CONTEXT-AWARE AND ADAPTABLE ARCHITECTURE CA^3

Stanimir Stoyanov

Abstract. *In the paper some important notions and features of context-aware and adaptable service provision have been discussed. An approach has been described which can be used to develop architectures with the mentioned features. The abstract architecture AC^3 and its application for implementing an eLearning environment have been described as well.*

Keywords: abstract architecture, eLearning , eServices, eContent
2010 Mathematics Subject Classification: 97U50, 97R40

1. Introduction

Currently, development of context-aware and adaptive environments, which deliver electronic services (eServices) in a flexible and personalized way, is in the focus of research interest. Such environments can be examined as combinations of infrastructure, eServices provision, electronic content (eContent) and user experience [1]. The rest of the paper is organized as follow. Section 2 presents briefly our concept of context-aware and adaptable service delivery. Section 3 describes an approach supporting developing of context-aware and adaptable software architectures. Section 4 presents the abstract architecture CA^3. A prototype implementation for eLearning is discussed in section 5. Finally section 6 concludes the paper.

2. Context-aware and adaptable service provision

As the proposed architecture is concerned with the context-awareness and adaptability aspects, it is important to first clarify these notions. In our development approach, we adopted the Dey's definition [2], which stipulates that "context is any information that can be used to characterize the situation of an entity". An entity could be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves. Context, however, can be a location, identity, activity, time etc. We use the Dey's definition as a basis for further discussions. In order to progress this definition as a concept for the creation of the desired middleware architecture, we first solidify the definition as presented below. We toned to clearly differentiate context-awareness from adaptability.

Context-awareness is the architecture's ability to identify changes in the environment/context as regards: User's/device's location (*device mobility*) – in some cases this change leads to changing the serving communication node; User device (*user mobility*) – this change offers different options for the service delivery to the user. It is essential to monitor for changes in the clients operating environment (user device capabilities), and to facilitate service content adaptation accordingly; Communication type – depending on the current prevailing wireless network conditions/constraints, the user may elect to utilize any one of a variety of wireless communications protocols; User preferences – service *personalization* may berequired to reflect changes made by the user to his/her own individual preferences, e.g., the way the service content is visualized to the user, etc.; Information environment; Goal-driven sequencing of tasks engaged in by the user; Environmental context issues such as classmates and/or student/educator interactions.

The goal of **adaptability** is to ensure trouble-free, transparent and adequate facilitation of user requests for services by taking into account the different aspects of the context mentioned above. In other words, after identifying a particular change in the environment, the middleware must be able to take compensating actions (counter-measures) such as handover of user service sessions from one IS to another, reformatting/transcoding of service content, service personalization, etc.

Our philosophy is that context-aware and adaptable service provision has to be supported at various levels and in different time periods by externalized architectural components. In this context any system, which facilitates a context-aware and adaptable service provision, can be built through the integration of two parts [3]: **A standardized middleware**, which is able to detect the dynamic changes in the environment during the processing of user requests for services (*contextawareness*) and correspondingly to ensure their efficient and intelligent execution (*adaptability*); **A set of electronic services** realizing the functionality of the application area (*business functionality*), which can be activated and controlled by the middleware. In our case, the application area is the mLearning domain.

3. Approach

In order to develop an software architecture supporting context-aware and adaptable service provision an approach has been proposed [4,5]. The approach uses some basic ideas of Model Driven Architecture (MDA) paradigm [6] which have been adapted and extended for our case. The approach envisages the existence of three architectural context-aware and adaptable levels (Fig. 1.). **Scenarios Level** presents the features and specifics of the underlying communication infrastructure in the form of different scenarios executed for the provision of eServices. **Middleware Level** is an agent-based multi-layered middleware playing a mediator role between the services level and the scenarios level. It offers shared functionality: on one hand, it contains agents needed for the execution of different scenarios; on the other hand, it specifies a set of agents assuring the proper provision of services. The level could be considered as a virtual (software) environment for service deployment. The main goal of the middleware level is to allow the architecture to execute/satisfy the user requests in a context-aware and adaptable fashion. **eServices Level** represents and models the functionality of the delivered services. The service functionality is modelled in UML by taking into account the fact that the service realization is not directly unfolded by the system software but rather is processed by the middleware. That's why it is very important to present the service as a composition of smaller parameterized activities, which could be navigated in different way.

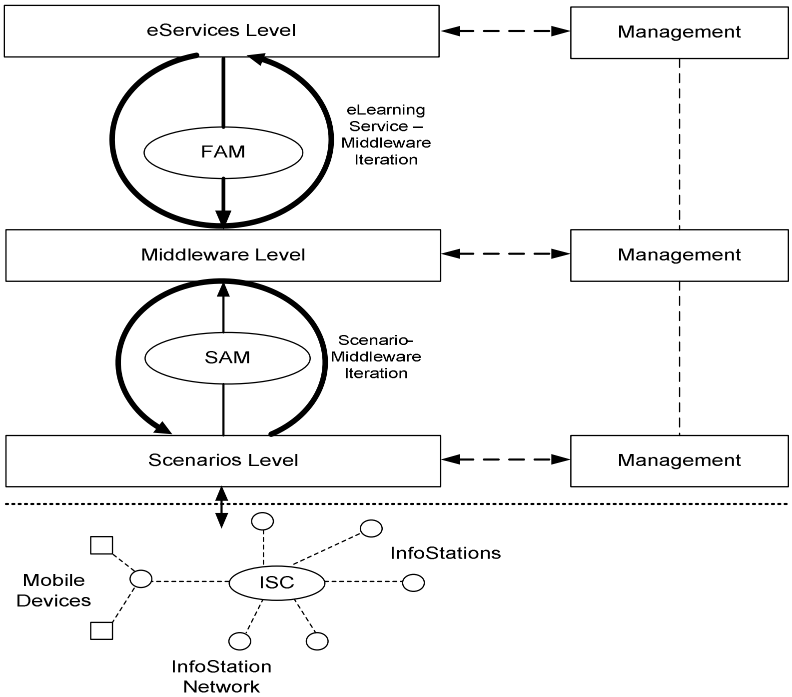

Fig.1. Multi-Layered Approach

There are two types of iterations in our approach (Fig.1.). *SM iterations* are relationships between the scenarios level and the middleware level. During each SM iteration, new scenarios that present/reflect particular aspects of the possible states and changes in the environment are develos). This way using the (formalized) presentation of scenarios, all corresponding middleware components needed for the support of these scenarios are developed step by step. *eLM iterations* are mappings of the services onto the middleware level, where the navigation model and parameterization of services are specified. Besides relationships the iterations present possible time periods and various aspects of the context-aware and adaptable service provision. Some examples of SM iterations: **Basic Architecture -** the iteration aims adaptation of the software architecture to the underlain communication network; **Time-based Management -** some important changes in the context during the user service request's execution can be detected and identified by the architecture only if the temporal aspects of this process are taken into consideration. Thus the goal of this iteration is to develop concepts and formal models allowing a temporal adaptation of the processes supported by the middleware; **Personalization -** this iteration is concerned with problems related to strengthening the architecture, e.g. to support personalization. For instance, in the eLearning domain personalized eLearning could be fully realized only by means of adaptive architectures, whereby the eLearning content is clearly distinguished from the three models influencing the learning process – the user model, the domain model, and the pedagogical model.

The approach has to ensure additional features of the architecture, e.g., user session management, user request globalization, resources management, service integration. In this way additional iterations can be defined, like resource deployment optimization, intelligent agents with new abilities (cloning, copying, mobility). Collaborative context-awareness and adaptation is a challenge too**.**

4. CA^3 layers

Following the approach described above the CA^3 has been developed consisting of three layers (Fig.2.). The first layer Middleware Activation Layer (MAL) operates as an interface between the architecture and the system software using in a CA^3 based application. MAL aims at activation of a multi-agent environment that the middleware can operate in.

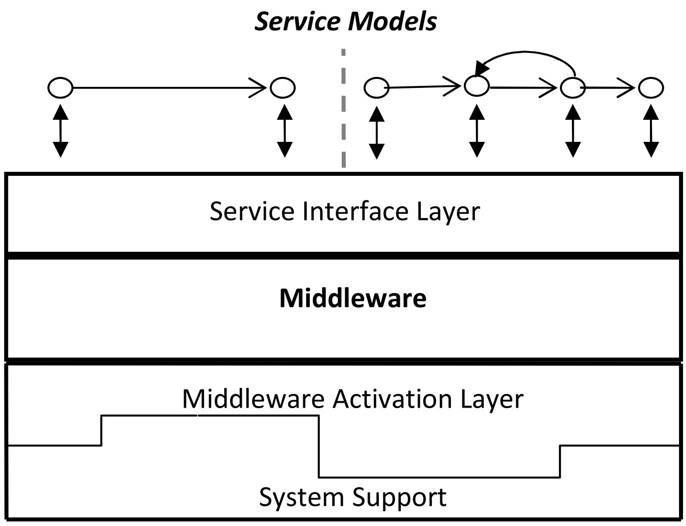

Fig.2. CA^3

The middleware consists of two parts: **Core** – contains agents, called persistent agents, that facilitate the context-awareness and adaptability of the architecture. These agents detect and identify various events within the environment, and dynamically generate other agents accordingly, which to process the execution of the user's service requests; **Dynamic part** – contains all agents, called operative agents, that are dynamically generated in response to the identified events (changes in the environment). These agents are responsible for the execution of scenarios and controlling the user's service sessions. After completing their task, these agents are removed from the system architecture.

The third layer Service Interface Layer chooses and activates service(s) corresponding to the current user request. We are going to investigate two service models which can be supported by CA^3: **Web Services like model** – the Web Services are wide used in the current service-oriented applications. However this model suffers from the lack of their ability to provide constructs and concepts that enable reasoning about runtime service behavior; **More adaptable service models** – in these models services are composed using a specification technique that characterizes ongoing behaviour of the services [7]. In these models context-awareness and adaptation can be enhance even at the service level.

5. Prototype implementation

In recent years the interest towards electronic education has been growing stronger. As a result of that many universities have developed and implemented their own systems for electronic and long-distance education. At the Faculty of Mathematics and Informatics of the University of Plovdiv a Distributed eLearning Center (DeLC) has been created. The DeLC infrastructure is introduced in various papers [8,9]. Here the logical structure of the center is described only briefly. DeLC is presented as a network, which consists of separate nodes, called eLearning (eLNs). The nodes model real units (laboratories, departments, faculties, colleges, and universities), which offer a complete or partial educational cycle. The functionality of an eLN is presented as a set of eServices. The configuration of the network edges is such as to enable the access, incorporation, use and integration of eServices located on the different eLNs. Electronic services are provided by physically separated servers or by a network of servers (eLearning Nodes, eLNs), which may dynamically connect to each other and form educational clusters (Fig. 1). Access to services can be either fixed or mobile. Remote eService activation and integration is possible only by means of previously defined virtual structures. In the network model we can easily create new structures, reorganize or remove existing structures (the reorganization is done on a virtual level, it does not affect the real organization).

In order to investigate workability of the proposed approach a prototype implementation of CA^3 is developing for DeLC application. Two eLearning nodes are under construction – a fixed access node in form of education portal and a mobile access node.

5.1. DeLC Education Portal. The education portal eLN^{fixed} (Fig.3.) supports a flexible delivery of teaching materials and eLearning services [10]. In many of the existing eLearning systems, the interaction with the teachers is somewhat static – it is achieved mainly through pre-defined templates for choosing information resources. The information resources are the electronic equivalent of the traditional students' books. Some of the existing systems use visualization and animation for improving the means of presenting the teaching materials. With DeLC portal we would like to develop an education environment which support an interactive, proactive and personalized process of education and stimulate the students' creative and innovative thinking and performance. For better perspicuity, the services included in the current version of the portal are grouped in the following modules: eTesting Module, eLectures Module, Planning & Control Module, Documentation & Statistics Module. The teaching material is presented according to the standard SCORM 2004 [11]. The reference model of SCORM is a set of interrelated technical standards, specifications and guidelines developed to meet the highest requirements for learning content and systems governing the content.

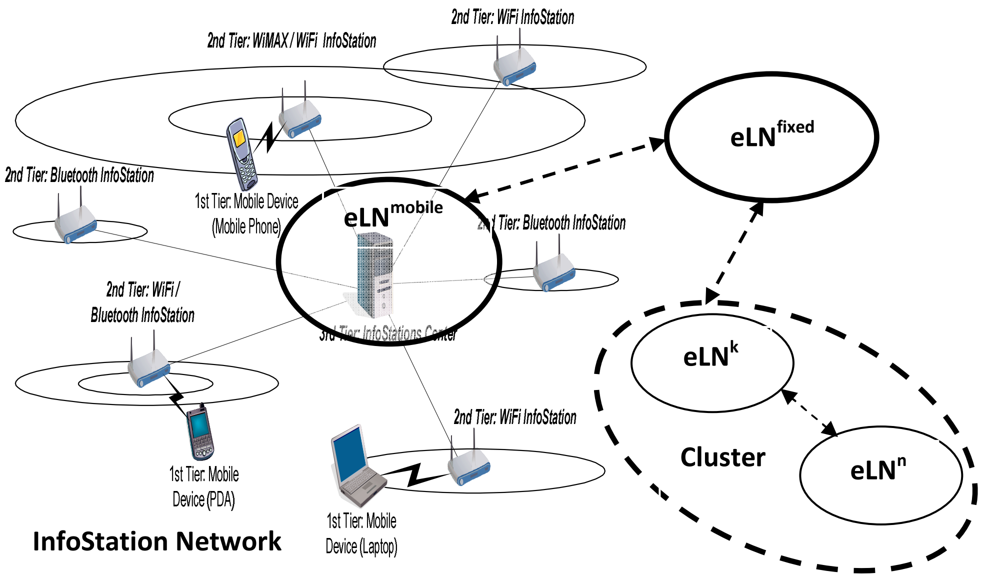

Fig.3. The DeLC network model

5.2. InfoStation-Based Network. An InfoStation-based network [12] is used as communication infrastructure for creating of the mobile eLearning Node eLN^{mobile} (Fig.3.). In order to adapt CA^3 to the communication network and due to device mobility (i.e. moving between geographically intermittent InfoStation cells) and user mobility (i.e. shifting to another mobile device) the following four basic scenarios are developed [13]: **'No change' scenario –** user doesn't change his/her location during service delivery;**'Change of device' scenario** - due to the user mobility, it is entirely possible that during a service provision, the user may shift to another mobile device;**'Change of InfoStation' scenario** - within the InfoStation paradigm, the connection between the InfoStations and user mobile devices is by definition geographically intermittent. With a number of InfoStations positioned around the University Campus, the users may pass through a number of InfoStation cells during the service session. This transition between InfoStation cells must be completely transparent to the user, ensuring the user has apparent un-interrupted access to the service; '**Change of device & InfoStation' scenario -** both procedures for device change and InfoStation change may be considered as autonomic procedures, independent of each other. Hence each of these may be executed and completed at any point inside the other procedure without a hindrance to it.

The middleware is implemented as a set of cooperating intelligent agents in order to model adequately the real distributed infrastructure, to allow for realization of distributed control models, to ensure pro-active middleware behavior and to use more efficiently and intelligently the information resources spread over different InfoStations. Moreover, this agent-oriented architecture could be easily extended with new agents (if necessary), which will cooperate with the existing ones and interact with them by means of a standardized protocol, e.g. the Agent Communication Language (ACL) [14]. Communication between a user mobile devices and a serving InfoStation could be realized in two ways: the InfoStation takes the initiative (the InfoStation's agents are pro-active), or the mobile device takes the initiative (the Personal Assistant agent is pro-active). In [15,16] the middleware is described in more detail.

6. Conclusion

Formalization of CA^3 is very important in order to develop supporting intelligent tools for a context-aware and adaptable delivery of eServices. First results are presented in [17].

7. Acknowledgement

The authors wish to acknowledge the support of the National Science Fund (Research Project Ref. No. ДО02-149/2008).

References

[1] B. Salem, J.A. Lino, M. Rauterberg, SmartEx: a case study on user profiling and adaptation in exhibition booths, Journal of Ambient Intelligennt Human Computing, Springer Verlag 2010, pp. 185-198

[2] A.K. Dey and G.D. Abowd. "Towards a better understanding of context and context-awareness," In: Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, New York, ACM Press, 2000.

[3] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, In: "Semantic Agent Systems: Foundations and Applications, Springer Verlag, A. Elci, M. T. Kone, M. A. Orgun (Eds.), Springer Verlag, 2010 (in print).

[4] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, An Approach for the Development of InfoStation-Based eLearning Architectures, Compt. Rend. Acad. Bulg. Sci., 61, No 9, 2008, Pp.1189-1198.

[5] Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma, An Approach for the Development of a Context-Aware and Adaptive eLearning Middleware, In: Intelligent Systems: From Theory to Practice, SCI 299. V. Sgurev et al. (Eds.). Springer-Verlag Berlin Heidelberg, 2010, ISBN: 978-3-642-13427-2. DOI: 10.1007/978-3-642-13428-9_26. Pp. 519-535.

[6] http://www.omg.org/mda/ (to date).

[7] M. Solanki, A. Cau, H. Zedan, Augmenting semantic web service descriptions with compositional specification, In: Proceedings of the 13th international conference on World Wide Web, pp. 544-552, ACM Press, 2004.

[8] S.Stoyanov, I.Ganchev, I.Popchev, M.O'Droma, From CBT to e-Learning, Journal "Information Technologies and Control", No. 4/2005, Year III, Pp. 2-10, ISSN 1312-2622

[9] S.Stoyanov, I.Popchev, O. Rachneva, A. Rachnev, DeLC – Technological Environment Supporting the Transition from CBT to eLearning, International Scientific Conference "Informatics in the Scientific Knowledge", 28-30 June, 2006, Varna Free University, 113-127.

[10] S. Stoyanov, I. Popchev, E. Doychev, D. Mitev, V. Valkanov, A. Stoyanova-Doycheva, V. Valkanova , I. Minov, DeLC Educational Portal, Cyberbetics and Information Technologies (CIT), 3/2010, Bulgarian Academy of Sciences (in print)

[11] SCORM 2004, http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/ (to date).

[12] R. H. Frenkiel and T. Imielinski, "InfoStations: The joy of 'many-time, many-where' Communications", WINLAB, Technical Report TR-119, April 1996.

[13] I. Ganchev, S. Stojanov, M. O'Droma, and D. Meere, "InfoStation-Based Adaptable Provision of M-Learning Services: Main Scenarios," International Journal "Information Technologies and Knowledge", Vol. 2 / 2008, Pp. 475-482.

[14] FIPA Agent Communication Language. http://www.fipa.org/repository/aclspecs.html (to date).

[15] S. Stoyanov, I. Ganchev, I.Popchev, M.O'Droma, V. Valkanova, Agent-Oriented Middleware for InfoStation-based mLearning Intelligent Systems, 5th IEEE International Conference on Intelligent Systems IS'10, 07.07. – 09.07.2010, London, IEEE Catalog Number: CFP10802-CDR, ISBN:978-1-4244-5164-7, Library of Congress:2009934065, pp. 91-95.

[16] I.Ganchev, S. Stoyanov, V. Valkanova, M. O'Droma, Service-oriented and Agent-based Architecture Supporting Adaptable Context-Aware Provision of Mobile e-Learning Services, IADIS e-Learning 2010 Conference, Freiburg, Germany, 26 - 29 July 2010, 97-104.

[17] M. Al-Sammarraie, F. Siewe, H. Zedan, Formalising Policies of a mLearning System using CCA, Proceedings of the 4th ACM International Workshop on Context-Awareness for Self-Managing Systems, Copenhagen, Denmark, 2010, ISBN:978-1-4503-0213-5

Stanimir Stoyanov
Faculty of Mathematics and Informatics
University of Plovdiv
236 Bulgaria Blvd.
4003 Plovdiv, Bulgaria
e-mail: stani@uni-plovdiv.bg