Orchestrating the Interoperability Workflow within a Transport Simulation Platform

Judicaël Ribault, Gregory Zacharewicz and Jean Christophe Deschamps
Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France
judicael.ribault@u-bordeaux1.fr, gregory.zacharewicz@u-bordeaux1.fr, jean-christophe.deschamps@u-bordeaux1.fr

Keywords: Workflow, Taverna, Interoperability, Discrete event simulation, G-DEVS, etc.

Abstract

The domain of logistics and transport is now gaining with the use of the web, geo positioning and RFID to improve the tracking and decision making for the product more appropriate routing in order to save time, cost and reduce impact on the environment. The combination of these software and hardware devices faces interoperability problems. This paper proposes to introduce a new simulation platform that will mix interaction with real world including sensor and human interfacing and simulation world. In detail, the proposition of this paper is to combine the Taverna Workflow, which handles and triggers the call of web services proposed by a platform, with several simulation models. In particular one drawback of several workflows orchestrator tools do not provide time management facilities to handle time and to rhythm simulation run. This paper introduces a clock ordering information defined in the G-DEVS formalism to give the beat the simulation of the system. The imbrication of G-DEVS modeling and simulation with the workflow Taverna shows the interoperability possible and complementarity of these approaches.

1. INTRODUCTION

The effectiveness of enterprise information technology system (IS) depends not only on its internal interconnectivity of its inner software components, but also on its ability to exchange data, so to collaborate, with every day new tools developed and updated in the environing digital world. This necessity led to the development of the concept called interoperability that allows improving collaborations between enterprises IS. No doubt, in such context where more and more networked enterprises are developed; that enterprise interoperability is seen as one of the most wanted solution in the development of an enterprise IS. Also the data treatment calls both human processing and automatic treatments. The sequencing of these actions is desired to be controlled or orchestrated by a high level application that can decide the component and/or human resource to solicit.

From a research point of view, several works has been launched since the beginning of 90's in the domain of Workflow. Workflow was first designed to formalize and improve enterprise business process. A product workflow is a set of linked steps required for developing a product until it gets into market [Weske 2012]. The workflow steps are based on observing a number of steps that are usually enchained manually and formalizing them. The research on the Workflow initiated by the WfMC was a premise to workflow modeling (e.g. with BMPN) and it permits the development of recent ERP systems in the enterprises. Nevertheless a clear distinction appeared in the late 90's between the theoretical approaches in this domain and applied approaches. In the theoretical approach, Modeling and Simulation (M&S) is a center consideration, while in the applied approaches execution is the core problem. Few approaches tried to compose M&S and real executions one main reason was the poor efficiency of these approaches were slowed down by the simulation engine that were constraining by the causality and time management [Chandy, 1979].

Recent improvement and web based development propose new facilities to connect in a more convenient way the applications. For instance the web services can support that question. We can classify the Web services into two categories:

- Web services of type "REpresentational State Transfer" (REST) [Richardson, 2007] whose main purpose is to manipulate XML representations of Web resources using a uniform set of HTTP operations (GET, PUT, POST, DELETE) and URI.
- Arbitrary Web services, which expose an arbitrary set of operations that can be executed remotely by using SOAP and WSDL standards that facilitate interoperability.

We propose to use web services and workflow for interoperability among simulation and real-world application. Web services enable the integration of applications or data from heterogeneous sources (i.e. Mash-up). This paper is proposing to apply the use of workflow Web services and simulation to the PRODIGE application.

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

2. BACKGROUND

In this section, we first present the interoperability concept. Then we quickly present the PRODIGE system and how workflow can be used for experimentation. Then we present the DEVS formalism and its interoperability through web services. Finally we present the Taverna workflow management system to orchestrate the experimentation.

2.1. Interoperability

Enterprise Interoperability [Chen, 2003] refers to the ability of interactions between enterprise systems. The interoperability is considered as significant if the interactions can take place at least at the three different levels: data, services and process, with a semantics defined in a given business context.

Interoperability extends beyond the boundaries of any single system, and involves at least two entities. Consequently establishing interoperability means to relate two systems together and remove incompatibilities. Incompatibility is the fundamental concept of interoperability. It is the obstacle to establish seamless interoperation. The concept 'incompatibility' has a broad sense and is not only limited to 'technical' aspect as usually considered in software engineering, but also 'information' and 'organisation', and concerns all levels of the enterprise.

Our goal is to tackle interoperability problems through the identification of barriers (incompatibilities) which prevent interoperability to happen. Basic concepts relating to enterprise interoperability are categorised into three main dimensions as described below.

Today, most of the approaches developed are unified ones such as for example in the domain of enterprise modelling, we can mention UEML (Unified Enterprise Modelling Language) and PSL (Process Specification Language) which aim at supporting the interoperability between enterprise models and tools.

Using the federated approach to develop enterprise interoperability is most challenging and few activities have been performed in this direction. The federated approach aims to develop full interoperability and is particularly suitable for an inter-organisational environment (such as networked enterprises, virtual enterprises, etc.). In the enterprise interoperability roadmap published by the European Commission in 2006, developing federated approach for interoperability is considered as one of the research challenges for the years to come.

From the state of the art of the Interoperability in federation domain and implementations experiences presented in the preceding points, we can define several directions of almost natural matching with HLA concepts in the following domains.

The first direction concerns the definition of commonly recognized paradigms and data structure because order and clarification is required to propose a sound paradigm. The second requirement not addressed at the enterprise modelling level is the synchronization of data. The order of data exchanged is important, ignoring this can lead to misunderstanding and wrong functioning of the model. Finally the enterprise modelling must consider the confidentiality management of data. The interoperability can be considered between concurrent enterprises in that context, a strategy of data sharing/not sharing between these must be defined. We present in the next points propositions to address these requirements.

2.2. PRODIGE

The PRODIGE project[1] aims to prepare the future of transport, placing the reflection at the level of the organization and control of the flow of commodities in order to provide a technical and organizational solution helping the reduction of kilometers, optimization of the tours, optimization of the volumes transported and taking into account new issues related to sustainable development.

The basement for this work is a Web application exposing its methods through the use of SOAP Web services in order to promote interoperability (set a tour, view the results, etc.).

2.3. Workflow

Workflows can quickly orchestrate several experiments (and optionally simultaneously) of the PRODIGE application. Indeed, computer experimentation has no time constraints which must face the real experiment: a tour of several hours can be simulated in a few seconds. Among the possibilities offered by computer experimentations, we can mention the possibility to verify and debug the PRODIGE application during its development. This parallelism of tasks saves time and resources allocated to the development of the PRODIGE application. Computer experimentation also allows to quickly test new features. Once all the features established and verified, computer experimentation can create scenarios of use mimicking the behaviour of different actors (manager, customers, and drivers in the case of PRODIGE). A scenario can have several objectives:

- Quantitative: you wish to calculate and compare several variables such as the number of kilometres travelled by products or the amount of CO2 emissions produced for a set of delivery
- Qualitative: you want to follow the different steps of the delivery of a product (e.g. respect of delivery times, compliance with the cold chain, etc.)

[1] anr-prodige.com/

- Analytics: you want to observe a special case not understand, difficult or impossible to reproduce with the real system, often for scientific purposes.

To this are added demonstrations scenarios, to explain PRODIGE to public audience and track the movement of vehicles depending on the scenario chosen.

2.4. DEVS M&S

Discrete EVent Specification (DEVS) was introduced by [Zeigler 00]. It describes systems with discrete event approach. In particular it represents state life time and takes into account elapsed time while firing a state transition triggered by an event received.

Genralized DEVS (G-DEVS) emerged with the drawback that most classical discrete event abstraction models (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 00]. Thus, G-DEVS defines the coefficient-event as a list of values representing the polynomial coefficients that approximate the input–output trajectory. Therefore, a DEVS model, from the founding point of view, is a zero order G-DEVS model (the input–output trajectories are piecewise constants).

G-DEVS keeps the concept of the coupled model introduced in DEVS [Zeigler 00]. Each basic model of a coupled model interacts with the others to produce a global behaviour. The basic models are either atomic or coupled models that are already stored in the library. The model coupling is done with a hierarchical approach (due to the closure under coupling of G-DEVS, models can be defined in a hierarchical way).

On the simulation side, G-DEVS models employ [Zeigler 00] an abstract simulator 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 insure the simulation of atomic models, Coordinators, which insure the routing of messages between coupled models, and the Root Coordinator, which ensures global simulation management. The simulation runs by sending Imessage to all Coordinators and Simulators, and continues by exchanging specific messages (*message for internal event, Xmessage for external event and Ymessage for output event) between the different processors. The specificity of G-DEVS model simulation is that the definition of an event is a list of coefficient values as opposed to a unique value in DEVS.

2.5. Services Simulation

We use discrete-event simulation results to mimic the behavior of certain elements of the PRODIGE system. In [Al-Zoubi and Wainer 2010a] the authors discussed the advantages and disadvantages of several modeling and simulation environments, including the High Level Architecture (HLA) [Kuhl et al., 2000], CORBA, SOAP-based Web-services, etc. As discussed there, most of these distributed simulation middleware still lack of plug-and-play interoperability, dynamicity, and composition scalability. Based on this conclusion, they designed the first existing RESTful Interoperability Simulation Environment (RISE) middleware [Al-Zoubi and Wainer 2010b].

The main goal of RISE is providing simulation interoperability and mash-up regardless of their formalism, theory or implementation. Access to RISE is done through Web resources (URIs like a classic website URL) and XML messages using HTTP channels: GET (to read a resource), PUT (to create new resource or update existing data), POST (to append new data to a resource), and DELETE (to remove a resource). RISE allows modellers to run any number of experiment instances, whose settings and resources (URIs) are persistent and repeatable (unless deliberately removed or updated). An interface between RISE and CD++ [Wainer, 2002] allows running distributed simulations using the CD++ simulation engine.

We need to orchestrate various services to simulate the use of the PRODIGE application. We want to use the results of simulations to drive the PRODIGE application through formalized scenarios. This formalization and orchestration of services corresponds to the use of workflow of services. Workflows of services can be useful for computer experimentation by promoting replayability, sharing and interoperability [Ribault and Wainer 2012a].

2.6. Workflows of services

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. They determine that Taverna provides a more compact set of primitives than BPEL and a functional programming model that eases data flow modelling. Due to our needs, we identify that a workflow management system such Taverna would be a better alternative than BPEL to illustrate the feasibility of our approach.

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:

- A service capable of running Java code directly within Taverna.

- A service to run a remote application via the REST protocol.
- A service to run a remote application via the SOAP/WSDL protocol.

A Taverna service can take inputs and produce outputs. The value of an entry can be part of the workflow (hardcoded) or a parameter to provide information during the execution of the workflow. A REST service returns systematically 2 outputs predefined: the return value of the Web service (404 if the resource is not found, 200 if everything went well, etc...), and the contents of the response (XML, HTML, ZIP, etc.). Figure 1 represents a REST service in Taverna. The number of input arguments is variable and chosen by the developer of the workflow. The number of output arguments is fixed.

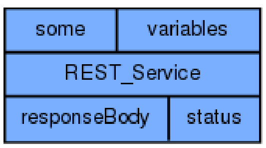

Figure 1: Taverna REST service.

In contrast, a WSDL service will find automatically, thanks to the WSDL file, the number and type of input and output. Figure 2 represents a Taverna workflow with a WSDL service in green in the middle of the figure. The service is available in Taverna after the addition of the URL of the WSDL file (such as http://xxx.xxx.xxx.xxx:8080/WS-PRODIGE/services/Identification?wsdl). Taverna offers the possibility to automatically format the input and output based on the type of parameters required by the Web service. In this example, the Web service "identificationChauffeur" that allows a driver to identify within the PRODIGE application takes as input a data type 'identificationChauffeur_input' that encapsulates id, imei, and pwd. The Web service "identificationChauffeur" produce as output a data type 'identificationChauffeur_return' that encapsulates various data such as firstName, lastName, login, etc.

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 interfaces for interacting with the user using boxes dialog, lists, etc. A Taverna workflow can also contain nested workflows in a hierarchical manner. In this way, a set of simple workflows easily allow to design more complex workflows. These workflows can then be shared, reused, and adapted to new needs.

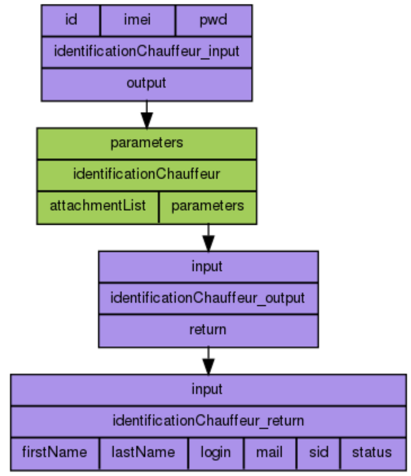

Figure 2: Taverna WSDL service.

3. CONTRIBUTION

We propose to use workflow of services as the interoperability layer among several services. In addition, we propose to integrate the G-DEVS engine as a specific workflow engine. G-DEVS is a formalism based on a state machine automaton. Workflows differ from state machine as state machine can be cyclic graphs while workflows are usually acyclic. Workflow proceeds down different branches until done. Thus, using G-DEVS coupled to another workflow engine to process a workflow could benefit from the DEVS formalism while keeping the top to bottom behaviour of the main workflow manager. Interoperability among workflow engines and applications are done using web services.

3.1. Workflow Orchestration Architecture

The Figure 3 presents the orchestration architecture based on the workflow architecture by the WfMC.[2]

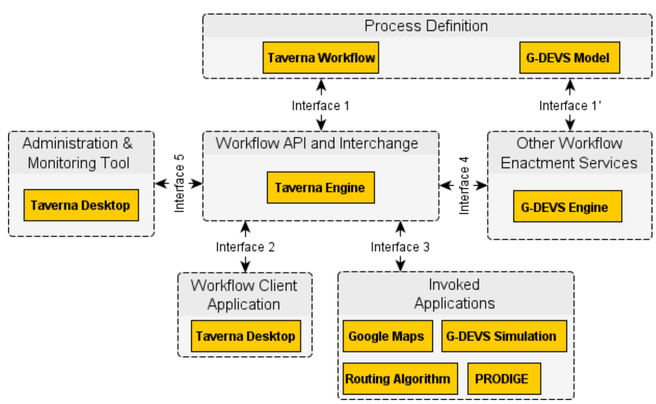

Figure 3: Workflow orchestration architecture.

[2] http://www.wfmc.org/

We propose to use Taverna and G-DEVS as the process definition formalism to express workflows. Taverna workflow represents the main workflow that organizes all tasks and enables interoperability between services. Taverna workflow process definition will be executed by the Taverna Engine (Interface 1). G-DEVS process definition will be executed by the G-DEVS Engine (Interface 1') as an others workflow enactment services. Communication between both engines (Interface 4) will be granted by web services thanks to RISE. Taverna interprets G-DEVS workflow event and enables the interoperability with other services using RESTful or SOAP/WSDL Web services protocols. Taverna ensures interoperability between workflows (Interface 4) and among invoked applications (Interface 3) such as Google Maps, G-DEVS Simulation, Routing Algorithm and PRODIGE. Interface 2 allows Taverna workflow to interact with users through the use of the Taverna Desktop.

3.2. Taverna Workflow Model

We want to test the PRODIGE application before moving to a phase of real experimentation. Then, we want to be able to quickly test algorithm, compare studies without having to drive trucks and monopolize drivers. Taverna is used to create scenarios using the PRODIGE application through workflow showing the behaviour of the users involved in the scenario such as customer who will apply for delivery of a point to another, managers who will validate and create the tour, and the drivers who will drive trucks.

Figure 4 presents one of the workflows we have created under Taverna which play the role of customers and manager in order to setup a tour. Data related to experimentation is stored in the first box "Prodige_Data" at the top of the workflow. The workflow will, from these data, connect to the system as a manager and register the different elements that compose the PRODIGE system (smart device, driver, customer (crossing point, vehicle and deposit)). Once these steps are complete, "DistanceComputation" will calculate distances between crossing points for the entire PRODIGE system. Then the workflow will mimic the behaviour of a customer by making a GS1 file (which is a standard format to place an order with a delivery service) and then submit it to the PRODIGE system through the box "Create_Command". Then, the workflow will play the role of the manager again and list and validate deliveries added to the system and create a tour taking into account all of the deliveries through an optimization algorithm wrapped in the PRODIGE application. Finally, the workflow will list and validate created tours as the manager would have done.

To simulate the movement of vehicles within the PRODIGE application, we retrieve all of the GPS coordinates from a route calculated through the use of the Google Maps API Web services[3]. Google Maps exposes its API through the use of REST Web service. The GPS coordinate are then sent to the G-DEVS workflow simulation model which takes into account the generation of errors to test possible transport problems. All of these simulation steps are formalized inside a workflow of services that guarantee the replayability of the experimentation. The workflow will also mimic the time of loading and unloading at each waypoint and hazards that may occur during delivery. The management of the time being naturally included in DEVS, we decided to mix the ease of organization and interoperability of Taverna with the time management capabilities of DEVS to express a workflow.

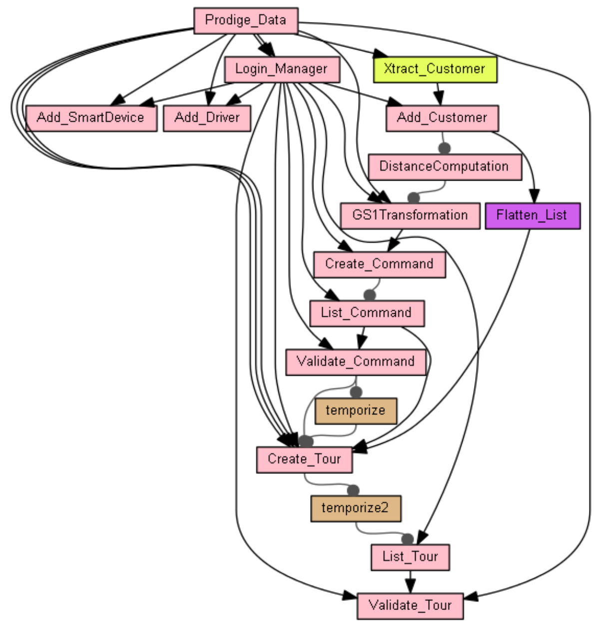

Figure 4: Taverna workflow to setup the PRODIGE system.

3.3. G-DEVS Workflow Model

In a previous work [Zacharewicz, 2011] several G-DEVS models were introduced to represent the behaviour of the various actors of the PRODIGE system.

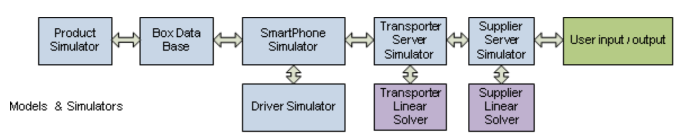

Figure 5 Workflow main components

The main components of the PRODIGE workflow have been proposed in G-DEVS models For instance the

3 https://developers.google.com/maps/documentation/webservices/

smartphone has been described. It detail the behaviour of the smartphone and in particular it précises how this device is reacting from its environment. In this approach the synchronization was given by an HAL RTI.

3.4. G-DEVS Clock and Sorting Model

In this paper the interoperability is assumed by the Taverna engine that calls the services and links the different applications. Nevertheless this tool does not provide time synchronization. Two options have been envisaged. The first was using a RTI to build an HLA federation [Al-Zoubi, 2011]. This option requires reusing an existing RTI that can set up rapidly a simulation but this kind of configuration can cause overheads in the communication like discussed earlier.

Because [Zacharewicz, 2011] already use G-DEVS models and simulators to simulate the behavior of several components in the PRODIGE environment, the idea proposed in this research is to define a G-DEVS model dedicated to be the clock of the PRODIGE workflow. This model will define the ordering of the actions regarding time. It also can be considered has the time driver of the simulation. In other terms G-DEVS, that is originally designed to run event driven simulation, is used in that case to run a time driven simulation.

In detail we propose in this paper a G-DEVS model that collects messages, sorts them and trigger right in time the services call to the PRODIGE server or forwards the message to the G-DEVS models that simulate the behavior of the PRODIGE components recalled in the Figure 5. This model can receive messages both from the server has a service answer or from a G-DEVS model that send an output message as a simulation result of a local behavior. The messages received from the server are service answers. They possess time stamp information to be used by the clock model to add the message at the right place in the queue. Then depending on the execution state of the clock it will sort the message and direct it to the proper receiver. The state of the clock can be processing a message or being available. In the first case, the approach is inspired from the conservative algorithm of [Chandy 79], in particular if a message is arriving late it will be ignored. The receiver can be the server. In that case it prepares an output message. This output message is addressed to Taverna that transforms it to service call and then trigger the PRODIGE server. If the message is addressed to a G-DEVS model to trigger component behavior, the message is directly sent to the appropriate G-DEVS component using the coupled model structure. In the second case (no input event to be treated) the state is transient and after a definite life time it automatically goes to another state. During transition to this state, an output message is generated in order to give the order to refresh the positioning of the trucks and product to the server according to the roadmap and geographical information extracted from Google maps. During the setting of the simulation the pace can be tuned in order to accelerate the simulation execution. Also at simulation time the execution can be stopped to show a particular case.

3.5. Interoperability

The interoperability between G-DEVS workflow model and invoked application such as PRODIGE are ensured by the Taverna workflow. Figure 6 present the sequence diagram among Taverna workflow, G-DEVS workflow and PRODIGE application. The Taverna workflow represents an experimentation scenario that is executed automatically by the Taverna Engine to test the PRODIGE application and is represented by the first column of the sequence diagram. The G-DEVS workflow model represents the workflow of a smart device sending every 30 seconds a couple of GPS coordinates to simulate truck move. The G-DEVS simulation is represented by the last two columns of the sequence diagram. Finally, PRODIGE is represented by the second column.

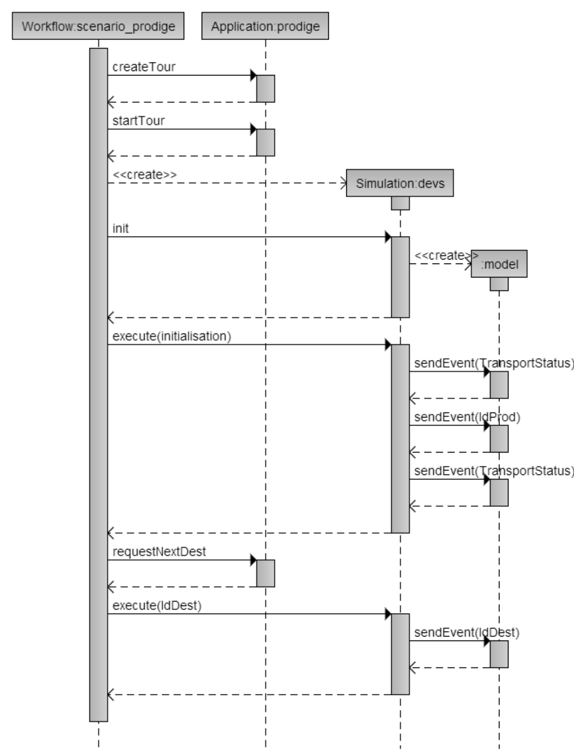

Figure 6 Interoperability sequence diagram.

The sequence is expressed as follow:

1. The Taverna workflow scenario invokes the PRODIGE application to setup a new round.
2. The Taverna workflow scenario invokes and initializes the G-DEVS simulation that will create in turn the workflow model.

3. The Taverna workflow scenario executes the G-DEVS workflow. The G-DEVS simulation engine interacts with the workflow model (sendEvent).
4. The Taverna workflow scenario gets in return what is needed by the G-DEVS workflow to continue its execution (i.e. the next destination of the truck).
5. The Taverna workflow scenario invokes the PRODIGE application to request the next destination of the truck associated with the smart device simulated by the G-DEVS workflow.
6. The Taverna workflow scenario continues the execution of the G-DEVS workflow by passing the next destination event to the G-DEVS simulation.

4. FRAMEWORK

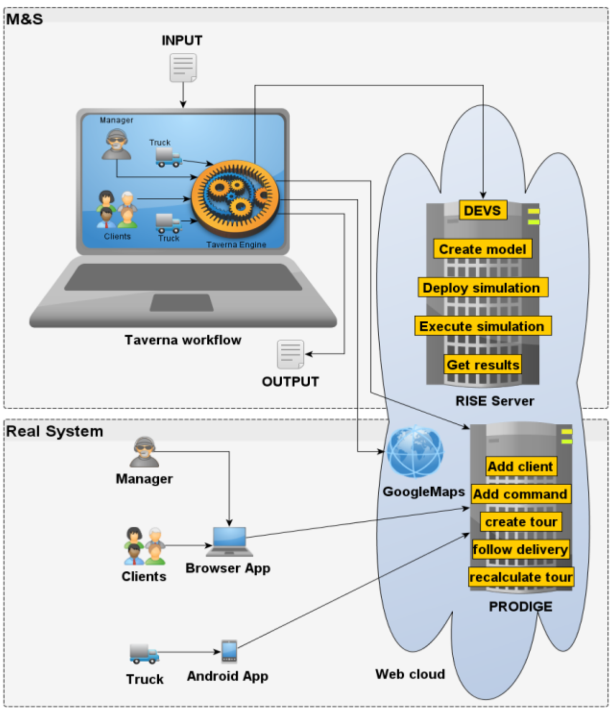

Figure 7 PRODIGE and simulation framework architecture.

We have implemented the architecture and concept described in the previous section. Figure 7 represents the solution framework. The top M&S box illustrates the virtual experiment while the bottom Real System box illustrates the real experiment. The virtual experiment is defined using Taverna workflow and DEVS simulation. The Taverna workflow mimics the behaviour of managers, clients and drivers while the DEVS simulation mimics the behaviour of smart devices. Communication between Taverna and DEVS are done through Web services thanks to RISE. The Taverna workflow communicates with the real PRODIGE application and Google Maps in the Cloud through Web service. The real experiment needs real human to manage the PRODIGE application (manager, clients) and drive trucks (divers). Communication between human and PRODIGE are done using a light web application (manager, client) or a mobile application on smart device (driver).

4.1. Scenario PRODIGE

We created several data inputs as well as several workflows to simulate different situations and experience the PRODIGE solution before placing it onto market. Packages must be picked up and delivered and:

- the delivery time windows are wide enough for it to be feasible with a single truck;
- the delivery time windows overlap and several trucks are needed to make the delivery on time;

Those two situations are done using the same generic workflows. We built another workflow to take into account hazards such as traffic jam or truck failure. Indeed, in those cases the workflow must take into account specific decision that could involve building new delivery.

4.2. Execution Example

Main experimentation workflow takes in input a XML configuration file that describes the whole experimentation. The workflow plays the role of all the actors (manager, clients, and drivers) and fills the PRODIGE system. Then, the workflow execute in parallel tours for each driver involved. The workflow retrieves the information needed on Google Maps and using G-DEVS simulation to mimic the behaviour of a real truck. The result of the execution of this workflow is directly visible in the PRODIGE web application on which you can view figure

Figure 8 the current path of a truck making its tour in the region of Bordeaux, France. We can also imagine an experiment mixing virtual truck and real truck since there is no difference from the PRODIGE perspective.

5. CONCLUSION

This work has permitted to introduce a new platform for simulation of logistic It recalled existing works that proposed to use the G-DEVS formalism for the description of the logistic platform components. Then it introduces the Taverna tool that will be the interoperability link to connect the services and the simulation components. Then it describes the G-DEVS model that has been proposed to serve has the clock ordering component in the system since the TAVERNA and more generally the services does not address the time synchronization. The main demonstration of this paper was to show the interest of interoperability in simulation. Here the approach was still pragmatic but the future works will propose to make the G-DEVS Clock model more generic to be reused in several service handling tools.

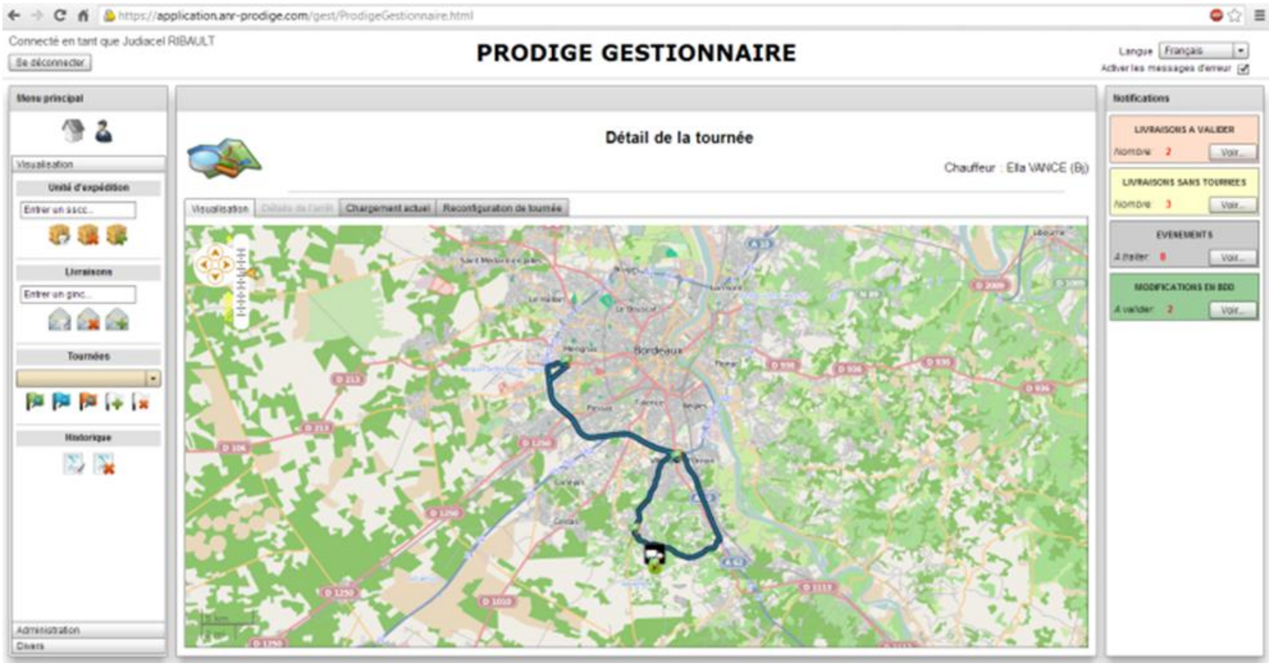

Figure 8 PRODIGE Web application.

ACKNOWLEDGEMENT

This research was partially funded by the PRODIGE project. PRODIGE is a French National Research Agency funded project (ANR-09-VTT-09-01) Nov@log labelled.

REFERENCES

Chandy, K. M., & Misra, J. (1979). *Distributed simulation: A case study in design and verification of distributed programs.* Software Engineering, IEEE, (5), 440-452.

Chen, D., Doumeingts, G. 2003. European Initiatives to develop interoperability of enterprise applications - basic concepts, framework and roadmap, *Journal of Annual reviews in Control*, 27, no.2, 151-160.

Giambiasi N., Escude B., Ghosh S. (2000) *GDEVS A Generalized Discrete Event Specification for Accurate Modeling of Dynamic Systems*. SCS Transactions, 17, no.3, 120-134.

Richardson, L., & Ruby, S. (2007). RESTful web services. O'Reilly Media, Incorporated.

Weske, M. (2012). Business Process Management Archi-tectures. Business Process Management, 333-371.

Ribault, J., & Wainer, G. (2012, March). Using workflows and web services to manage simulation studies (WIP). Proceedings of the 2012 Symposium on TMS/DEVS Integrative M&S Symposium (p. 50). SCS International.

Ribault, J., & Wainer, G. (2012, May). Simulation Processes in the Cloud for Emergency Planning. In Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 886-891). IEEE Computer Society.

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.

Tan, W., Missier, 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.

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

Kuhl, F., Dahmann, J., & Weatherly, R. (2000). *Creating computer simulation systems: an introduction to the high level architecture.* Prentice Hall PTR.

Al-Zoubi, K., & Wainer, G. (2010, December). *Managing simulation workflow patterns using dynamic service-oriented compositions.* Proceedings of the 2010 Winter Simulation Conference (WSC), (pp. 765-777). IEEE.

Al-Zoubi, K., & Wainer, G. (2010, December). *Rise: Rest-ing heterogeneous simulations interoperability.* In Simulation Conference (WSC), Proceedings of the 2010 Winter (pp. 2968-2980). IEEE.

Zacharewicz G., Deschamps J.C., François J., (2011). Distributed platform for advanced freight transportation systems, Special Issue on Grand Challenges for DELS, Computers in Industry 62, 6 597-612.