

ОБУЧЕНИЕ И ТЕСТВАНЕ В КОНТЕКСТНО-ЗАВИСИМА УЧИЛИЩНА УЧЕБНА СРЕДА

Тодорка Глушкова

Резюме. Създаването на контекстно-зависими системи, които се адаптират динамично към параметрите на околната среда е предизвикателство и необходимост в съвременното развитие. Целта на статията е да се представи един модел на контекстно-зависима среда за електронно обучение в средните училища и неговата CCA¹ формализация.

Ключови думи: CCA, Context-aware, SCORM, e-learning

1. Въведение

Динамиката в развитието на съвременното общество изисква създаване на нови условия за обучение през целия живот. Новият закон за предучилищното и училищно образование [9] регламентира в значителна степен този процес. Законът регламентира правилата по които ще се обучават ученици на дневна, задочна, самостоятелна, индивидуална, дистанционна, комбинирана и пр. форми на обучение. Всичко това дава основание за концентриране на вниманието върху създаването на училищни среди и системи за електронно и дистанционно обучение, които да предоставят учебните ресурси и услуги в зависимост от знанията, целите и потребностите на всеки ученик, независимо от неговата форма на обучение. Тези среди трябва да се адаптират към динамично променящата се околна среда на обучаемия и да отговарят адекватно на тези промени, т.е. трябва да бъдат контекстно-зависими.

¹ CCA – Calculus of Context-aware Ambients

Контекстно-зависимите системи могат да адаптират поведението си в зависимост от конкретния контекст и вече се прилагат в различни области от живота- здравеопазване, транспорт, комуникации и пр. Съществуват различни дефиниции на понятието „контекст“ – като активен процес по разпознаване на заобикалящата среда; като знанията на компютъра за елементите на заобикалящата потребителя среда и пр. Ние ще базираме нашите изследвания върху дефиницията на Deu [1, 2] според който контекст е всяка информация, която може да се използва за категоризиране на състоянието на една идентичност (entity) – човек, място или обект, които се смятат за свързани с взаимодействието между потребител и приложение, включително и самите потребител и приложение.

Контекстната зависимост на една система означава, че тя следи промените в околната среда, и използва информацията от тази среда, за да се адаптира, така че да осигурява информация или услуга на потребителя. Улавянето на контекста става с помощта на сензори, които могат да са физически, виртуални или логически. В разработвания модел на училищна учебна среда ще разгледаме контекстната зависимост като способността на системата да открива, идентифицира и интерпретира събитията от околната среда и според естеството им да предприема компенсиращи действия. Основните компенсиращи действия са персонализацията и адаптацията. Персонализацията е способността на системата да се пригоди към индивидуалните особености, характеристики, желания, намерения и цели на учениците, а адаптацията е способността за самонастройка на системата към останалите контекстни области като учебен предмет, местоположение на ученика, устройство за достъп и т.н. Формализацията на контекстно-зависимата система се представя чрез математическа нотация за моделиране на мобилни и контекстно-зависими системи CCA (Calculus of Context-aware Ambients), която се базира на контекстно-зависими амбиенти. Един амбиент е ентити, което се използва, за да опише някой обект или компонент (например личност, процес, устройство, местоположение и т.н. [5]). Той има име, граници и може да съдържа други амбиенти в себе си. Може да бъде мобилен и има способността да комуникира с други амбиенти.

В статията се разгледаме един модел на контекстно-зависима система за самостоятелно обучение на ученици в средното училище, която да отчита времевите характеристики, промяната в местоположението на ученика, вида на устройството му за достъп до образователните ресурси, неговата форма на обучение и пр. като динамично се самонастройва и адаптира към тези промени. Математическата нотация CCA ще използваме за формализация на отделните

учебни сценарии. Ще се коментират възможностите за промяна и усъвършенстване на модела, както и възможностите за неговото прототипиране.

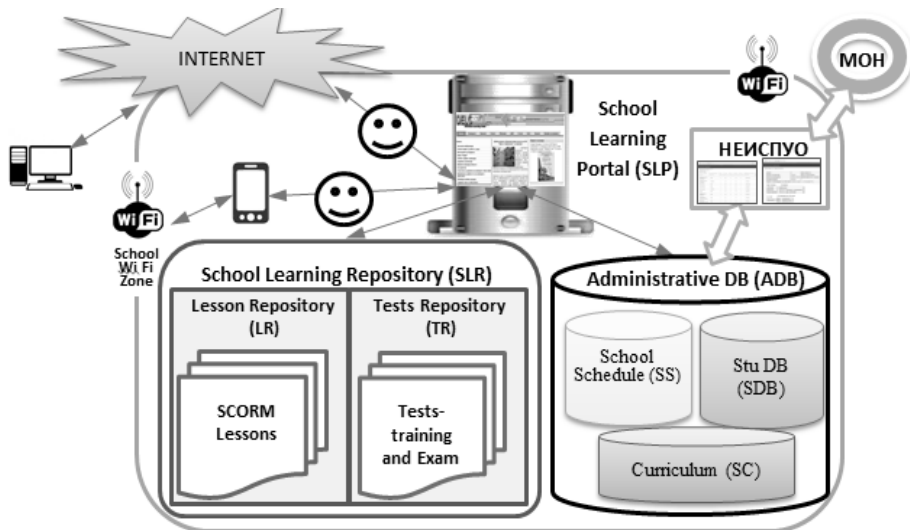
2. Архитектура на училищна образователна среда

Според ЗПУО за средното училище са дефинирани различни форми на обучение, които можем да обобщим в две по-големи групи- присъствени и неприсъствени. В статията ще концентрираме вниманието си върху учениците от неприсъствените форми на обучение – самостоятелно, индивидуално, дистанционно и комбинирано, които използват образователните ресурси от своите стандартни или мобилни устройства, както отдалечено, така и в района на училищната WiFi зона. Те трябва да могат да се обучават, да проверяват знанията си чрез тренировъчни тестове, да бъдат информирани за датите и мястото на изпитите, консултациите и други училищни събития.

Училищната образователна среда има разпределена архитектура (Фигура 1.) Отделните компоненти се свързват и обменят информация по различен начин, в зависимост от текущия сценарий. Училищният образователен портал изпълнява основна координираща роля в модела. В неговия run-time е вградена SCORM LMS, която поддържа изпълнението на избрания от ученика учебен процес. Порталната архитектура е многослойна като всеки от слоевете е свързан с изпълнението на определена функционалност- оторизиране на потребителя и поддръжка на неговия профил; категоризиране на учебните ресурси и услуги; търсене на подходящи уроци и тестове; интеграция, управление на локални и отдалечени услуги и пр.[3] Средата използва и външно административно хранилище от данни с информация за учениците и учебния план НЕИСПУО, чрез която се осигурява директен обмен на данни с МОН. Осигурено е и хранилище съдържащо текущите събития – изпитни сесии, съобщения и пр. – SchoolSchedule. Електронните уроци се съхраняват в отделно хранилище (Lesson Repository-LR) с описание на мета информация за всеки урок- за кой клас, учебен предмет, тема, тип на ученика, тип на устройството – GSM, Tablet или PC. По аналогичен начин е изградено и хранилището с тестове (Test Repository-TR).

Устройството, с което ученикът достъпва да учебните ресурси и услуги също е от съществено значение за модела. Различните мобилни и стандартни устройства имат различни характеристики и могат да визуализират по различен начин получаваната информация. Според този контекст, уроците и

тестовите могат да се разработят в три различни варианта – за телефон, таблет или PC.



Фигура 1. Архитектура на средата и основни компоненти

Достъпът до средата може да се осъществи по два начина- чрез персоналния асистент PA на мобилното устройство, когато влезе в обхвата на School Wifi Zone, или отдалечено чрез Интернет достъп директно към училищния портал. И в двата случая ученикът ще се представлява в системата от своя персонален асистент (PA) [4]. Асистентът е интелигентен агент, който е мобилен и може да променя местоположението си според текущия сценарий [7]. Всяко мобилно или стационарно устройство на ученика има инстанция на този агент. При идентификация в системата PA комуникира с останалите компоненти и осигурява подходящо представяне на услугите и учебните ресурси на ученика. След края на сесията, PA се актуализира като съхранява променения профил на потребителя. Негова инстанция се съхранява на съответното устройство, като по този начин се гарантира актуализиране на информацията за ученика при следващото му влизане в системата.

Нека разгледаме някои услуги, които могат да се предоставят от образователната среда:

- Електронен урок (e-Lesson) – тази услуга реализира търсене на подходящ урок за съответния клас и учебен предмет. Ако ученикът прави заявка за този урок от своето мобилно устройство, системата

трябва да му предостави този урок в подходящ за визуализация на съответното устройство вид

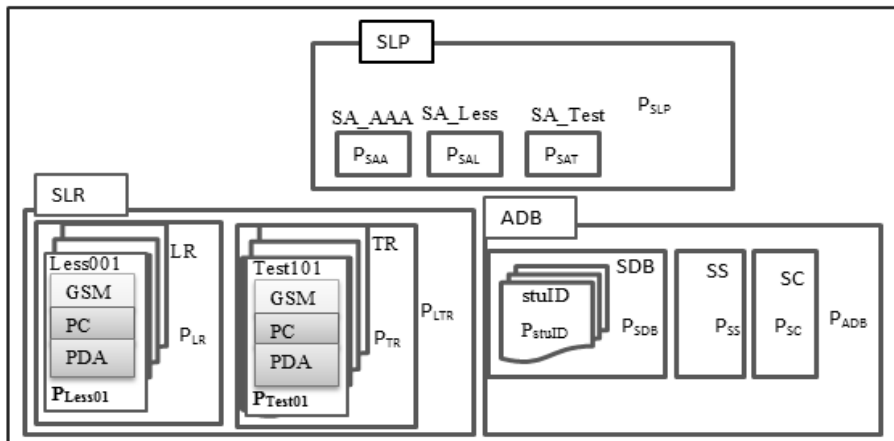
- Електронен тест (e-Test) – системата е проектирана да осигурява два вида тестове- тренировъчни и изпитващи. Първите дават възможност на учениците да проверят своите знания и да продължат обучението си, според тях. Изпитващите тестове се стартират от преподавател и се провеждат в сградата на училището. Според типа на устройството за достъп, услугата може да предложи подходящ тест, според неговите характеристики.
- Разписание на учебните занятия, изпитни сесии и други събития (e-Schedule) – тази информация се извлича от административната база от данни (ADB) и се предоставя на ученика в зависимост от неговия профил, форма на обучение, вид на устройството за достъп и времевите характеристики. Тази информация се генерира и подава на ученика като списък от събития с характеристики дата, час, местоположение, вид на събитието, информация за събитието.

3. Моделиране на контекстно-зависимата учебна среда

Моделирането на системата ще се реализира чрез Calculus of Context-aware Ambients – CCA, в който основна роля има амбиентът. Понятието „ambient“ означава околна, обкръжаваща. Това е ентити, което се използва да се опише някой обект или компонент- процес, устройство, местоположение и пр. Той може да се определи като ограничено пространство, в което се извършват определени действия. Всеки амбиент има име, граници и може да съдържа други амбиенти в себе си, както и да бъде включен в друг амбиент. Между два амбиента има три възможни взаимоотношения – parent, child and sibling. Те са мобилни и могат да излизат и влизат в други амбиенти. Амбиентите могат да изпращат съобщения помежду си чрез механизма на ръкостискането (handshaking). Символът „:“ означава амбиенти роднини; „<“ – изпращане на съобщение, а „(“ – получаване на съобщение. CCA ще използваме за моделиране на амбиенти в смисъл на процес, местоположение и възможност. Процесът е основната функционалност на един амбиент. С $P|Q$ се индикира, че двата процеса се извършват паралелно. Процесът 0 не извършва нищо и приключва веднага. Знакът за удвояване (replication)! се използва за създаване на ново копие на процеса (!P), когато това е необходимо. Процесът $n[P]$ обозначава амбиент с име n и процес P, който описва неговото поведение. Последователните процеси се отделят със символа „.“. Местоположението се използва при комуникацията между амбиентите. Символът „↑“ означава

обръщение към родител; “↓” – към дете, а “...” – към роднина. Възможностите са свързани с движението на амбиента – in и out според другите амбиенти. Възможността del n позволява унищожаване на амбиент с име n, но само ако е празен, т.е. n[0].

Според представената на фигура 1 архитектура на системата, основните компоненти са School Learning Portal (SLP), School Learning Repository (SLR), Administrative Data Bases (ADB), както и стандартните и мобилни устройства на учениците. За да моделираме чрез CCA е необходимо да представим тези компоненти като амбиенти. Ученикът ще бъде представен в модела чрез своето устройство за достъп –PC, GSM, Tablet. Училищният образователен портал ще бъде представен чрез амбиента SLR. Той ще съществува паралелно с SLR и ADB. Lesson Repository (LR) and Tests Repository (TR) са деца амбиенти на SLR, а Students DB (SDB), School Schedule (SS) и School Curriculum (SC) са деца амбиенти на ADB. CCA моделът е представен графично на Фиг.2.

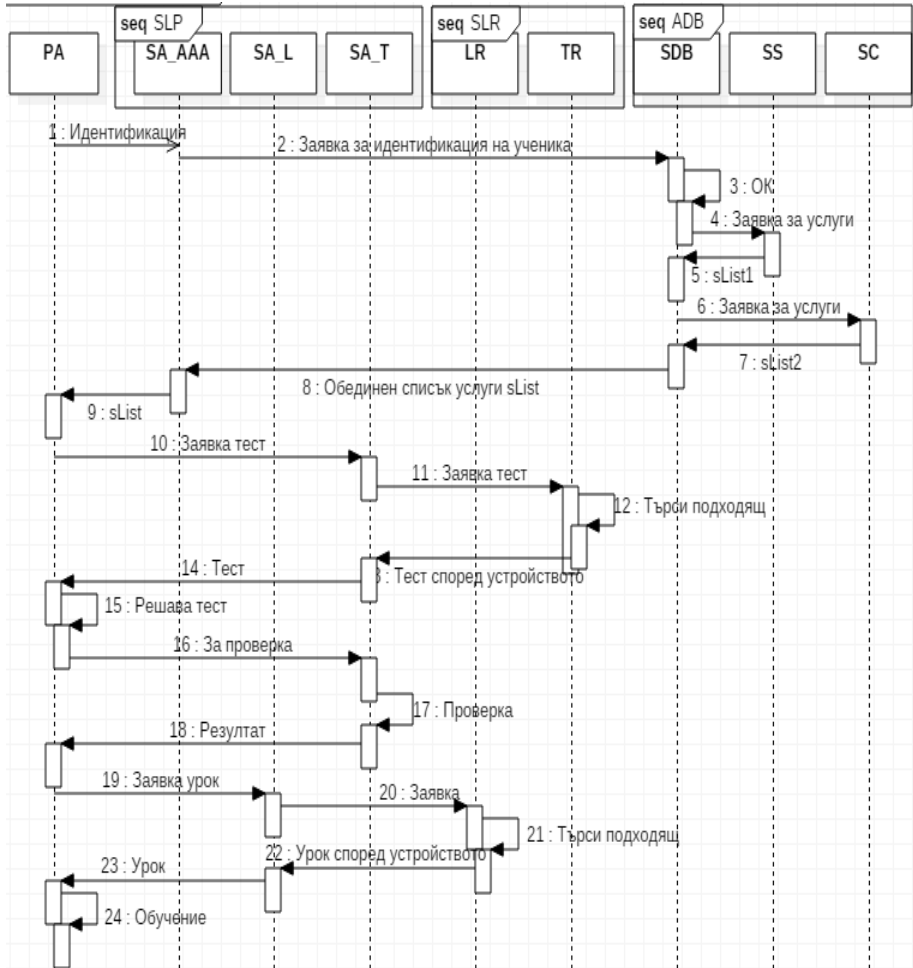


Фиг.2. CCA модел на училищната образователна среда

Основните компоненти на SLP са амбиент на заявка за идентификация SA_AAA, амбиента за уроци SA_Less и за тестове – SA_Test. Потребителските устройства няма да се моделират като амбиенти, тъй като те могат да променят местоположението си. Амбиентът, представляващ потребителя в системата ще е неговия персонален асистент PA_i. Грите постоянни амбиента, които се намират в SLP – SA_AAA, SA_Less и SA_Test са отговорни за реализирането на съответната услуга – идентификация, доставка на лекция или тест. Всяка една от тези услуги се управлява от интелигентен агент-специалист

(SA) [6, 8]. Тези три амбиента са роднини помежду си. Персоналният асистент PA_i също е дете на SLP.

Реализацията на сценария, при който ученикът се идентифицира, получава списък от подходящите за него услуги, избира тест, системата му връща теста в подходящ за устройството му формат, прави теста и получава резултат, после избира урок, получава го в подходяща форма и стартира обучението си е представена на Диаграма 1.



Диаграма 1. UML диаграма на учебния сценарий за обучение и тестване

Всеки ученик трябва да направи своята идентификация в системата, за да може да ползва съответните услуги. За да се стартира процесът по регистрация, амбиентът SA_AAA трябва да получи заявка със следните параметри: stuID, dType, PA_i, която препраща към родителя си SLP. Параметърът stuID служи за проверка правата на този ученик, dType- за получаване на съответстващ на устройството списък от услуги, а PA_i – за да знае къде да изпрати отговора на заявката. Когато амбиентът получи отговор от SLP, той го препраща чрез PA_i към потребителското устройство. След това ученикът може да избере някоя от предложените услуги – електронен урок, тест или информация за предстоящи събития.

SA_Less амбиентът отговаря за заявките и стартирането на електронните уроци. Той изпраща заявка до своя родител SLP със следните параметри: lessID, stuID, dType, PA_i, където lessID е избрания от ученика урок. Отговорът от SLP съдържа три параметъра (lessID, reply, PA_i), като lessID е заявения урок, а reply е или съдържанието на този урок или отказ за неговото ползване. Амбиентът SA_Test има подобно поведение на SA_Less. Изпитването на ученици в неприсъствените форми на обучение се осъществява само в присъствието на квестори и преподаватели. Тестовите, както и уроците са свързани с учебния предмет и съответния клас на ученика. Тогава:

Нека сега разгледаме поведението на SLP амбиента по отношение на заявките и комуникацията му както с амбиентите ADB и SLR, които са негови роднини, така и с SA_AAA, SA_Less, SA_Test, които са негови деца. След като получи заявка за идентификация от SA_AAA, SLP я препраща към ADB, където трябва да се провери дали потребителят е ученик в училището. Ако е така, проверява неговия клас и форма на обучение и според типа на персоналното му устройство, му връща списък с подходящи за него услуги, като междувременно проверява дали ученикът все още очаква отговора, или е излязъл от системата (с което е унищожена и инстанцията на неговия PA_i). Комуникацията с амбиентите SA_Less и SA_Test се обработва по сходен начин. Сценарият за SLP включва следните действия: след като получи заявка от SA_Less или SA_Test, амбиентът изпраща заявка до LTR, където след проверка за наличие на такава услуга връща резултат. Заявката включва следните параметри:

- lessID или TestID – търсения урок или тест;
- stuID – идентификатор на ученика;
- dType – тип на устройството;
- PA_i – персоналният асистент.

Моделираме AMBIENTът SLP, неговите деца и комуникацията му с неговите родители SLR и ADB по следния начин:

$$P_{SA_AAA} \triangleq ! :: (stuID, dType, PA_i).SLP \uparrow \langle stuID, dType, PA_i \rangle . \\ .SLP \uparrow (ack, PA_i, sList).PA_i :: \langle ack, sList \rangle .0$$

$$P_{SAL} \triangleq ! :: (lessID, stuID, dType, PA_i).SLP \uparrow \langle lessID, stuID, dType, PA_i \rangle .0 | \\ !SLP \uparrow (lessID, reply, PA_i).PA_i :: \langle lessID, reply \rangle .0$$

$$P_{SAT} \triangleq ! :: (testID, stuID, dType, PA_i).SLP \uparrow \langle testID, stuID, dType, PA_i \rangle .0 | \\ !SLP \uparrow (testID, reply, PA_i).PA_i :: \langle testID, reply \rangle .0$$

$$P_{SLP} \triangleq \left(\begin{array}{l} \left(!SA_AAA \downarrow (stuID, dType, PA_i).ADB :: \langle AAAreg, stuID, dType, PA_i \rangle .0 | \right. \\ \left. !ADB :: (ack, PA_i, sList).(has(PA_i)) ? SA_AAA \downarrow \langle ack, PA_i, sList \rangle .0 \right) | \\ \left(!SA_Less \downarrow (lessID, stuID, dType, PA_i).SLR :: \langle lessID, stuID, dType, PA_i \rangle .0 | \right. \\ \left. !STR :: (lessID, reply, PA_i).(has(PA_i)) ? SA_Less \downarrow \langle lessID, reply, PA_i \rangle .0 \right) | \\ \left(!SA_Test \downarrow (testID, stuID, dType, PA_i).SLR :: \langle testID, stuID, dType, PA_i \rangle .0 | \right. \\ \left. !STR :: (testID, reply, PA_i).(has(PA_i)) ? SA_Test \downarrow \langle testID, reply, PA_i \rangle .0 \right) \end{array} \right)$$

След като получи заявка от SLP AMBIENTът SLR я препраща до някой от вътрешните си AMBIENTи LR или TR, които я обработват и връщат отговор, който съдържа съответния учебен ресурс в подходяща за потребителското устройство форма, или отказ за предоставяне на същия. AMBIENTът SLR връща този отговор на SLP. Последователно ще моделираме AMBIENTите LTR, LR, TR и техните вътрешни AMBIENTи. В зависимост от вида на заявката, получена от SLP ($lessID, stuID, dType, PA_i$) or ($testID, stuID, dType, PA_i$), AMBIENTът LTR я препраща съответно към вътрешните си AMBIENTи LR или TR. От тях получава отговор съответно във вид ($lessID, stuID, dType, reply, PA_i$) или ($testID, stuID, dType, reply, PA_i$). Стойността на параметъра reply е или съдържанието на самия учебен ресурс, или отказ за неговото ползване. AMBIENTът LR отговаря за предоставянето на заявеният урок в подходящ за вида на потребителското устройство вид. Той получава заявка от родителя си SLR във вид ($lessID, stuID, dType, reply, PA_i$), проверява дали такъв урок съществува и изпраща съобщение на вътрешния си AMBIENT LessID. В резултат получава отговор във вида ($reply, dType, PA_i$), където reply е или съдържанието на урока в съответния за dType вид или информация за липса или отказ за ползване на този урок. След това връща отговор на SLR ($lessID, stuID, dType, reply, PA_i$). Аналогично

моделираме и амбиентът TR. Амбиентите LessID и TestID получават заявка от LR или TR. Проверяват дали съществува заявения учебен ресурс в нужния формат във вътрешните си амбиенти PC, PDA или GSM и връща резултат. Ще обработваме групово амбиентите PC, PDA and GSM като dType. Щом dType получи заявка с параметър (PA_i), връща отговор ($dType, reply, PA_i$). Поведението на SLR, неговите деца и взаимодействието му с SLP и ADB ще моделираме по следния начин:

$$P_{SLR} \hat{=} \left(\begin{array}{l} !SLP :: (lessID, stuID, dType, PA_i).LR \downarrow < lessID, stuID, dType, PA_i > .0 | \\ !LR \downarrow (lessID, reply, PA_i).SLP :: < lessID, stuID, dType, reply, PA_i > .0 \end{array} \right) | \\ \left(\begin{array}{l} !SLP :: (testID, stuID, dType, PA_i).TR \downarrow < testID, stuID, dType, PA_i > .0 | \\ !TR \downarrow (testID, reply, PA_i).SLP :: < testID, stuID, dType, reply, PA_i > .0 \end{array} \right)$$

$$P_{LR} \hat{=} \left(\begin{array}{l} !\uparrow (lessID, stuID, dType, PA_i). \\ \left(\begin{array}{l} has(LessID)? LessID \downarrow < dType, PA_i > .0 | \\ !LessID \downarrow (dType, reply, PA_i).SLR \uparrow < lessID, stuID, dType, reply, PA_i > .0 \end{array} \right) | \\ \left(-has(LessID)? SLR \uparrow < lessID, stuID, dType, NULL, PA_i > .0 \right) \end{array} \right)$$

$$P_{LessID} \hat{=} \left(\begin{array}{l} !\uparrow (dType, PA_i). \\ \left(\begin{array}{l} has(dType)? dType \downarrow < PA_i > .0 | \\ !dType \downarrow (reply, PA_i).LR \uparrow < dType, reply, PA_i > .0 \end{array} \right) | \\ \left(-has(dType)? LR \uparrow < dType, NULL, PA_i > .0 \right) \end{array} \right)$$

Амбиентът ADB е отговорен за проверка на данните за ученици, съответния за класа на ученика учебен план, както и за разписанието на консултациите, изпитните сесии и други училищни събития. Амбиентът ADB е роднина на SLP и има вътрешни амбиенти SDB (Students DB), SC(School Curriculum) и SS (School Schedule). Всички тези амбиенти се осигуряват от хранилището на данни на софтуерния продукт AdminL, който е задължителен за всички училища в страната. Информацията е достъпна както локално, така и през интернет (<http://am.mon.bg>). Амбиентът ADB получава заявка за идентификация на ученик от SLP със следните параметри ($AAA_{reg}, stuID, dType, PA_i$). Той препраща тази заявка към SDB. В резултат получава отговор със следните параметри: ($reply, dType, PA_i, sList$), като $reply$ има стойност АСК или NULL, а $sList$ е списъкът от услугите, които може да ползва конкретния

ученик, в зависимост от неговия клас и форма на обучение. След това той изпраща отговор на SLP, който съдържа (ack, PA_i, sList).

Амбиентът SDB получава заявка за идентификация на ученик от родителя си ADB с параметри (AAAre_g, stuID, dType, PA_i). Той проверява дали съществува такъв ученик като изпраща заявка към съответния амбиент-дете StuID. В резултат получава отговор с параметри (reply, classID, PA_i), където reply е утвърдителен отговор ACK, или отрицателен NULL, а classID е класа на съответния ученик, който се използва за да се генерира съответния списък с услуги. Ако отговорът е NULL, веднага се връща отговор към ADB с параметри (NULL, dType, PA_i, sList), като sList е празния списък. Ако отговорът е утвърдителен, амбиентът изпраща заявка (classID, dType, PA_i) към SC, с който са роднини. В отговор получава списък sList1 с услуги за съответния клас и форма на обучение, които са във вид, подходящ за потребителското устройство. Успоредно с това изпраща заявка (stuID, classID, dType, PA_i) към амбиента SS, който също му е роднина с искане да се предостави списък с училищните събития, валидни за конкретния ученик. Отговорът съдържа списък от предстоящите събития за съответния ученик-sList2. От двата списъка се сформира общ списък с всички услуги и събития sList, който се връща като отговор на ADB. Амбиентът StuID има комуникация само с родителя си SDB, от който получава заявка и връща отговор с параметри (ack, classID, PA_i). ССА моделирането на амбиента ADB и неговите деца можем да представим така:

$$P_{ADB} \triangleq \left(\begin{array}{l} !SLP :: (AAAre_g, stuID, dType, PA_i).SDB \downarrow \langle AAAre_g, stuID, dType, PA_i \rangle .0 \\ !SDB \downarrow (reply, dType, PA_i, sList).SLP :: \langle ack, PA_i, sList \rangle .0 \end{array} \right)$$

$$!ADB \uparrow (AAAre_g, stuID, dType, PA_i).$$

$$P_{SDB} \triangleq \left(\begin{array}{l} -has(stuID) ? .ADB \uparrow \langle NULL, dType, PA_i, sList \rangle .0 \\ has(stuID) ? .stuID \downarrow \langle PA_i \rangle .stuID \downarrow (ack, classID, PA_i) .0 \\ \left(!SC :: \langle classID, dType, PA_i \rangle .0 \right) \parallel \\ \left(!SC :: (dType, PA_i, sList1) .0 \right) \parallel \\ \left(!SS :: \langle stuID, classID, PA_i \rangle .0 \right) \parallel \\ \left(!SS :: (dType, PA_i, sList2) .0 \right) \parallel \\ SDB \uparrow \langle ack, dType, PA_i, sList \rangle .0 \end{array} \right)$$

$$P_{\text{StudID}} \triangleq \left(\begin{array}{l} \{!SDB \uparrow (PA_i).0\} \\ \{!SDB \uparrow \langle ack, classID, trType, PA_i \rangle .0\} \end{array} \right)$$

4. Заключение и бъдещи планове

Представеният модел на училищна контекстно-зависима среда представя услуги за обучение и тестване на ученици от неприсъствените форми на обучение. Предоставените уроци и тестове зависят от вида на устройството за достъп, неговия клас, профил и пр. Моделът може да се доразвие по отношение на локацията на ученика, като ще се очаква, когато ученикът се намира в района на училището да получава съобщения, свързани с провеждане на изпити, промяна в разписанията и други събития, които касаят конкретния ученик, а когато ученикът е далеч от училището – съобщения за това къде и кога трябва да бъде в следващите минути и часове през деня. Също така интересни от гледна точка на ССА моделирането са и услуги като учителски бележник, ученически картон и ученическа книжка, които да отразяват промените в знанията на учениците и се адаптират според промените в околната среда.

Верификацията на модела преминава през неговото прототипиране. За целта на изследването може да се използва езикът за програмиране на ССА-ссаPL. Създаването на прототипна анимирана среда, в която да се реализират основните сценарии, ще даде възможност да се верифицира модела, преди да се премине към неговото реализиране.

Предложеният модел на училищната учебна среда е базиран на концептуалната рамка на Virtual Educational Space VELSPACE и DeLC. Към момента е създаден прототипен SCORM-базиран образователен портал, който се използва в реалния учебен процес [10].

Благодарности

Авторът изказва благодарност към научен проект ИТ 15-ФМИИТ-004 „Изследвания в областта на иновационните ИКТ с ориентация към бизнеса и обучението“ към Фонд „НИ“ на ПУ „Паисий Хилендарски“, за частичното финансиране на настоящата работа.

Литература

- [1] Dey A., and G. 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.
- [2] Dey, A., Understanding and Using Context, *Personal and Ubiquitous Computing Journal*, Vol. 5, No. 1, 2001, pp.4–7..
- [3] Glushkova, T., *Adaptive model for e-Learning in secondary school*, E-Learning book “Long-Distance and Lifelong Perspectives”, 2011, ISBN 978-953-51-0250-2.
- [4] 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.
- [5] 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.
- [6] Stoyanov, S., 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*, Vol. 3, Dynamic Publishers, Inc., USA, (2011), ISSN 2150-7988, pp. 771–779.
- [7] 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>
- [8] Stoyanov, S., Context-Aware and Adaptable eLearning Systems, *Internal Report*, Software Technology Research Laboratory, De Montfort University, Leicester, UK, August, 2012.
- [9] ЗПУО – <http://mon.bg> (посетен на 28.10.2016 г.).
- [10] Училищен SCORM-базиран образователен портал – <http://sou-brezovo.org> (посетен на 29.10.2016 г.).

Факултет по математика и информатика
Пловдивски университет „Паисий Хилендарски“
Бул. „България“ 236, 4003 Пловдив, България
e-mail: glushkova@uni-plovdiv.bg

TRAINING AND TESTING IN CONTEXT-AWARE SCHOOL LEARNING ENVIRONMENT

Todorka Glushkova

Abstract. The creation of context-aware systems that dynamically adapt to environmental parameters is a challenge and a necessity in the modern development. The aim of the article is to present a model of context-aware environment for e-learning in secondary schools and its CCA -formalization.