Real Time Self Adaptable Web Services to the Context: Case Study and Performance Evaluation

Faîçal Felhi, Jalel Akaichi
Institute University of Tunis, ISG
Computer Science Department, BESTMOD Laboratory, Tunis
Tunisia
felhi_fayssal@yahoo.fr, jalel.akaichi@isg.rnu.tn 7

ABSTRACT: *Web services are a solution for the integration of distributed information systems, autonomous and in real time, heterogeneous and auto adaptable to the context. This impact can resolve many problems in different system based on SOA and Web services. In this paper, we are interested in defining an approach to provide the different needs of a self-adaptability of Web services to context based on workflow and present its performance through an evaluation study, define the real time goal in our approach and showing the feasibility of this approach in an e-healthcare study.*

Keywords: Web services, SOA, Real time, Self-adaptability, Context, Performance evaluation

Received: 11 September 2014, Revised 19 October 2014, Accepted 26 October 2014

1. Introduction

Web systems based-on ubiquitous computing [32] must meet some specific constraints surrounding self-adaptation to the context in real time in the systems based on SOA and Web services. Computing applications now operate in a variety of new settings; for example, embedded in cars or wearable devices. They use information about their context to respond and adapt to changes in the computing environment. They are, in short, increasingly context aware. Considerable approaches related to adaptability with different modes of implementation such as: Aspect Oriented Programming [15]. This aspect used by various platforms on the goal to adapt the Web service [35] to the context dynamic changes of environment. Web services, like any other middleware technologies, aim to provide mechanisms to bridge heterogeneous platforms, allowing data to flow across various programs. The Web services technology looks very similar to what most middleware technologies looks like. The emergence of Web services as a model for integrating heterogeneous Web information has opened up new possibilities of interaction and adaptability to context when offered more potential for interoperability. However, from a set of requirements on SOA (Service Oriented Architecture) [7], and to provide self adaptation to the context of Web services, we need to integrate more generic connector that takes into account all ambient or distant events in real time.

The SOA offer great flexibility that is a great ability to functional and technical changes. Moreover, this type of architecture is most often used as Web services support, which provide the flexibility and interoperability expected, that is the ability to communicate between heterogeneous systems. The application in such information systems that incorporate SOA need to communicate across the exchange software (middleware or platforms). These middleware are the source of our work. It is on them that will think the same expectations in terms of flexibility, interoperability and adaptability.

The rest of this paper is organized as follows: In Section 2, we review previous research on adaptability of Web services, context awareness and real time impact in their approach. In section 3 we present our approach for a self adaptability of Web services to the context in the real time and the context awareness meta-model for them. In Section 4 we illustrate our solution by feasibility in ambulance trajectory case study. Finally, we summarize our work and discuss future research in Section 5.

2. Related works

Best fitted technology for implementing service-oriented architectures (SOA) offering flexibility and interoperability is Web services. Web services provide a minimalist mechanism to interconnect different applications. But one fundamental point is the importance of the WSDL [36] being the exact interface of the system. WSDL is responsible for the message payload, itself described with the equally famous protocol SOAP (Object Access Protocol) [25], while data structures are explained by XML (eXtended Markup Language) [37]. Very often, WS are stored in UDDI (Universal Description Discovery and Integration) [29] registry.

The context awareness [18][19][20] of such applications is the subject of a recent field of studies in pervasive computing called: context-aware systems. In [21][22], authors define context-awareness as the ability of a program or device to sense or capture various states of its environment and itself. Referring to these latter definitions a context-aware application must have the ability to capture the necessary contextual entities from its environment, use them to adapt its behavior (real time environment) and finally present available services to the user. In [13], the authors introduce another definition in which they insist on the use of context and the relevance of context information. The authors consider a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevance depends on the user's task. In [8], the authors considered context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. The authors give a general definition that can be used in a wide range of context-aware applications. In [33], the author approves this definition and claims that it covers all proposed works in context. However he considers it as a general definition that does not limit a context. We stress that the notion of hierarchy (structure) of context introduced by [33] is important. The definition proposed in [6] also presents the context as hierarchically organized. In this work the authors differentiate between environmental information that determines the behavior of mobile applications and that which is relevant to the application. They thus define the context as the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user.

Many approaches treat the adaptability of SOA in joining with Web services, to context. Charfi and al. approach [5] propose a framework that provides support for middleware BPEL (Business Process Execution Language) [4] engines. The authors apply the concepts of deployment descriptor and container for the Web service composition. Ferraz Tomaz and al. approach [11] proposed a tool for weaving aspects for a simple adaptability of the Web services, implementing aspects of the services as loosely coupled, where aspects are woven dynamically. In this approach, aspects are themselves Web services, thus they are independent of languages and platforms. Ben Hmida M. Et al., approach [2] extended the solution proposed by [11] to specify BPEL processes adaptable, that is to say, the adaptability of complex services. Hence the need to extend the semantic aspects and Web services, which resulted in the ASW (Aspect Service Weaver). Aspects are themselves loosely coupled Web services, they are independent of languages and platforms, but, this approach has limitations.

Adaptation to context is not taken into account, that is to say, if an event occurred during a search on a Web service, this approach does not take into account this event. In the other approaches we find those based on context adaptation [12][3][1][23]. The ambient computing encourages the proliferation of associated devices. We cited WComp approach [26][27][28] which represents the implementation of experimental models for lightweight components for service composition SLCA (Service Lightweight Component Architecture) which enables the design of ambient computing applications by assembling software components, orchestrating access to services through infrastructure devices from ambient. WComp supports protocols such as UPnP (Universal Plug and Play) [30] and Web services, allowing components through the proxy to interact with them. To promote adaptation to context WComp uses Aspect Assembly (AA) paradigm. Aspect Assemblies can either be selected by a user or fired by a context adaptation process.

Several approaches have investigated the impact of real-time SOA. In [17], the approach supports the ability to publish and search prescriptive and descriptive metadata of services. The prescriptive metadata is the descriptive and domain-specific information about the functionality of a service. For instance, prescriptive metadata of a geospatial service could be the capabilities file describing the geospatial data presented by that service. The descriptive metadata, i.e. quality of service attributes, into discovery process. The geospatial data being provided by a geospatial service may be fitted with client's request, however, this does not necessarily guarantee whether the service is sufficient for the desired quality of service requirements. RTSOA approach [27], presents a software infrastructure for industrial automation based on SOA for easing the problem of identification, discovery and communications among networked components, to allow for the configuration of the system at run-time. In [16], authors carried out a demonstration project under iLAND [14] based on SOA to show the feasibility of challenge such when using real-time operating systems and networks. they used a Relatively easy video streaming Application That highlights the fast reconfiguration capabilities of the middleware iLAND.

In our research work [9][10], we presented a proposal to a self-adaptable SOA to the context based on workflow [34] by presenting the functional and technical architecture of our approach. In this architecture we have given different features in terms of the needs of self-adaptability offered by the integration of workflow, which allows the management rules [24] and a kind of security and administration of Web services. This solution which can offer management rules that deal with business logic. Business logic can help in the development and optimization of these assemblies separating the events produced by the components of Web services.

3. Real Time Impact in Web Services Self-adaptable to the Context

3.1 System Architecture

In figure (Figure 1) we presented our research results for a self adaptability of service oriented architecture to the context. This architecture is based on workflow when we can use the management's rules and when we aid to treat external and internal events in the context. Web services are a sometimes block not change as an access provider, which is why we integrate this rules in Web services block to change the behavior of Web services at an event to finally determine good self adapt these web services and then the SOA to the context. These management rules also provide us with real-time processing. The impact of real time treatment is during extractions new needs and events not processed by Web services through well to technologies dedicated in the context.

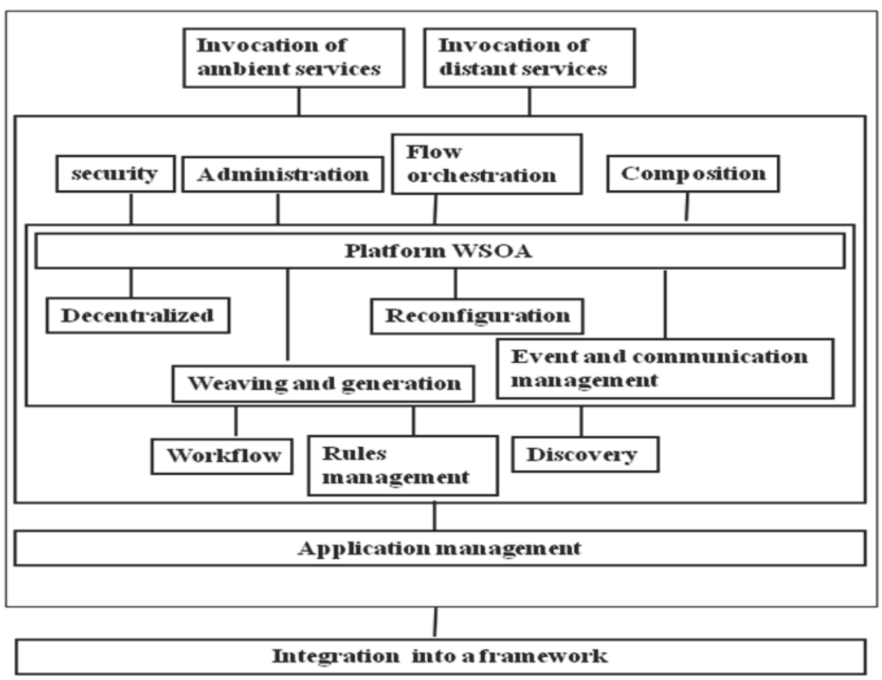

Figure 1. SOA self-adaptable to the context

Our architecture is based on objects or components to make the dynamic reconfiguration of components using more advanced mechanisms. It qualifies the distribution of applications across multiple servers and not the increase in service levels. There is a distributed architecture whose purpose is to deliver services to their audience and they will be accessible from any types of clients. Security and administration are offered by this system in treating the business logic from the workflow and rules.

Contextual resource discovery is the use of context data to discover other resources within the same context. The invocation of distant and ambient services is also permitted by this architecture using technologies dedicated to each type of invocation.

3.2 Feasibility

We opted to show the feasibility of our approach by the integration in a platform allowing the processing of Web services and workflow based on SOA. Technical architecture shown in Figure 2, allows the structuring of technical capabilities and infrastructure in our new approach WComp. In this architecture, except for the different needs initially used by WComp (service invocation ambient and remote data orchestration ...), we integrated connector's rules engine that communicates with a workflow engine in framework .NET. In this rules engine we need to define the rules that manage the data flow to finally produce events providing services to the customer.

The information shall be provided from a component that specifies the service to send it to another component by assembling them in a container through the language of Aspect of Assembly.

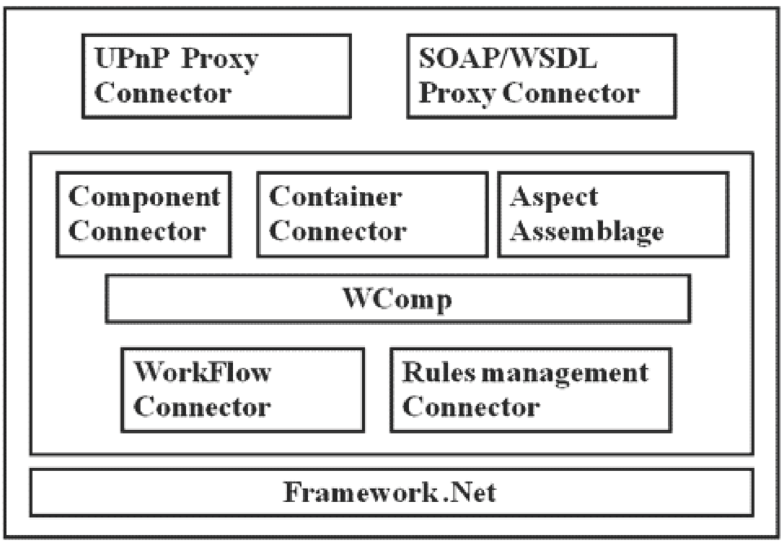

Figure 2. Technical architecture SOA_Workflow

4.3 A Healthcare Case Study

4.3.1 Scenario System

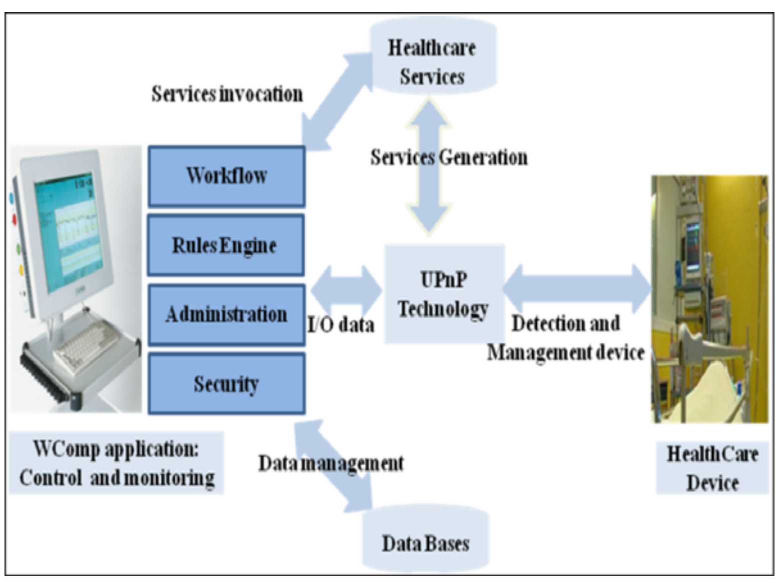

Figure 3. A E-Healthcare monitoring system

Figure.3 present our case studies for pervasive e-healthcare system. This architecture represents the different tools and components necessary for the conduct of monitoring and supervision between a doctor in his office and the room where they are many devices to monitoring.

Our system is based on a workflow; this workflow can test a request from a doctor by a rules engine that will transform the requests in the form of rules. A middleware is required to detect and manage devices "Healthcare Devices" distance in an ambient or distant space based on real time events. This middleware can also give and automatically generate Web services that represent different functionality to handle this device and host under a registry "Healthcare Services". This register is used by our application to invoke Web services dedicated and necessary for monitoring of device detected. The information for each device is provided by the middleware stored in a database for subsequent needs management.

Our solution helps doctors and health workers to not force them to move patients to make inquiries or analyzes of selected devices installed within the patient in real time. Our approach also minimizes the time cost analysis, preparation, resolution of some disease in a short time.

Under WComp we have integrated a rule engine that can provide management rules that deal with business logic. The rules engine can communicate with a workflow engine, which helps optimize and evolution of these assemblies separating the events produced by the components defined in an application WComp. Our system uses UPnP technology for detecting and handling equipment through the network.

4.3.2 Modeling

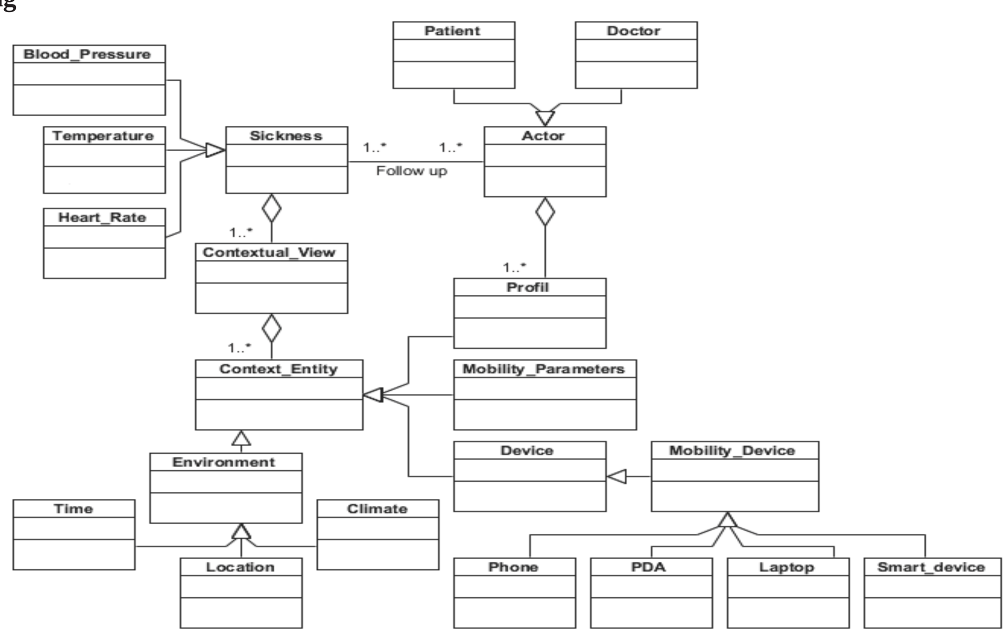

Figure 4. Healthcare model using context meta model

By using the meta model of context, Figure 4 represent the model of our solution to help doctors and health workers to not force them to move patients to make inquiries or analyzes of selected devices installed within the patient. Our approach also minimizes the time cost analysis, preparation, resolution of some sickness or disease (temperature, heart rate, blood pressure ect…) in a short time.

4.3.3 Implementation

Figure 5 shows the user interface of our application. Through a network access to a room where it is installed equipments for patient care, the user can choose a service controlled by clicking one of the buttons "Heart rate", "Blood pressure" or "Temperature". In this interface we presented an example of real time service by displaying on a screen the heart rate away from a device installed in a patient without the need to move near patient or to device for view these frequencies. These results are obtained through the use of Web services generated by the UPnP technology in our system.

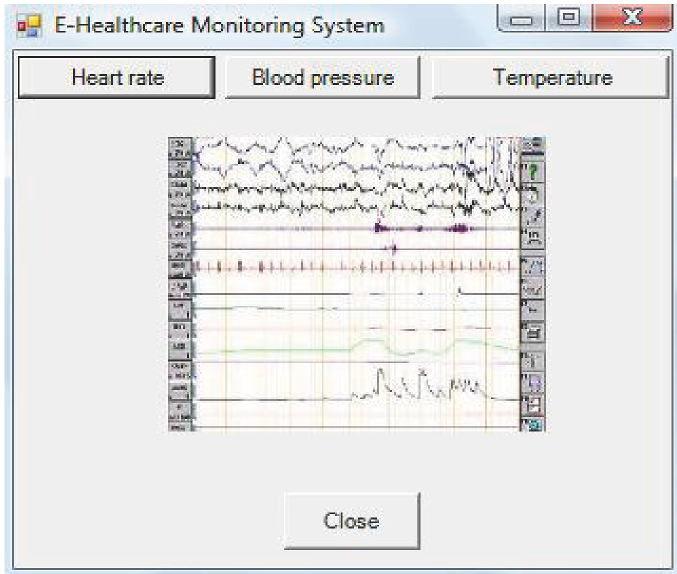

Figure 5. Pervasive E-Healthcare System Interface

5. Performance Evaluation

Needs		WComp	CORTEX	WSOA_Workflow
Architecture	Centralized			
	Decentralized	X	X	X
Service		X		X
Event Management		X	X	X
Object/Component	Object		X	X
	Component	X		X
Reconfiguration	Reflexivity		X	
	Assembly	X		X
Discovery	Dynamic	X	X	X
	By user	X		X
Interoperability		X	X	X
Adaptability	MOP		X	
	Weaving	X		X
	Generation	X		X
Security and administration				X
Management rules				X
Workflow				X
Business logic				X

Table 1. Performance evaluation

To show the performance of our approach, we summarize our studies on platforms adaptability by performance evaluation between our approach and other approaches based on needs of self adaptability to the context presented in Table 1.

In this evaluation we can approved some assets of our approach. These assets are:

- Decentralization and reconfiguration: Our architecture is based on objects or components to make the dynamic reconfiguration of components using more advanced mechanisms. He describes the distribution of applications across multiple servers and not the increase in service levels. It is a distributed architecture whose purpose is to provide services to the public and will be accessible from all types of customers.

- Event and Communication management: The events sent by the external environment are self adapted to the context. This architecture must include an inference engine, which specifies the behavior of applications in a given context and uses the event execution-condition stock model. Communication management is based on events that are created dynamically during system operation. Analysis of the communication is based on the modeling of communication events in composite services for surrounding areas.

- Weaving and generation: weaving is the process that takes as input a set of aspects and a database application, and issues a request whose behavior and structure are extended by aspects. The code generation correspond the component sets, the manipulation of graphic representation and generation in the application executable. Weaving can occur when compiling, the loading time or execution.

- Workflow and policy management: The workflow engine can then connect to the engine rules and "know" what option to take over a process. The rules engine also allows users to expose simple interfaces to generate these rules.

- Composition and flow orchestration: enables the design of ambient computing applications by assembling software components, orchestration and service access by the devices of the ambient infrastructure.

- Security and Administration: Offered by this system in the treatment of business logic and workflow rules.

- Discovery: Discovery of contextual resources is the use of context data to discover other resources in the same context.

- Invocation of ambient and remote services: Invoking remote and ambient services is also authorized by this architecture using technologies for each type of call.

6. Conclusion

In this paper we presented a performance evaluation of our approach for a self adaptability of real time SOA to context. We presented the generic meta-model used in any area of application. We have shown in this paper the interest of self adaptability of Web services in SOA since it often involves multiple heterogeneous systems well as the real time impact in Web systems based on workflow. Real time impact can help us deal with the demands of Web services at any time and taking into account the internal and external events related to context which gives a self adaptability of SOA. We have shown the feasibility of this idea and through modeling and implementation of e-healthcare case study.

References

[1] Anastasopoulos, M., Klus H., Koch J., Niebuhr D., Werkman E. (2006). DoAmI – a middleware platform facilitating re-configuration in ubiquitous systems, System Support for Ubiquitous Computing Workshop, *In*: The 8th Annual Conference on Ubiquitous Computing (Ubicomp) Publisher.

[2] Ben Hmida M., M., Ferraz Tomaz R, F., and Monfort V. (2006). Applying AOP concepts to increase Web services flexibility, *Journal of Digital Information Management* (JDIM).

[3] Biegel., G., Cahill. V. (2004). A framework for developing mobile, context-aware applications, *In*: 2nd IEEE Conference on Pervasive Computing and Communication, p.361–365.

[4] BPEL (2003). Available in Web URL - http://www6.software.ibm.com/software/developer/library/ws-bpel.pdf

[5] Charfi., A., Mezini., M. (2004). Aspect-Oriented Web Service Composition with AO4BPEL, 2nd European Conference on Web Services (ECOWS) Publisher, vol. 3250 of LNCS, Springer, p. 168-182.

[6] Chen., G., Kotz, D. (2000). A survey of context-aware mobile computing research, Tech. rep., Dept. of Computer Science,

Dartmouth College.

[7] Curbera, F., Khalaf, R., Mukhi, N. (2008). Quality of Service in SOA Environments. An Overview and Research Agenda (Quality of Service in SOA-Umgebungen). IT - Information Technology 50 (2) 99-107.

[8] Emanuele, J., Koetter, L. (2007). Workflow opportunities and challenges in healthcare, BPM & Workflow Handbook.

[9] Felhi, F., Akaichi, J., Adaptation of Web services to the context based on workflow: Approach for self-adaptation of service-oriented architectures to the context, *International Journal of Web & Semantic Technology* (IJWesT) Vol.3, No.4, October 2012 Publisher.

[10] Felhi, F., Akaichi, J. (2012). Towards the self-adaptability of Service-Oriented Architectures to the context based on workflow, *International Journal of Advanced Computer Science and Applications* (IJACSA) 3 (12)

[11] Tomaz, R, Ferraz., Hmida, Ben M., M., Monfort, V., (2006). Concrete Solutions for Web Services Adaptability Using Policies and Aspects, *Journal of Digital Information Management*.

[12] Garlan, D., et al. (2012). Aura : Toward distraction free pervasive computing, *IEEE Pervasive Computing*, 2012 Publisher.

[13] Gu, T., Pung, H., Zhang D., Q. (2005). A service-oriented middleware for building context-aware services, *Journal of Network and Computer Applications*, 28 1-18.

[14] iLAND (2013). http://www.iland.com/

[15] Kiczales, G., et al. (1997), Aspect-oriented programming, Proceedings European Conference on Object-Oriented Programming (ECOOP'97), V. 1241, p. 220–242. Springer-Verlag, Berlin, Heidelberg, and New York .

[16] Luís, S., Ricardo, M., Luís, A. (2012). Demonstrating real-time reconfiguration in service-oriented distributed systems, 8th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'2012) Demos session, p. 295-296, Hangzhou, China.

[17] Mehmet S., A., Galip A., Geoffrey C. F., Harshawardhan G., Marlon P., Ahmet S. (2005). Information Services for Grid/Web Service Oriented Architecture (SOA) Based Geospatial Applications, CGL Technical Report.

[18] Monfort, V., Felhi, F. (2010). Context Aware Management Platform to Invoke remote or local e Learning Services Application to Navigation and Fishing Simulator, International Symposium on Ambient Intelligence, ISAMI'10 Publisher, special volume *In*: Advances in Intelligent and Soft Computing (Springer), Guimarães, Portugal.

[19] Monfort, V., Felhi, F. (2010). A contextual approach to invoke intelligent house Services: an application to help physically handicapped persons, 1rst International Workshop on Recent Trends in SOA Based Information Systems in conjonction with ICEIS, Funchal Madeira, Portugal, Juin 2010.

[20] Monfort V., Khemaja M., Ammari N., Felhi F. (2010). Using SaaS and Cloud computing For "On Demand" E Learning Services : Application to Navigation and Fishing Simulator short paper *In*: The 10th IEEE International Conference on Advanced Learning Technologies, Sousse, Tunisia, July 5-7.

[21] Monfort, V., Hammoudi, S. (2010). When Parameterized MDD Supports Aspect Based SOA, *IJEBR International Journal of E-Business Research*.

[22] Monfort, V., Hammoudi, S. (2009). ICSOC, Towards Adaptable SOA: Model Driven Development, Context and Aspect, The 7th International Conference on Service Oriented Computing, November 23-27, Stockholm, Sweden.

[23] Roman, M., Islam, N. (2004). Dynamically programmable and reconfigurable middleware services, Middleware Publisher, Vol. 3231 in LNCS, p. 372–396, Springer.

[24] Rules (2010). Available in Web URL -http://www.vdocsoftware.com/vdoc/easysite/InVDOC2010/news/innovation/agilite-regles-metiers.

[25] SOAP (2007). Available in Web URL - http://www.w3.org/TR/SOAP

[26] Tigli J.-Y., Lavirotte S., Rey G., Hourdin V., Riveill M. (2009). Lightweight Service Oriented Architecture for Pervasive Computing, *IJCSI International Journal of Computer Science* 4 (1) 694-0814.

[27] Tigli J.-Y., Lavirotte S., Rey G., Hourdin V., Riveill M. (2009). Context-aware Authorization in Highly Dynamic Environments, IJCSI *International Journal of Computer Science* 4 (1).

[28] Tigli J.-Y., Lavirotte S., Rey G., Hourdin V., Cheung-Foo-Wo D., Callegari E., Riveill M., (2009). WComp Middleware for

Ubiquitous Computing: Aspects and Composite Event-based Web Services, *Annals of Telecommunications*, 64 (3-4) 197.

[29] UDDI (2004). Available in Web URL - http://www.uddi.org/pubs/uddi_v3.htm

[30] UPNP (2012) Available in Web URL - http://www.upnp.org/

[31] WComp2.4 (2013). Available in Web URL - http://www.wcomp.fr/

[32] Weiser, M. (1993). Some Computer Science Issues in Ubiquitous Computing, *Communications of the ACM,* 36 (7) 75–84.

[33] Winograd, T. (2001). Architectures for context, *Human-Computer Interaction (HCI) Journal* p.16.

[34] Workflow (2006). Available in Web URL - http://www.bpmbulletin.com/2006/06/21/difference-entre-workflow-et-moteur-de-regle/

[35] WS (2004). available in Web URL - http://www.w3.org/TR/ws-arch/.

[36] WSDL (2007). Available in Web URL - http://www.w3.org/TR/wsdl20/.

[37] XML (2012). Available in Web URL - http://www.w3.org/XML/