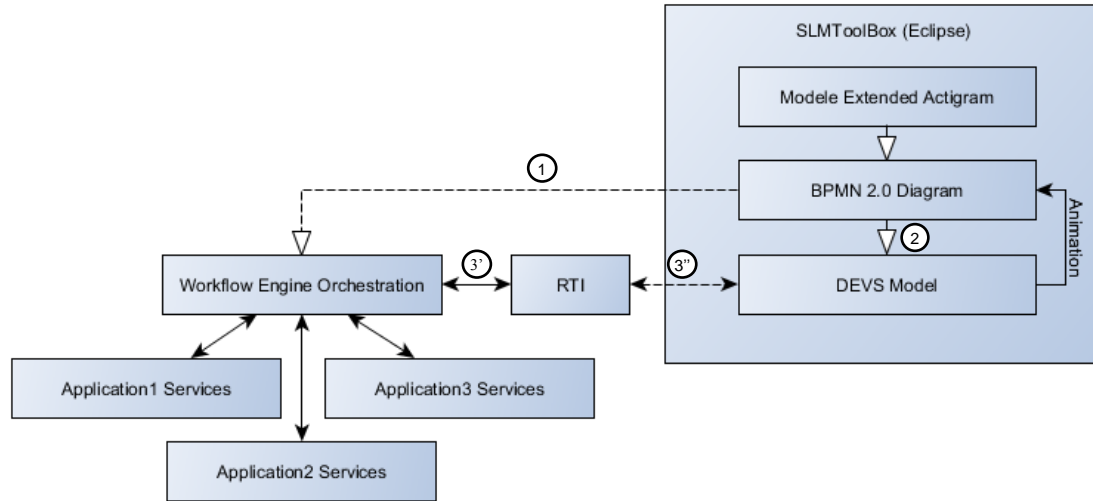
### **3.3 Mixing Simulated Models and Calls to services**

#### **3.3.1 Concepts and Architectural Framework**

The user of the SLMToolBox was restraint originally transforming all EA\* and BPMN model components to DEVS models. This is not realistic, new services are commonly coupled with existing ones. To improve the integration with its environment and the validation of the conceptual model, we propose to compose a simulation model that simulates non-existing or unavailable enterprise services while also being coupled and reusing existing enterprise service. 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 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 service process model at BSM and TIM is composed of both existing services and new ones to be developed. The new concepts will have to match with existing technical services. This first issue can be addressed by transforming BPMN existing concepts into executable Workflow of services as described in the Figure 7 circle 1 with the dashed link and white arrow point 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 is tackled using DEVS models running the behavior of enterprise BPMN services (Figure 7 circle 2). Then to make them interoperable, the authors proposed to make services calls (Figure 7 circle 3') and DEVS simulation (Figure 7 circle 3'') compliant with HLA to communicate with other distributed and heterogeneous components. This is represented in the Figure 7 circles 3 with RTI link both from DEVS Workflow 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 prepares the fields for 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. The previous works recalled in section 2.4.2 have already put the first stones in this domain. This step will go further in this direction, guided by the MDSEA lifecycle, by proposing to generate real calls to services and external systems.





**Figure 7** Mashing up Workflow Models and Real Web Services

### 3.3.2 Architecture Implementation

Up to now, the SLMToolBox is preparing the models to be ready for use at TIM level. The first problem is the matching between the concepts announced in the enterprise models of services and the technical services. This TIM 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. However it is interesting to test by simulation the system in its future real environment. So the TIM models could be enriched with technical interfaces, provisionally and in the objective of test, to be connected directly with other software or material components to support a concrete service calls implementation. The idea proposed in this paper is to reuse the experience acquired when connecting DEVS models with service calls (Ribault et al., 2014). The interoperability can gain from this service architecture and distributed interoperability simulation architecture proposed previously. The missing element in the Workflow chain is the component that links the model with the service call. This work permits the user, after the edition of the BPMN model, to annotate the model and to separate two categories in the model. It distinguishes the parts that will be transformed to simulation models and the parts that will be transformed into Workflow of services.

Concerning the transformation of BPMN portions to executable Workflow, the XML Meta model approach has been preferred. The reason comes also from the fact Taverna saves similarly the Workflows in XML. So we have transformed the BPMN sequence and message flow part that link activities inside lanes or pools into Taverna Workflow abstract service calls thanks to XSLT. The user is still having the charge to fulfill the query detail for the service call but the BPMN communication actions prepare the primitives for these service calls. The interoperability has been facilitated by the model structure already saved with the XML format in the Eclipse standard. Concretely a resulting executable Workflow model is generated from the considered extract of the BPMN model. This model is played with Taverna that sequences the service calls and answers. It facilitates the interoperability with existing services and permits to integrate them in the hybrid behavioral simulation.

On the other side, unavailable services are transformed into DEVS models to be simulated. Nevertheless, the time synchronization between these heterogeneous components is required. This question is detailed in the following section.

### 3.3.3 Introducing HLA Compliance to SLMToolBox

The functional and syntactical interoperability with web service providers is assumed by the Taverna engine that handles services calls and callback and links the different applications. This tool allows defining a sequence of service calls. Nevertheless this tool, as most WFMS does not deal with time constraints that exist in a sequence mixing real/simulated execution. In consequence it does not provide time synchronization for dialoging with the simulation of services calls in a defined time dependent sequence. For instance, the access to a specific data in advance or too late can be a problem, i.e. with obsolete values or future values. This problem arises systematically when parallel process is executed. To address this issue, a solution that has been proposed in (Ribault et al., 2014); it has been reused and merged with the SLMToolBox features.

The idea has consisted in using Taverna WF interoperability facility as the main interoperability layer between services (including applications and simulations) and simulated service processes of the SLMToolBox. We have reused (Ribault et al., 2014) bridge developed in Java between Taverna and the SLMToolBox. It is executing the service call scenario scripts when a message is sent out of the simulation models. To keep the simulation models synchronized, we used the Java based DEVS/HLA architecture proposed in (Zacharewicz et al., 2008) upon the DEVS simulator of the SLMToolBox to have a time based message scheduler for the WF scenario. In that case the use of the RTI is not systematic; it is only solicited when a simulated component is communicating with a service WF. In (Tu et al., 2014), the poRTico RTI (poRTico, 2015) supporting HLA 1516 has been used in a Java platform to facilitate the connection between RTI and the calls to web services. It has been partially extended to support service approach as preconized in 1516 Evolved. This RTI implementation has been reused in this work; it has been facilitated by the fact all component where java based coded (SLMToolBox, Taverna, DEVS/HLA).

In details, the service process DEVS models have been embedded into HLA federates. These federates gain interoperability properties to communicate within the distributed simulation, thanks to HLA. On one side, The DEVS federates publishes or subscribe to HLA “flow” messages as an activity input/output is connected to a service in the sequence (or message) flow within a time synchronized environment. The HLA FOM describes solely basic flow messages structures that are used to trigger services. It includes the web services parameters, query or answer to the query and a timestamp to be published. Nevertheless, these models should, before to launch the simulation even synchronized, need to be enriched to have the primitives for “real” services calls and communication including how to form a query and how to reach the URL of the service. On the other side, the HLA add-on for Taverna proposed in (Ribault et al., 2014) subscribes to this published information to form the service call. On the way back, the callback publishes “flow” messages destined to wake up and continue the simulation.

The execution can be summarized as follows. On demand, the DEVS models are playing the behavior of the service components with simulating the time spent achieving the service process action that attends to reproduce the real reaction delay. The local synchronization algorithm embedded in the DEVS models have reused (Tu et al., 2012) approach that already combined DEVS M&S and HLA RTI simulators to simulate the behavior of several enterprise process components. The next section details how it mixes HLA and web services following the time management proposition enounced in a prior study (Zacharewicz et al., 2008).

### **3.3.4 Time Orchestration**

The distributed simulation principle proposed by (Tu et al., 2012) is based on the original pessimistic algorithm described in (Chandy and Misra, 1979), but adding more recent

advances on lookahead introduced in (Zacharewicz et al., 2008). The RTI is defining the ordering of the actions regarding their occurrence time. It stores the information before releasing them regarding the scenario definition played in DEVS and Taverna. It can be also considered as the script clock and blocker/releaser of the simulation. Regarding time synchronization, the DEVS/HLA models have been already instrumented in (Zacharewicz et al., 2008) to inform the RTI about their Lower Bound on Time Stamps (LBTS) (IEEE, 2010) to compute the Lookahead (minimal treatment delay) and unblock the simulation. Taverna was not defined for that. The idea has been to define minimum treatment duration in each Workflow step to be communicated to the RTI. Thanks to this information taken by the RTI as the Taverna LBTS, the distributed simulation can be run without deadlocks.

In detail, in this approach, the RTI collects DEVS simulation output messages, sorts them and triggers the web services call right in time. Then it forwards the message back to the DEVS service models in the SLMToolbox that simulate the behavior of the service process according to defined scenario, i.e. a timed sequence of chained service activities. So the RTI can receive messages both from a service provider as a service answer or from a DEVS model that sends an output message as a simulation output of a local behavior. The messages received from the server are service answers (callback). They both possess time stamp information to be used by the RTI to rank the message at the right place in the queue. Then depending on the execution state of the global clock it will sort the message and deliver (publish) it to a pending receiver. The approach is based on the conservative algorithm of (Chandy and Misra, 1979); it used the DEVS/HLA algorithm, proposed in (Zacharewicz et al., 2008). For e.g., if a message arrives late, it suspends temporarily the simulation, it is not ignored. Then the simulation is unblocked by identifying the lower time stamped message and passes to process this next message, it shows the interest of providing an accurate value of LBTS to the RTI. The final target of the message can be a web server through Taverna. In that case the RTI publishes a message. This message is received by Taverna since it was subscribing to it, then it transforms it to a service call and then triggers a server web service. If the message is a callback addressed to a DEVS model to trigger a service component behavior, the message is sent through the RTI to the appropriate DEVS component in the service process using the coupled model structure.

#### 4. CONCLUSION

---

This paper has proposed to implement MDSEA principle in a mechanism to generate from a BSM conceptual, a more IT oriented model and then a simulation model to observe behavior of the service model at TIM level. The methodology attempts to support the integration, testing and interoperability within the existing enterprise services system. This contribution has tried to answer to two major problems: (1) conceptual models are mostly blueprint and they contain components with implicit behaviors that need to be explicated if simulated in order to test and verify their correctness in the global behavior in the enterprise system, and (2) simulated conceptual models does not embrace all enterprise service system, thus they must be able to interact with existing enterprise services. The first problem has been previously tackled down to TIM level thanks to model transformation proposed in the SLMToolBox architecture. The second problem has also been addressed in composing service calls Workflow and DEVS M&S. 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 - automated MDSEA approach. We described the basis for this new approach. In particular we have proposed a method for decomposing BPMN models into, on one side simulation components and, on the other, real service interaction. These two sides are then coupled again in a HLA based distributed testing system that composes simulation

models with concrete service calls. This work still needs significant and frequent human actions in the loop. This will be extended in order to generate a more achieved technical Workflow of services calls using semantics to recover information and to compose the service call.

## 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 May 2014)
- BPMN2 Modeler official site: <http://eclipse.org/bpmn2-modeler/> (accessed October 2015)
- Chandy, K.M. and Misra, J. Distributed simulation: A case study in design and verification of distributed programs. *IEEE Transactions on Software Engineering*, 5 (5). 440-452, 1979.
- 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.
- Eclipse Juno, 2015, <https://eclipse.org/juno/>, (Accessed October 2015)
- IEEE std 1516<sup>TM</sup>-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.
- Jouault Frédéric, Freddy Allilaire, Jean Bézivin, Ivan Kurtev, ATL: A model transformation tool, *Science of Computer Programming*, Volume 72, Issues 1–2, 1 June 2008, Pages 31-39, ISSN 0167-6423
- FP7 – FoF-ICT-2011.7.3. 2012. *Manufacturing SService Ecosystem Project- Annex 1 description of work*; July 29, Available from: <http://msee-ip.eu/> (accessed May 2014)
- Goult, R. *Introduction to ISO 9001.2008 Handbook*. Available from : <http://fr.scribd.com/doc/171647046/ISO-9001-2008-Handbook> (accessed October 2015)
- 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 October 2015)
- 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.
- 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.
- National Institute of Standards and Technology (NIST). *IDEF, Integrated DEFinition Methods*. Available from <http://www.idef.com/IDEF0.htm> (accessed May 2014).
- OMG 2011a. *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 October 2015)
- OMG 2011b. Business Process Model and Notation (BPMN) Version 2.0. <http://www.omg.org/spec/BPMN/2.0/PDF/> (Accessed Mars 2014).
- OMG 2012. Diagram Definition (DD) Version 1.0. Available from <http://www.omg.org/spec/DD/1.0/> (accessed Mars 2015).
- OMG. MDA Guide Version 1.0. 2005 <http://www.omg.org/cgi-bin/doc?omg/03-06-01> (accessed October 2013).
- Papyrus official site: <http://www.eclipse.org/papyrus/> (accessed October 2015).
- Portico, <http://www.porticoproject.org/> (accessed October 2015)
- 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
- 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.
- Touraille Luc, Mamadou K. Traoré, and David R. C. Hill. 2011. A model-driven software environment for modeling, simulation and analysis of complex systems. In *Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium* (TMS-DEVS'11). Society for Computer Simulation International, San Diego, CA, USA, 229-237.
- Weske, M. 2012. *Business Process Management Architectures*. Business Process Management, pp 333-371.
- WfMC Workflow Management Coalition. 2005. Workflow Process Definition Interface—XML Process Definition Language. WfMC-TC-1025, October.
- 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 recent PhD from 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 RIBAUT** is associate researcher fellow at the University of Bordeaux IMS Lab. 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 HDR at University of Bordeaux (IUT MP / IMS) 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. 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.