ПРОТОТИП НА КОНТЕКСТНО-ЧУВСТВИТЕЛЕН ТУРИСТИЧЕСКИ ПЪТЕВОДИТЕЛ

Тодорка Глушкова

Резюме: *Математическата нотация на Calculus of Context-aware Ambients (CCA) е подходяща за моделиране на контекстно-чувствителни мобилни системи. Тъй като е трудно да се възпроизведе синтаксисът на CCA в среда за програмиране поради специфичните символи, които се използват в нотацията, за създаване на симулатор на контекстно-чувствителен туристически пътеводител ще използваме езика ccaPL, който е четима за компютъра версия на синтаксиса на CCA. В статията ще се представят процесите на верификация и валидация на сценариите на различни услуги чрез модифициране и разработка на анимиран ccaPL симулатор за прототипиране на системата.*

Ключови думи: *Calculus of Context-aware Ambients (CCA), ccaPL, DeLC*

Въведение

Процесите на създаване и развитие на разпределени системи, в които дигиталните устройства доставят на потребителите нужната информация и услуги, отчитайки динамичната промяна на околната среда (Schmidt, Laerhoven, 2001) са свързани с новата парадигма на интернет на нещата. В това направление са изследванията на екипа на DeLC от ФМИ при ПУ „Паисий Хилендарски“ за моделиране, прототипиране и реализиране на виртуални среди, които отчитат динамичните промени на физическата среда и се адаптират към тях (Stoyanov, Popchev, Orozova, 2016). Като резултат от тези изследвания се изгражда Виртуално Образователно Пространство (Stoyanov, 2016, Стоянов, Попчев, 2016) като IoT екосистема. Част от пространството е проектът за дигитално представяне на културно-историческото наследство на България.

В статията ще разгледаме моделирането и прототипирането на контекстно-чувствителна система за предоставяне на туристически услуги, която ще отчита времевите характеристики, промяната в местоположението на потребителя, надморската височина, температурата, времето като част от денонощието и др. Ще използваме сензорите на мобилното устройство на туриста за получаването на актуалната информация за текущия контекст.

Съществуват различни дефиниции на понятието контекст. В разработката на модела и прототипа на туристическия пътеводител ще използваме дефиницията на Dey (A.K. Dey, G.D. Abowd, 2000), (A.K. Dey, 2001), 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. Формализацията на модела е реализирана чрез математическата нотация Calculus of Context-aware Ambients (CCA) (Siewe F., A. Cau, H. Zedan, 2009, 2011), (Alotaibi, Hind, 2012). В статията ще се коментират основните сценарии на услугите в модела, както и възможностите за верификацията на този модел.

Верификацията и валидирането гарантират, че всяка фаза при разработката на системата е реализирана според предварително поставените изисквания. Верификацията е качествен процес, по време на който се извършва оценка на това дали системата или услугата отговарят на регламентите, спецификациите и условията, т.е.

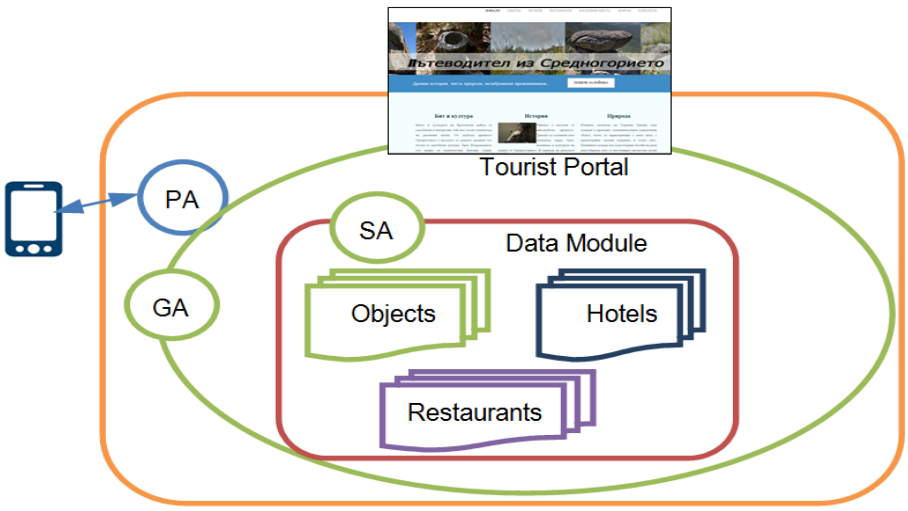
може да се опише с въпроса: „Дали изграждаме системата правилно?“. От друга страна валидирането е процес, който осигурява доказателство, което да гарантира високо ниво на сигурност, че системата или услугата покрива заложените параметри, т.е. може да се опише с въпроса: „Дали изграждаме правилната система?“. Процесът на валидиране представлява предизвикателство за контекстно-зависимите системи според (H. Zedan, F. Siewe, and A. Cau, 2008), тъй като съществуващите техники за валидиране като проверка на модели, тестване и симулацията не могат да осигурят 100%-ва сигурност за точността и производителността на приложението, защото при тестването на една контекстно-зависима система се изисква динамична промяна на контекстната информация. За достигане на по-голяма сигурност се препоръчва използването на симулатори и създаването на прототипи.

ССА моделиране на системата

Амбиент-ориентираното моделиране дава възможност за описание на характеристиките и поведението на различните идентичности в контекстно-чувствителна среда. Един амбиент (ambient) е идентичност, която няколко основни характеристики: *ограниченост* до местоположението си; *вложеност* в други амбиенти в йерархична структура; *мобилност* на амбиента от едно местоположение в друго като едно цяло. Всеки амбиент може да бъде представен като структура, съдържаща: идентификатор за контрол на достъпа; кореспондиращо множество от локални процеси; множество от прилежащи подамбиенти.

Туристическата информационна система е проектирана да осигури подходящи за конкретния потребител услуги, в зависимост от промените в околната му среда. На фигура 1 е представена архитектурата и основните компоненти на част от системата, свързано с реализирането на модела.

Средата за туристически услуги има разпределена архитектура. В зависимост от текущия сценарий, отделните модули се свързват по различен начин,. Туристическият портал има многослойна архитектура, като всеки слой да е специализиран в изпълнението на определена функционалност. Моделът на данните включва няколко хранилища за информация, свързани с туристическите обекти (Objects), хотелите и къщите за гости (Hotels), както и ресторантите и местата за отдих (Restaurants). Мобилното устройство на потребителя осигурява контекстната информация за околната среда чрез сензорите си за местоположение, надморска височина и време като част от денонощието.



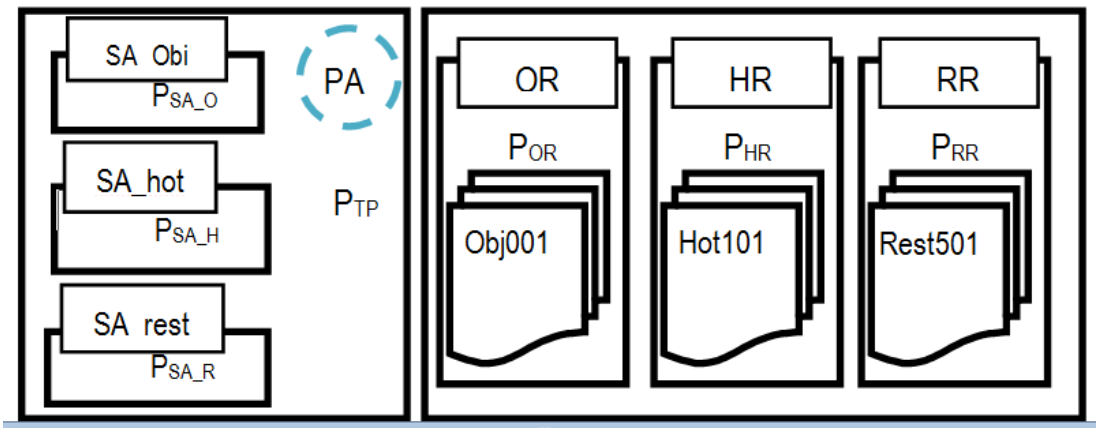
Фигура 1. Архитектурен модел на системата

Асистентите са автономни компоненти, които комуникират помежду си чрез обмен на съобщения. По своята същност те са интелигентни агенти, които отговарят за изпълнението на определени услуги (Valkanov, 2013), (Stoyanov, 2012). Ще използваме три вида асистенти: персонални асистенти (PA), които подпомагат работата на потребителите в средата; асистенти специалисти (SA), които осигуряват специализирана помощ и услуги на потребителите при изпълнението на техните заявки и гардове (GA), които имат специфична функционалност, свързана със сигурността по време на изпълнението на услугите. Моделирани са следните услуги:

- ***Посещение на туристически обект:*** Тази услуга реализира търсене на туристически обект по заявка на потребителя. След откриване на обекта, системата трябва да предостави информация за неговото местоположение, кратко и пълно описание, както и най-близките хотели и ресторанти. Информацията трябва да се визуализира във вид, подходящ за типа на потребителското устройство. Използвайки интеграция с други приложения се предоставя информация за най-удобните маршрути за посещение на обекта.
- ***Туристически пътеводител:*** Тази услуга изисква непрекъснато проследяване на параметрите на динамично променящата се среда. Според местоположението на туриста се реализира търсене на най-близкия обект и информация за него в графичен, тектов и аудио формат. При движението на туриста се осигурява информация за маршрута, местата за отдих, забележителни точки по маршрута или такива, които могат да се видят от разстояние. При получаване на информация за рязка промяна в температурата и вероятността за буря, GA активира услугата търсене на най-близки хотели, заслони или места за отдих и предлага навигация на туриста до тях. В случай на извънредна ситуация (ниска температура, буря, висока надморска височина и преустановяване на движението на туриста) GA може да сигнализира спасителните служби.

Моделиране на първата услуга

Сценарият за реализиране на първата услуга стартира с изпращане на заявка от потребителя за търсене на определен туристически обект. От модулът за данни се търси и извлича списък от обекти, които отговарят на определените критерии, както и на параметрите на потребителското устройство за достъп. От този списък туристът избира обект и се стартира процес на доставка на информацията за този обект. Ако списъкът с обекти е празен, системата връща съобщение към потребителя. На фигура 2 са представени амбиентите, които ще бъдат използвани при реализацията на сценариите.



Фигура 2. Амбиентите, осигуряващи услугата на туриста

SA_Obj Ambient – този амбиент е отговорен за изпълнението на услугата за търсене, доставка и получаване на информация за определен туристически обект. Заявката съдържа следните параметри: ObjID, dType, PAi, където ObjID е търсения обект; dType е типа на устройството, на което ще се използва информацията, а PAi е персоналния асистент, който представлява потребителя. Това поведение може да се моделира така:

$$!:: \ (ObjID,\ dType,\ PA_i).TP \uparrow < ObjID,\ dType,\ PA_i > .\ 0 \qquad (1)$$

Отговорът от TP съдържа три параметъра (ObjID, reply, PAi), като ObjID е заявения обект (тъй като може да са заявени няколко обекта), reply е или информация за този обект или информация за неговата липса.

$$!\ \mathrm{TP} \uparrow \ (ObjID,\ reply,\ PA_i).\ PA_i \ ::\ < ObjID,\ reply > .0 \qquad (2)$$

Тогава процесът на този амбиент е: $P_{SA\_Obj} \stackrel{\wedge}{=}$ Eq.(1) | Eq. (2)

TP Ambient – той комуникира с неговия амбиент-роднина DM, както и с SA_Obj, който е негово дете. След като получи заявка от SA_Obj, той я предава към DM, където се проверява наличието на такъв обект и според типа на потребителското устройство за достъп се връща информация за търсения туристически обект. Преди да върне отговора TP проверява дали потребителят все още очаква отговора (т.е. дали PAi още е активен).

$$P_{TP} \Leftrightarrow \begin{pmatrix} !SA\_Obj \downarrow (ObjID, dType, PA_i).DM ::< ObjID, dType, PA_i > .0 \mid \\ !DM :: (ObjID, reply, PA_i).(\mathrm{has}(PA_i))?SA\_Obj \downarrow < ObjID,\ reply, PA_i > .0 \end{pmatrix}$$

DM Ambient – щом получи заявката от TP, амбиентът DM я насочва към вътрешния си амбиент Objects Repository (OR) и получава отговор във вида (ObjID, dType, reply, PAi).

$$P_{DM} \Leftrightarrow \begin{pmatrix} !TP :: (ObjID, dType, PA_i).OR \downarrow < ObjID, dType, PA_i > .0 \mid \\ !OR \downarrow (ObjID, reply, PA_i).\mathrm{TP}:: < ObjID, \mathrm{dType}, reply, PA_i > .0 \end{pmatrix}$$

Моделиране на втората услуга

Мобилното устройство подава информация за местоположението на туриста, за надморската височина и за времето от денонощието. Като външна услуга порталът извлича информация за температурата на въздуха в тази точка и вероятността за буря. Информацията за местоположението се изпраща на модула за данни, който го препраща към амбиента OR. Оттам се извлича информация за най-близкия обект, която се изпраща на потребителя. Информацията за надморската височина, температурата, вероятността за буря и времето от денонощието се изпращат на GA, който е роднина на TR. Ако времето от денонощието е след 19 ч., или температурата на въздуха е под 2^0 C, вероятността за буря е над 50% и надморската височина е над 800м., GA изпраща заявка за търсене на най-близките хотели и ресторанти към DM. След като получи отговора на заявката я препраща към PA_i като извънредно съобщение.

SA_Obj Ambient – този амбиент е отговорен за изпълнението на услугата за търсене и получаване на информация за най-близкия туристически обект.

$$P_{SA\_Obj} \Leftrightarrow \begin{pmatrix} !:: \ (\ PA_i, location, time, altitude).TP \uparrow < \ PA_i, location, time, altitude > .\ 0 \mid \\ !TP \uparrow (reply, PA_i).PA_i ::< reply > \end{pmatrix}$$

Променливата reply е информацията за обекта, съдържаща кратка и по-обстойна информация за най-близкия обект в графична, текстова и аудио форма.

SA_hot Ambient – този амбиент е отговорен за търсене и получаване на информация за най-близкия хотел. Неговото поведение може да се моделира така:

$$P_{SA\_hot} \Leftrightarrow \begin{pmatrix} !:: \ (PA_i, location).TP \uparrow < PA_i, location > .0 \mid \\ !TP \uparrow (list1, PA_i).PA_i ::< list1 > \end{pmatrix}$$

Аналогично за SA_rest:

$$P_{SA\_rest} \Leftrightarrow \begin{pmatrix} !:: \ (PA_i, location).TP \uparrow < PA_i, location > .0 \mid \\ !TP \uparrow (list2, PA_i).PA_i ::< list2 > \end{pmatrix},$$

където list1 и list2 са списъците на най-близките хотели и ресторанти.

TP Ambient – той комуникира със своите деца SA_Obj,SA_hot и SA_rest, както и със своите роднини GA и DM. Поведението можем за опишем така:

$$P_{TP} \Leftrightarrow \begin{pmatrix} !SA\_Obj \downarrow (location, altitude, time, PA_i). \\ GA ::< location, altitude, time, temperature, procent, PA_i > .0 \mid \\ !GA :: (reply, PA_i).DM ::< location, PA_i > .0 \mid \\ !DM :: (list, PA_i).PA_i \downarrow < list > .0 \end{pmatrix}$$

GA Ambient – този амбиент получава информация от TP и при отчитане на евентуална опасност за туриста, изпраща съобщение <location, PAi> към портала с искане за извличане на списък с най-близките хотели и ресторанти.

$$P_{GA} \Leftrightarrow \begin{pmatrix} !TP :: (location, altitude, time, temperature, procent, PA_i). \\ ((time > 19) or ((temperature < 2) and (procent > 50) and (altitude > 800))) \\ ?\,\mathrm{TP}::<\mathrm{reply},\mathrm{PA_i}>.0 \end{pmatrix}$$

DM Ambient – когато се получи искане от TP, този амбиент търси и формира списък на най-близките до текущото местоположение на туриста хотели и ресторанти:

$$P_{DM} \Leftrightarrow \begin{pmatrix} !TP :: (location, PA_i).(HR \downarrow < location, PA_i > .0 \mid RR \downarrow < location, PAi >) \mid \\ (!HR \downarrow (\mathrm{list1}, PA_i) \mid (!RR \downarrow (list2, PAi)).T\mathrm{R}::<\mathrm{list},PA_i>.0 \end{pmatrix}$$

Език за програмиране на CCA – ccaPL

Трудно е да се възпроизведе синтаксисът на CCA като се използва текстов редактор, поради специфичните символи, които са включени като например „↑“ и „↓“. Езикът ccaPL е четима за компютъра версия на синтаксиса на CCA. *Таблиците 1, 2, 3* представят синтаксиса на ccaPL що се отнася до процеси, възможности и местоположение. С h се дефинира превеждаща функция от CCA на ccaPL, т.е. за процеса P в CCA, h(P) е кореспондиращата ccaPL нотация.

CCA: M	ccaPL: h (M)	**Описание**
in n	in n	Влез в амбиент n
out	out	Излез
del n	del n	Изтрий амбиент n
α (ý)	h(α)recv(ý)	Получаване на списък от съобщения ý от α
α <ý>	h(α)send(ź)	Изпращане на списък от съобщения ź до α

Таблица 1. Възможности на ccaPL

CCA: P	ccaPL: h (P)	**Описание**
0	0	Пасивност
n[P]	n[h(P)]	Амбиент
P|Q	h(P)|h(Q)	Паралелно изпълнение
!P	!h(P)	Удвояване
k?M.P	<h(k)>h(M).h(P)	Контекстно гарантиращо предусловие

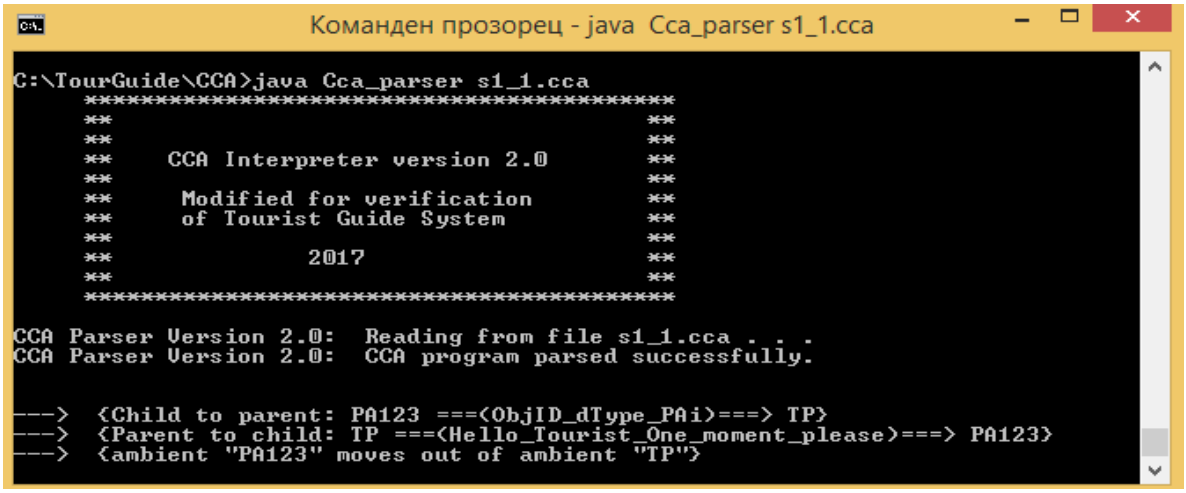
Таблица 2. Процеси на ccaPL

CCA: α	ccaPL:h (α)	**Описание**
↑	@	Кой да е родител
n ↑	n@	Родител n
↓	#	Кое да е дете
n↓	n#	Дете n
::	::	Кой да е роднина
n::	n::	Роднина n

Таблица 3. Местоположение при ccaPL

Изпълнимата среда на ccaPL е разработена като Java приложение. Въз основа на базовата версия е разработена специална среда за верифициране на контекстно-чувствителен туристически пътеводител. Например, за първоначалното взаимодействие между персоналния асистент на туриста (PA123) и туристическия портал (TP) се създава CCA-файл със следната ccaPL програма, резултатът от която виждаме на фигура 3:

```
TP[ PA123[ TP@send(ObjID_dType_PAi).TP@recv(x).out.0]      |
PA123#recv(y).<y=ObjID_dType_PAi>PA123#send(Hello_Tourist_One_moment_please).0]
```

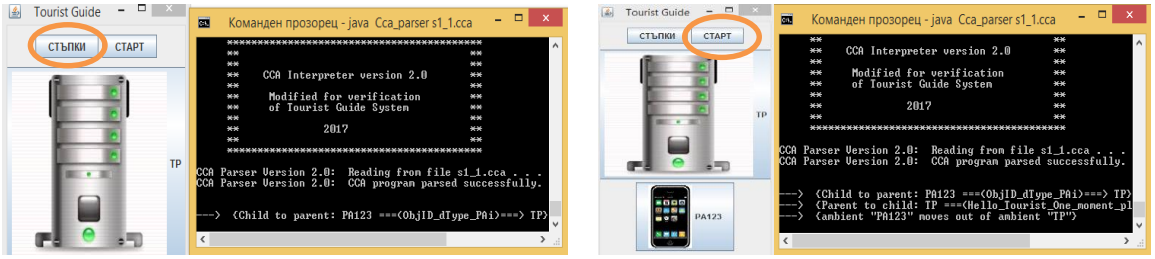


Фигура 3. Изпълнение на ccaPL скрипт

Всеки ред от изходните данни на изпълнението на програмата се четат като: знакът „--->“ представя редукционните взаимоотношения на CCA (Siewe, Zedan, Cau, 2010). Обяснението на всеки преход е дадено в двойка кадрави скоби {}. Нотацията „A ===(X)===> B” означава, че амбиент „A“ изпраща съобщение „X“ до амбиент „B“. Нотациите „Child to parent“, „Parent to child“ и „Sibling to sibling“ дават информация за взаимовръзката между изпращача „A“ и получателя „B“ по отношение на йерархията на амбиентите. За улеснение на разчитането е разработен аниматор (Mohammed H. Al-Sammarraie, 2011) към средата на ccaPL за прототипиране на системата. Целта на аниматора е да представи графично амбиентите и техните процеси. Визуализират се

реходите между процесите; как и кога всяка възможност на даден амбиент се изпълнява; как амбиентите се движат в заобикалящата ги среда. Аниматорът представя амбиентите йерархично като основният е представен като контейнер, който съдържа в себе си всички деца амбиенти. Всяко дете амбиент е представено чрез бутон с име и икона. Иконите, които са свързани с туристическия пътеводител са: PA, GA, TP, SA_Obj, SA_Hot, SA_Rest, DB, OR, HR, RR.

Вътрешните амбиенти също са представени като бутони. Контейнерът на основният амбиент съдържа първоначално два бутона „СТАРТ“ и „СТЪПКИ“. При стартиране на програмата се изпълнява първия преход и се визуализира анимацията. След това потребителят продължава тестването на сценария стъпка по стъпка или изцяло. Мобилните преходи в ccaPL-аниматора за разгледания пример са представени на фигура 4.



Фигура 4. Визуализация на мобилните преходи в симулатора

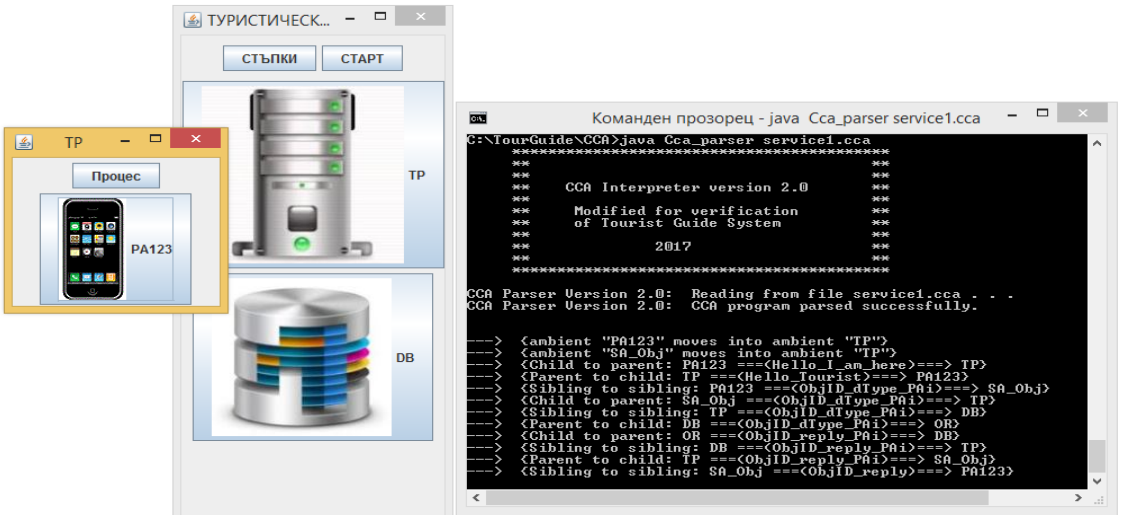
Валидиране

Валидирането на сценариите в системата са от особено значение за нейната реализация. Целта на този процес е да докаже, че системата работи коректно в съответствие с разработения модел.

Първият сценарий потвърждава услугата за откриване на туристически обект. Съгласно CCA модела, тази услуга има следната ccaPL програмна реализация:

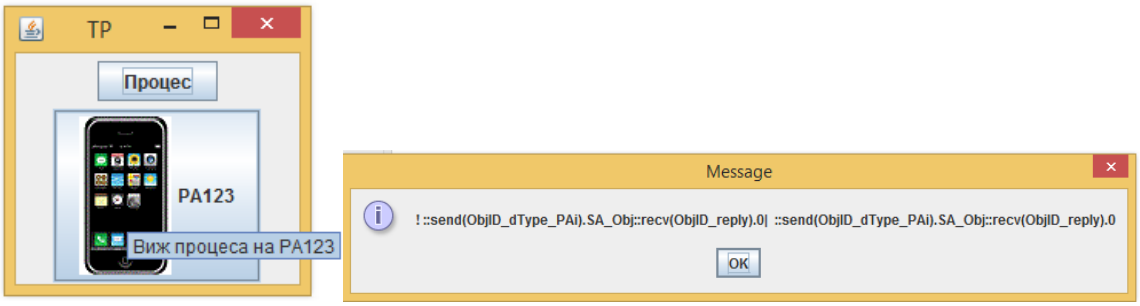
```
TP[
   PA123[TP@send(Hello_I_am_here).TP@recv(x1) .0]
       | PA123#recv(y1).<y1=Hello_I_am_here>PA123#send(Hello_Tourist).0
       | SA_Obj[PA123 :: recv(ObjID_dType_PAi). TP @ send(ObjID_dType_PAi). 0]
       | SA_Obj[TP @ recv(y5).PA123 :: send(ObjID_reply).0]
       | DB::recv(y5).<y5=ObjID_reply_PAi>SA_Obj#send(ObjID_reply_PAi).0
       | SA_Obj#recv(y2).<y2=ObjID_dType_PAi>DB::send(y2).0
       | SA_Obj[TP@recv(y3).<y3=ObjID_reply_PAi>PA123::send(ObjID_reply).0]
       | PA123[!::send(ObjID_dType_PAi).SA_Obj::recv(ObjID_reply).0]
   ]
   |DB[
   OR[!@recv(ObjID_dType_PAi).DB@send(ObjID_reply_PAi).0]
   |!::recv(y6).<y6=ObjID_dType_PAi>OR#send(y6).OR#recv(y7).<y7=ObjID_reply_PAi>
   TP::send(ObjID_reply_PAi).0
   ]
```

Изпълнението в ccaPL програмната среда (Фигура 5) дава възможност да се проследят процесите на участващите амбиенти и последователността на изпращаните и получавани съобщения между тях. Аниматорът позволява да се визуализират участващите амбиенти, тяхното местоположение и в хода на изпълнението на сценария да се проследят техните процеси.



Фигура 5. Изпълнение на първия сценарий в симулатора

Процесите на всички участващи амбиенти могат да се проследят при всяка стъпка от изпълнението на сценария, което дава възможност веднага да се установят несъответствия, грешки и неточности (Фигура 6).



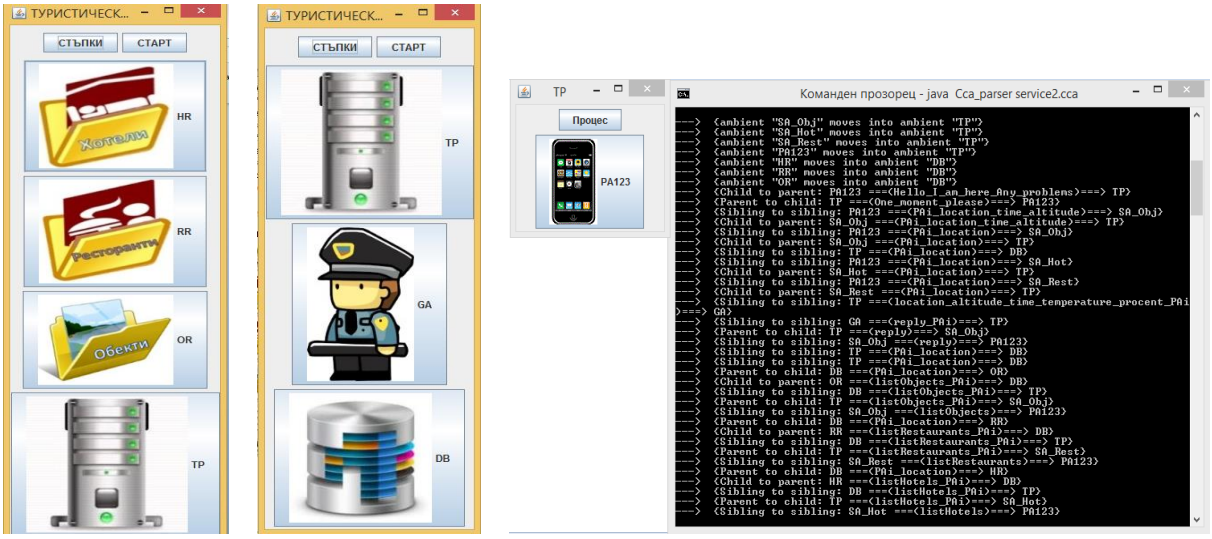
Фигура 6. Валидиране на отделните процеси и съобщения

Втората моделирана услуга предоставя на туриста възможност при подадена от неговото мобилно устройство информация за местоположението, надморската височина и за времето от денонощието, системата да определи има ли опасност за потрелителя и изпраща извънредно съобщение заедно със списъка на най-близките хотели и местта за отдих. Ако няма опасност за туриста, системата му изпраща списък на най-близките туристически обекти. Валидирането на тази услуга се формализира чрез следния скрипт:

```
SA_Obj[in TP.0]|SA_Hot[in TP.0]|SA_Rest[in TP.0]|
PA123[in TP.0]|
TP[PA123[TP@send(Hello_I_am_here_Any_problems).TP@recv(x1) .0]    |
PA123#recv(y1).<y1=Hello_I_am_here_Any_problems>PA123#send(One_moment_please).0|
 SA_Obj[PA123 ::
recv(PAi_location_time_altitude). TP @ send(PAi_location_time_altitude). 0|
TP@recv(reply).PA123::send(reply).0|
PA123::recv(PAi_location).TP@send(PAi_location).0|
TP@recv(listObjects_PAi).PA123::send(listObjects).0]|
SA_Hot[PA123::recv(PAi_location).TP@send(PAi_location).0|
TP@recv(listHotels_PAi).PA123::send(listHotels).0]|
 SA_Rest[PA123::recv(PAi_location).TP@send(PAi_location).0|
TP@recv(listRestaurants_PAi).PA123::send(listRestaurants).0]|
```

```
 PA123[ SA_Obj::send(PAi_location_time_altitude).SA_Obj::recv(reply).0|
SA_Obj::send(PAi_location).SA_Obj::recv(listObjects).0|
SA_Hot::send(PAi_location).SA_Hot::recv(listHotels).0|
SA_Rest::send(PAi_location).SA_Rest::recv(listRestaurants).0]|
SA_Obj#recv(PAi_location_time_altitude).
GA::send(location_altitude_time_temperature_procent_PAi).0|
GA::recv(reply_PAi).SA_Obj#send(reply).0|
SA_Obj#recv(PAi_location).DB::send(PAi_location).0|
DB::recv(y10).<y10=listObjects_PAi>SA_Obj#send(listObjects_PAi).0|
SA_Rest#recv(z3).<z3=PAi_location>DB::send(z3).0|
DB::recv(z4).<z4=listRestaurants_PAi>SA_Rest#send(listRestaurants_PAi).0|
SA_Hot#recv(x11).<x11=PAi_location>DB::send(x11).0|
DB::recv(x12).<x12=listHotels_PAi>SA_Hot#send(listHotels_PAi).0 ]|
GA[!::recv(x7).<x7=location_altitude_time_temperature_procent_PAi>TP::
      send(reply_PAi).0]|
DB[OR[!@recv(PAi_location).DB@send(listObjects_PAi).0]
|TP::recv(y6).<y6=PAi_location>OR#send(y6).OR#recv(y7).
<y7=listObjects_PAi>TP::send(listObjects_PAi).0|
RR[!@recv(PAi_location).DB@send(listRestaurants_PAi).0]
|!::recv(y8).<y8=PAi_location>RR#send(y8).RR#recv(y9).
<y9=listRestaurants_PAi>TP::send(listRestaurants_PAi).0|
HR[!@recv(PAi_location).DB@send(listHotels_PAi).0]
|!::recv(y12).<y12=PAi_location>HR#send(y12).HR#recv(y13).
<y13=listHotels_PAi>TP::send(listHotels_PAi).0]
```

Симулаторът в ccaPL програмната среда (Фигура 7) дава възможност да се проследи последователността на изпращаните и получавани съобщения между амбиентите, съгласно моделирания сценарий.



Фигура 7. Изпълнение на втория сценарий в симулатора

Въз основа на валидирането на моделираните услуги чрез ccaPL симулатора се разработи прототип на туристическа система, която реализира отделни елементи от моделираните услуги (Фигура 8).

Фигура 8. Прототип на туристическия пътеводител с частична реализация на моделираните услуги

Заключение и бъдещи планове

Моделирането на контекстно-чувствителна туристическа информационна система чрез нотацията ССА дава възможност за представяне и реализация на различни услуги, отчитащи динамичната промяна в параметрите на околната среда. Създадени са два прототипа за частично верифициране на двата представени сценария. Верификацията на модела изисква създаването на симулатор на процесите и разработката на прототип. За целта на изследването е използван езикът за програмиране на ССА – ccaPL. Създаването на анимирана среда за симулация на сценариите дава възможност за проследяване на процесите и обмяната на съобщения между участниците в системата и спомага за верификацията и валидацията на разработения модел, преди да се премине към неговото цялостно реализиране.

Благодарности

Авторът изказва благодарност към научен проект ФП17-ФМИ-008 „Иновационни софтуерни инструменти и технологии с приложения в научни изследвания по математика, информатика и педагогика на обучението“ към Фонд „НИ“ на ПУ „Паисий Хилендарски“, за частичното финансиране на настоящата работа.

Литература

Schmidt, A. and K. Laerhoven, How to build smart appliances, *IEEE Personal Communications*, 2001, 66–71, ISSN 1070-9916.

Stoyanov, S., A Virtual Space Supporting eLearning, *Proceedings of the Forty Fifth Spring Conference of the Union of Bulgarian Mathematicians*, Pleven, 2016, 72–82.

Стоянов, С. и И. Попчев, Инфраструктури за електронно обучение, списание „*Техносфера*“, БАН, 2015, 4 (30), 38–45.

Dey, A. and G. Abowd, Towards a better understanding of context and context-awareness, *Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness*, New York, ACM Press, 2000.

Dey, A., Understanding and Using Context, *Personal and Ubiquitous Computing Journal*, Vol. 5, No. 1, 4–7. 2001.

Siewe F., Z. Hussein and C. Antonio, The Calculus of Context-aware Ambients, *Journal of Computer and System Sciences*, Vol. 77, Issue 4, July 2011, 597–620, doi:10.1016/ j.jcss.2010.02.003

Siewe F., A. Cau and Z. Hussein, CCA: aCalculus of Context-aware Ambients, Conference: Advanced Information Networking and Applications Workshops, 2009, WAINA '09. International Conference, DOI: 10.1109/WAINA.2009.23 Source: IEEE Xplore

Alotaibi, H., Context-Aware and Secure Workflow Systems, Ph.D Thesis, Software Technology Research Laboratory, De Montfort University, Leicester, United Kingdom, 2012.

Zedan, H., F. Siewe and A. Cau, The validation of context-aware systems, Technical Report, Software Technology Research Laboratory, 2008.

Valkanov, V., *Context-oriented management of electronic services*, Academic Publishing House “Prof. Marin Drinov”, Sofia, Bulgaria, 2013, ISBN 978-954-322-701-3.

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

Siewe, F., H. Zedan and A. Cau, The calculus of context-aware ambients, *Journal of Computer and System Sciences*, In Press, Corrected Proof, 2010.

Al-Sammarraie, M., *Policy-based Approach For Context-aware Systems*, PhD Thesis, Software Technology Research Laboratory, De Montfort University, Leicester – United Kingdom, July 2011.

Stoyanov, S., I. Popchev and D. Orozova, Virtual Education Space – present and future, *International Conference “The New Idea in Education”*, Burgas, 410–418, 2016.

Пловдивски университет „П. Хилендарски“,

Факултет по математика и информатика

бул. „България“ № 236, 4003, Пловдив

glushkova@uni-plovdiv.bg, 0878212950

A PROTOTYPE OF CONTEXT-AWARE TOURIST GUIDE

Todorka Glushkova

Summary: *The Mathematical Notation of Context-aware Ambients (CCA) is suitable for modeling of contex-aware mobile systems. Since it is difficult to reproduce the syntax of CCA in a programming environment due to the specific symbols used in the notation, we will use the ccaPL language, which is a computer-readable version of the syntax of the CAA, to create a contex-aware travel guide simulator. The article will present the processes of verifying and validating the scenarios of various services by modifying and developing an animated simulator for prototyping the system.*

Key words: *Calculus of Context-aware Ambients (CCA), ccaPL*