

## A Development of AWSCPM Framework for Automation of Web Services Composition Processes

<sup>1</sup>Nouha Adadi, <sup>1</sup>Mohammed Berrada, <sup>1</sup>Driss Chenouni and <sup>2</sup>Mohamed Halim  
<sup>1</sup>IPI Laboratory, Sidi Mohammed ben Abdellah University, Fez, Morocco  
<sup>2</sup>Couches Minces et Traitement de Surface par Plasma Laboratory,  
Sidi Mohammed Ben Abdellah University, Fez, Morocco

---

**Abstract:** The composition of web services that is the combination of several services to obtain new features becomes more and more popular and presents a necessary stage for the realization of the collaboration inter-companies (B2B). To implement this collaboration, a developer has to elaborate a specification which allows the modeling of the global behavior of the system to verify formally this model to assure the quality of the system then pass to the implementation of the composed service. This study presents a summary of our developed approach of web services composition this approach is separated into three tasks: specification using BPMN notation and Multi-Agent Reactive Decisional (MARDS) Model, formal verification using LOTOS language and implementation using BPEL language. Then this study will introduced the framework AWSCPM “Automatic Web Services Composition Processes based on MARDS”, it is developed as part of this research and that automates the generation of different processes: business process BPMN, executable process BPEL and formal process LOTOS.

**Key words:** Web services composition, framework AWSCPM, business process BPMN, executable process BPEL, formal process LOTOS, processes

---

### INTRODUCTION

Web services are defined as software components which can be invoked by application programs through a stack of internet standards. Once deployed, web services provided by various organizations can be inter-connected in order to implement business collaborations, leading to composite web services. Web service composition raises the need to provide the mechanism to fulfill the complexity of the execution of business processes. Several organizations are currently working on new service composition proposals and various approaches and formalism have been proposed and used for web services composition, among these approaches we have the Models Driven Approach (MDA) which concentrates on the realization of abstract models. Thus, the phase of specification represents an important part of the cycle of development of composite web service. To proceed to this cycle of development, a developer has to elaborate a specification which allows the modelling of the global behaviour of the system to verify formally this model for assuring his quality, then pass to the implementation of the composed service.

This study presents, first a summary of our developed approach of web services composition; second it exhibits the framework AWSCPM developed as this research constituent in order to automate the generation of processes in the three tasks of our approach: specification

using BPMN notation (Business Process Model and Notation) (Chand and Chircu, 2012) and MARDS Model, formal verification using LOTOS language (Richard and Jirachiefpattana, 1998) and implementation using BPEL language (Business Process Execution) (OASIS, 2007). As part of a case study, we consider the hospital information system as an illustrative example to apply the concepts of our approach.

### MATERIALS AND METHODS

**Developed approach:** In this study, we present a summary of our developed approach based on MDA and we explain the process of development of composite service. Figure 1 shows the steps involved in the development process (specification, formal verification and implementation) to better understand how to proceed.

**Phase of specification:** In the process presented in Fig. 1, once the requested services are selected by the directory we pass to the specification stage. At this level we propose a modelling based on MARDS Model and using the BPMN notation. The MARDS Model constitutes an approach among the newest and most useful ones for the composing and modeling of complex system (Aaroud *et al.*, 2005; Berrada *et al.*, 2007). We have used this system in our approach because it allows to model the composition of services in a simple and powerful way and

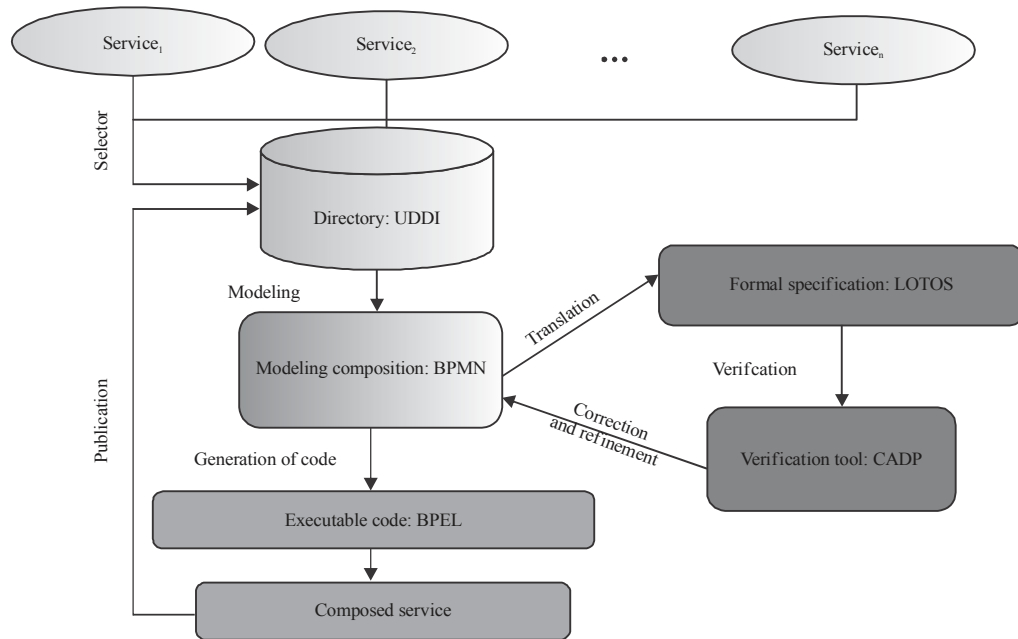


Fig. 1: Process of development of composite service

in well-structured architecture. The BPMN notation is a modeling language, it is more adapted to the domain of the web services, legible and sufficiently precise and expressive to allow the generation of executable code from it. We have used this notation for modeling the processes generated from the composed web services on orchestration mode. This modeling phase is described in detail by Adadi *et al.* (2014).

**Phase of verification and implementation:** This approach considers not only the specification of composite services but also their verification. As it is better to detect errors as early as possible in the cycle of development from the specification stage, the next step is the qualitative formal verification of our proposed model. Our specification is described by the BPMN notation but this language is often criticized for its lack of formality. One proposed solution is to transform the BPMN Model in formal specification. Any formal specification language is susceptible to agree but we propose the use of the process algebra LOTOS which has the advantage of being supported by free formal verification tools such as CADP (Garavel *et al.*, 2013) toolbox. Due to CADP, it is possible to validate automatically the behavioral properties. In case where errors are detected, the developer is responsible for correct and refine its model to arrive at a model proven correct. The formal verification step is the object by Adadi *et al.* (2016) where there is more detail and description.

When the composition model is validated, the next step is the implementation of the system by generating

BPEL code from the BPMN specification. Finally, once the composed service is implemented, the last step is usually to publish it in the directory to facilitate its future use. More details and descriptions of this step of implementation are gives by Adadi *et al.* (2015).

In this study we focus on the automatic generation of business model and automatic transformation of BPMN models into LOTOS Models and BPEL code. This research is important to facilitate the task of the developer and to make the steps of specification, formal verification and implementation simple, rapid and completely automatic. We will present in the next section the framework AWSCPM which allows the automatic generation of the different process.

**Presentation of the framework AWSCPM:** The AWSCPM is a framework developed as part of this research. The functionalities of AWSCPM can be grouped into three categories: ARD (Agent Reactive Decisional) management, MARDS management and appropriate process generation (business processes BPMN, BPEL processes and LOTOS processes) each category is presented by one or a set of interfaces. These functionalities are presented through the electronic management process in a hospital, called “e-Health” developed by Adadi *et al.* (2017).

**General description of the process “e-Health”:** The current challenge in the health sector is not the search for strategies for the implementation of information systems within hospitals but rather the search for the necessary

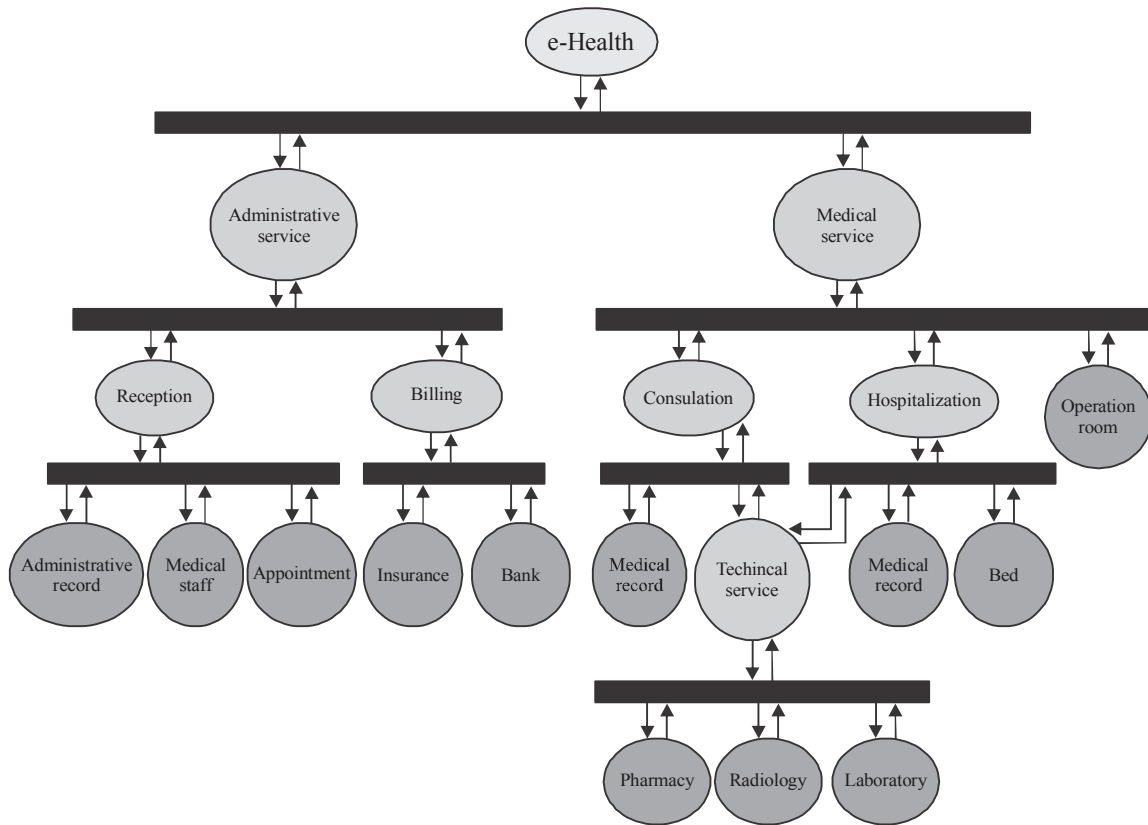


Fig. 2: Web service composition model based on MARDS

means to better govern the interoperability between these different systems and propose models for sharing patient data.

The data of the patient can be shared within the same institution between several units and even with different other institution, this sharing and this possibility of remote and electronic access to the data of the patient in order to solve his problems and meet his needs are summarized by the name “e-Health”.

Several scenarios of composition can be adopted according to the need, the expected result and the mode of circuit realized by the patient in the hospital. In this study, we present a classic scenario which can be useful to apply our approach of web services composition. We justify the choice of this simple scenario by the fact that it is capable (in a limited context) of tracing the process of the management of a patient in a sufficiently significant way without needing several services.

The process is triggered by the appointment request from the patient. Once the patient is admitted, the hospital takes of his charge. A medical secretary reveals the patient’s administrative record (if it exists if not he creates it) and leads the patient to consultation with the physician in charge with whom he made an appointment. In order to carry out its consultation the physician may need to access other services such as medical record service in order to

consult and update the patient’s data, status and its history, pharmacy service to help him to prescribe his medical notice, radiology and laboratory services to request analyses and radios and to receive the results. If it is necessary, other services and processes can be triggered after this consultation as the hospitalization and operating block services depending on the case and the need of the patient. In all cases and at each stage, billing service is necessary to automatically appreciate the benefits and consumptions of the patient. The edition of the invoices can intervene other services like insurance and bank.

## RESULTS AND DISCUSSION

**Web service composition model based on MARDS:** In this approach, it is necessary to define beforehand the Decisional Reactive Agents (DRA) which will participate in the development of each process. Then, it is possible to model these processes by the implementation of its agents in the framework AWSCPM.

Applying the rules and methods described by Adadi *et al.* (2014) to the “e-Health” scenario, we obtain the MARDS structure shown in Fig. 2.

In this model of service composition, the basic components are: “Administrative Record” (AR); “Medical Staff” (MS); “Appointment”; “Insurance”; “Bank”;

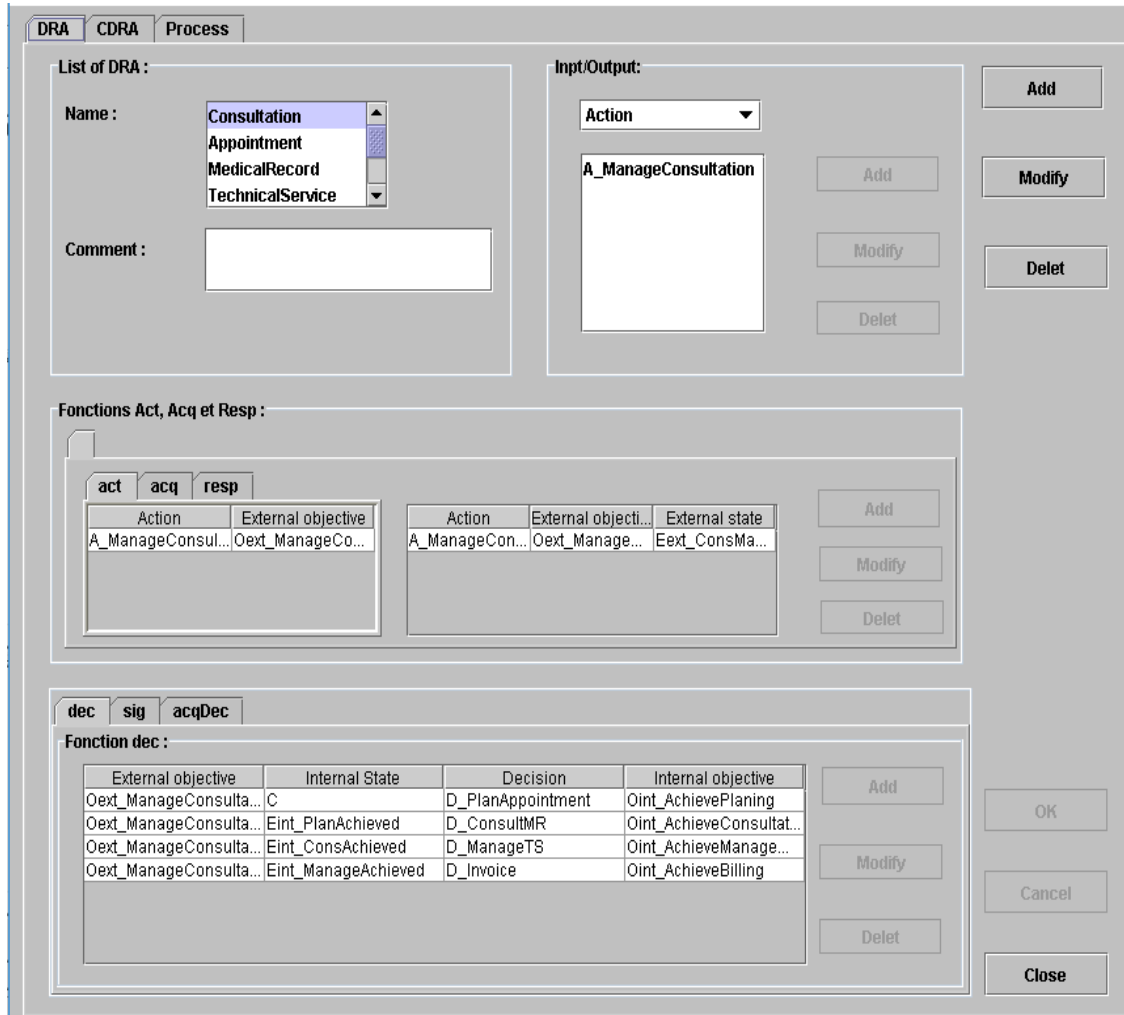


Fig. 3: DRA management interface

“Medical Record” (MR); “Radiology”; “Laboratory”; “Pharmacy”; “Bed” and “Operating Block” (OB). The intermediate components are: “Home”; “Billing”; “Consultation”; “Hospitalization”; “Technical Service” (TS); “Administrative Service” (AS) and “Medical Service” (MS). The main composite component is “e-Health”.

**DRA management:** The functionalities of this interface (Fig. 3) allow the realization of all the operations “addition, modification, deletion, validation” on the agents DRA. Besides, this interface is characterized by 4 panels.

The first panel displays all the agents DRA defined by the user. The second panel shows the internal/external events specific to every agent. These events are action, decision, signaling, external state, internal state, external objective and internal objective.

The third panel allows us to choose the events corresponding to the functions of action (act), acquittal of external objective (acq) and response to an action (resp).

The last panel allows us to complete the tables by the events corresponding to the functions of decision (dec), signaling (sig) and acquittal of decision (acqDec).

**MARDS management:** To specify new systems (MARDS agents), we have developed this interface which consists of three main panels (Fig. 4). The first panel displays the name and the type (composite or elemental) of each MARDS system. The second panel Displays the Agent Supervisor (DRAS). The last panel allows the user to choose the lower level agents as well as to fill the messages of the communication interfaces (decision and signaling).

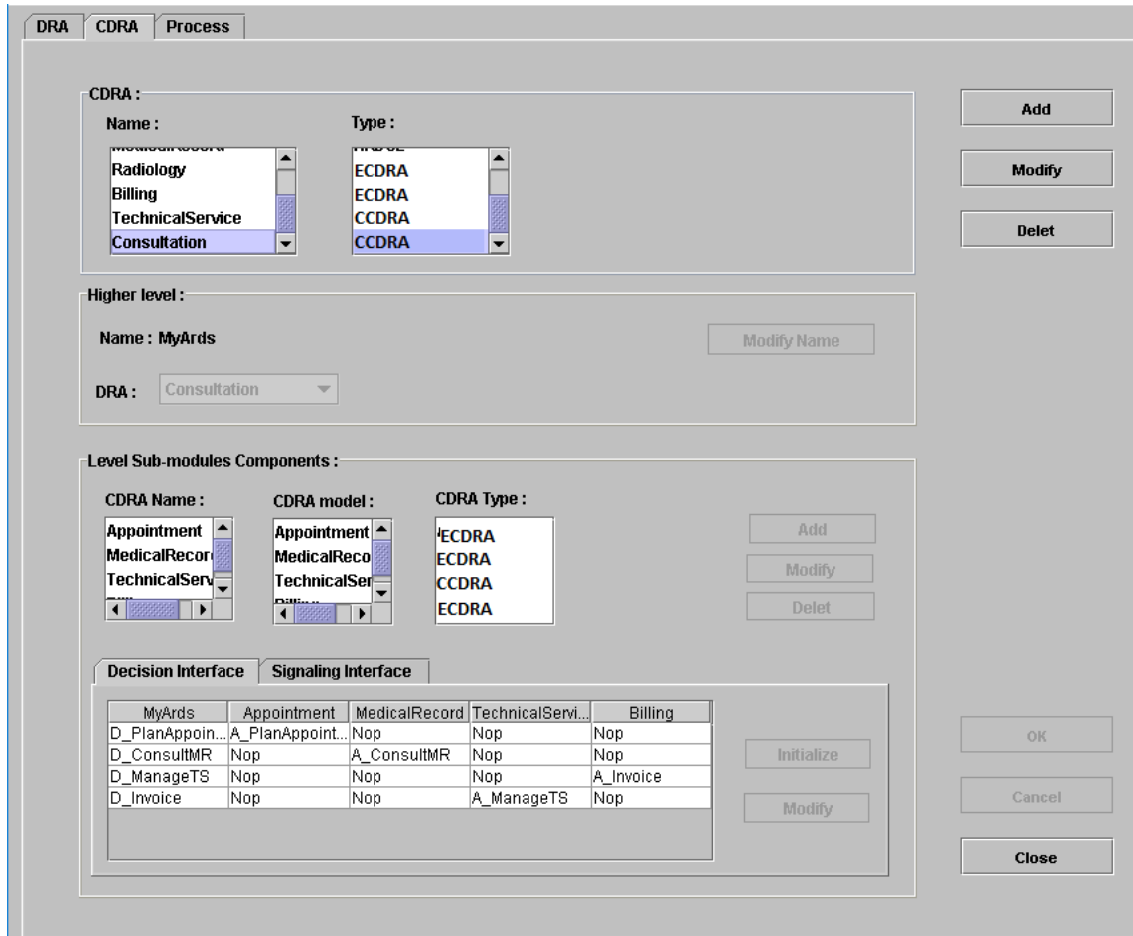


Fig. 4: MARDS management interface

**Process generation:** This interface (Fig. 5) is composed of two panels, the first panel allows us to choose the parameters to generate in the second panel the description of the MARDS system, the action and decision traces, the BPMN business process, the BPEL process and the LOTOS process. Each process is displayed in a different tab.

This tab (Fig. 5) displays the organizational structure of the “Consultation” MARDS. The hierarchy of this system is composed of 3 levels.

**The BPMN business model:** The business model displayed depends on the parameters chosen by the user. We present in the interface (Fig. 6) the model of the process “Consultation” with the parameters: “action = A\_ManageConsultation”; “internal state = C” and “level = 3”. This model displays the tasks completed by the agents participating in this process “Consultation” and belonging to the entire structure of the system (level = 3).

The action “A\_Manage Consultation” received by the “Consultation” component generates a decision

“D\_Achieve Consultation”. On his part, this decision generates four parallel sub-actions {A\_PlanAppointment; A\_ConsultMR; A\_ManageTS; A\_Invoice} for the “Appointment”; “MedicalRecord”; “TechnicalService” and “Billing” components. The two sub-actions correspond to the process of action “A\_Manage Consultation”.

The sub-action “A\_ConsultTS” received by the “Technical Service” component generates a sub-decision “D\_ManageTS”. From its role this sub-decision generates three parallel sub-actions {A\_Consult Pharmacy; A\_ConsultRadio; A\_ConsultLabo} for “Pharmacy”, “Radiology” and “Laboratory” components. The three sub-actions correspond to the sub-process of the “A\_ConsultTS” sub-action.

The sub-actions {A\_ConsultPharmacy; A\_ConsultRadio; A\_ConsultLabo; A\_ConsultMR; A\_PlanAppointment; A\_Invoice} received respectively by the basic components “Pharmacy”; “Radiology”; “Laboratory”; “MedicalRecord”; “Appointment” and “Billing” generate the external states

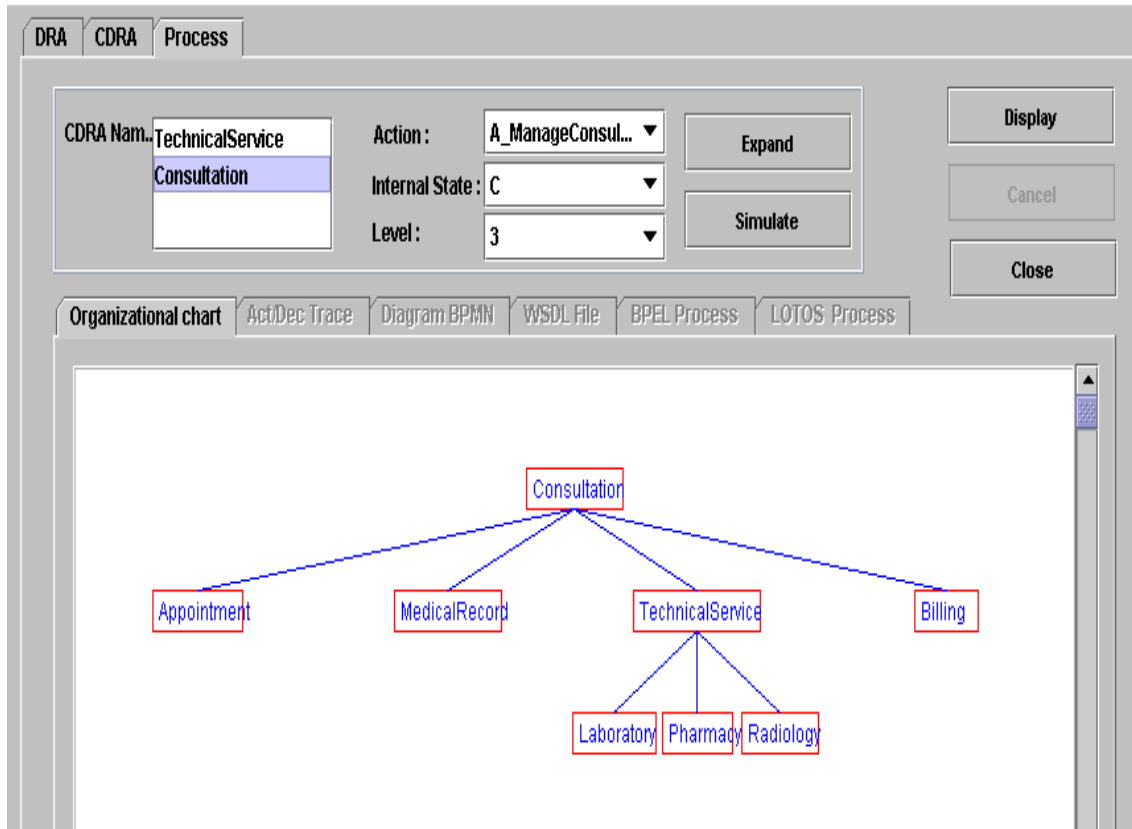


Fig. 5: Organizational structure of MARDS

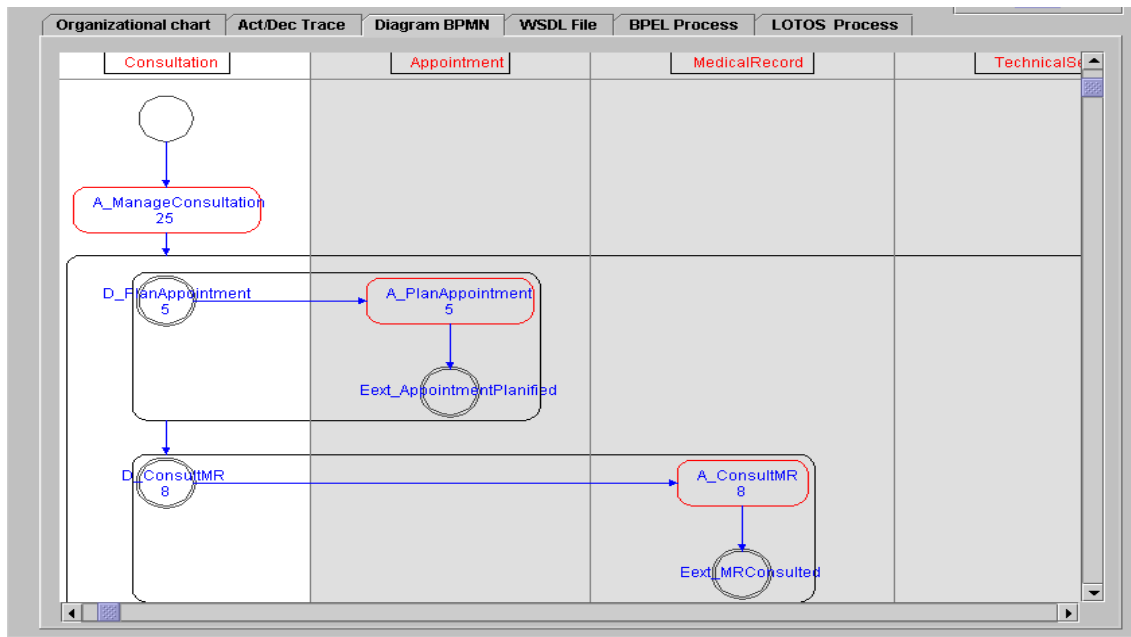


Fig. 6: BPMN modeling tab

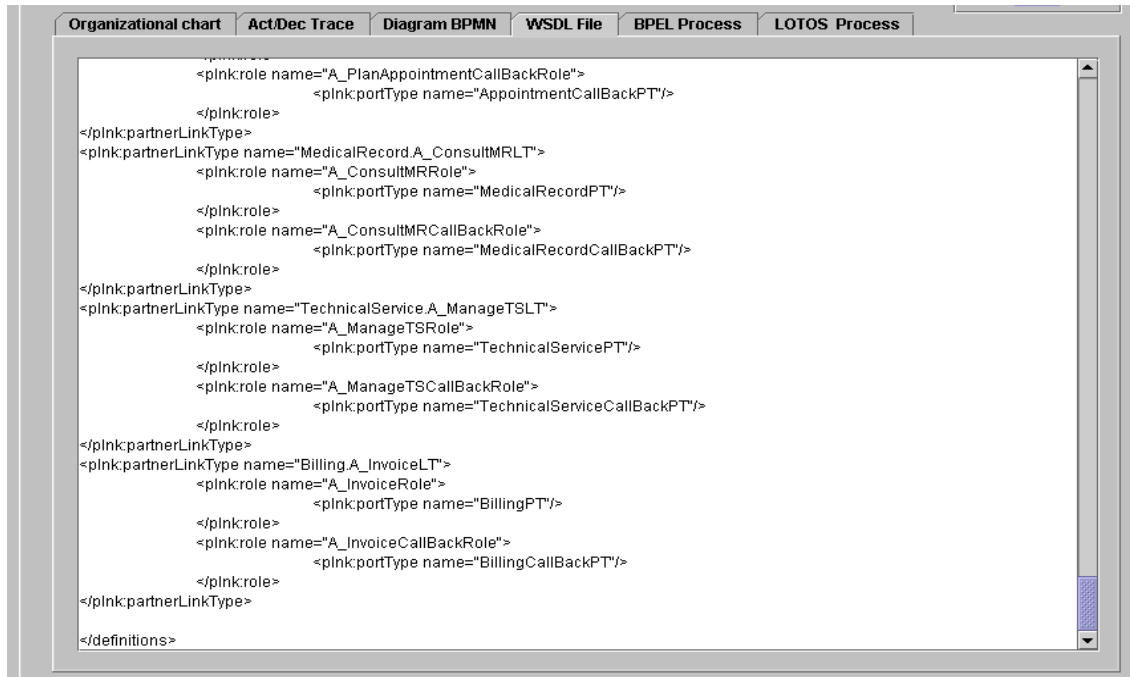


Fig. 7: WSDL source tab

{XS\_Pharmacy Results; XS\_Radio Results; XS\_Labo Results; XS\_MR Consulted; XS\_Appointment Planified; XS\_Expenses Paid }.

**BPEL translation:** The WSDL description and the BPEL file of the “Consultation” process are presented in the following two interfaces.

The interface (Fig. 7) defines all the messages, port type and partner link types associated with the different web services involved in the composition of the “Consultation” process

In the interface (Fig. 8), we describe the behavior of the main process corresponds to the system “e-Health” (agent composite of the first level) by the definition of the various partners participating in this process by the declaration of variables used by the operations defined in the appropriate WSDL interfaces and by the sequence of the different activities required to complete this process. This description should also include those of lower level sub-processes (level 2).

**LOTOS translation:** The formal LOTOS (Fig. 9) file of the “Consultation” process is presented in the following interface. We obtain this LOTOS specification by applying the rules and methods described by Adadi *et al.* (2016) to the BPMN Model presented in Fig. 6. Figure 10 summarizes all of the AWSCPM functionalities presented above.

**Comparison of AWSCPM with other BPM solutions:** Service orchestration tools are generally, part of workflow

applications and their presence in large numbers demonstrates the importance of development in this application domain.

These products are characterized by different levels of maturity. Thus, they come in the form of commercial solutions and/or software packages in open source software such as ActiveBPEL Enterprise [Active Endpoints] Intalio[Intalio]and Oracle BPEL process manager [Oracle Corporation].

We will not list here all the web services execution platforms but only present two platforms whose selection criteria are based on their availability, their state of development and the gravitation of community around these products.

**ActiveBPEL:** The application server included in the ActiveBPEL tools is a free licensed server. It is embellished with tools, developed by active endpoints of design and development of BPEL services that are paid for.

It is based on the Apache Tomcat server which is also free. Adding the ActiveBPEL server to Tomcat is relatively simple. In addition, once deployed, it has an administration interface that is fully usable by not only a web interface but also by web services provided with it. This interface enables the deployment and management of services on the server.

**BPEL process manager:** Oracle offers the BPEL Process Manager tool for designing and executing BPEL processes.



```
Organizational chart Act/Dec Trace Diagram BPMN WSDL File BPEL Process LOTOS Process
<copy>
    <from expression="A_ConstultRadio"/>
    <to variable="Action_Radiology" part="Message"/>
</copy>
</assign>
<invoke partnerLink="Radiology.A_ConstultRadio"
    portType="Radiology.A_ConstultRadio_PT"
    operation="A_ConstultRadio"
    inputVariable="Action_Radiology" />
<receive partnerLink="Radiology.A_ConstultRadio"
    portType="Radiology.A_ConstultRadioCallBack_PT"
    operation="A_ConstultRadioCallBack"
    Variable="EtatExterne_Radiology" />
</sequence>
</Flow>
<assign>
<copy>
    <from expression="Eext_TSManged"/>
    <to variable="EtatExterne_TechnicalService" part="EtatExt"/>
</copy>
</assign>
<invoke partnerLink="Client_TechnicalService.A_ManageTS"
    portType="TechnicalService.A_ManageTSCallBack_PT"
    operation="A_ManageTSCallBack"
    outputVariable="EtatExterne_TechnicalService" />
</sequence>
</process>
```

Fig. 8: BPEL source tab

The design tool, BPEL designer has the advantage of working directly on BPEL descriptions in native mode. It is therefore, the closest to the BPEL standard. It interfaces with known Java publishers such as Eclipse. The BPEL Execution Engine executes directly BPEL services with the backup of the server state in a database, a necessary step in a long interaction between multiple processes or services. This is a very good product, available under Windows and some Linux.

**Comparison of AWSCPM with active BPEL designer:** Business analysts and managers can create, model and deploy business processes using graphical tools included in the Active BPEL designer editor.

On the other hand, these managers can define agents, compose systems and generate business models and BPEL processes of the same business processes using the framework AWSCPM. However, generated models are composed of static and non-interactive objects.

For any change in these models users must first modify the specifications (the internal and external states) of the agents that make up the basic system.

However, this AWSCPM framework differs from the different BPM editors including active BPEL designer by the implementation of multi-agent systems and the automatic generation of LOTOS formal specifications which allows the verification and validation of the business model before its implementation.



```

where
process BranchementMultiple [ENV, REC] (Emt_Id:Int, dsts_Id:IntSet): exit :=
E[empty(dsts_Id)] -> exit
[]
[not(empty(dsts_Id))] ->
(let dst:Int=pick(dsts_Id) in
ENV !dst ! Emt_Id !RUN !void;
BranchementMultiple [ENV, REC](Emt_Id, remove(dst, dsts_Id))
)
endproc

where
process Sequence [ENVRadiology, RECRadiology] (Emt_Id:Int, dst_Id:Int): exit :=
ENVRadiology !dst_Id !Emt_Id !RUN !void; exit
endproc

processBilling [ENV0,REC0] (Id:Int) : exit:=

processLaboratory [ENV1,REC1] (Id:Int) : exit:=

processPharmacy [ENV1,REC1] (Id:Int) : exit:=

processRadiology [ENV1,REC1] (Id:Int) : exit:=
    
```

Fig. 9: LOTOS specification

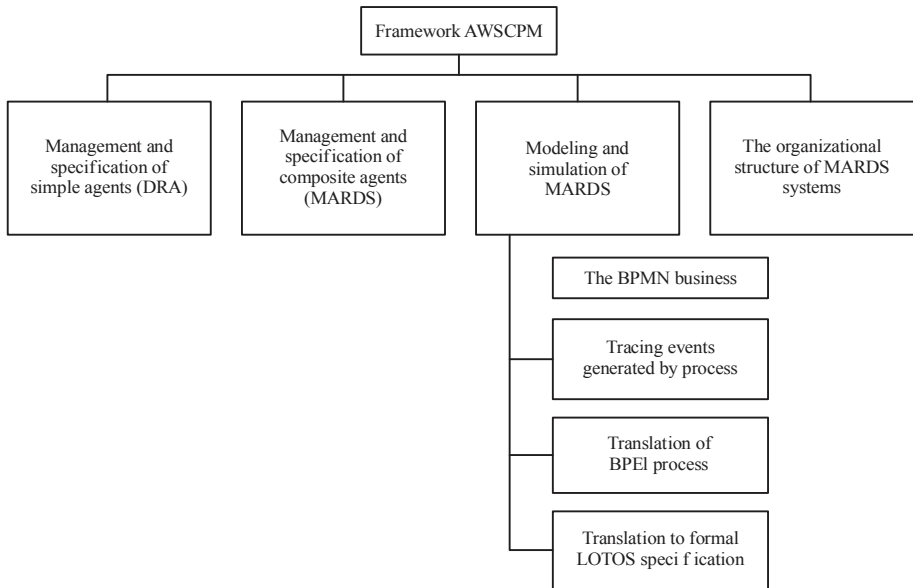


Fig. 10: Functional architecture of AWSCPM

**CONCLUSION**

This document consists of the presentation of the developed AWSCPM. The main functionalities of this framework are: the modeling of the business processes of a MARDS, the translation of these models to a formal specification LOTOS for the

verification of the composition system of web services, then to BPEL processes for the implementation of the system.

The use of this orchestration framework concerns designers, software developers and business analysts to automate the modeling, implementation and verification of the web services composition system.

**REFERENCES**

- Aaroud, A., S.E. Labhalla and B. Bounabat, 2005. Modelling the handover function of global system for mobile communication. *Int. J. Modell. Simul.*, 25: 99-105.
- Adadi, N., M. Berrada and D. Chenouni, 2016. Formal specification of web services composition using LOTOS. *Int. J. Comput. Technol. Appl.*, 7: 636-642.
- Adadi, N., M. Berrada, D. Chenouni and B. Bounabat, 2017. Proposition of web services composition approach basing of model-driven approach and multi-agent systems. *Comput. Modell. New Technol.*, 21: 12-19.
- Adadi, N., M. Berreda, D. Chenouni and B. Bounabat, 2014. Multi-agent architecture for business modeling of web services composition based on WS2JADE framework. *Int. Rev. Comput. Software*, 9: 1667-1674.
- Adadi, N., M. Berreda, D. Chenouni and B. Bounabat, 2015. Modeling and simulation of Web Services composition based on MARDS model. *Proceedings of the 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA'15)*, October 20-21, 2015, IEEE, Rabat, Morocco, pp: 1-6.
- Berrada, M., B. Bounabat and M. Harti, 2007. Modeling and simulation of multi-agent reactif decisionnal systems using business process management concepts. *Int. Rev. Comput. Software*, 2: 159-169.
- Chand, D.R. and A.M. Chircu, 2012. Business Process Modeling. In: *Business Enterprise, Process and Technology Management: Models and Applications*, Shankararaman, V., J.L., Zhao and J.K. Lee (Eds.), IGI Global, Pennsylvania, USA., pp: 32-50.
- Garavel, H., F. Lang, R. Mateescu and W. Serwe, 2013. CADP 2011: A toolbox for the construction and analysis of distributed processes. *Int. J. Software Tools Technol. Transf.*, 15: 89-107.
- OASIS., 2007. Web services business process execution language version 2.0. *Advancing open standards for the information society*, Burlington, Massachusetts, USA. <https://scholar.googleusercontent.com/scholar?q=cache:izRh9S01M2EJ:scholar.google.com/+OASIS+Standard.+Web+services+business+process+execution+language+version+2.0&hl>
- Richard, L. and A. Jirachiefpattana, 1998. *Communication Protocol Specification and Verification*. Vol. 464, Springer, Boston, Massachusetts, USA., ISBN-13: 978-0792382843, Pages: 304.