Context-Aware Services: A Survey on Current Proposals

Alfonso García de Prado
University of Cádiz
Cádiz, Spain
alfonso.garciaprado@mail.uca.es

Guadalupe Ortiz
Quercus Software Engineering Group
Mérida, Spain
gobellot@unex.es

Abstract— **Web services provide a successful way to communicate distributed applications, in a platform-independent and loosely coupled manner. Even though there are examples of good practice for the design, development and management of web services, there are scopes in which web service adaptation is required, such as context adaptation. Context-awareness is a complex topic to deal with but grants added value to any service which provides it. In this regard, there are multiple proposals in the recent literature which face the problem from different perspectives and using different technologies. This paper aims to show an overview of the most relevant approaches in this area, with a strong emphasis on those particularly related to the authors' work on the topic, namely using model-driven and/or aspect-oriented approaches.**

Keywords- Context-awareness, web services, model-driven development, aspect-oriented programming, context ontologies.

I. INTRODUCTION

Context and its use in software systems is a topic about which multiple research studies have been done in the last decade. This is not a surprising fact, since the design of applications and their communications whilst taking context into account permits the optimization of the use of information technologies in several respects: on the one hand, we can reduce the information submitted through communication lines to avoid their overhead; on the other hand; we will be able to save considerable resources in the client side and even in the servers', thanks to avoiding processing information which is not relevant for the device; finally, we will improve the user experience offering him a personalized service according to his requirements.

Context processing as well as adaptation to context is a hard task mainly due to the inherent complexity of the context itself and the multiple ways of managing it. In this regard, in this paper we will describe context state of the art, mainly focusing on those approaches which use a model-driven and/or aspect oriented development and on their usability for web services and those clients which access services from mobile devices.

We have focused our latest research on context adaptation for web services, specifically making the adaptation in the service-side and making it transparent for the client, which implementation would only have to provide the context information [1]. Our proposal is based on the use of aspect-oriented programming (AOP) for the decoupleness of the adaptation code and model-driven development (MDD) to simplify the system design without focusing on the final implementation or device requirements. We are currently extending this approach for other context issues and this is why we are specially interested in those research works where all these technologies interact in order to carry out context adaptation or context-awareness, although we will cover a wider area in this study of the state of the art.

In this sense, context adaptation in web services can be classified in two groups: first of all, adaptation in the service-side, where the process of transforming, selecting and adapting information depending on the client context is carried out in the service side. Secondly, client-side adaptation, which would in this case be the one in charge of following the mentioned process with all the information received from the service. Of course, these two options are not exclusive and there are some approaches which propose mixed models or approaches that, for instance, use a proxy or facade to develop the adaptation.

This paper is structured as follows: Section 2 introduces the definition and classifications of context. Section 3 provides an overview of recent research in the context scope, specially focusing on papers related to service adaptation and paying special attention to those which use model-driven or aspect-oriented techniques. Finally, Section 4 depicts a few conclusions in relation to the research works described.

II. CONTEXT BACKGROUND

Multiple definitions and discussions on the term context can be found ([2],[3], [4]), the one provided by Dey et al. in [5] being specially well-known –page 3, section 2.2: "*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*". One of the particular features of context information is that it is specific of each system, so that one specific type of information can be considered as part of the context in a particular system but not in a different one.

The term context-awareness supports the fact that the information provided about context by the client is properly used by the system so as to improve the quality of the interaction with it. That is, it means using information such as location, social attributes and other information from the user environment to foresee its necessities so that we can offer more personalized and easier to use systems. Therefore, a system is context-aware if it uses the context to provide relevant information or services to the user, adapting the system behavior to the particular needs of a specific user.

It is important to highlight that context information should be optional in context-aware applications, so that the

 ISBN: 978-1-61208-152-6

application or service can still be delivered even if a lack of context information means it is not personalized.

It is difficult to establish a context classification since the term covers a wide range of topics, but we can distinguish three general types of context:

- Device-related context: it describes those features specific to system devices and communications among them, providing information about their current state (for instance *use*, *load*, etc.), capabilities and configurability. Such types of information, for example, consist of available networks and services, screen size and its orientation, available memory and battery, et cetera.
- Environmental context: it describes the environmental conditions in which user and devices are. Sensors are normally used in order to provide such kind of information as location, temperature, noise, et cetera.
- User context: users can specify their preferences in relation to configurable properties in their devices, be it personal data, office, hobbies, needs, et cetera.

Dealing with this context so that applications are aware of it implies distinguishing which part of the application is impacted. In this regard, we discern three different groups:

- First of all, the user interface; the way to represent the information and the way in which application and user can interact may vary depending on the context (if the device can represent images or not, if it is tactile, etc.).
- Secondly, the information itself may be affected by the context; for instance, if location is taken into account the result when searching a cinema would be restricted to the current city, or when checking the playbill from a mobile device, it would be better to avoid film trailers.
- Finally, we also have to take account of the changes in the functionality that a context can cause: for instance, if context is based on user preferences, when buying a cinema ticket online, one user's preference could be to pay online and obtain the ticket on a pdf file, but another user might wish to pay for and collect the ticket at the ticket office before the film starts.

Besides, we can also classify how we deal with the information in two different options:

- On the one hand, we find dealing with the amount of information depending on the context: for instance if the context information provides us with the screen size, when looking for information about running films, we may want to skip comments from other users, actor profiles, etc., for smaller screens.
- On the other hand, dealing with information content: in the same scenario, depending on user preferences we can provide information about one type of films or another.

Thus, we can assert that there is no doubt of the benefits of being able to adapt services to the context, yet the problem is the complexity of dealing with this adaptation.

III. CONTEXT-AWARENESS STATE OF THE ART

A. *Frameworks, middlewares and tools*

In general terms we can find some tools and middlewares which tend to support the development of context-aware applications and services, some of them mentioned in [6] and [7]. Yau and Karim [6] propose RCSM, a middleware with support for context-aware applications. RCSM provides a language for the specification of context requirements. RCSM obtains context real time data from different sources and provides them to objects which are analyzing the status. This system is for both conventional computers and PDAs.

There are also proposals specific to a particular programming language: Bardram provides a framework named JCAF which helps in the development of Java-based context-aware applications [9]. JCAF obtains information from context sensors and generates the appropriate classes to manage context in the final application.

PACE is a middleware proposed by Henricksen et al. [10] which provides support for user preferences and context. They also provide tools to facilitate the use of context information by the applications. PACE supports context based on user preferences.

SOCAM is a middleware presented by Gu et al. [11] which supports context logic and modeling based on OWL (Ontology Web Language).

The research pieces analysed so far are not specially designed to be integrated with web services, being more general approaches. In the following paragraphs we will summarize those which are specifically designed to deal with web services.

Keidl et al. introduce a framework which facilitates context adaptation in web service development [12]. Context information, provided through the SOAP message, can be processed in the service, the client or automatically by the dedicated framework.

The Akogrimo project presented by Osland et al. in [13] help mobile device users access and compute information in grid systems; it focuses on the location and environment context information, collecting the mentioned information and sending it through the context manager.

Chen et al. propose CA-SOA [14]; a context model is provided to describe the service and client-side context information. Based on the model, CA-SOA provides different components in order to facilitate discovery and access to context-aware services.

Anyserver platform, Han et al. [15], helps with context management for mobile services for varied context information such as device information and networks.

De Almeida et al. propose Omnipresent [16], a LBS (Location-Based Service) context adaptation system for web services. The context is modeled based on OWL; besides, different services to provide information based on location (such as maps or routes) are offered.

Truong et al. have several publications on the topic in question such as ESCAPE framework [17] and inContext Project [18], both designed for web services in workgroups and collaborative environments. They focus on emergency situations and provide techniques for modeling, storing and exchanging context information among web services.

Finally, CoWSAMI is a middleware proposed by Athanasopoulos et al. [19] to support context through the use of a *Context Manager* which deals with the different context sources, where context information can be queried.

B. Context Ontologies

Context ontologies deserve special attention; in this section we will examine some representative approaches:

Chen at al. propose CORBA-ONT [20], an ontology for the support of context-based systems expressed in OWL. It is a collection to describe locations, agents and events and their corresponding properties. A logic reasoning engine is also provided in order to deal with context information.

Korpipää et al. provides a context-based framework and ontology in [21], where a semantic definition is provided to manage multiple sources context information.

To end with, Ying and Fu-Yuan describe an ontology based on afore-mentioned SOCAM architecture, where the context is represented using OWL [22]. The proposed ontology focuses on intelligent home environments.

Even though context ontologies are a relevant topic, the approaches we find normally focus on very specific domains and do not solve the general problem of web service contexts. In contrast, W3C proposes Delivery Context Ontology [23], a more general approach providing a formal model for representing environment features for devices interacting on the Web. The proposed ontology includes, among other features, device characteristics and network used for connection.

C. Client-Side and Proxy-Based Adaptations

We can find several papers focused on the client-side or proxi-based adaptation; the following ones deserve special attention:

Laakko and Hiltunen present a content-based adaptation of information through a proxy [24]. They focus on converting XHTML (Extensible Hypertext Markup Language) into XHTML MP (XHTML mobile profile) and into WML (Wireless Markup Language).

URICA [25] is a technique for content adaptation for mobile devices. Mohomed et al. base this proposal on the system learning through its interaction with the user, identifying the most relevant context for the adaptation.

The previously mentioned paper from Korpipää et al. [21] focuses on the client-side context adaptation proposing a framework for mobile devices.

Lastly, Carton et al. propose a model-driven aspect-oriented development for generating context-aware applications for mobilephones ([26], [27]), although they do not deal with web services.

D. MDD and Context Services

Sheng et al. [28] propose a UML-based modeling language – ContextUML- for the model-driven development of context-aware web services. They show how UML can be used for dealing with context-information in a simple and flexible way. The proposed metamodel displays several classes:

a) First of all, a class identifying the context, which is extended by subtypes *AtomicContext* and *CompositeContext*: the first one represents a low level context and the second a higher level one which might be composed of other *Atomic* or *Composite* ones. For instance, temperature would be an atomic context, but weather would be composite.

b) Seconly, class *ContextSource* extended by *ContextService* and *ContextServiceCommunity* in a similar way as in the first topic.

c) The third class is *CAMechanism*, which is the one that formalizes context-awareness through two possible subtypes: *ContextBinding* and *ContextTriggering*; the latter would be formed by a set of *ContextConstrains* and another set of *Actions*.

d) Finally, class *CAObject* is the base class for any kind of element in a *ContextUML* model and shows four different subtypes, namely *Service*, *Operation*, *Message* and *Part*.

Sheng et al have presented a more recent paper in which a platform for the development of context-aware services is provided [29]. This platform, named ContextServ and based on ContextUML, provides an integrated environment where developers can specify and deploy context-aware services. The three main objectives of this platform are a) providing context definitions and facilities for specifying different context types; b) defining context-aware web services, for which purpose a graphic interface is provided; c) transforming the service model into the corresponding BPEL code.

E. MDD, Aspects and Context-Awareness

Carton et al. suggest, as previously mentioned, combining model-driven development and aspect-oriented programming [27] through the use of a set of tools: Eclipse Modelling Framework (EMF) is proposed for defining the metamodels. Theme/UML [30] is used for extending UML2.0 and OCL (Object Constraint Language) will allow developers to establish model restrictions. Then, Java Emitter Templates (JET) will permit the transformation of models into J2ME-based code. Finally, Graphical Modelling Framework (GMF) provides the way of defining models with a graphical representation.

An extension of the previous research is presented in [26] by Carton et al., where they provide a utility for model-driven transformations for mobile-based context-aware applications. They integrate Theme/UML through XMI (XML Metadata Interchange) using the UML editor MagicDraw. The openArchitectureWare tool is used for the generation of code from the XMI file.

Three phases are differentiated in their development process: first of all, Theme/Uml is used in the modeling stage, at the end of this phase and through the use of MagicDraw, they will obtain two files –Theme/UML MarkingProfileFile and UML2 Class Diagram File. Secondly, two transformations will be carried out at the composition phase: the input are the two files obtained from the modeling stage for the first one and a composition model is created from them; the second transformation uses the latter model to produce an EMF-based model with a platform independent oriented model. The final stage, through the use of XPand [31], consists of transforming the object-oriented platform-independent model into a platform-specific one, refining low level details until the final transformation of the platform-specific model into code.

F. MDD, Aspects and ContextServices

We can find some papers in which web services, model-driven development and aspect-oriented programming appear together:

Prezerakos et al. propose decoupling the main service functionality from the behavior related to the context using a model-driven development based on contextUML [31]. Thus, service business logic and context management are treated as separate issues at modeling and code level, where context code is encapsulated through an aspect-oriented implementation.

They modify the contextUML metamodel to utilize stereotypes with a view to: (1) reducing association between modeled services and context; (2) removing unnecessary relations; (3) simplifying the metamodel semantic in order to facilitate the model to code transformation.

Grassi and Sindico provide support for context adaptation in [33] by decoupling the adaptation process from the application business logic, their scope being service-oriented applications. For this purpose they define a framework based on model-driven and aspect-oriented software development (AOSD). Context and adaptation to context are modeled in separate sections.

As far as context modeling is concerned, the authors distinguish two types of context associated to entities: firstly, status-based context, which consists of a set of relevant attributes for the entity. An attribute in this context can be defined according to other attributes and can be associated to the source providing this information. Secondly, event-based context consists of a group of relevant events for the entity. In parallel, two types of constraint are defined as key elements for the introduction of context-awareness in the application, These are *state constraint* (defined by the logic predicate of the context value based on its state) and *event constraint* (defined as an event-based pattern).

Regarding the adaptation modeling, two mechanisms are provided to introduce context into the application: the first one (context-aware binding) is defined by a pair formed by an entity and a set of values, enabling the creation of different adaptation types depending on the entity type. The second one (context-aware insertion) is based on AOP. We can make two types of insertions (structural and behavioral) and in both cases the value to be inserted and location are provided. The structural insertion is equivalent to AOP intertype declaration using the location to specify the part of the application to be linked, the value specifying the elements to be injected. The behavioral insertion is equivalent to the advice concept in AOP, location being used to specify where to inject the new functionality (AOP join point) and the value to specify the new functionality itself.

Vale and Hammoudi [34] focus on the context-aware development of distributed applications proposing the use of model-driven engineering and separating matters of interest in different models. They focus on web service implementation and how to adapt them to changing contexts based on OMG EDOC-ECA [35] principles for context modeling and context-aware architectures. The result is CSOA, a *Context-aware Service Oriented Architecture*, based on the EDOC_ECA metamodel and providing business, context and composition views as platform-independent models and adaptation and service views as the platform-specific models, as we explain in the following paragraphs.

1. Business view provides traditional business logic.
2. Context view represents context information through the use of ontologies. A metamodel for context definition is also provided. Specifically W3C DRF (*Resouce Description Framework*) is used for the representation of context information at model level.
3. Composition view separates business from context logic in two different component types: business process components for the implementation of business logic, and contextual process components to provide the application's adaptation to context. The composition identifies which are the connections and interactions among all components.
4. Adaptation view provides a composition of one or more business and context components and a connection one. This composition is an abstraction of the process component and describes how process component instances are configured and connected for implementing the composition.
5. Service view is based on the WSDL metamodel. Services can be formed by other services represented in the context service composition.

Monfort and Hammoudi's proposal [36] shows two approaches to facilitate web service adaptation; the first one is based on an aspect-oriented implementation; the second one on the use of model-driven development for the context. For aspect-oriented implementation ASW (*Aspect Service Weaver*) is presented: it is a utility that lets us intercept SOAP messages between client and service adding new behaviors through the use of AOP, using XPath [37] for selecting WSDL methods to be intercepted. The second implementation represents a context metamodel identifying those issues considered more relevant for mobile devices. They also provide context parametrized transformations. Finally, they propose the combination of both techniques.

G. MDD, Aspects and Context –Awareness for Mobile Devices

We can also find some proposals which deal with context adaptation. In this line of work, Menkhaus presents an architecture for decoupling user interfaces in web applications from the application logic [37]. Dockhorn et al. propose an architecture to support mobile context-aware applications through a publish-subscribe mechanism [39]. None of these approaches deal with the adaptation of web service responses. Additionally, Pashtan et al. propose [40] the adaptation of web applications' content depending on the device, but do not tackle web service applications. On the other hand, Schomohl et al. provide context-aware mobile services [41], but they focus mainly on the creation of location-based services, rather than the adaptation of service responses to client requirements.

The work from Keith et al., whose framework was already mentioned [12], present an approach for services to deal with client contextual information through a context framework [42]. Context is always included in the client SOAP header as well as in service messages. This implies that not only services, but also clients have to process the context included in the header, however the proposal does not explore how the client can deal with the received context. In our proposal, the answer provided by the service is already adapted to client requirements, thus can be processed normally. Besides, their framework allows client context processing through the use of context plugins or context services. Context plugins have to be installed locally, which is improved by context services, available anywhere. A plugin and service have to be developed for each context and must be compatible with all services, which is extremely difficult and costly. Song et al. extend the latest work to preserve the client context privacy in [43], yet do not provide any further advantage to our proposal.

Concerning the client side, the proposal from Zhang et al. allows reengineering PC-based systems into a mobile product line by using a meta-programming technique. In their approach, systems are firstly developed for PC environments and then evolved to mobile device platforms by generating specific components from generic metacomponents [44]. The approach from Alves deals with existing variations in different mobile devices' models. He uses AOP to refactorize the variations and therefore decouple them from the core of the mobile application [45]. The idea of Blechsschmidt et al. is based on allowing the personalization of mobile device applications based on the end user profile [46]. For this purpose user information is collected and stored in XML files which are precompiled with the applications' core code in order for this information to be considered in the application during execution.

Finally, a relevant piece of work is the one presented by D. Zhang in [47]. He provides an approach for web content adaptation to meet user needs, suit characteristics of individual mobile devices, and adjust to dynamic contexts. His approach mainly focuses on web applications and an interactive adaptation in which users have to take part during the invocation to obtain the information they want. In the same line we can find the work from Niederhausen et.al. [48], where a framework for web applications adaptation is provided. The framework allows the developer to adapt web application content depending on different adaptation concerns, such as device adaptation, through the use of what they call adaptation aspects.

IV. CONCLUSIONS

Having studied the different approaches in related literature we are ready to depict a few conclusions:

Client-side adaptation does not remove unneeded information traffic and, even worse, overheads computation in the client side. In many cases, this computation becomes too complicated and even impossible to perform due to the excessive amount of unnecessary and useless data; for instance, if we wished to obtain home delivery restaurants in Cádiz (Cádiz being our context), a context-aware web service would remove those restaurants which are outside Cádiz from the initial list.

The use of a facade or proxy for the adaptation would imply the same problem. Client-side adaptation benefits from the advantage of not having to send the context to the service, but if we do it through a proxy then the mentioned information will be required. Finally, introducing a new element between service and client to manage the context creates a more complex architecture system, making its development and maintenance more difficult.

These significant drawbacks are solved when using service-side adaptation, in spite of having the inconvenience of having to send context information from the client to the service. For the adaption to context we consider that using model-driven development and aspect-oriented programming allows an easier implementation both at design and development phase, as well as at maintenance time. We also consider of special relevance the use of an ontology or other formalized classification and definition for context concepts. Based on these assumptions we plan to extend our previous work [1] for context adaptation as explained in [49].

ACKNOWLEDGMENT

The second author acknowledges the support from Ministerio de Ciencia e Innovación (TIN2008-02985) and Fondo Europeo de Desarrollo Regional (FEDER).

REFERENCES

[1] Ortiz, G. and Garcia de Prado, A. "Improving Device-Aware Web Services and their Mobile Clients through an Aspect-Oriented, Model-Driven Approach". Information and Software Technology Journal, Vol. 52, Issue 10, 2010, pp.1080-1093.

[2] Dey, A.K., "Understanding and Using Context", Personal and Ubiquitous Computing, vol. 5, 2001, pp. 4-7.

[3] Chen, G. and Kotz, D., A Survey of Context-Aware Mobile Computing Research, Hanover, NH, USA: Dartmouth College, 2000.

[4] Baldauf, M., Dustdar, S., and Rosenberg, F., "A survey on context-aware systems", International Journal of Ad Hoc and Ubiquitous Computing, vol. 2, 2007, pp. 263.

[5] Abowd, G.D., Dey, A.K., Brown, P.J. Davies, N. Smith, M. and Steggles, P. "Towards a Better Understanding of Context and Context-Awareness", Handheld and Ubiquitous Computing, vol. 1707, 1999, pp. 304-307.

[6] Truong, H.L and Dustdar, S. (2009) "A survey on context-aware web service systems", International Journal of Web Information Systems, Vol. 5 Iss: 1, pp.5 - 31

[7] Kapitsaki, G.M. Prezerakos, G.N. Tselikas, N.D. and Venieris I.S. "Context-aware service engineering: A survey", Journal of Systems and Software, vol. 82, no. 8, pp. 1285-1297, 2009.

[8] Yau, S.S. and Karim, F., "A context-sensitive middleware for dynamic integration of mobile devices with network infrastructures", Journal Parallel Distributed Computing, vol. 64, 2004, pp. 317.

[9] Bardram, J.E., "The Java Context Awareness Framework (JCAF) – A Service Infrastructure and Programming Framework for Context-Aware Applications", Pervasive Computing, Gellersen, H.W., Want, R., and Schmidt, A., Eds., Springer, 2005, pp. 98-115.

[10] Henricksen, K., Indulska, J., McFadden, T., and Balasubramaniam, S., "Middleware for Distributed Context-Aware Systems", Int. Symp. on Distributed Objects and Applications, 2005, pp. 846–863.

[11] Gu, T., Pung, H.K., and Zhang, D.Q., "A service-oriented middleware for building context-aware services", Journal of Network and Computer Applications, vol. 28, 2005, pp. 1-18.

[12] Keidl, M. and Kemper, A., "A Framework for Context-Aware Adaptable Web Services", EDBT, 2004, pp. 826-829.

[13] Osland, P.O., Viken, B., Solsvik, F., Nyngreen, G., Wedvik, J., and Myklbust, S.E., "Enabling context-aware applications", Proceedings of ICIN2006: Convergence in Services, Media and Networks, 2006.

[14] Chen, I., Yang, S. and Zhang, J., "Ubiquitous Provision of Context Aware Web Services", 2006 IEEE Int. Conf. on Services Computing, Chicago, USA: 2006, pp. 60-68.

[15] Han, B., Jia, W., Shen, J. and Yuen, M.-C., "Context-Awareness in Mobile Web Services", Parallel and Distributed Processing and Applications, Cao, J., Yang, L., Guo, M., and Lau, F., Eds., Springer Berlin / Heidelberg, 2005, pp. 519-528.

[16] De Almeida, D.R., Baptista, C.D., Da Silva, E.R., Campelo, C.E.C., De Figueiredo, H.F. and Lacerda, Y.A., "A Context-Aware System Based on Service-Oriented Architecture", Int. Conf. on Advanced Information Networking and Applicationsm, V. 1, Vienna, Austria: 2006, pp. 205-210.

[17] Truong, H.-L., Juszczyk, L., Manzoor, A. and Dustdar, S., "ESCAPE – An Adaptive Framework for Managing and Providing Context Information in Emergency Situations", Smart Sensing and Context, Kortuem, G., Finney, J., Lea, R., and Sundramoorthy, V., Eds., Springer Berlin / Heidelberg, 2007, pp. 207-222.

[18] Truong, H.-L., Dustdar, S., Baggio, D., Corlosquet, S., Dorn, C., Giuliani, G., et al., "inContext: A Pervasive and Collaborative Working Environment for Emerging Team Forms", Int. Symposium on Applications and the Internet, Turku, Finland: 2008, pp. 118-125.

[19] Athanasopoulos, D., Zarras, A., Issarny, V., Pitoura, E. and Vassiliadis, P., "CoWSAMI: Interface-aware context gathering in ambient intelligence environments", Pervasive and Mobile Computing, vol. 4, 2008, pp. 360-389.

[20] Chen, H., Finin, T. and Joshi, A., "An ontology for context-aware pervasive computing environments", The Knowledge Engineering Review, vol. 18, 2003, pp. 197-207.

[21] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H. and Malm, E.-J., "Managing context information in mobile devices", IEEE Pervasive Computing, vol. 2, 2003, pp. 42-51.

[22] Ying, X. and Fu-yuan, X., "Research on Context Modeling Based on Ontology", 2006 Int. Conf on Computational Inteligence for Modelling Control and Automation and Int. Conf.on Intelligent Agents Web Technologies and International Commerce, Sydney, Australia: 2006, pp. 188-188.

[23] Cantera Fonseca, J.M. and Lewis, R., "W3C - Delivery Context Ontology", http://www.w3.org/TR/dcontology/, 2009. Last access 06/2011

[24] Laakko, T. and Hiltunen, T., "Adapting Web Content to Mobile User Agents", IEEE Internet Computing, vol. 9, 2005, pp. 46-53.

[25] Mohomed, I., Cai, J.C., Chavoshi, S. and de Lara, E., "Context-aware interactive content adaptation", Int Conf on Mobile systems, applications and services , Uppsala, Sweden: 2006, pp. 42.

[26] Carton, A., Driver, C., Jackson, A. and Clarke, S., "Model-Driven Theme/UML", Transactions on Aspect-Oriented Software Development VI, Katz, S., Ossher, H., France, R., and Jézéquel, J.-M., Eds., Springer Berlin / Heidelberg, 2009,p p. 238-266.

[27] Carton, A., Clarke, S., Senart, A. and Cahill, V., "Aspect-Oriented Model-Driven Development for Mobile Context-Aware Computing", Int. W. on Software Engineering for Pervasive Computing Applications, Systems and Environments, USA, 2007, pp. 5-5.

[28] Sheng, Q.Z. and Benatallah, B., "ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services Development", International Conference on Mobile Business, Sydney, Australia: 2005, pp. 206-212.

[29] Sheng, Q.Z., Pohlenz, S., Yu, J., Wong, H.S., Ngu, A.H.H. and Maamar, Z., "ContextServ: A platform for rapid and flexible development of context-aware Web services", Int. Conf. on Software Engineering, Vancouver, BC, Canada: 2009, pp. 619-622.

[30] Clarke, S. and Baniassad, E., Aspect-Oriented Analysis and Design: The Theme Approach, Addison-Wesley Professional, 2005.

[31] XPand. http://www.eclipse.org/modeling/m2t/?project=xpand, [Last access: August 2011

[32] Prezerakos, G.N., Tselikas, N.D. and Cortese, G., "Model-driven Composition of Context-aware Web Services Using ContextUML and Aspects", IEEE Int. Conf. on Web Services, Salt Lake City, USA: 2007,pp. 320-329.

[33] Grassi, V. and Sindico, A., "Towards model driven design of service-based context-aware applications", Int. W on Engineering of software services for pervasive environments, Dubrovnik, Croatia, 2007, pp. 69-74.

[34] Vale, S. and Hammoudi, S., "Model Driven Development of Context-aware Service Oriented Architecture", Int. Conf. on Computational Science and Engineering - Workshops, San Paulo, Brazil, 2008, pp. 412-418.

[35] OMG EDOC-ECA. http://www.omg.org/spec/EDOC/1.0/ [Last access: August 2011]

[36] Monfort, V. and Hammoudi, S., "Towards Adaptable SOA: Model Driven Development, Context and Aspect", International Joint Conference on Service-Oriented Computing, 2009, pp. 175 - 189.

[37] Xpath. http://www.w3.org/TR/xpath/ [Last access: August 2011]

[38] Menkhaus, G. "Architecture for client-independent Web-based applications". Proceedings of the Technology of Object-Oriented Languages and Systems, Zurich , 2001, pp. 32-40.

[39] Dockhorn Costa, P., Ferreira Pires, L., van Sinderen, M. and Pereira Filho, J., "Towards a Service Platform for Mobile Context-Aware Applications". Int. W. on Ubiquitous Computing, Portugal, 2004, pp.48-61.

[40] Pashtan A., Kollipara S. and Pearce, M. "Adapting Content for Wireless Web Services", IEEE Internet Computing, V. 7, N. 5, 2003, pp. 79-85.

[41] Schomohl, R. and Baumgarten, U. "Mobile Services based on client-server or P2P architectures facing issues of context-awareness and heterogeneous environments". Int. Conf. on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 2007.

[42] Keidl, M. and Kemper, A. "Towards Context-Aware Adaptable Web Services".Int. WWW Conference on Alternate, New York, 2004, pp.55-65.

[43] Song, Y. J., Lee, D.H., Yim, J.G. and Nam, T. Y. "Privacy Aware Adaptable Web Services Using Petri Nets". Int.Conf. on Convergence Information Technology, Korea (South), 2007, pp. 19-24.

[44] Zhang, W., Jarzabek, S. Loughran, N. and Rashid, A. "Reengineering a PC-based System into the Mobile Device product Line". Int. W. on Principles of Software Evolution, Helsinki, Finland, 2003, pp.149-160.

[45] Alves, B. "Identifying Variations in Mobile Devices". Journal of Object technology, Vol 4, Nº 3, 2004, 47-52.

[46] Blechsschmidt, T., Wieland, T., Kuhmunch C. and Mehrmann, L.. "Personalization of End User Software on Mobile Devices". Int. W. on Mobile Commerce and Services, München, Germany, 2005, 130-137.

[47] Zhang, D. "Web Content Adapatation for Mobile Handheld Devices". Communications of the ACM, Volume 50, Issue 2, February 2007, pp. 75-79.

[48] Niederhausen, M., Fiala, Z., Kopcsek, N. and Meissner, K., "Web Software Evolution by Aspect-oriented Adaptation Engineering", IEEE Int. W. on Web Site Evolution, Paris, France: 2007, pp. 3-7.

[49] Ortiz, G. and Garcia de Prado, A. "Web Service Adaptation: A unified approach versus multiple methodologies for different scenarios", ICIW 2010, the International Conference. on Internet and Web Applications and Services, Barcelona, Spain, 2010, pp.569-572.