Context Aware Management Plateform to Invoke Remote or Local e Learning Services: Application to Navigation and Fishing Simulator

Valérie Monfort and Fayssal Felhi

Abstract. Many companies aim to use Web services to integrate heterogeneous and /or remote application in SOA (service oriented architecture) contexts. The SaaS (Software as a Service) economical model allows to link service consumption and pricing. We aim to consider e learning as a set of services, hosted according to Cloud Computing techniques. Moreover, we are convinced some local or remote services may be invoked according to context and training orchestration may be dynamically adapted on run time. We developped a concrete industrial product development based on Web services and WComp plate form for adaptability.

Keywords: Context awareness, SOA, SAAS, Cloud Computing, Web services, e learning.

1 Introduction

Economical context impacts companies and their information system (IS). Companies take over other competitors or develop new business skills, delocalize whole or part of their organization and their production. IS are faced to these genuine constraints and have to overcame these changes. Service oriented architecture (SOA)

Valérie Monfort
MIRACL, Université de Sfax, Tunisie

Valérie Monfort
Université Paris 1 - Panthéon – Sorbonne
e-mail: valerie.Monfort@univ-paris1.fr

Fayssal Felhi
ISIG Kairouan, Khemais Elouini Road, Tunisie
e-mail: felhi_fayssal@yahoo.fr

J.C. Augusto et al. (Eds.): ISAmI 2010, AISC 72, pp. 157–165.
springerlink.com

[17] offers a great flexibility to IS because each application owns interfaces masking implementation details. So, applications own interfaces including services and are seen as black boxes independently connected to a middleware as enterprise application integration bus (EAI) with its connectors and adaptors. However, this integration solution does not allow connecting heterogeneous applications or infrastructures, as distant IS. It is the reason why Web services are based on standards and are till now the cheapest and simplest solution to support interoperability between platforms. Based on Web services, enterprise services bus (ESB) [18] is a kind of services Web based EAI and allows loosely coupling with low costs.

Faced to these technologies, the SaaS (Software as a Service) model allows associating a price to a service consumption. Moreover, Cloud Computing, very linked to SOA and SaaS paradigms, offers delivering IT resources "on demand" by using virtualization paradigm. We are convinced SaaS and Cloud Computing are the fitted solutions to be used for e learning domain where any users as: companies, schools,… could pay services to train learners via Web services and ESB. Therefore, we are developing a new kind of e learning platform, based on orchestrated services, with services used "On Demand", according to training needs and learners skills. This solution aims to address: i) Full services oriented solution to provide full interoperability between authoring tools based on specific e learning standards and any kind of execution platforms. ii) Services consumption according to SaaS model and Cloud computing approach, iii) An execution platforms simplification. Platforms become middleware with orchestration engine for distant, remote, simple or complex services hosted locally. Moreover, some services may be invoked according to context (as weather, date, time, user profile, …) and/or specific events as mobility [23]. Specific context aware plate forms perform events to adapt system behavior. We used WComp [25][27] plate form as a middleware for context adaptation. The background of our research work is industry and users are learners employed by navigation and fishing companies, to train employees on simulators. We shall process as followed. The second section shows the case study. The third section explains Web services, SaaS and cloud computing. The fourth section addresses the adaptability. The fourth section shows related works.

2 Case Study

We aimed to design and implement a genuine fishing simulator. This software was intended to navigation and fishing schools or fishing ship owner companies. Following partial UML model (Fig.1) shows the different services proposed by the navigation and fishing e learning system. This e learning platform provides fisherman diploma for students (learners) in fishing schools and certificate for companies. The learner can subscribe, have a course, and get a diploma after an exam. The Teacher may be human or not. The system can take decision and can send specific events to complicate lesson if the student has a good level. Fig. 2 shows a process where the Teacher is preparing training and is sending it to the student who identifies himself by invoking identification and authentification services, linked to rules manager. While training learner may receive specific events during

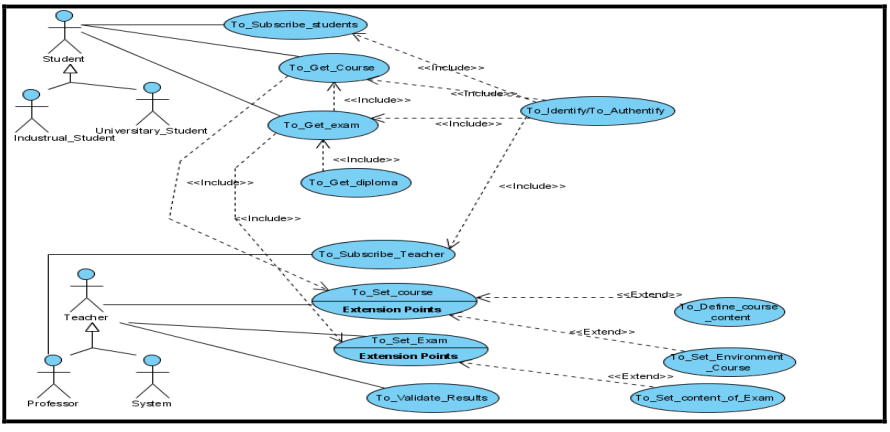

Fig. 1 E learning Services modeling

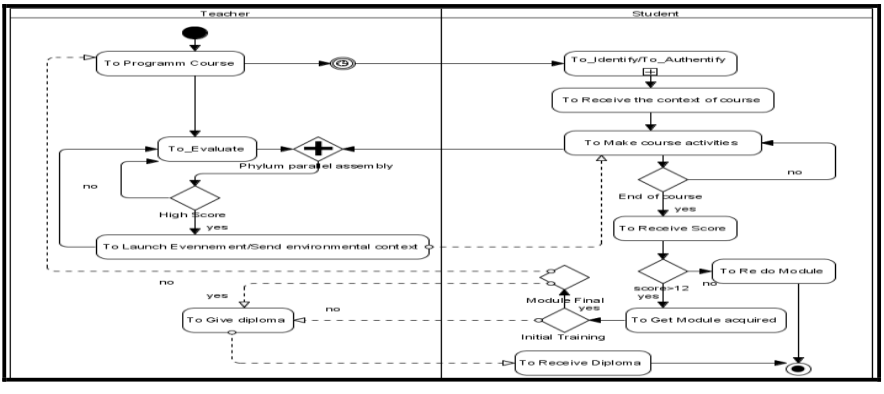

Fig. 2 Training services according to BPMN

current lesson to assess learner's skills. The learner is evaluated at any time. The following e learning architecture (Fig.3) shows services proposed by providers. Actors as learner and teacher can work anywhere and use different media. Services are available according to SaaS model and managed by providers according to Cloud Computing principles as explained in section 3. Contents are defined by the teacher with authoring tools according to standards (see previous section). He defines e learning tasks sequences, and so, e learning services orchestrations. He informs the services providers by loading training content and the concerned learner. The learner may use genuine navigation equipments and/or simulation. The genuine navigation equipments are linked to a middleware able to interpret signals coming from equipments and to send them to Execution platform. The learner may also use PDA or mobile phone. Learner receives the training scenario and while training sends information and invokes services. The e learning

Fig. 3 Introducing WComp

middleware (ESB) manages: routing messages, transporting messages and transforming exchanged data.

The providers repositories manage at least following services as: business, training supervisor, course virtual management, planning management, collaborative management, subscribing management, time/tracking management.

3 Web Services

3.1 Web Services Definition

Web services (WS) [7], like any other middleware technologies, aim to provide mechanisms to bridge heterogeneous platforms, allowing data to flow across various programs[10][11][12]. The WS technology looks very similar to what most middleware technologies looks like. Consequently, each WS has an Interface Definition Language, namely WSDL (Web Service Description Language) [14], that is responsible for the message payload, itself described with the equally famous protocol SOAP (Object Access Protocol)[9], while data structures are explained by XML (eXtended Markup Language) [15]. Very often, WS are stored in UDDI (Universal Description Discovery and Integration) registry [13]. Web services standards are gathered in WSA (Web Service Architecture)[7][8]. WS-BPEL [16] provides a language for the specification of Executable and Abstract business processes. Many integration solutions are based on an ESB. It has services repositories and services orchestration (based on BPMN [19] as Business Process Modeling Notation) that are modeled in a specific modeling tool linked to the environment. Providers use this technology to make increasing business by using SaaS model. Let us see Saas model in following section.

3.2 The SaaS Model

Software as a Service (SaaS) [1] is a model of software deployment whereby a provider licenses an application to customers for use as a service on demand. SaaS software vendors may host the application on their own web servers or download the application to the consumer device, disabling it after use or after the "on-demand" contract expires. The "on-demand" function may be handled internally to share licenses within a firm or by a third-party Application Service Provider (ASP) sharing licenses between firms. "On-demand" licensing and use alleviates the customer's burden of equipping a device with every conceivable application. It also reduces traditional End User License Agreement (EULA) software maintenance, ongoing operation patches, and patch support complexity in an organization.

"On-demand" licensing enables software to become a variable expense, rather than a fixed cost at the time of purchase. It also enables licensing only the amount of software needed versus traditional licenses per device. SaaS also enables the buyer to share licenses across their organization and between organizations, to reduce the cost of acquiring EULAs for every device in their firm.

3.3 Cloud Computing

Cloud computing [8][20][21] is a model for enabling convenient, "on-demand" network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models as: On-demand self-service, Ubiquitous network access, Resource pooling, Location independence, Homogeneity, Rapid elasticity, Measured service. Cloud computing brings a new level of efficiency and economy to delivering IT resources on demand. It offers efficiency and agility.

Cloud computing is typically divided into three levels of service offerings (fig. 5): Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a service (IaaS). The middle layer, or PaaS, is the encapsulation of a development environment abstraction and the packaging of a payload of services. PaaS offerings can provide for every phase of software development and testing, or they can be specialized around a particular area, such as content management. IaaS is at the lowest layer and is a means of delivering basic storage and compute capabilities as standardized services over the network. Servers, storage systems, switches, routers, and other systems are pooled (through virtualization technology, for example) to handle specific types of workloads from batch processing to server/storage augmentation during peak loads.

4 Context Adaptation

4.1 The WComp Plate Form

WComp is a prototyping "development" environment for context-aware applications. The WComp Architecture is organized around Containers and Designers paradigms. The purpose of the ***Containers*** is to take into account system services required by ***Components*** of an assembly during runtime: instantiation, destruction of software ***Components*** and Connectors. The purpose of the ***Designers*** allows configuring assemblies of through ***Containers***. To promote adaptation to context WComp uses Aspect [26] Assembly paradigm. Aspect Assemblies can either be selected by a user or fired by a context adaptation process. It uses a weaver that allows adding and or suppressing components. A container includes a set of (Beans) components and each bean has: properties, input methods that use received input information, and output Methods to send to another bean, for instance, output information. Aspect Assemblies allow defining links between Beans by using input and output information. WComp uses UPnP (Plug and Play) technology to detect locally whether the device is active or not and to define input methods and sent events for each component. With this architecture WComp allows: i) managing devices heterogeneity and dynamic discovering by using UPnP, ii) events driven interactions with devices, iii) managing dynamic devices connection and disconnection (dynamic re configuration on run time) in infrastructure. Let us see now the proposed solution.

4.2 Reviewed Architecture

Following architecture shown in Fig. 3, expresses WComp plate form can be imbedded in the execution plate form. WComp intercepts all the events coming from I/O middleware (learner interactions with the equipments) and learner interactions with different media as keyboard and mobile phone. According to these contextual parameters it invokes remote services and new services orchestrations. Let us see the implementation step.

4.3 Implementation

Each Bean owns properties, methods and events that define its status, its role, and its Input/output data (Get/Set). In our example we show the component Bean Boat. Its input method: Valaction expresses a data showing the action to do or the status of moorings or visibility. This component sends a displayed event showing the received data.

Concerning code generated from previous modeling, The lines 46 and 47 of the Fig. 4 show the assembly definition between beans "Boat" and status change (by new events called "eventHandler") in the container. "EventHandler" gets the value

```
46  this.checkBox1.CheckedChanged += new
47  System.EventHandler(this.__checkBox1_to_boat_0);
48  this.boat.EmitStringValue += new
49  Bateau.Bateau.StringValueEventHandler(this.__boat_to_label1_1);
50  [.....]
51  }
```

Fig. 4 WComp implementation for breaking mooring, assembly part

of the "checkedChanged" event that is emitted by "checkbox1" to the input method of the Bean "Boat" called "this_checkBox1_to_boat_0 and that is the parameter of the "EventHandler" method. The lines 48 to 49 are the definition of the assembly between the components beans "Boat" and "label1" in the container. Assembly part allows to re configure components according to training context.

5 Related Works

Several e learning research works [4][5][6] use Web services to get interoperability. [22][24][26] proposed a prototyped architecture with WComp without any access to remote services based infrastructure. No one use SaaS and Cloud paradigms that are the genuine architecture aims of Web services for the industry.

6 Conclusion

We proposed Web services based e learning architecture and we used SaaS and Cloud Computing paradigms to implement a navigation and fishing simulator. The main advantages of this architecture are : i) The interoperability of the platforms and the applications. Applications are considered as black boxes offering services without considering the way they are coded and on which platform they are. Interoperability is promoted by Web services XML based standards. ii) To take into account new needs in terms of new lessons, and to add or modify lessons (flexibility). iii) Lessons and exercises may be accessible via Intranet, Extranet, Internet, also with mobility via different media as lab top, PDA, mobile phone, … iv) This e learning Architecture is fully distributed, each knowledge component may be supported by one server independently. Based on previous research works, we aim now to propose technical architecture and solutions for context awareness as weather changes during training.

References

[1] The Benefits of SaaS, Web site retrieved from, http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=107276

[2] Web site retrieved from, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf
[3] Vossen, G., Westerkamp, P.: E Learning as a Web Service, ideas. In: Seventh International Database Engineering and Applications Symposium (IDEAS 2003), p. 242 (2003)
[4] Chepegin, V., Aroyo, L., De Bra, P., Houben, G.J.: CHIME: Service-oriented Framework for Adaptive Web-based Systems. In: De Bra, P. (ed.) Proceedings of Dutch national conference InfWet, Eindhoven, the Netherlands, November 20 (2003)
[5] Ullrich, C., Borau, K., Luo, H., Tan, X., Shen, L., Shen, R.: Why Web 2.0 is Good for Learning and for Research: Principles and Prototypes. In: Proceedings of the 17th International World Wide Web Conference, ACM, New York (2008)
[6] Lu, T., Ullrich, C., Grabowski, B.: Course Generation as a Web-Service for e Learning Systems. In: The 2nd European Young Researchers Workshop on Service Oriented Computing, Leicester, UK, pp. 63–69. University of Leicester (2007)
[7] Web site retrieved from WSA, http://www.w3.org/TR/ws-arch/
[8] Web site retrieved from, http://javaboutique.internet.com/articles/WSApplications/
[9] Web site retrieved from, http://www.w3.org/TR/SOAP
[10] Web site retrieved from, http://www.service-architecture.com/
[11] Web site retrieved from, http://www-106.ibm.com/developerworks/lib-rary/ws-polfram/
[12] Web site retrieved from, http://msdn.microsoft.com/webservices/build-ing/wse/
[13] Web site retrieved from, http://www.uddi.org/pubs/uddi_v3.htm
[14] Web site retrieved from, http://www.w3.org/TR/wsdl20/
[15] Web site retrieved from, http://www.w3.org/XML/
[16] Web site retrieved from, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
[17] Curbera, F., Khalaf, R., Mukhi, N.: Quality of Service in SOA Environments. An Over-view and Research Agenda (Quality of Service in SOA-Umgebungen). IT - Information Technology 50(2), 99–107 (2008)
[18] Chappell, D.: Enterprise Service Bus. O'Reilly, Alexander Ryan (2006)
[19] Web site retrieved from, http://www.bpmn.org/
[20] Miller, M.: Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online. QUE Publisher (2008)
[21] Reese, G.: Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. O'Reilly Publisher, Sebastopol (2009)
[22] Tigli, J.Y., Lavirotte, S.: Mobility in e learning context: Distributing Information Depending on Geographical Localization and User Profile. In: Proceeding of the 21th ICDE World Conference on Open Learning and distant Education, Hong-Kong, February 18-21 (2004)
[23] Menzel, K.: Mobile Devices in the classroom- Potentials and Requirements. In: ITC@EDU 1st International Workshop on Construction Information Technology in Education, Portoroz, Slovenia (2002)

[24] Stromboni, J.P., Tigli, J.-Y., Sander, P.: Some Experiments for Computer Aided Teaching. Poster in the 20 ICDE World Conference on Open, Dusseldorf, Germany, April 01-05 (2001)

[25] Tigli, J.-Y., Lavirotte, S., Rey, G., Hourdin, V., Riveill, M.: Lightweight Service Oriented Architecture for Pervasive Computing. IJCSI International Journal of Computer Science Issues 4(1) (September 2009) ISSN (Online): 1694-0784, ISSN (Print): 1694-0814

[26] Kiczales, G., Lamping, J., Maeda, C., Lopes, C.: Aspect-oriented programming. In: Aksit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)

[27] Web site retrieved from, `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsfed`