

Web Services Composition in E-Learning platform

Mohamed HALIM¹, Nouha ADADI², Driss CHENOUNI³, Mohammed BERRADA⁴

¹IPI Laboratory, Sidi Mohamed Ben Abdellah University -Fez, Morocco, mohamed.halim@usmba.ac.ma

²IPI Laboratory, Sidi Mohamed Ben Abdellah University -Fez, Morocco, nouhaadadi@gmail.com

³IPI Laboratory, Sidi Mohamed Ben Abdellah University -Fez, Morocco, Morocco, d_chenouni@yahoo.fr

⁴IASSE Laboratory, Sidi Mohamed Ben Abdellah University -Fez, Morocco, Morocco, mohammed.berrada@gmail.com

ABSTRACT

Since the emergence of Service Oriented Architecture and its implementation with the Web services technology, combining several Web services to response to a complex request presented a hard challenge. This is an area that has attracted the interest of many research organizations and manufacturers. This paper firstly describes the developed approach of web services composition based on Model Driven Approach, and using different languages and systems, like Multi-Agent Reactive Decisional system for the modeling task and Business Process Execution language for the implementation task. Secondly, the paper seeks to presents the application of this approach in the online learning domain. In brief, this document proposes the design and implementation of an e-learning platform based on Web services composition.

Key words: Web services composition, Model Driven Approach, Multi-Agent Reactive Decisional system, e-learning platform.

1. INTRODUCTION

Many organizations and institutions use e-learning because it can be as effective as traditional training at a lower cost. Indeed, the costs of broadcasting an e-learning course (including fees for Web servers and technical assistance) are considerably lower than those which involve the use of classrooms, the remuneration of trainers, the funding of participants' travel and the time lost to attend training sessions. In addition, e-learning allows to reach a wider audience, by integrating learners who have difficulties to participate in conventional training. As a result, e-learning applications is becoming more and more popular, there are now schools operating only in e-learning, and delivering a diploma recognized by the state. Hence the need to develop this domain and use new and powerful technologies to improve its performance and increase profitability.

The evolution of the Internet and the competitiveness between companies have been factors of the explosion of Web services. Indeed, Web services can be a contribution of speed and efficiency for e-business. The concept of Web service means an application made available on the Internet by a provider, and accessed by customers through standard

Internet protocols. Their peculiarities, compared to other distributed computing technologies, is that they offer a model of weakly coupled components using Internet technology as the infrastructure for communication. The composition of Web services is a natural evolution of this technology that allows the integration of applications on the web to achieve better solutions.

The objective of this work is to offer an e-learning application based on web services composition concepts to take advantage and reuse existing services in the web.

2. PROPOSITION OF WEB SERVICES COMPOSITION APPROACH

A growing number of companies are using Web services to make their expertise and data available through the network. The current problem is the integration of these services in order to implement inter-company collaboration. Constructing a composition approach is not a trivial task and presents the subject of many research work. The approaches presented in the literature to solve this problem can be grouped into four classes: approaches by planning techniques [1], [2], workflow approaches [3] [4], approaches by dependence graphs [5], [6] and model-driven approaches (MDA) [7]. All of these approaches differ in how they perceive the composition of Web services and use techniques that are compatible with that vision. In particular, they apply techniques from artificial intelligence, software engineering, graph theory or specification. Certainly these approaches can all generate composition plans and they are mature and equivalent if we consider their degree of adaptation to Web service composition problem solving and the proven maturity of the techniques they use. We find that all these approaches can generate composition plans. In our work we choose to adopt the model-driven approach. Certainly the other approaches allow to generate composition plans, however their major limitation lies in the dependence of proposed composition solutions on the semantic models of Web services description, as well as the level of composability considered which is often limited to the matching of input and output services, and not treating the different facets of verification between two or more services to connect.

In our work we adopt the model-driven approach, 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. In this section, we pre-sent a summary of our developed approach based on MDA. The figure 1 shows the cycle of development of the approach.

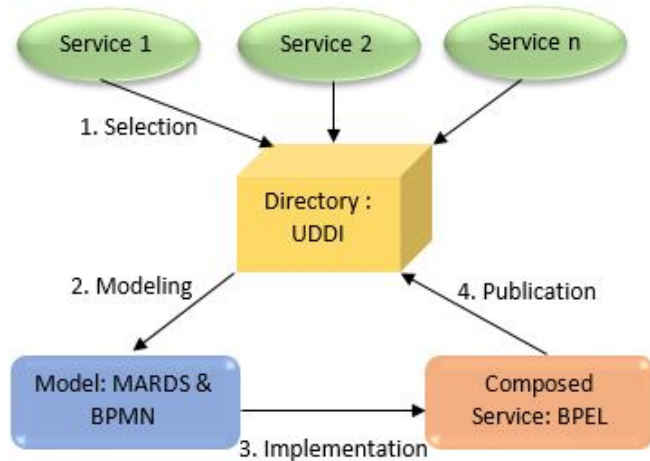


Figure 1: Cycle of development of proposed approach of web services composition

2.1. Phase of Modeling

The specification part consists in modeling the business processes of the composition of the web services. These processes can be described using modeling languages associated with the BPM concept. This modeling becomes complex and unstructured when services number is increasing. Indeed, the implementation phases become difficult to achieve. Multi-agent systems are particularly suitable for modeling composite and complex information systems. They are composed of autonomous agents that coordinate and interact to achieve their intentions. This makes them a good solution for the problem of web services composition, they provide a well-structured architecture that allows services to be composed in a simple, powerful and transparent way, by adding and removing services without affecting other services. Some multi-agent models, such as the MARDS [8], have a well-structured hierarchical architecture and can be used to model business processes in a simple, powerful and transparent way to facilitate constraint checking and generation of executable code. This specification phase is therefore based on the MARDS model while using the BPM standards and especially the BPMN [9]. It is called BPMN-MARDS profile. This profile is designed to improve the level of expressiveness of models in complex systems such as automated production systems, mobile systems [10] and the organization system [11]. This profile is customized for the domain of web services composition in order to facilitate their understanding as well as their transformation into code such as BPEL [12].

The specification phase of our approach is based on the MARDS model and using BPMN standards. It is called BPMN-MARDS profile. In the following we present the prototype modeling of MARDS by BPMN.

A. BPMN-MARDS profile:

The business model shown in Figure 2 represents a prototype model for any MARDS agent. The initial action generates several decisions {dec_i, i = ... m} each of which must trigger one or more actions {act_{ki}} intended and completed in parallel by the agents {MARDS_k}.

So, any decision represents the starting event; actions are parallel activities; external states are intermediate events; the signaling translated by the signaling interface, is intended for the supervisor as a final event. All these interrelated elements, as they are described in sequence, compose an independently modeled sub-process.

Thus, the modeling of the last sub-process, triggered by the last decision, will be followed by the final event that represents the final external state of the base process.

Concerning the activities achieved by {MARDS_k} agents, they can be modeled as simple tasks or sub-processes according to the nature of the agents in question. For a composite MARDS_k agent, it uses other sub agents to perform an activity that it will be modeled as a sub process. In addition, an activity performed by a simple DRA agent will be modeled by a simple task.

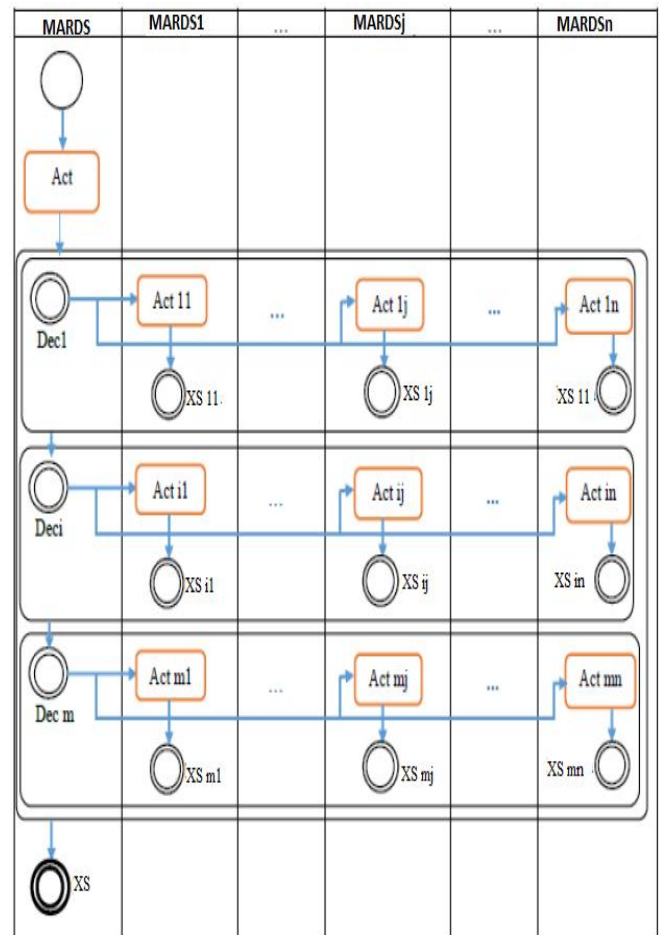


Figure 2: Business model of MARDS agent

2.2. Phase of Implementation

When the composition model is achieved, the next step is the implementation of composed Web service. Our work is to simulate the MARDS using the BPEL. This simulation is achieved by the translation of the BPMN-MARDS model into BPEL. The structure of this simulation is divided into two parts, the definition part (WSDL interface) and the process part (BPEL file).

A. WSDL interface:

WSDL interface describe the services as a set of abstract messages and operations, connected network servers. This description contains operational information about the service such as:

- Definition of messages: Input message (action: text) and output message (external state: text).
- portType: A first portType "MARDS_PT" which groups operations "action_i" to receive requests. A second portType "MARDSCallBack_PT" that groups operations "action_iCallback" to respond to requests.
- partnerLinkType. A partnerLinkType "MARDS_action_i_LT" for each action with two roles: "action_i_Role" and "action_iCallback_Role".

B. BPEL file

The BPEL file include the definition of elements (variables, partnerLink, initial activity *receive* and last activity *invoke*) and the description of the process and the sequence of his actions.

```
<Flow>
<!-- Realization of the action Actionp by the agent MARDSi -
>
<sequence>
<!-- Initialization of the variables -->
<assign>
<copy>
<from expression="Actionp"/>
<to variable="Action_MARDSi" part="action"/>
</copy>
</assign>
<invoke partnerLink="MARDSi.Actionp"
portType=" MARDSi_PT"
operation=" Actionp"
inputVariable= "Action_MARDSi" />
<receive partnerLink=" MARDSi.Actionp "
portType=" MARDSCallBack_PT "
operation=" ActionpCallBack"
Variable= "XS_MARDSi"/>
</sequence>
<!-- Realization of the action Actionk by the agent MARDSj ->
<sequence> ...
</sequence>
...</Flow>
```

3. CASE STUDY: E-LEARNING

Our approach of web services composition is based on powerful, mature and standardized techniques and languages;

consequently it can solve problems of web services composition in different domains of application. We propose the learning sector due to the spread of this method of learning in the last years, and its need for development to improve the quality and efficiency of its services. The development of giant, complex and robust application while keeping the cost low is one of the main issues of online learning. We believe that current computing means can be used to define a teaching that adapts, to a certain extent, to existing web services [13], [14].

3.1. Functional specifications

Our modeling presented in the use case (Figure 3) is composed of three main actors: learners, teachers and administrators.

A. Learner

Once the learner has registered, he can consult the courses and exercises that are offered to him and which correspond to his level. Pass tests and at the end pass an exam to validate the training. The learner has the opportunity to participate in the forums, send an email to a teacher or another learner, and attend a virtual classroom (learners can see the screen of their tutor where he explains and shows in more detail the important points of the course).

B. Teacher

As you can see on the use case diagram (Fig 3), teachers have multiple roles. The author, the tutor and manager. The author is responsible for putting his courses, exercises and quiz online. The tutor can help learners by talking to them through a forum on the platform or via a virtual classroom. As for the manager, he poses the final exam of the training and manages the results of the learners, he is considered as the supervisor of other teachers since he can accept or modify teachers and judge the effectiveness of their courses from the results of the learners.

C. Administrator

As teachers, administrators can have multiple roles. The technical manager, the financial manager, the schooling manager and the personnel manager. We chose to have several categories of administrator in order to have a better organization and thus to better manage access to the data present in the system. The Technical manager is primarily responsible for the technical part of the platform. The financial manager is responsible for settling learner payments and teacher salaries, preparing budgets and their follow-ups, negotiating purchases, and managing treasury, accounting and relations with banks. As for the schooling manager, his role is important, especially for management of learner registrations, deliver of diplomas management of virtual classroom sessions schedule. For the personnel manager, he is responsible for recruiting teachers and delivering staff attestations and contracts.

D. Adapter

The adapter is the central role of this system. He adapts the courses according to the profile of the learner, and consults

the results of a test performed by the new student, for him offers a suitable course with his level.

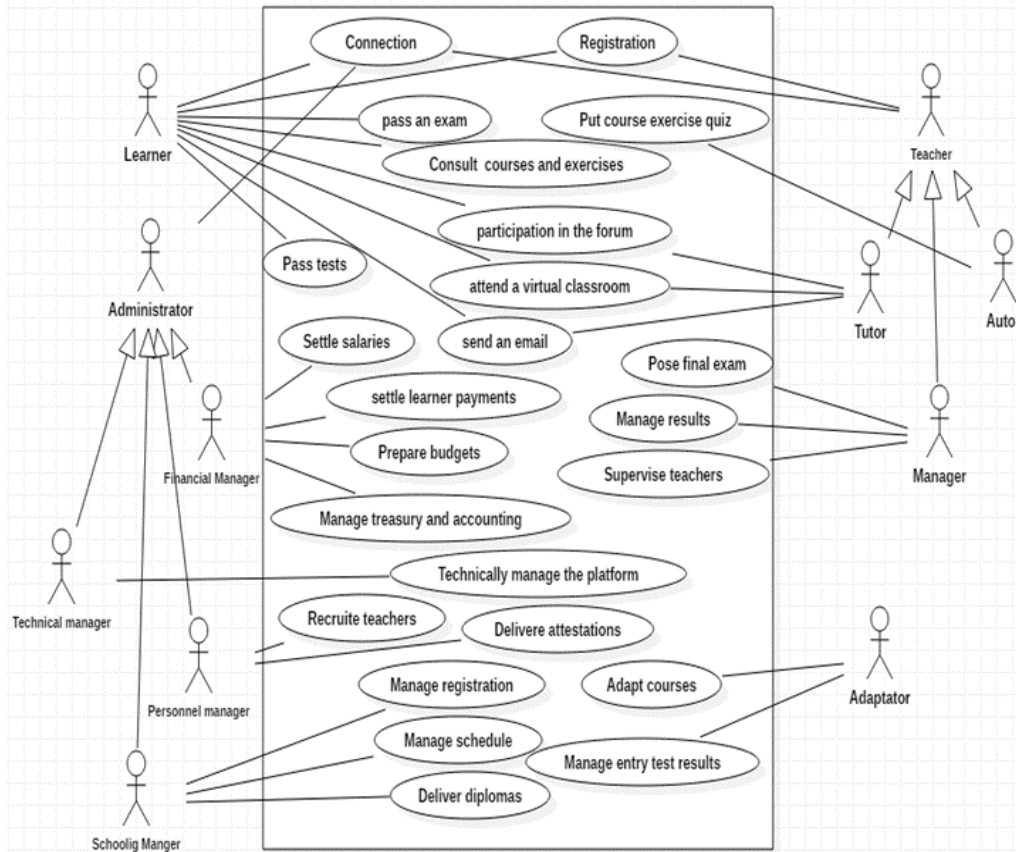


Figure 3: Use case diagram of the e-learning system

3.2. Modelling Phase

A. Structure of Web services composition

To model the functional specification, presented in the previous section, using the MARDS model we will follow the following steps:

- Organize agents in layers according to the activities that they execute.
- Identify atomic agents which execute simple and basic tasks (DRA agents).
- Specify the intermediate composite agents (MARDS agent) of each layer.

- Identify the first and main composite agent which receives the principal composite request from client.

Applying these steps to the e-learning specification, we obtain the MARDS structure of e-learning composite service shown in figure 4.

In this model of service composition, the basic components are: ("Technical service", "Inscription", " Schedule", " Diplomas", " Economic service", " Personnel service", " Course/ Exercises", " Quiz", " Forum ", " Virtual class", " Assessment" and "Adaptation". The intermediate composite components are: "Administration"; "Schooling service"; " Teaching"; " Training" and " Communication". The main composite component is "E-learning".

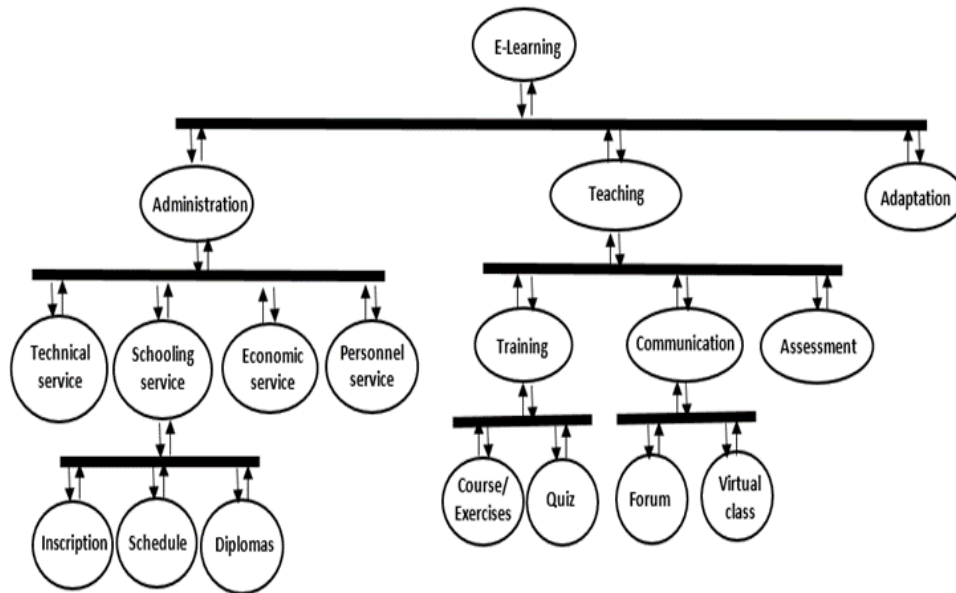


Figure 4: Web services composition model based on MARDS

B. Business Model of web services composition

Figure 5 and 6 shows the BPMN- MARDS profile of the composite service E-learning.

The main composite component “E-learning” receive the action “A_Online learning” and generates three decisions “D1_ Manage Administration”; “D2_ Teaching” and “D3_Manage adaptation”. Each decision triggers one or more actions intended and completed concurrently by MARDS agents “Administration”, “Teaching” and “Adaptation”. So, any decision represents the starting event; actions are parallel activities; external states are intermediate events; and the signaling translated by the signaling interface, which is intended for the supervisor as a final event. All these interrelated elements compose an independently modeled sub-process.

In the sub process “Administration”, the component “Administration” receive sub-action “A_Manage Administration” and generates four sub-decisions “D1_ Manage Schooling”; “D2_Manage the platform”; “D3_Manage financing” and “D4_Manage personnel”. These sub-decisions generate respectively the sub-actions “A_Manage Schooling” for the “Schooling service” component, “A_ Manage the platform” for the “Technical service” component, “A_Manage financing” for the “Economic service” component, and “A_ Manage personnel” for the “Personnel service” component. The sequencing of the four sub-decisions corresponds to the sub-process of the “A_Manage Administration” sub-action.

In the sub process “Teaching”, the component “Teaching” receive sub-action “A_Teach” and generates three sub-decisions “D1_Manage discussion”; “D2_ Manage training” and “D3_Manage assessment”. These sub-decisions generate respectively the sub-actions “A_Manage discussion” for the “Communication” component, “A_Manage training” for the “Training” component and “A_Achieve exam” for the “Assessment” component. The sequencing of the three sub-decisions corresponds to the sub-process of the “A_Teach” sub action.

In the sub process “Schooling”, the component “Schooling service” receive sub-action “A_Manage Schooling” and generates one sub-decisions “D_ Manage Schooling” which generates the sub-actions “A_Register” for the “registration” component, “A_Plan the timetable” for the “Schedule” component and “A_Deliver diplomas” for the “Diplomas” component.

In the sub process “Training”, the component “Training” receive sub-action “A_Manage training” and generates one sub-decisions “D_ Manage training” which generates the sub-actions “A_Consult course” for the “Course” component, “A_Pass quiz” for the “Quiz” component.

In the sub process “Communication”, the component “Communication” receive sub-action “A_Manage discussion” and generates one sub-decisions “D_Communicate” which generates the sub-actions “A_Communicate by Forum” for the “Course” component, “A_Communicate by virtual class” for the “Virtual class” component.

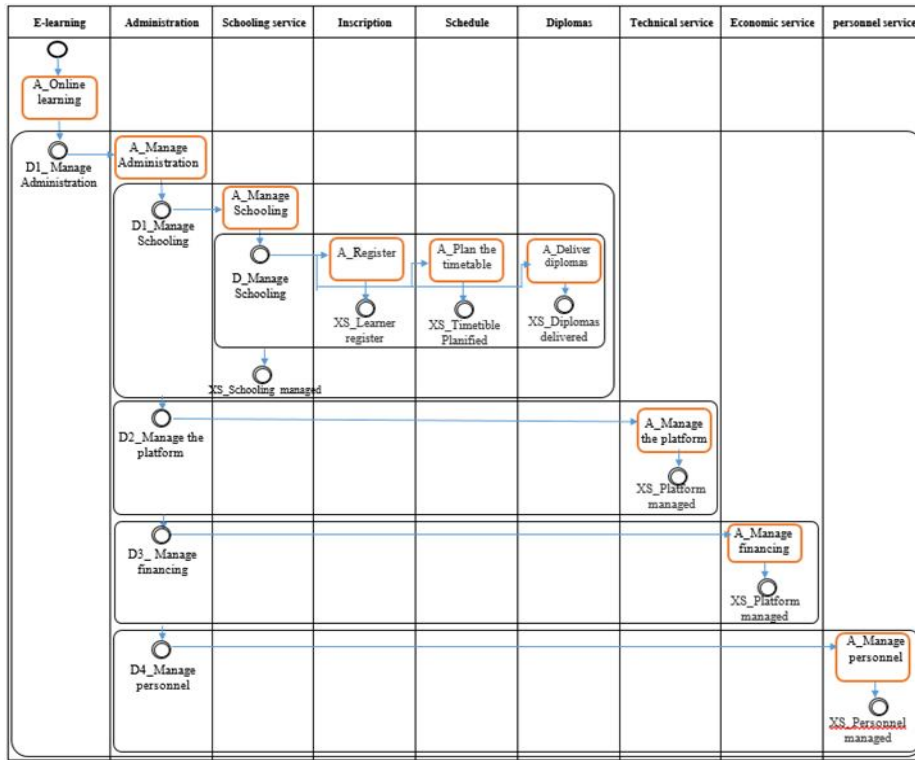


Figure 5: BPMN Diagram of "e-learning" Composition Scenario (Part 1)

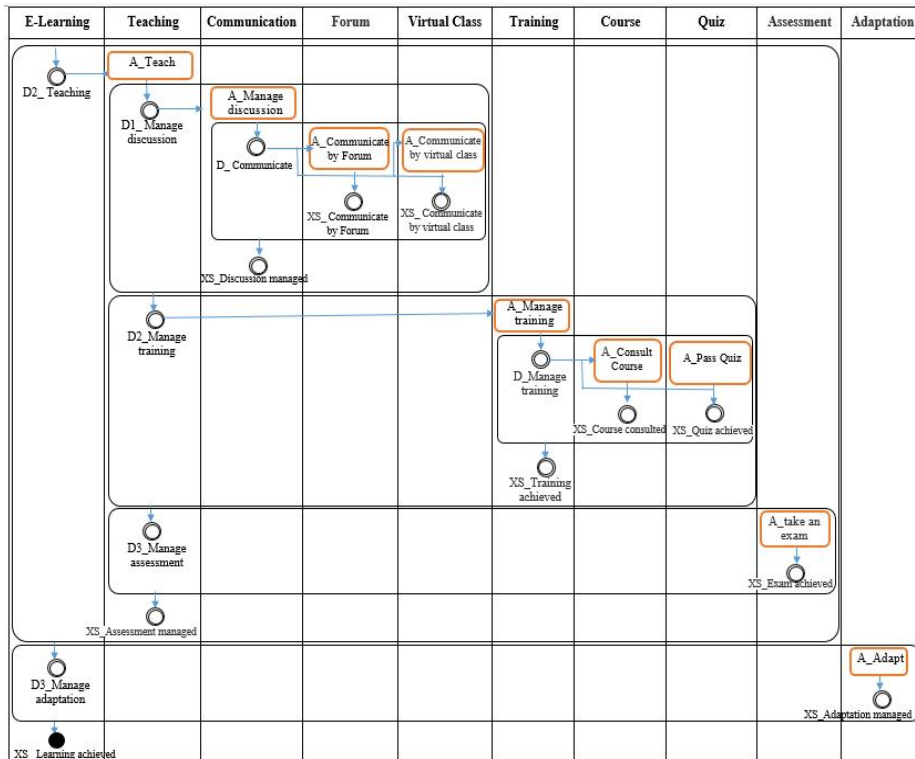


Figure 6: BPMN Diagram of "e-learning" Composition Scenario (Part 2)

3.3. Phase of implementation

Applying the methods described in Section 3.2 to the BPMN models presented in Figures 5 and 6, we obtain the following BPEL process.

```

<process name= "Online_Learning" process>
<!-- declaration of PartnerLinks -->
<PartnerLinks>
<PartnerLink name="E_Learning.A_Onlinelearn"
partnerLinkType="E_Learning.A_Onlinelearn_LT"
myRole = "A_Onlinelearn_Role"
partnerRole = " A_OnlinelearCallabck_Role"/>
<PartnerLink name = "Administra-
tion.A_ManageAdministration"
partnerLinkType = "Administra-
tion.A_ManageAdministration_LT"
partnerRole = " A_ManageAdministration_Role"
myRole="A_ManageAdministrationCallabck_Role" />
...
</PartnerLinks>
<!-- Declaration of variables -->
<variables>
<!--input and output for E-learning process-->
<variable name="Action_E-learning"
messageType="StartState"/>
<variable name="ExternalState_E-learning"
messageType="EndState"/>
<!-- input and output for Administration process -->
<variable name="A_ManageAdministration"
messageType="StartState"/>
<variable name="ExternalState_Administration"
messageType="EndState"/>
<!-- input and output for Teaching process -->
<variable name="A_Teaching"
messageType="StartState"/>
<variable name="ExternalState_Teaching"
messageType="EndState"/>
...
</variables>
<!--Definition of the BPEL process main body -->
<sequence>
<receive partnerLink= "E-
Learning.A_Onlinelearn"
portType= "E-Learning.A_Onlinelearn_PT"
operation= "A_Onlinelearn "
Variable= "Action_E-Learning"
CreateInstance="Yes" />
<!-- Decision Name =D_ManageAdministration-->
<Flow>
<!-- Action = "A_ManageAdministration",Agent =
Administration-->
<sequence>
<assign>
<copy>
<from expression=
"A_ManageAdministration"/>
<to variable="Action_Administration" part="Message"/>
</copy>
</assign>
<invoke partnerLink=
"Administration.A_ManageAdministration"

```

```

portType="
Administration.A_ManageAdministration_PT"
operation="A_ManageAdministration"
inputVariable="Action_Administration"/>
<receive partnerLink=
"Administration.A_ManageAdministration"
portType=" Administration.
A_ManageAdministrationCallBack_PT"
operation="A_ManageAdministrationCallBack"
Variable="ExternalState_Administration" />
</sequence>
</Flow>
<!-- Decision Name = D_Teaching -->
<Flow> ... </Flow>
<!-- Decision Name = D_ManageAdaptation -->
<Flow> ... </Flow>
<invoke
partnerLink="E_Learning.A_Onlinelearn"
portType= "E_Learning.A_OnlinelearnCallBack_PT"
operation="A_OnlinelearnCallBack"
outputVariable="ExternalState_E-Learning"/>
</sequence>
<!--Definition of Administartion process -->
<sequence>...
</sequence>
<!--Definition of SchoolingService process -->
<sequence>...
</sequence>
<!--Definition of Teaching process -->
<sequence>...
</sequence>
<!--Definition of Communication process-->
<sequence>...
</sequence>
<!--Definition of Training process-->
<sequence>
<receive
partnerLink="Training.A_ManageTraining"
portType="Training.A_ManageTraining_PT"
operation="A_ManageTraining"
Variable="Action_Training"
CreateInstance="Yes" />
<!-- Decision Name = D_ManageTraining -->
<Flow>...
</Flow>
<invoke
partnerLink="Training.A_ManageTraining"
portType="Training.A_ManageTraining CallBack_PT"
operation="A_ManageTraining"
outputVariable="ExternalState_Training" />
</sequence>
</process>

```

4. CONCLUSION

Web services technology has brought several benefits to the field of e-learning such as integrating content into learning systems as services, making maintenance easier because it is stored once on the provider server, and learners connect to the same content. Updates of this content are done only once, and the configuration of the platforms according to the user needs and their functionalities can be reused in different

platforms. This justifies the continued growth of the number of e-learning services on the web, hence the need for the composition of different services to respond to a complex query.

In this work, an approach of Web services composition is proposed. It is a model-driven approach that is faithful to the MDA principles. Therefore, the cycle of development of our approach is composed of the specification and implementation phases. The specification phase insists on modeling the business processes of web services composition. These processes have been described using BPM standards and especially BPMN notation and using the SMARD model. The implementation phase consists of generating the BPEL executable code from the specification. For the perspectives, we plan to consider the dynamism of the composition. In this work, we consider only the static composition of services. The services are selected as early as the specification step and the composition scenario is predetermined. We plan to extend our approach to the paradigm of dynamic service composition. Indeed, only the composition scenario will be predefined at the time of specification, the services are discovered and selected dynamically at runtime through discovery mechanisms of the UDDI directory [15] and semantic Web description languages such as OWL-S.

REFERENCES

1. R. Ben Lamine, R. Jemaa and I. Amous, "Graph Planning Based Composition For Adaptable Semantic Web Services", *Procedia Computer Science*, vol. 112, p. 358-368, September, 2017.
<https://doi.org/10.1016/j.procs.2017.08.016>
2. Praveen Kumar Kollu and R. Satya Prasad, "Intrusion Detection System Using Recurrent Neural Networks and Attention Mechanism", *International Journal of Emerging Trends in Engineering Research (IJETER)*, vol. 7, no. 8, pp 178 - 182, 2019.
<https://doi.org/10.30534/ijeter/2019/12782019>
3. J. Zhang, Q. Bao, X. Duan, "Collaborative Scientific Workflow Composition as a Service: An Infrastructure Supporting Collaborative Data Analytics Workflow Design and Management", In *Proc. 2nd IEEE International Conference on Collaboration and Internet Computing*, Pittsburgh, PA, pp. 219-228, 2016.
4. K. Velin, K. Radoslava, S. Nin, " An Analysis of a Web Service based Approach for Experimental Data Sharing", *International Journal of Online Engineering (iJOE)*, vol. 14, no. 9 pp. 19, September 2018.
5. H. Elmaghraoui, L. Benhlima and D. Chiadmi, "Dynamic web service composition using AND/OR directed graph", In *Proc. 3rd International Conference of Cloud Computing Technologies and Applications*, Morocco, pp. 1-8, 2017.
6. N. Arch-int, S. Arch-int, S. Sonsilphong and P. Wanchai, "Graph-Based Semantic Web Service Composition for Healthcare Data Integration", *Journal of Healthcare Engineering*, vol. 2017, pp. 1-19, August 2017.
7. D. Sferruzza, J. Rocheteau, C. Attiogbe, A. Lanoix, "A Model-Driven Method for Fast Building Consistent Web Services in Practice", In *Proc. 6th International Conference on Model-Driven Engineering and Software Development*, Portugal, 2018, pp. 15-24.
<https://doi.org/10.5220/0006531900150024>
8. B. Bounabat, "Méthode d'analyse et de conception orientée objet décisionnel. Application aux langages synchrones et aux systèmes répartis", doctoral thesis, Cadi Ayyad University, Faculty of sciences, Marrakech, Morocco, 2000.
9. BPMI.org, "Business Process Modeling Notation Specification", *OMG Final Adopted Specification*, 2006.
10. A. Aaroud, S. E. Labhalla, B. Bounabat, "Modelling the handover function of global system for mobile communication", *International Journal of Modelling and Simulation*, vol. 25, no. 2, pp. 99-105, 2005
11. M. Berrada, "Business Modeling of Enterprise Architecture Based on Multi-Agent System", *International Journal of e-Education, e-Business, e-Management and e-Learning*, vol. 3, no. 6, December 2013.
<https://doi.org/10.7763/IJEEEE.2013.V3.281>
12. C. Barreto, V. Bullard and T. Erl, "Web services business process execution language version 2.0", *OASIS*, April 2007.
13. Arba Asha Altaye and Dr. J. Sebastian Nixon, " A Comparative Study on Big Data Applications in Higher Education", *International Journal of Emerging Trends in Engineering Research (IJETER)*, vol. 7, no. 12 pp. 739 – 745, 2019.
<https://doi.org/10.30534/ijeter/2019/027122019>
14. A. Vijaya Lakshmi, K. V. T. Nagendra Babu, "A Machine Learning based Approach for Defect Detection and Characterization in Non-Linear Frequency Modulated Thermal Wave Imaging", *International Journal of Emerging Trends in Engineering Research (IJETER)*, vol. 7, no. 11 pp. 517 - 522, 2019.
<https://doi.org/10.30534/ijeter/2019/197112019>
15. T. Bellwood, L. Clément, D. Ehnebuske, "UDDI Specification v3.0.2", *OASIS*, February 2005. Accessed on: Dec. 8, 2019. [Online]. Available: <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>