MODELING OF CONTEXT-AWARE SCHOOL E-LEARNING ENVIRONMENT

Todorka Glushkova

Plovdiv University "Paisii Hilendarski", 236 Bulgaria bul., Plovdiv, Bulgaria

Abstract

In recent years it becomes increasingly important to develop context-aware systems that can adapt their behaviors depending on the context and already applied in various areas of life- health, transport, communications, training and so on. The capture of context takes place by means of sensors that can be physical, virtual, or logical. The report will present a model of context-aware system for e-learning in secondary school. Formalization will be presented by mathematical notation for modeling mobile and context-aware systems - CCA (Calculus of Context-aware Ambients).

Key words: *CCA, Context-aware, SCORM, e-learning*

1. INTRODUCTION

The new Bulgarian law on pre-school and school education [13] sets up new horizons for secondary school in the country. According to Article 5. the main purpose of school education is intellectual, emotional, social, spiritual, moral and physical development, and support each student in accordance his age, needs, abilities and interests, as well as the formation of persistent attitudes and motivation for lifelong learning. According to Art. 6 the school education system must ensure validation of competences acquired through non-formal learning and informal learning as the focus is on the digital competences and learning skills.

All this gives grounds for concentrating attention on creating school environments and systems for electronic and distance learning, to provide educational resources and services depending on background knowledge, goals and needs of each student [4], [5]. These environments need to adapt to dynamically changing environment of the learner and to respond adequately to these changes, ie must to be context-sensitive [1].

The article presents a model of context-aware system for e-Learning [8] in secondary school, in that will take into account the change of location of the learner, his form of education, the type of the user device etc. In depending on the context, the learning environment will adapt to these changes. We will base our research on the definition of the Dey ([2], [3]), according to which the context is any information that can be used for categorization of the status of an identity (entity) - person, place or object, which are related to the interaction between user and application, including the user and application [10].

Formalization of the model will be represented by the mathematical notation Calculus of Context-aware Ambients (CCA). Will comment on the possibilities for change and improvement of the model and the possibilities for its prototyping.

2. MODEL OF SCHOOL E-LEARNING ENVIRONMENT

In the aforementioned law for pre-school and school education are defined various forms of teaching - daily, evening, extramural, individual, distance, combined etc. We can summarize these forms of training in two groups- attendance and non-attendance. The modeled school environment must provide services and educational resources of both groups of students as follows:

- For students in attended forms of training- lessons and tests for self-study and further training; the information for the weekly schedule, syllabus, dates of tests, consultations and other events related to their training in school; electronic student's school report book.

- For students in remote non-attendance forms of training – tutorials and e-lessons; training tests and examination tests in the school building; information about the curriculum, schedule of exams and consultations; electronic student's school report book

The Figure 1 is presented the architecture and key components of the school e-learning system.

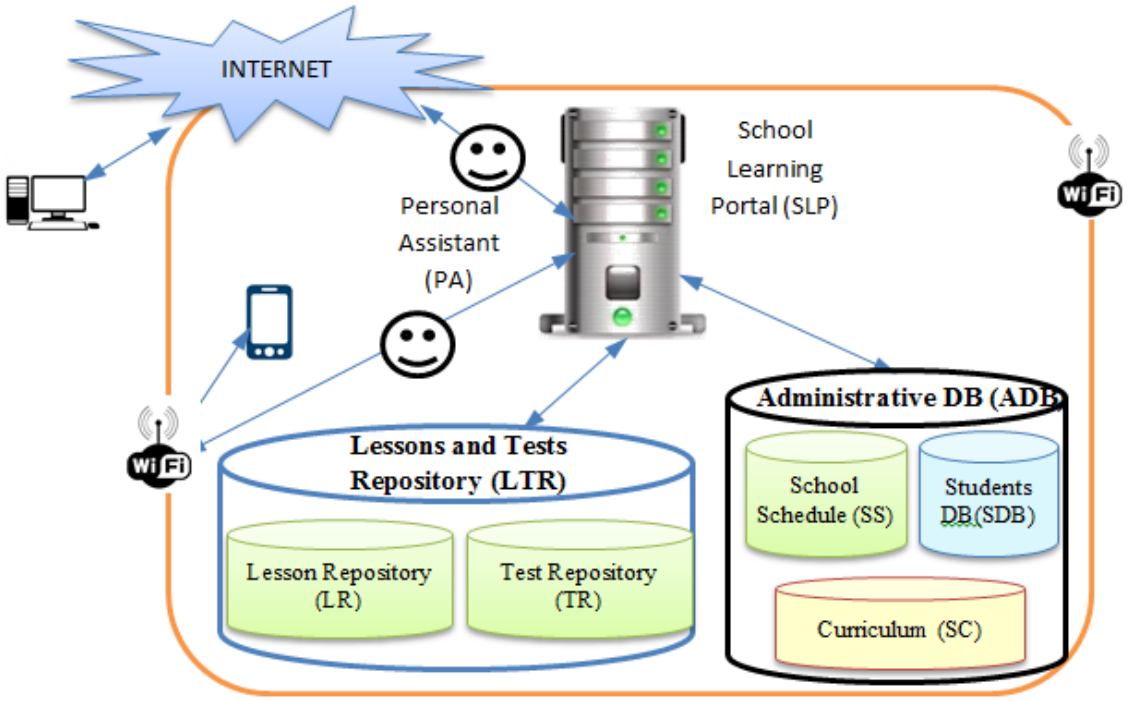


Figure 1. Architecture and main components of the system

2.1. Main components of the School Learning Environment

The school learning environment has a distributed architecture. The individual components are connected in a different way, depending on the current scenario. The school portal plays a key coordinating role in the model. In its runtime is integrated Learning Management System (LMS), which supports the implementation of the selected learning process. Portal architecture is layered as each layer is linked to the performance of a particular functionalities as user authorization and support of his profile; categorization of learning resources and services; search appropriate lessons and tests; collaboration, integration, local and remote services etc.

The system uses external administrative repository of data about students (Students DB-SDB) and the School Curriculum (SC) and repository with weekly schedule for regular students, exam sessions for independent and non-attendance students, current school events (SchoolSchedule), etc. Electronic lessons are stored in a separate repository (Lesson Repository-LR) with a description of meta information for each lesson- for which class, school subject, topic, type of student, type of user device - GSM, Tablet, PDA or PC. Similarly, repository is built and for tests (Test Repository-TR).

The device, by which the student accesses to learning resources and services, is also essential for the model. The standard and mobile devices have different characteristics and may differently display received information - lessons, tests, schedule and so on. Under this context, lessons and tests can be developed in three different variants: for GSM, PDA or PC.

Access to the learning environment can be achieved in two ways - through mobile device when it enters within the School Wifi, or by remote access directly to School Learning Portal. In both cases, the student will be presented into the system through his personal assistant (PA) [11].

The personal assistant is an intelligent agent that is mobile. Any mobile or stationary user device of the student has an instance of this agent. Upon identification in the e-Learning system PA communicates with other components and provides the necessary type of services and educational resources to the student. After the session, PA is updated by storing the changed profile information. The new PA instance is stored on the device, thus ensuring updating the user's profile in the next learning session.

2.2. Some services presented in the model

Let's look at some services that will be provided by the system:

- Electronic lesson (e-Lesson): This service realized searching for a suitable lesson, according to student's characteristics as grade, school subject, type of user device and form of education. If the student makes a request for this lesson from his mobile device, the system must provide this lesson in proper visualization of the device type
- Electronic test (e-Test) - this service is important to the learning process, testing is a tool for assessing students' knowledge and give feedback on the results. The electronic test allows users to practice and pass their exams through their standard or mobile devices. According to the type of device, the service can offer a suitable test, according to the characteristics of the device itself.
- Schedule of lessons, examination sessions and other events (e-Schedule) - This information is derived from the administrative database (ADB) and is provided of the student depending on his profile, form of education, type of device access and time characteristics. This information is generated and submit to the student as list of events.

3. CCA MODELING

The modeling of the system will be implemented through Calculus of Context-aware Ambients- CCA. The term “Ambient” plays a crucial role in this mathematical notation. This is the entity, which is used to describe an object or component - process, device, location, and so on. It can be defined as the limited space in which to perform certain actions. Each ambient has a name, boundaries and may contain other Ambients in yourself, and be involved in other ambient. Between the two Ambients there are three possible relationships: parent- child; child-parent and sibling-sibling. They are mobile and can enter into other ambient and get out from other one. Ambients can send messages to each other through the mechanism of handshaking. The symbol "::" means ambient relatives (sibling); "<>" - sending a message, "()" - receiving a message. CCA will use for ambient modeling in terms of process, location and capability.

The process is the core functionality of an ambient. With P|Q will indicate that two processes P and Q are performed in parallel. The process 0 doesn't perform anything and ends immediately. Label doubling (or replication) is "!". It is used to create a new copy of the process (! P), when it is necessary. The process n [P] denotes the ambient named “n” and process P that describes its behavior. Successive processes are separated by the symbol “.”

The location is an important parameter that is used in communication between Ambients. The symbol "↑" means parent; "↓" means child, and "::" - communication between siblings.

The possibilities are related to the movement of ambient in and out according to the others ambient. The possibility "del n" allows destruction of ambient named n, but only if it is empty, i.e. n [0].

According to present in Figure 1 architecture, the main components are School Learning Portal (SLP), Lessons and Tests Repository (LTR), Administrative Data Bases (ADB) and mobile or standard student’s devices. To create CCA model it is necessary to present these components as ambients [6]. The student will be represented in the model through its device for access to the e-learning environment - PC, GSM or PDA. The school educational portal is the most important component in the model and will be represented by ambient SLR. It will coexist with its siblings Lessons and Tests Repository (LTR) and Administrative Data Bases (ADB), represented by ambients LTR and ADB. Lesson Repository

(LR) and Tests Repository (TR) are children of ambient LTR. Ambients Students DB (SDB), School Curriculum (SC) and School Schedule (SS) are children of ambient ADB. The CCA model is represented graphically in Figure 2.

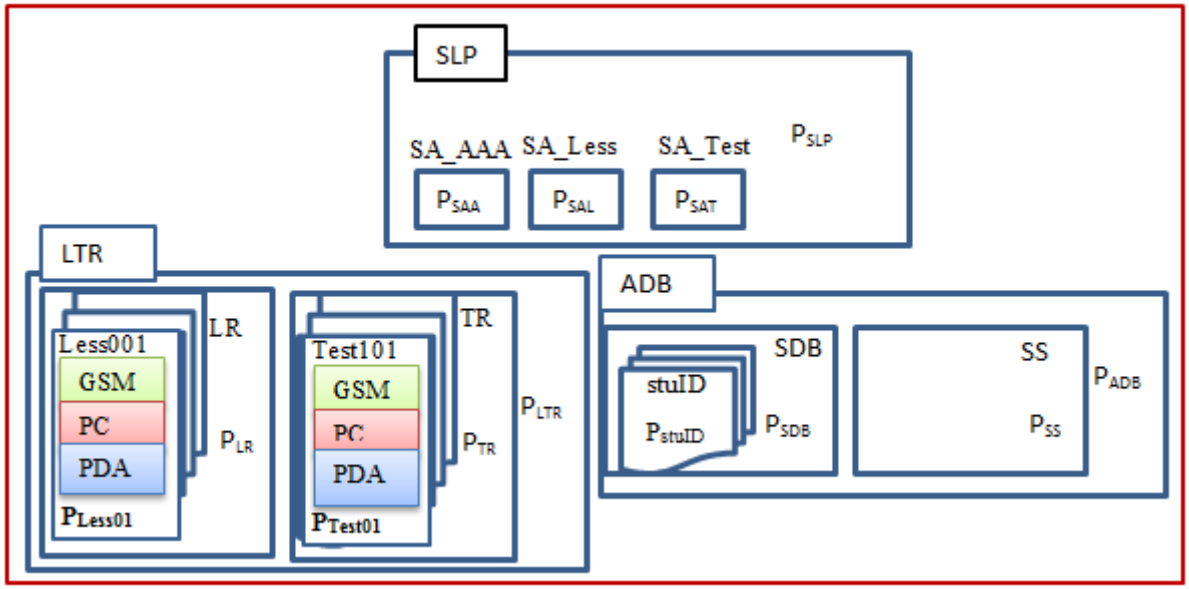


Figure 2. CCA model

Text description of CCA model requires a description of the notation, according to the presented model. The constants and variables are presented in Table1 and Table2.

Notation	Description	Notation	Description
SLP	School Learning Portal	LR	Lessons Repository
LTR	Lesson Test Repository	TR	Tests Repository
GSM	Mobile Phone	SDB	Students Data Base
PC	Personal Computer	ADB	Administrative Data Base
PDA	PDA device	Null	NULL
SLIST	Services List	ACK	acknowledgment
SS	School Schedule	SC	School Curriculum

Table 1. Constants in the model

Notation	Description	Values	Notation	Description	Values
stuID	Student ID	S201, S202	reply	Reply	OK, Denied, Content
dtype	Device type	GSM,PDA,PC	content	Lesson or Test	CONTENT
aname	Ambient name	PC201	sList	List of services	SLIST
lessID	Lesson ID	L001, L002…	ack	acknowledgment	ACK
testID	Test ID	T101, T102	PA	Personal Assistant	PA201, PA202…

Table 2. Variables in the model

3.1. CCA modelling of School Learning Portal - SLP

The main components of SLP are: ambient for identification SA_AAA; ambient for request of e-lesson SA_Less and ambient for test request - SA_Test. The ambient that representing the user in the system will be his personal assistant PA_i. The three permanent Ambients located in SLP are SA_AAA, SA_Less and SA_Test. They are responsible for the implementation of the relevant service - identification, delivery of lesson or test. Each of these services is managed by intelligent Agent Specialist (SA) . These three Ambients are siblings to each other. The personal assistant PA_i is child of SLP and also is sibling of SA_AAA, SA_Less and SA_Test.

3.1.1. SA_AAA Ambient

Each student must make its authorization, authentication and accountability(AAA) in the system to be able to use relevant services. To start the process of identification, ambient SA_AAA must receive a request with the following parameters: stuID, dType, PA_i and send it to parent SLP. The parameter StuID is used to check the rights of that student, dType- to obtain the list of services, corresponding with device type; and PA_i- to know where to send the response to the request. When this ambient receives a response from SLP, it forwards the result through PA_i to the user device.We can model this behavior by the following process:

$$P_{SAA} \triangleq !::(stuID, dType, PA_i).SPL \uparrow < stuID, dType, PA_i > .SPL \uparrow (ack, PA_i, sList).PA_i ::< ack, sList > .0$$

The process of sending the request and receiving the response are performed as successive processes. The personal assistant PA_i receives the reply of the request and send it to the student. He can choose any of the proposed services- e- lesson, e-test or information about upcoming school events.

3.1.2. SA_Less Ambient

This ambient is responsible for all requests to launch an electronic lesson. The request contains the following parameters: lessID, stuID, dType, PA_i. LessID is parameter for selected lesson; stuID- student, dType- type of user device and PA_i-personal assistant of the student who will receive the answer to the query. The behavior is modelled as follows:

$$!::\ (lessID,\ stuID,\ dType,\ PA_i).SLP \uparrow < lessID,\ stuID,\ dType,\ PA_i > .\ 0 \quad (1)$$

The response from SLP has three parameters (lessID, reply, PA_i), as lessID is requested lesson (because this student may to requested some lessons), "reply" is content of this lesson or refusal to be used and PA_i is personal assistant of the student who will receive the answer.

$$!\ SLP \uparrow\ (lessID,\ reply,\ PA_i).\ PA_i\ ::\ < lessID,\ reply > .0 \quad (2)$$

Then the overall behavior of SA_Less ambient is described as follows:

$$P_{SAL} \triangleq Eq.\ (1)\ |\ Eq.\ (2)$$

3.1.3. SA_Test Ambient

This ambient is responsible for the execution of all requests for tests of students in the school. Due to the characteristics of Bulgarian school education, the tests in the system are just for training. The tests, by which are assessed students' knowledge in all forms of training is carried out only in the presence of invigilators and teachersThe tests, like the lessons are related to selected school subject and the class of the student. SA_Test ambient gets a request from the personal assistant PA_i of student in which are included the next parameters: testID- identifier of selected test; stuID- identifier of the student; dType - the type of user device and PA_i. Then the behavior of SA_Test is modelled as follows:

(3) $! :: (testID,\ stuID,\ dType,\ PA_i).\ SLP \uparrow < testID,\ stuID,\ dType,\ PA_i > .\ 0$

The answer that the ambient SA_Test gets from the parent ambient SLP includes three parameters (testID, reply, PAi). Similarly to previous requests testID is the identifier of the requested test; reply is one of these entities: or content of the test or refusal for its providing and using. PA_i is personal assistant of the student who will receive this answer.

(4) $!SLP \uparrow\ (testID,\ reply,\ PA_i).\ PA_i :: < testID, reply > .0$

Then the overall behavior of the SA_Test ambient can be described as parallel processes (3) and (4) as follows:

$$P_{SAT} \triangleq Eq.\ (3)\ |\ Eq.\ (4)$$

3.1.4. SLP Ambient

The next step in creating of the model is to look at the behavior of SLP Ambiente and his communication with ambients ADB and LTR, who are its relatives; as and with SA_AAA, SA_Less, SA_Test, who are its children. Now we will describe the communication between this ambient and its internal components:

- SA_AAA – After receiving the request from SA AAA, the ambient SPL forward it to ambient sibling ADB, where you need to check whether the user is a student at the school. If so, check its class and form of education. Then depending on the type of personal user device, returns a list of appropriate services for this student and his type of user device (PC,GSM or PDA). Prior to forward the list of services need to check whether the student still awaiting answers or he already is come out of the system (i.e. is destroyed and the instance of his PA_i). Communication can be described as follows:

(5) $$\begin{pmatrix} !SA_AAA \downarrow (stuID, dType, PA_i).ADB :: < AAAreg, stuID, dType, PA_i > .0 \mid \\ !ADB :: (ack, PA_i, sList).\left(has\left(PA_i\right)\right)?\ SA_AAA \downarrow < ack, PA_i, sList > .0 \end{pmatrix}$$

- SA_Less and SA_Test- Communication between ambient SLP with children ambients SA_Less and SA_Test are handled similarly. The scenario for SLP includes the following steps: After receiving the request from SA_Less or SA_Test, ambient sends a request to LTR, where after checking for the presence of such a service, the result is returned. The request includes the following parameters: lessID or testID-identifier of selected lesson or test; stuID- identifier of the student; dType - type device and PAi- personal assistant. According to the scenario, we can model the behavior of this ambient as follows:

(6) $$\begin{pmatrix} !SA_Less \downarrow (lessID, stuID, dType, PA_i).LTR :: < lessID, stuID, dType, PA_i > .0 \mid \\ !LTR :: (lessID, reply, PA_i).(\text{has}(PA_i))?SA_Less \downarrow < lessID,\ reply, PA_i > .0 \end{pmatrix}$$

and

(7) $$\begin{pmatrix} !SA_Test \downarrow (testID, stuID, dType, PA_i).LTR :: < testID, stuID, dType, PA_i > .0 \mid \\ !LTR :: (testID, reply, PA_i).(\text{has}(PA_i))?SA_Test \downarrow < testID,\ reply, PA_i > .0 \end{pmatrix}$$

We can model the behavior of SLP ambient by the following process:

$$P_{SLP} \triangleq Eq.\ (5)\ \big|\ Eq.\ (6)\ \big|\ Eq.\ (7)$$

3.2. CCA modelling of Lesson and Test Repository-LTR

Lesson and Test Repository (LTR) ambient will also modeled as ambient that provides searching and supply of educational resources - lessons and tests. After receiving the request from the SLP, the ambient forward it to some of its internal ambient LR or TR, who cultivate it and return an answer. This response contains the appropriate learning resource suitable for user device form or refusal thereof. Ambiente LTR returns this response to the SLP. Will describe modeling of LTR, ambients LR and TR and their internal ambients.

3.2.1. LTR Ambient

Depending on the type of request received from the SLP (lessID, stuID, dType, PAi) or (testID, stuID, dType, PAi), ambient LTR forward it to the appropriate internal ambient LR or TR. As a result LTR receives answer in format respectively as (lessID, stuID, dType, reply, PAi) or (testID, stuID, dType, reply, PAi). The value of the parameter "reply" is or content on the learning resource, or refusal to use.

$$\begin{pmatrix} !SLP::(lessID, stuID, dType, PA_i).LR\downarrow<lessID, stuID, dType, PA_i>.0 \mid \\ !LR\downarrow(lessID, reply, PA_i).\text{SLP}::<lessID,\ \text{stuID,dType},reply, PA_i>.0 \end{pmatrix} \quad (8)$$

$$\begin{pmatrix} !SLP::(testID, stuID, dType, PA_i).TR\downarrow<testID, stuID, dType, PA_i>.0 \mid \\ !TR\downarrow(testID, reply, PA_i).\text{SLP}::<testID, stuID, dType, reply, PA_i>.0 \end{pmatrix} \quad (9)$$

Then the behavior of SLP ambient we can describe as:

$$P_{LTR} \triangleq Eq.\ (8)\ \mid Eq.\ (9)$$

3.2.2. LR and TR Ambients

The LR ambient is responsible for providing the requested lesson in proper form, depending on the type of user device. The algoritm includes the next steps: LR receives a request from its parent LTR in the form (lessID, stuID, dType, reply, PA_i); verifies that in the Lesson Repository has such a lesson and sends a message to child ambient LessID. As a result receive a response in format (reply, dType, PAi), where "reply" is either: the content of the lesson in the appropriate form for dType; or information about lack or refusal to use this learning resource. Finaly LR returns response to parent LTR in format (lessID, stuID, dType, reply, PA_i).

$$P_{LR} \triangleq \left(!\uparrow(lessID, stuID, dType, PA_i). \left(\begin{pmatrix} has(LessID)?LessID\downarrow<dType, PA_i>.0 \mid \\ !LessID\downarrow(dType, reply, PA_i).\text{SLP}\uparrow<lessID, stuID, dType, reply, PA_i>.0 \end{pmatrix} \mid \left(\neg has(LessID)?\text{SLP}\uparrow<lessID, stuID, dType, NULL, PAi>.0 \right) \right) \right)$$

Similarly, we will model the behavior of ambient TR:

$$P_{TR} \triangleq \left(!\uparrow(testID, stuID, dType, PA_i). \left(\begin{pmatrix} has(TestID)?TestID\downarrow<dType, PA_i>.0 \mid \\ !TestID\downarrow(dType, reply, PA_i).\text{SLP}\uparrow<testID, stuID, dType, reply, PA_i>.0 \end{pmatrix} \mid \left(\neg has(TestID)?\text{SLP}\uparrow<testID, stuID, dType, NULL, PAi>.0 \right) \right) \right)$$

3.2.3. LessID and TestID Ambients

LessID and TestID ambients receive a request from LR or TR. They check whether there is a requested learning resource in the required format in their internal ambients PC, PDA or GSM and return the result. This result may or content of the lesson (or the test), or information that in the Lesson Repository lack resource in required for visualization format. We will process as group named dType ambients PC, PDA and GSM, as they have similar characteristics. The behavior of dType will describe as follows:

$$P_{LessID} \triangleq \left(\begin{array}{l} !\uparrow (dType, PA_i). \\ \left(\left(\begin{array}{l} has(dType)?\, dType \downarrow < PAi > .0 \mid \\ !dType \downarrow (reply, PAi).LR \uparrow < dType, reply, PAi > .0 \end{array} \right) \mid \right) \\ \left(\neg has(dType)?\, LR \uparrow < dType, NULL, PAi > .0 \right) \end{array} \right)$$

Similarly:

$$P_{TestID} \triangleq \left(\begin{array}{l} !\uparrow (dType, PA_i). \\ \left(\left(\begin{array}{l} has(dType)?\, dType \downarrow < PAi > .0 \mid \\ !dType \downarrow (reply, PAi).TR \uparrow < dType, reply, PAi > .0 \end{array} \right) \mid \right) \\ \left(\neg has(dType)?\, TR \uparrow < dType, NULL, PAi > .0 \right) \end{array} \right)$$

3.2.4. dType Ambient

The ambient dType is summarized presentation of ambients PC, PDA, GSM, who are children of any ambient LessID or TestID. Once received request with parameter (PA_i), returns response in format (dType, reply, PA_i):

$$P_{dType} \triangleq !LessID \uparrow (PA_i).LessID \uparrow < dType, reply, PAi > .0$$

or

$$P_{dType} \triangleq !TestID \uparrow (PA_i).LessID \uparrow < dType, reply, PAi > .0$$

3.3. CCA modelling of Administrative Data Bases(ADB)

Administrative Data Bases will be present in the model as ambient ADB, which is responsible for checking the data for students in the school; the curriculum, according to the class of individual student; as well as weekly schedule of lessons; examination sessions and other school events. Ambiente ADB is a sibling of SLP and has internal ambients: SDB (Students DB); SC (School Curriculum) and SS (School Schedule). All these ambients are provided by the data store of software product AdminL, which is mandatory for all schools in the country. The information is available both locally and through the Internet at http://am.mon.bg ([14]).

3.3.1. ADB Ambient

ADB ambient receives a request for identification of student from sibling ambient SLP with the following parameters (AAAreg, stuID, dType, PAi). He forwards this request to the sibling ambient SDB. As a result, receive a response from SDB with the following parameters: (reply, dType, PAi, sList). Parameter "reply" has value ACK or NULL, and sList is a list of services that can be appropriate for this student, depending on his class and form of education. It then sends a reply to the SLP, which contains parameters (ack, PAi, sList):

$$P_{ADB} \triangleq \begin{pmatrix} !SLP::(AAAreg, stuID, dType, PA_i).SDB\downarrow< AAAreg, stuID, dType, PA_i > .0 | \\ !SDB\downarrow(reply, dType, PA_i, \text{sList}).\text{SLP}::< ack, PA_i, sList > .0 \end{pmatrix}$$

3.3.2. SDB Ambient

SDB ambient receives a request to identify of the student from parent ambient ADB with parameters (AAAreg, stuID, dType, PA_i). It checks if there is such a student by sending a request to the respective ambient - child StuID. As a result receives a response in format (reply, classID, trType, PA_i), where value of "reply" is affirmative ACK, or negative NULL; classID the class of this student and trType is a type of training for the student (regular, evening, extramural, individual, distance etc.). The last two parameters are necessary to generate the corresponding list of services. If the value is NULL, immediately returns a response to the parent ambient ADB with parameters (NULL, dType, PA_i, sList), as sList is an empty list. If the answer is affirmative ACK, the SDB ambient sends a request to the sibling SC ambient. The request contains the following parameters: classID, trType, dType, PA_i.

In response gets list of learning resources sList1 with lessons and tests in all school subjects of the curriculum for the class and type of training that are in a form suitable for the user device. In parallel, sends a request to ambient sibling SS with a request to provide a list of weekly scheduled lessons, examination sessions and other events appropriate to the student depending on his class and form of education. The parameters of this query are: stuID, classID, type, dType, PA_i. The response contains a list of upcoming events for the student - sList2. From both lists sList1 and sList2 SDB ambient forms a common list of all services and events sList, which returns as a response to ADB:

$$P_{SDB} \triangleq \begin{pmatrix} !ADB\uparrow(AAAreg, stuID, dType, PA_i). \\ \begin{pmatrix} \neg has(stuID)?.ADB\uparrow< NULL, dType, PA_i, \text{sList}).0| \\ \begin{pmatrix} has(stuID)?.stuID\downarrow< PA_i > .stuID\downarrow(ack, classID, trType, PA_i).0| \\ \begin{pmatrix} !SC::< classID, trType, dType, PA_i > .0| \\ !SC::(dType, PA_i, sList1).0| \end{pmatrix}| \\ \begin{pmatrix} !SS::< stuID, classID, trType, PA_i > .0| \\ !SS::(dType, PA_i, sList2).0| \end{pmatrix}| \\ SDB\uparrow< ack, dType, PA_i, sList > .0 \end{pmatrix} \end{pmatrix} \end{pmatrix}$$

3.3.3. StuID Ambient

The ambient StuID communicates only with its parent SDB. It receives a request and returns the response with parameters (ack, classID, trType, PAi) as follows:

$$P_{StuID} \triangleq \begin{pmatrix} !SDB\uparrow(PA_i).0| \\ !SDB\uparrow< ack, classID, trType, PA_i > .0 \end{pmatrix}$$

3.3.4. SC (School Curriculum) Ambient

SC ambient communicates only with his sibling SDB. It receives a request from the SDB with parameters (classID, trType, dType, PAi) to retrieve the list sList1 with educational resources suitable for the student, his class, form of education and type of user device. The response has parameters (dType, PA_i, sList1) as follows:

$$P_{SC} \triangleq \begin{pmatrix} !SDB :: (classID, trType, dType, PA_i).0 \mid \\ !SDB ::< dType, PA_i, sList1 > .0 \end{pmatrix}$$

3.3.5. SS (School Schedule) Ambient

SS ambient is responsible for providing a list of events that relate to the student. When receives a request from the SDB with parameters (stuID, classID, trType, PAi), retrieves the list of events sList2. The answer has parameters (dType, PAi, sList2) as follows:

$$P_{SS} \triangleq \begin{pmatrix} !SDB :: (stuID, classID, trType, PA_i).0 \mid \\ !SDB ::< dType, PA_i, sList2 > .0 \end{pmatrix}$$

4. CONCLUSIONS AND FUTURE PLANS

The presented CCA-model of context-aware system of e-learning in secondary schools could be the basis for the development of real school environment. The model includes only the basic services as: searching and supply a lessons or tests and receiving a list of school events relating to the characteristics and requirements of the individual student. This simulated learning environment depends on the form of training of every student (under the new law of education it will be frequently changing parameter); from his class; type of device for access to learning resources; and partially from his location. We plan to develop a context-aware model according to the location and scenarios that will allow the student if he is in the area of the school to receive messages concerning his personal involvement in the school events, and if not - messages for that where and when he need to be in the next minutes. Some interesting for CCA-modeling services are also: student's notebook and teacher's diary, which will dynamically reflect the student's knowledge, in line with dynamically-changing environment.

The next step is formalization of the model. We use Programming Language for CCA - cca PL. Through this language we will create a prototype that would allow verification of the model [7].

This School Learning Environment is based on the conceptual framework of Virtual Educational Space VELSPACE, inherited DeLC [9], [12]. We created SCORM-based school education portal, which is used in the real learning process at school. [4]. The portal's learning resources are based on standard SCORM, but we plan to create an intelligent digital library with lessons and tests, as well as multi-agent model of the educational space to meet more fully the set objectives.

ACKNOWLEDGEMENT

The report is part of the work on the project IT 15- FMIIT-004 "Research in the field of innovative ICT business orientation and training" to fund "Scientific Researches" of Plovdiv University "Paisii Hilendarski".

REFERENCES

[1] A. Schmidt and K. V. Laerhoven. How to build smart appliances. IEEE Personal Communications, pp. 66-71, ISSN 1070-9916, 2001.

[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] A. K. Dey, Understanding and Using Context, Personal and Ubiquitous Computing Journal, Vol.5, No.1, pp.4-7. 2001

[4] Glushkova, T., Adaptive model for e-Learning in secondary school, E-Learning, book" Long-Distance and Lifelong Perspectives", ISBN 978-953-51-0250-2, 2011, http://www.intechopen.com/download/get/type/pdfs/id/31951

[5] Glushkova, T. Personalization and User Modeling in Adaptive E-Learning Systems for Schools, E-Learning - Instructional Design, Organizational Strategy and Management, Dr. Boyka Gradinarova (Ed.), ISBN: 978-953-51-2188-6, InTech, DOI: 10.5772/61084. Available from: http://www.intechopen.com/books/e-learning-instructional-design-organizational-strategy-and-management/personalization-and-user-modeling-in-adaptive-e-learning-systems-for-schools

[6] Mohammed H. Al-Sammarraie, Policy-based Approach For Context-aware Systems, PhD Thesis, Software Technology Research Laboratory, De Montfort University, Leicester - United Kingdom, July 2011.

[7] S. Stoyanov, I. Ganchev, D. Mitev, V. Valkanov, M. O'Droma, Service-oriented and Agent-based Architecture Supporting Adaptable, Scenario-based and Context-aware Provision of Mobile e-Learning Services, International Journal of Computer Information Systems and Industrial Management Applications, Volume 3, Dynamic Publishers, Inc., USA, (2011), ISSN 2150-7988, pp. 771-779.

[8] S. Stoyanov, V. Valkanova, I. Ganchev, M. O'Droma, An Approach and Architecture Supporting Context-Aware Provision of mLearning Services, IEEE Second International Conference on Mobile, Hybrid, and On-line Learning, 10-16 February 2010, St. Maarten, pp.1116.

[9] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, A. Elci, M. T. Kone, M. A. Orgun (Eds.): "Semantic Agent Systems: Foundations and Applications", Book Series: Studies in Computational Intelligence, Vol. 344, Springer Verlag, 2011, ISBN: 978-3-642-18307-2

[10] Valkanov, V., Context-oriented management of electronic services. Sofia, Bulgaria: Academic Publishing House "Prof. Marin Drinov ", 2013, ISBN 978-954-322-701-3.

[11] Kehayova, I., P. Malinov, S. Stoyanov, Intelligent Personal Assistants in a Virtual Learning Space, DeLC – Past, Present, Future/ In: Proceedings of the International Conference – From DeLC to VELSPACE, Plovdiv, 26-28 March 2014, 29-40. Third Millennium Media Publications. ISBN: 0-9545660-2-5, http://fmi-plovdiv.org/GetResource?id=1809

[12] S. Stoyanov, Context-Aware and Adaptable eLearning Systems, Internal Report, Software Technology Research Laboratory, De Montfort University, Leicester, UK, August, 2012.

[13] Law on pre-school and school education - http://mon.bg/?go=page&pageId=7&subpageId=57, visited on 17 March 2016

[14] Information system for Bulgarian schools - http://am.mon.bg, visited on 16 March 2016