

*Пловдивски университет „Паисий Хилендарски“
Факултет по математика и информатика
Катедра „Компютърни системи“*

КОНТЕКСТНО-ЗАВИСИМО МОДЕЛИРАНЕ В КИБЕР-ФИЗИЧЕСКО ПРОСТРАНСТВО

АВТОРЕФЕРАТ

*на дисертационен труд за присъждане на образователна и научна
степен „доктор“ в област*

*4. Природни науки, математика и информатика, професионално
направление:*

*4.6. Информатика и компютърни науки, докторска програма:
Информатика*

Докторант: Константин Николаев Русев

Научен ръководител: доц. д-р Тодорка Атанасова Глушкова

Рецензенти:

1. Чл. кор. Проф. д.н. Любка Атанасова Дуковска
2. Доц. д-р Ирина Александровна Радева
3. Проф. д-р Мария Петкова Христова
4. Доц. д-р Тодорка Атанасова Глушкова
5. Проф. д-р Станимир Недялков Стоянов

Дисертационният труд е обсъден и насочен за защита пред научно жури на заседание на катедра „Компютърни системи“ при Факултета по математика и информатика на ПУ „Паисий Хилендарски“ на 24.10.2022 г.

Дисертационният труд съдържа **153 страници**. Библиографията включва **111 източника**. Броят на авторските публикации е 4.

Защитата на дисертационния труд ще се състои на 12.01.2023 год. в Заседателната зала на Нова сграда на ПУ „Паисий Хилендарски“, гр. Пловдив.

Материалите по защитата са на разположение на интересуващите се в секретариата на ФМИ – каб. 330 в Нова сграда на ПУ „Паисий Хилендарски“, всеки работен ден от 8:30 до 17:00 часа.

Автор: Константин Николаев Русев

Заглавие: Контекстно-зависимо моделиране в кибер-физическо пространство

Пловдив, 2022 г.

Съдържание

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД	4
Основни задачи	4
Методика на изследването	5
Структура на дисертацията	5
КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД	7
Глава 1. Състояние на изследвания проблем	7
1.1. Кибер-физически и социални пространства	7
1.2. Амбиент-ориентирано моделиране. ССА	8
Глава 2. Амбиент-ориентирано моделиране във Виртуално-физическо пространство	10
2.1. Виртуално-физическо пространство	10
2.2. Концептуален модел на Компонент за контекстно-чувствително ССА-моделиране	11
2.3. Структура на компонент за контекстно ССА моделиране	13
Глава 3. Разработка на компонент за ССА моделиране	14
3.1. Софтуерна архитектура	14
3.2. Жизнен цикъл	16
3.3. Прототип	17
Глава 4. Тестване на компонента за моделиране	21
4.1. ССА моделиране в земеделието	21
4.2. ССА моделиране в образованието	22
4.3. ССА моделиране в интегрирани приложни области	23
АВТОРСКА СПРАВКА ЗА РЕЗУЛТАТИТЕ В ДИСЕРТАЦИОННИЯ ТРУД	26
Основни задачи, решени в дисертационното изследване	26
Предимства на създадения компонент за ССА моделиране	27
Бъдещо развитие	28
Приноси	28
Апробации	29
ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД	30
Библиография	31

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

Тенденциите в развитието на дигиталните технологии в ерата на Четвъртата индустриална революция поставят нови хоризонти в създаването на интелигентни пространства, в които хора, дигитални устройства и обекти от физическия и виртуалния светове си взаимодействат, за да предоставят необходимата информация и подходящи услуги на крайния потребител [13]. Контекстно-чувствителните системи са тези системи, които могат да адаптират поведението си в зависимост от конкретния контекст. Тези системи са твърде комплексни за непосредствена разработка и изискват задълбочен процес на предварително моделиране, тестване и верифициране на ключовите процеси, услуги и сценарии. Моделирането и представянето на пространствените аспекти на „нещата“ е от ключово значение при създаването на динамични, контекстно-чувствителни системи [16]. Съществуват различни формални системи за улавяне, обработката и моделиране на контекстно-чувствителната информация.

През последните няколко години във Факултета по математика и информатика на Пловдивския университет „Паисий Хилендарски” се разработва референтна архитектурна рамка на виртуално-физическо пространство (ViPS), която позволява адаптиране към различни приложни области [10]. При разработката на ViPS за моделиране на пространствените аспекти на „нещата“ е избрана формалната семантика на амбиент-ориентираното моделиране и в частност на Calculus of Context-Aware Ambients (CCA), което дава възможност за улавяне и моделиране на контекстната информация в динамично променяща се среда. Референтната ViPS архитектура дефинира необходимостта от разработката на специален компонент за амбиент-ориентирано моделиране, чиято задача е да предостави необходимите инструменти и средства за предварително моделиране, тестване, оптимизиране и съхранение на основните процеси и сценарии в контекста на конкретния домейн – образование, туризъм, интелигентно селско стопанство и т.н.

Основна цел на дисертацията

Да се разработи концептуален модел и прототип на компонент за контекстно-зависимо моделиране на процеси, услуги и сценарии във виртуално-физическо пространство.

Основни задачи

Задача 1: Разработване на модел и архитектура на компонент за контекстно-чувствително CCA-моделиране във виртуално-физическо пространство.

Задача 2: Разработка на прототип на визуален ССА редактор за предварително моделиране на процеси и сценарии във виртуално-физическото пространство.

Задача 3: Интегриране на прототипната реализация на ССА редактора в компонента за моделиране.

Задача 4: Провеждане на експерименти с използването на Компонента за моделиране при моделирането на процеси в контекста на един домейн и в контекста на интегрирани домейни.

Методика на изследването

- *Разработване на концепция, модел и архитектура на компонент за контекстно-чувствително ССА-моделиране във виртуално физическо пространство ViPS;*
- *Разработка на първа версия на прототип на ССА визуален редактор за контекстно-чувствително моделиране;*
- *Интегриране на прототипната реализация на ССА редактора в компонента за моделиране;*
- *Разработка на втора версия на ССА редактора, който дава възможност за моделиране в интегрирани домейни;*
- *Тестване на създадената система за контекстно-зависимо моделиране в контекста на отделни или на интегрирани домейни.*

Структура на дисертацията

Увод – В уводната част на дисертацията се мотивира необходимостта от провеждане на изследването, представено в дисертацията. Формулирана е целта на дисертацията, определени са задачите за постигане на целта и е представена методологията за провеждане на изследването.

Глава 1. Състояние на изследвания проблем – разглежда се прогресът и текущото състояние на изследвания проблем. Представя се парадигмата и подходът, който ще се използва за моделиране на системи от различни приложни области.

Глава 2. Амбиент-ориентирано моделиране във Виртуално-физическо пространство – разглежда се ViPS (*Virtual-Physical Space*) пространството и всички подсистеми, които са част от него. Обръща се детайлно внимание на Аналитичното подпространство от ViPS архитектурата, където важна роля заема амбиент-ориентираното моделиране. Представя се концептуален модел за разработка на модул за контекстно-ориентирано ССА моделиране като част от тази архитектура.

Глава 3. Разработка на компонент за ССА моделиране – представя се реализацията на прототип, неговите възможности и функционалности, чиято

цел е да подпомогне разработката на сценарии и модели, чрез предоставяне на помощни средства, чрез които моделиращият има възможност да разработи даден сценарий без да притежава експертни знания за системата, парадигмата и езика за моделиране. Представени са и първите няколко версии на прототипа.

Глава 4. Тестване на компонента за моделиране – описват се редица сценарии и модели от различни приложни области (*домейни*), които са изцяло разработени с помощта на разработения компонент за моделиране с цел да се онагледят и докаже неговата полезност и ефективност.

Заклучение – заключението обобщава резултатите от дисертацията и дава идеи и насоки за бъдещото ѝ развитие.

КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

Глава 1. Състояние на изследвания проблем

Четвъртата индустриална революция (*Industry 4.0*) споделя разбирането, че огромният потенциал на изкуствения интелект, виртуалната реалност, връзката между физическия и виртуалния светове и пр. могат да се използват за подобряване на качеството на живота на хората в съвременния свят [12]. Счита се, че ключова роля в Четвъртата индустриална революция ще изиграят трите тясно свързани парадигми на „Интернет на нещата“ (*IoT*), кибер-физически системи (*CPS*) и кибер-физическите-социални системи (*CPSS*) [20]. В тази глава ще бъдат разгледани различни аспекти, свързани с интеграцията между физическия и виртуалния светове, както и изграждането на контекстно-зависими интелигентни пространства. Ще бъдат разгледани особеностите и ключовите характеристики на *CPS* и *CPSS* пространства.

Поради високото ниво на комплексност, създаването на такива системи изискват задълбочено предварително проучване, моделиране и тестване на основните взаимодействия, сценарии и услуги. В тази глава ще бъде разгледан формалният модел на амбиент-ориентираното програмиране, част от което е математическата нотация на *Calculus of Context-Aware Ambients*.

1.1. Кибер-физически и социални пространства

Според дефиницията на National Science Foundation (*NSF*) [15] *кибер-физическите системи* са проектирани така, че да осигуряват безпроблемното интегриране на изчислителни алгоритми и физически компоненти. *Кибер-физическо-социалното пространство* е сливане на физическото пространство, киберпространството и социалното пространство. Еволюцията от *CPS* към *CPSS* [24] е дълъг процес, който е свързан с решаването на предизвикателства и проблеми от различен характер. *CPSS* обръща внимание на дълбокото взаимодействие между физическия, кибернетичния и социалния свят чрез интегриране на физическия и социалния чрез интелигентното взаимодействие „човек-машина“ в киберпространството за постигане на адекватно управление и контрол на сложни социално-технически системи [25].

През последните години във Факултета по математика и информатика на Пловдивския университет се разработва референтна архитектура на Виртуално-физическо пространство (*ViPS*) [19]. По своята същност това е архитектура на *CPSS* пространство, която може да се адаптира към различни приложни области – електронно обучение [7], [22] земеделие [18], туризъм [5] и др.

Казваме, че едно пространство е интелигентно, ако е в състояние да наблюдава собственото си поведение, да отчита настъпващите промени и да реагира на тези промени. Това е предпоставка за възникването на специфични

проблеми и предизвикателства, едно от най-значимите, от които е контекстът и осигуряването на контекстна зависимост.

Съществуват различни дефиниции на понятието контекст. В дисертацията ще се използва дефиницията на Дей (Dey) [4], където **контекст** е всяка информация, която може да се използва за категоризиране на състоянието на една идентичност. Идентичност може да бъде човек, място или обект, който се смята за свързан с взаимодействието между потребител и приложение, като включва самите потребител и приложение.

Формализацията на контекстно-зависимите системи е от съществено значение за тяхното представяне, проектиране и разработка. Съществуват различни формални методи като: *Метод базиран на модели*; *Алгебрични методи*; *Методи базирани на процесното смятане (process calculus)*; *Логически методи*; *Методи базирани на мрежи* и др. В дисертацията са представени методите на процесното смятане и по-конкретно на амбиент-ориентираното моделиране.

1.2. Амбиент-ориентирано моделиране. ССА

Един **амбиент** (*ambient*) е идентичност, която притежава следните характеристики [2]: ограниченост, вложеност и мобилност. За целите на моделирането един амбиент може да бъде представен като структура със следните елементи: *идентификатор*, *кореспондиращо множество от амбиентни процеси* и *множество от прилежащи под-амбиенти*.

Основополагащ формализъм, подходящ за амбиент-ориентирано моделиране, е *π-калкулът* [14], който представя вид процесно смятане. Този формализъм позволява да се опишат едновременно изчисления, чиято мрежова конфигурация може да се промени по време на изчислението. Едно съществено разширение на *π-калкула*, познато като Ambient Calculus (AC) е представено в [2], където е възприета парадигмата за мобилност. В нея амбиентите са йерархично структурирани, агентите са ограничени до амбиенти и амбиентите се движат под контрола на агенти. Наследник на AC е Calculus of Boxed Ambients (CBA) [1].

За моделирането на кибер-физическо и социално пространство може да се използва Calculus of Context-aware Ambients (CCA), който надгражда CBA с нови конструкции, за да се осигури възможност на мобилните амбиенти да реагират в зависимост от промените на средата, в която се изпълняват. За представяне на свойствата на CCA процесите се използва логически език, в който основна роля играе контекстния израз. Контекстните изрази се използват в CCA, за да се гарантира изпълнението на определена възможност само при наличието на определени условия на средата, т.е. в определен контекст.

ССА-амбиентът е идентичност, която се използва за описание на обекти или компонент – процес, устройство, местоположение и пр. Всеки AMBIENT притежава име, граници и може да съдържа други AMBIENT в себе си, както и да бъде включен в друг AMBIENT. Между два AMBIENT има три възможни взаимоотношения – родител, дете, роднина. Всеки AMBIENT може да комуникира с AMBIENTите около себе си. AMBIENTите могат да обменят съобщения помежду си. Процесът по обмяна на съобщения се извършва чрез процеса „ръкостискане“ (*handshaking*). В ССА-нотацията:

- „::“ е символ за роднини AMBIENT;
- „↑“ и „↓“ са символи за родител и дете;
- „< >“ означава изпращане;
- „()“ – получаване на съобщение.

Един AMBIENT може да бъде мобилен, т.е. да се движи в рамките на заобикалящата го среда. При ССА има две възможности за движение: навътре (*in*) и навън (*out*), които позволяват на AMBIENTите да се придвижват от едно местоположение в друго. В ССА-нотацията могат да се разграничат 4 синтактични категории: *процеси* „P“; *възможности* „M“; *местоположения* „a“ и *контекстни изрази* „k“.

Програмният език „ssaPL“ е компютърно четима версия на синтаксиса на ССА. Интерпретаторът на ssaPL е реализиран на Java и позволява симулация и анимация на ССА процеси. Програмирането се реализира в текстов файл с разширение „*.ssa“. Тъй като е трудно да се възпроизведе ССА синтаксисът с помощта на текстов редактор, поради специфичните символи, като например „↑“ и „↓“, се налага да се модифицира синтаксисът на ssaPL. Базовата версия на интерпретатора на ssaPL работи в конзолен режим. Всеки ред на изходните данни на програмата представлява връзките и взаимодействията на AMBIENTите в ССА-модела.

Глава 2. Амбиент-ориентирано моделиране във Виртуално-физическо пространство

2.1. Виртуално-физическо пространство

Виртуалното физическо пространство (ViPS) се изгражда като референтна архитектура, която може да бъде адаптирана към различни CPSS приложения в редица приложни области. ViPS-архитектурата трябва да осигурява виртуализация на реални обекти, които са значими за разглеждания домейн. Виртуализацията на „нещата“ се поддържа от междинния софтуер на ViPS, включващ две основни подпространства- „Аналитично подпространство“ и „Подпространството на цифровите библиотеки“, както и от „Мениджър на събитията“ [21] и оперативни асистенти с различни функционалности и отговорности.

Аналитичното подпространство предоставя средства за дигитализация и представяне на информацията, свързани с време, пространство и местоположение на обектите. В този смисъл аналитичното подпространство е изградено от следните базови компоненти:

- TNet (*Temporal Net*) предоставя възможност за представяне на временните аспекти на „нещата“, събитията и местоположенията;
- AmbiNet (*Ambient Net*) осигурява възможността пространствените характеристики на „нещата“ и събитията да бъдат моделирани чрез контекстно-чувствителни амбиенти.

Мениджърът на събитията моделира и интерпретира различни видове събития, които са значими за разглежданото поле на интерес. Работата с тези компоненти се поддържа от специализирани интерпретатори, внедрени във ViPS като оперативни асистенти, базирани на официалните спецификации на Interval Temporal Logics (ITL) [3], [23], Calculus of Context-aware Ambients (CCA) [17] и Event Model (EM) [21].

AmbiNet е структура, моделираща съответния домейн (*интелигентен град, туристическа зона, университетски кампус, средно училище и др.*). Според основната ѝ задача AmbiNet трябва да работи в симулативен режим и да реализира своите функционалности чрез мрежова структура от „амбиенти“ [9].

TNet е компонент на аналитичното подпространство, който поддържа времевите аспекти на изследваните процеси и тестовите сценарии. Той базира своите функционалности на формалната семантика на интервалната и темпорална логика (*Interval Temporal Logic – ITL*).

Взаимодействието между двата компонента на аналитичното подпространство (*AmbiNet* и *TNet*) може да се разглежда от различни аспекти. Нека насочим вниманието си към фазата на превантивно планиране преди

началото на изпълнението на определено събитие. На този етап личният асистент на потребителя трябва да подготви план за постигане на конкретна цел, чрез взаимодействие с аналитичното подпространство. По-специално, планът се генерира с помощта на AmbiNet, работещ в режим на моделиране. Като цяло планът се представя под формата на възможен маршрут, който трябва да осигури присъствието на потребителя на необходимото място навреме за участие в предстоящото събитие. Следователно моделирането на процесите и оценката на отделните планове в Аналитичното подпространство изисква създаването на **концептуален модел** на компонент за контекстно-чувствително моделиране в рамките на AmbiNet.

2.2. Концептуален модел на Компонент за контекстно-чувствително ССА-моделиране

Взаимодействието между отделните компоненти в Аналитичното подпространство се извършва в определен контекст, който се дефинира според текущата информация от околния свят – физически и виртуален. Според приетата дефиниция базовият контекст се представя като наредена четворка: (*идентичност, местоположение, активност, време*). Тоест:

$$k = (identity, location, activity, time) \quad (1)$$

Всяка промяна на тези категории, променят и контекста. Да разгледаме множеството от различни контексти в един определен сценарий $K = \{k_1, k_2, k_3, \dots, k_n\}$, като $k_i \in K$ и $k_i = (ID_i, LOC_i, ACT_i, TIME_i)$, за всяко $i=1, \dots, n$.

В реално време гардовете следят за промените в параметрите на околната среда, което автоматично води до промяна на текущия контекст. **В режим на моделиране** е необходимо да се симулира промяната на параметрите на средата, което да доведе до промяна на контекста.

AmbiNet работи в режим на моделиране. Амбиентите формализират идентичностите, включени в тестовия сценарий. Тъй като те унифицират обработката на обектите от физическия и виртуалния светове на CPSS пространството, е необходимо да разширим техните характеристики. По дефиниция амбиентите представят в системата отделни идентичности и техните локации по отношение на амбиентната йерархия. Тъй като във всеки момент в тази йерархия всеки амбиент има само един родител, или няма такъв, местоположението в йерархията еднозначно се описва с родителския амбиент (*ако няма родител, за родител записваме None*). Тъй като амбиентите са мобилни и могат да променят местоположението си в амбиентната йерархия, тази структура се променя динамично. Освен това всеки амбиент има дефинирано множество от процеси „ $P(a)$ “. Тогава амбиентът се представя като наредената тройка:

$$a = (Name, Parent, P(a)) \quad (2)$$

Освен това е необходимо да се укаже, дали представляваната от амбиента идентичност е от физическия, или от виртуалния свят. За физическите идентичности е съществено дали могат да променят местоположението си в реалния свят или не. Например амбиентът „Музей“ представя обект от физическия свят, който е неподвижен. Статичните физически амбиенти имат и конкретна географска локация. Тогава можем да **разширим представянето на амбиента** с още няколко категории. Така амбиентът може да се представи в AmbiNet като **наредена шесторка** (*идентичност, родител, тип, подвижност, локация, процеси*):

$$a = (Name, Parent, Type, Dynamicity, Location, P(a)) \quad (3)$$

В този запис: **Name** е уникално име на амбиент; **Parent** е името на родителя му в амбиентната йерархия. Ако няма родител, получава стойност *None*; **Type** е типът на идентичността, описвана с амбиента (*физическа или виртуална*); **Dynamicity** е параметър, определящ дали амбиентът е *статичен* или *подвижен*; **Location** е местоположението на обекта във физическия свят; **P(a)** са множеството от амбиентните процеси. Нека множеството на амбиентите, представящи идентичностите в един примерен сценарий е $A = \{a_1, a_2, a_3, \dots, a_m\}$, като $a_i = (ID_i, PAR_i, TYPE_i, DYNAM_i, LOC_i, P(a_i))$.

В ССА два амбиента могат да обменят съобщения само, ако се намират в едно от отношенията: „родител“, „дете“ или „роднина“. Тогава, амбиентите „ a_i “ и „ a_j “ могат да комуникират директно само, ако са в едно от тези отношения. В противен случай те предават съобщенията през амбиентната йерархия. $a_3 \xrightarrow{\langle message \rangle} a_2; a_2 \xrightarrow{\langle message \rangle} a_1$ (4)

ССА амбиентите са мобилни и могат да променят местоположението си в амбиентната йерархия. Ако амбиентът **b** е „дете“ на амбиент **a**, това означаваме като: $a = parent(b)$ (5)

Нека за **a** и **b** $\in A$, дефинираме следните две операции:

- $b \downarrow a(in)$ – амбиент **b** влиза в границите на **a** и $a = parent(b)$.
- $b \uparrow a(out)$ – амбиент **b** излиза извън границите на **a** и става негов „роднина“. Тогава $parent(a) = parent(b)$.

Следователно два амбиента могат да комуникират помежду си, като изпращат и получават съобщения, само ако е изпълнено условието:

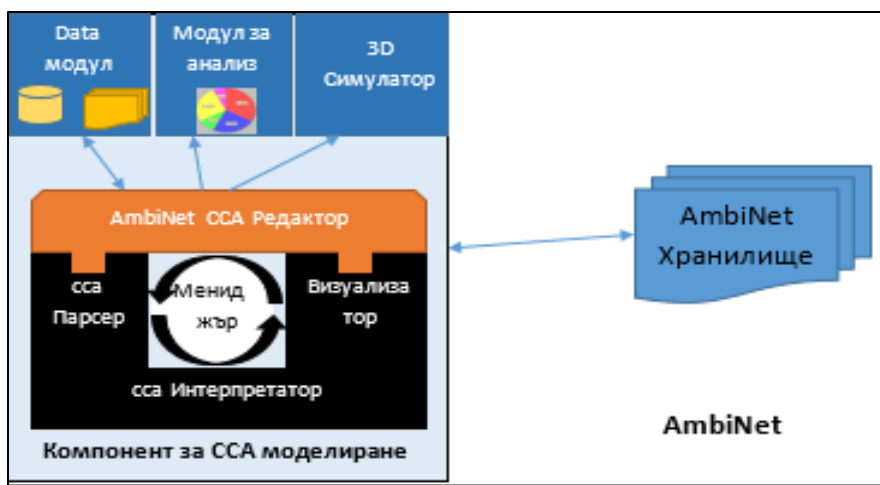
$$(a = parent(b)) OR (b = parent(a)) OR (parent(a) = parent(b)) \quad (6)$$

Всяка промяна на контекста, може да доведе до промяна на амбиентната йерархия, или на взаимодействието между амбиентите в моделирания сценарий. Следователно, в симулативен режим, **промяната на контекстите** се отразява при разработката на отделни базови сценарии за съответната

приложна област. Тези сценарии могат да се формализират чрез взаимоотношенията между ССА амбиентите, представени в концептуалния модел.

2.3. Структура на компонент за контекстно ССА моделиране

Поради своята специфика и целите, които трябва да изпълнява, разработваният компонент е част от AmbiNet на аналитичното подпространство. Въз основа на очакваната промяна на контекстната информация в моделираната предметна област, се разработват примерни базови сценарии, които описват процесите на реализация на очакваните функционалности на участващите идентичности. За ССА-базираното моделиране на тези сценарии се разработва **AmbiNet ССА Редактор**, който да предостави възможност за визуален, и в голяма степен интуитивен, процес на работа, съобразен с представения по-горе модел. Макар този редактор да предоставя допълнителни разширени функционалности, той трябва да поддържа изискванията на формализма ССА. Затова генерираните от него модели трябва да могат да се изпълняват, тестват и верифицират от класическия **ССА-Интерпретатор**. За осигуряване на възможност за работав интегрирани домейни, както и за използване и управление на допълнителните характеристики на амбиентите (*според концептуалния модел*) е необходимо да се създаде **Data модул**. В него ще се съхранява, както текущата информация за амбиентите, така и за приложните области, в чийто контекст се използват тези амбиенти. Освен това в този модул може да се съхранява оперативна информация за работата на редактора като сса-файлове, csv представяния на данни, съобщения и др. Разработката на **3D симулатор** към редактора ще осигури по-ясна визуализация на процесите в моделирания сценарий. Аналитичната обработка на моделираните процеси и нейното графично и таблично представяне е задача на **Модула за анализ**. Генерираните и тествани сценарии в подходящ за бъдещо използване вид се съхраняват в **AmbiNet Repository**.



Фигура 1. Структура на компонент за ССА моделиране

Глава 3. Разработка на компонент за ССА моделиране

С цел да се реализира **втората основната задача** от дисертацията се разработва софтуерно приложение, което да подпомогне и улесни процеса по разработката на ССА модели.

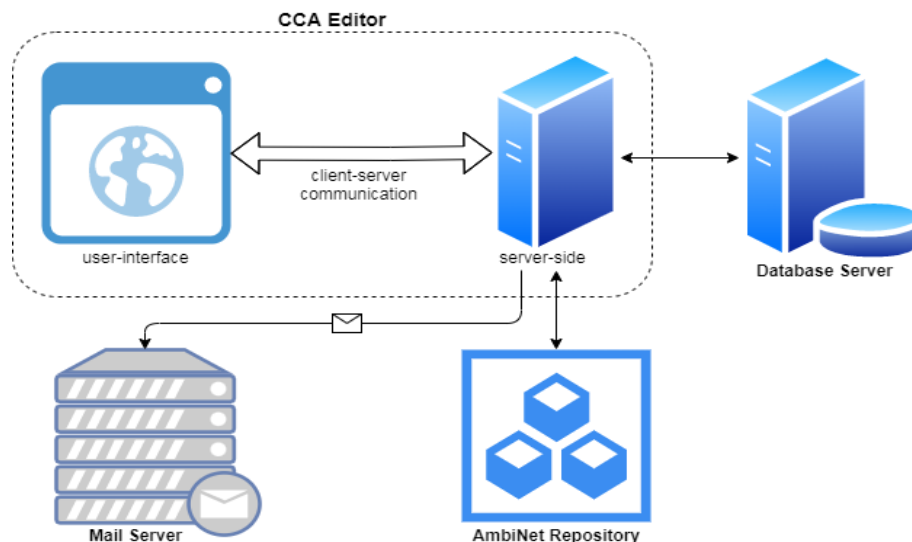
В **първата версия** на ССА редактора трябваше да се реализира функционалност, която да позволи да се създават базови модели и сценарии в термините на обучението във виртуалното образователно пространство, като в процеса на разработка започнаха да се появяват възможности и идеи за добавяне на нови функционалности към първоначалната версия.

Във **втората** си прототипна версия ССА редакторът имплементира възможности за моделиране на сценарии, не само свързани с виртуалното образователно пространство, но с в други приложни области като земеделие, туризъм и други. Това позволява да се оптимизират работните процеси в различните приложни области и да се установят проблеми още в зародиш, което спестява много разходи, усилия и време впоследствие.

В **текущата** си версия приложението се модифицира отново, като то вече заема централна роля в модула „AmbiNet“ на аналитичното подпространство от архитектурата на ViPS. С цел да се позволи създаването не само на базови сценарии, но и на по-комплексни такива, се разшири функционалността, като вече се поддържа създаването на модели и сценарии, които да поддържат йерархия на отделните „амбиенти“ в интегрирани домейни. Заедно с това се реализира и функционалност за предаване на съобщенията между тях по начин, който остава незабелязан от моделирания и улеснява процеса по комуникация между отделните идентичности.

3.1. Софтуерна архитектура

Архитектурата на компонента за моделиране и ССА редактора, като централна част в него, е разработена в съответствие с представения концептуален модел и с цел да се тестват сценарии от различните приложни области, в които приложението може да бъде използвано в режим „моделиране“. На фигура 2 е показана основната диаграма на архитектурата на модула за моделиране с четирите основни компонента (*CCA Editor*, *Database Server*, *Mail Server*, *AmbiNet Repository*) и взаимодействието между тях. Тези компоненти са в постоянна взаимовръзка с цел да осигурят интуитивна, лесна, удобна и бърза работа при създаването на различни ССА модели и сценарии в различни приложни области.



Фигура 2. Архитектура на компонента за моделиране и CCA Editor

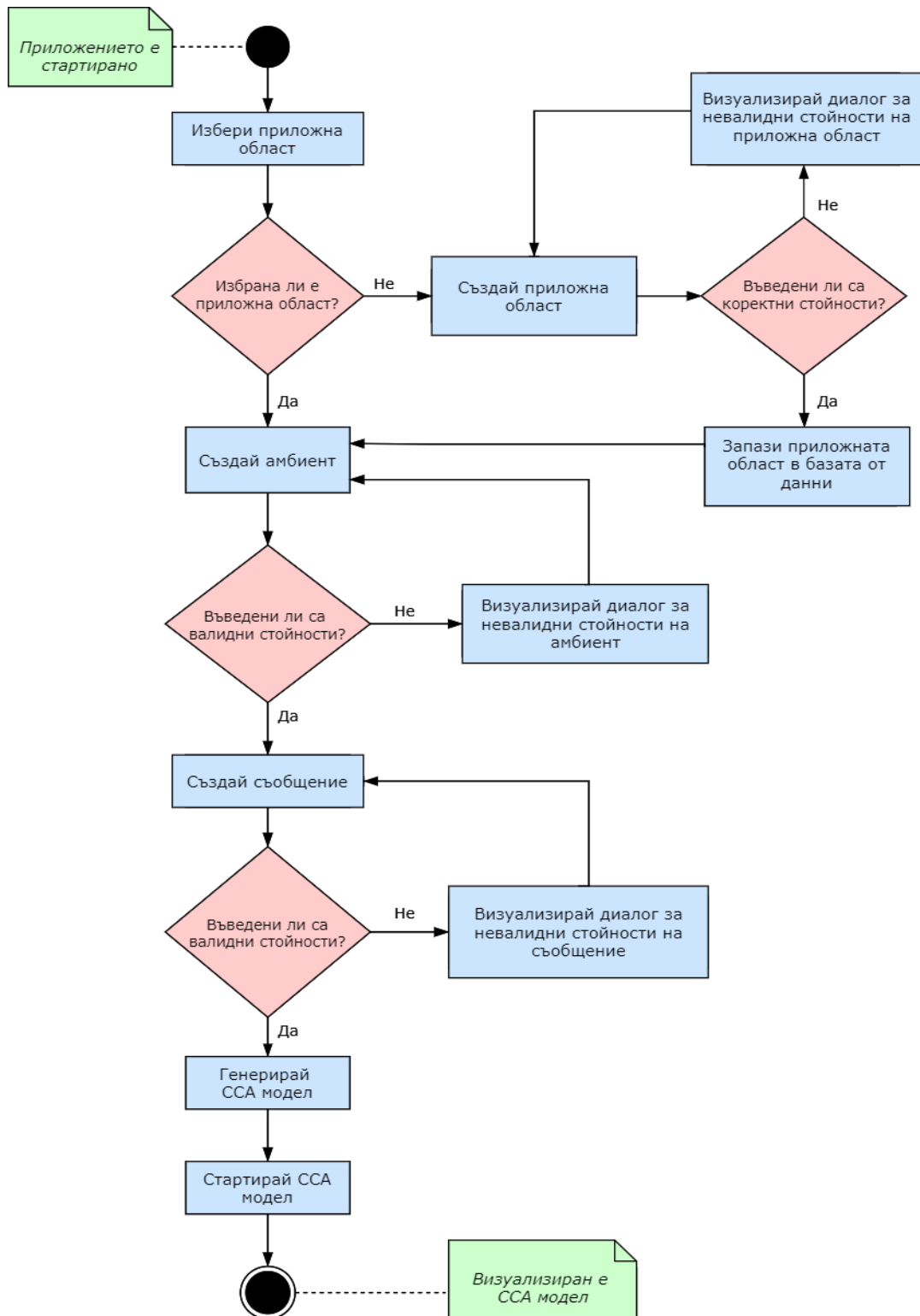
ССА редакторът е съставен е от два основни компонента – графичен потребителски интерфейс и сървърна част. Графичният потребителски интерфейс е разработен чрез Java-базираните библиотеки Swing и Abstract Window Toolkit (AWT). Сървърната част на CCA редактора (*server-side*) е разработена основно с помощта на обектно-ориентирания език за програмиране от високо ниво – Java. Този компонент от архитектурата на приложението има за цел да приеме и обработи информацията, въведена от крайния потребител, и да я съхрани в Data модула, с който е в постоянна комуникация. Освен с този модул, сървърът комуникира още с AmbiNet хранилището (*AmbiNet Repository*) с цел да се съхраняват вече разработените CCA модели и сценарии, както и със сървърът за изпращане на електронни писма (*Mail Server*), чрез който приложението предоставя възможност за виртуална комуникация между крайния потребител и администраторите на системата.

В **Data** модулът информацията се съхранява в структуриран и неструктуриран вид. За обработката на структурираната информация използваме релационни бази данни (PostgreSQL Server). Този модул е в постоянна комуникация със сървърната част на приложението с цел да съхрани създадените приложни области (*domains*), амбиенти (*ambients*) и съобщения (*messages*), които служат за комуникация между отделните амбиенти, както и да предостави информация за вече създадени такива.

AmbiNet хранилището е мястото, където се съхраняват всички разработени CCA модели и сценарии. Комуникацията между CCA редактора и хранилището се случва, когато редакторът иска да съхрани новосъздаден модел/сценарий и/или иска да извлече информация за вече съществуващ такъв. Съхранението на моделите и сценариите е изнесено във външно хранилище, с цел да може да бъде достъпвано и от други системи, с които редакторът няма пряка комуникация.

3.2. Жизнен цикъл

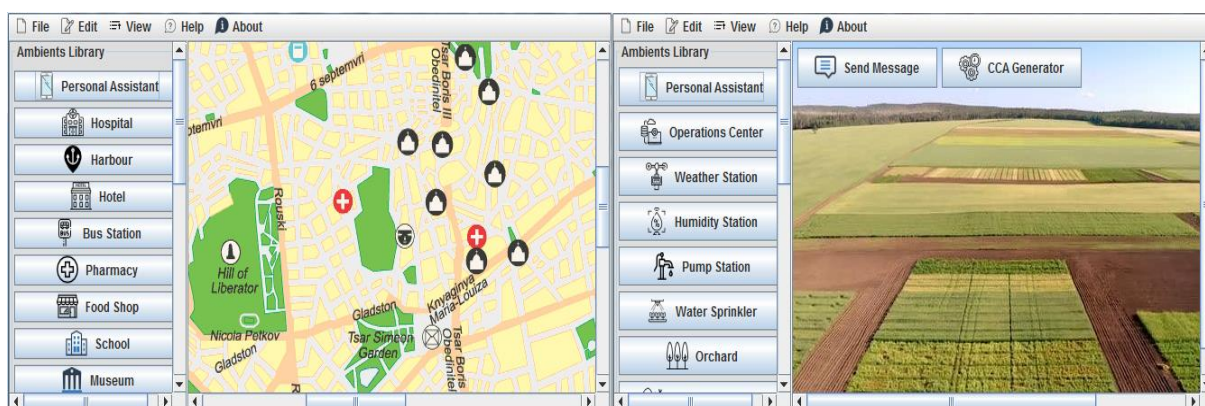
Жизненият цикъл на ССА редактора в режим „*моделиране*“ е **многостъпков последователен процес**, който е съставен от редица основни действия, които трябва да бъдат извършени с цел да бъде създаден коректно даден ССА модел в съответната приложна област (Фигура 3).



Фигура 3. Жизнен цикъл на CCA Editor

3.3. Прототип

Първоначално Компонентът за ССА моделиране беше замислен да оперира в само в един домейн. Впоследствие се породиха идеи да се добави функционалност, така че да могат да бъдат създавани модели и сценарии в различни интегрирани приложни области. Веднага след стартиране на приложението, моделиращият трябва да избере или създаде приложна област (*domain*), в която иска да моделира. Редакторът поддържа набор от предефинирани приложни области, които могат да се използват, но при необходимост се позволява и създаване на изцяло нови такива. Веднъж направил своя избор, моделиращият вижда началния екран (Фигура 4).



Фигура 4. Режими на работа на ССА редактора

На фигурата се вижда основният екран на ССА редактора, който е стартиран в режим моделиране за две абсолютно различни приложни области – „умен град“ и „умно земеделие“. **Библиотеката с амбиенти (Ambients Library)** – заема лявата част от основния екран на прототипа и съдържа набор от амбиенти, които са генерирани на база избраната предметна област. **Картата на приложна област** заема по-голямата част от основния екран на прототипа и зависи изцяло от избора на приложна област. Основната идея на картата е да онагледява предметната област и да визуализира местоположението на създадените амбиенти на база въведените, от потребителя географски, като те се преизчисляват динамично спрямо размерите на картата (*ширина и дължина*), потребителският екран и каква част от него е заета от приложението към момента. **Създаването на съобщение** се реализира след избора на бутон, който отвежда потребителя към друг екран, където да бъдат зададени специфичните характеристики, които притежава всяко едно съобщение, преди то да бъде официално генерирано. **Генерирането на ССА модела** също се реализира след избора на бутон (*CCA Generator*), намиращ се в непосредствена близост до бутона за създаване на съобщения (*Send Message*). Моделиращият трябва да е създал всички амбиенти и съобщения между тях преди да премине към тази стъпка, а тя ще го отведе до нов екран, предоставящ възможности за генериране, стартиране и визуализиране на конкретен ССА модел.

Създаването и модифицирането на амбиентите се реализира чрез шаблони, реализирани въз основа на концептуалния модел, представен в предходната част. Въвеждат се наименование, родителски амбиент, тип на амбиента, географско положение (*за физическите амбиенти*) и т.н.

Веднъж след като са създадени всички амбиенти, които са необходими за разработвания сценарий, от съответната приложна област, може да се премине към създаване на съобщения. За целта, трябва да се навигира до бутон „Send Message“, който ще отведе потребителя на изцяло нов екран, където ще трябва да се въведат всички необходими характеристики, така че дадено съобщение да може да се използва правилно при генерирането на ССА модела. Създаването на съобщения се реализира чрез използването на параметризирани шаблони, които осигуряват изпълнението на формалната семантика и синтаксис на ССА, според изискванията на представения в предходната част концептуален модел.

Следващата стъпка е генерирането и визуализацията на ССА сценария. Генерирането на ссаPL кода е сложен и многостъпков процес, като се наблюдават няколко основни етапа, през които трябва да се премине.

- **достъп до всички амбиенти и съобщения** (*които се съхраняват в Data модула*)
- **обработка на данните**
- **избор на подходящи шаблони**, които зависят от амбиентната йерархия, типа на съобщението и т.н. Шаблоните да се използват по време на процеса по генериране. Необходимостта от използването на шаблони произтича оттам, че ССА-нотацията и синтаксисът на математически базирания език „ссаPL“ са прекалено сложни. Шаблонът, сам по себе си, представлява фрагмент от код, който ще се използва по време на генериране на ССА сценария, като част от стойностите в него ще бъдат заместени с данни за конкретен амбиент и съобщение.

```
AMBIENT_SENDER
[
    AMBIENT_RECEIPIENT::send(SEND_PARAMETERS).0 |
    AMBIENT_RECEIPIENT::recv(RECEIVE_PARAMETERS).0
]
```

Фигура 5. Базов шаблон за генериране на ССА модел

- **генериране на ССА модел** – използва се вече избрания шаблон като всички маркери (*placeholders*) в него се заменят със стойностите за даден амбиент и съобщение. Напр., маркерът `AMBIENT_SENDER` ще бъде заменен от уникалния идентификатор (*наименование*) на амбиента (*подател*), който изпраща съобщението, `AMBIENT_RECEIPIENT` - с наименованието на амбиента получател, `SEND_PARAMETERS` ще бъде заместен от изпращаното

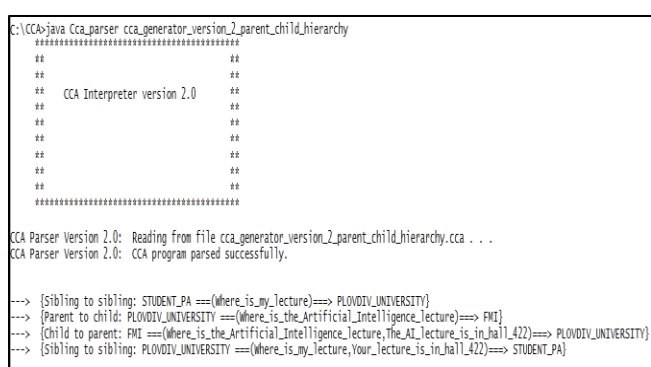
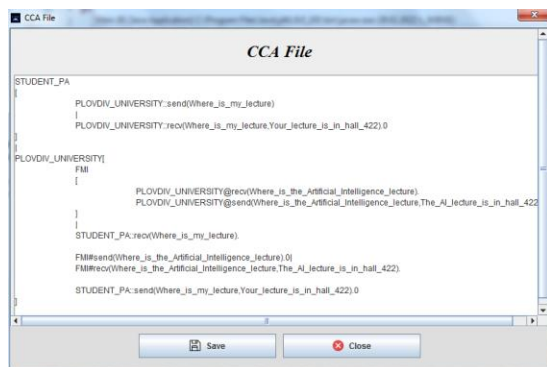
съобщение, в което могат да се включат редица други параметри (*идентификатор, местоположение, координати и други*), които е необходимо да се предадат от подателя към получателя и т.н. Всички останали символи, които са използвани при реализацията на шаблона, са част от синтаксиса на езика *ssaPL*, като те зависят от йерархията на амбиентите и това дали искаме да използваме паралелни процеси по време на комуникацията между амбиентите или не. Трябва да се има предвид, че интерпретаторът на програмния език „*ssaPL*“ не поддържа някои специални, които се заменят по време на процеса по генериране, така че да бъде възможно стартирането на сценария. Този процес се повтаря за всеки един амбиент и съобщение създадени в системата за приложната област, а когато приключи се преминава към фазата по съхранение.

- **съхранение на ССА модел** – Генерираният ССА модел се съхранява във файл с разширение „*.ssa“. Това е важно условие и се прави с цел сценарият да бъде разпознат от стандартизирания интерпретатора на езика „*ssaPL*“. Друго важно условие е моделът да се съхрани в един файл, а не в няколко такива. По време на този етап файлът се съхранява в специална директория в рамките на ССА редактора, а впоследствие може да бъде прехвърлен и в „*AmbiNet Repository*“. Налична е и функционалност за ръчно експортиране на файла в избрана, от моделиращия, директория на локалната му файлова система.

Втората версия на ССА генератора, поддържа реализация на по-комплексни модели и сценарии от различни и интегрирани приложни области, като вече се наблюдава реализация на препредаване на съобщения между амбиенти без изрична намеса от моделиращия.

Визуализацията на ССА модела е функционалност, която е недостъпна за моделиращия в началото на реализацията на конкретен сценарий и е налична в момента, в който се генерира ССА моделът. Чрез бутон „*Open CCA File*“ се визуализира нов екран, на който потребителят има възможност да види същинският ССА модел, който е генериран с помощта на ССА редактора. Предоставя се и допълнителна функционалност за редакция, но тъй като синтаксисът на програмния език „*ssaPL*“ е строго специфичен, са необходими допълнителни знания, които моделиращият трябва да притежава преди да премине към фаза на редакция.

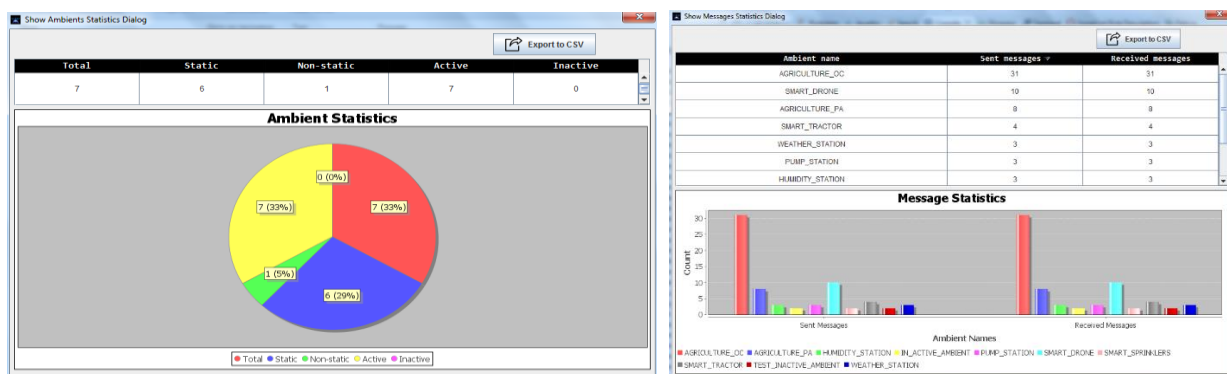
Веднъж след като са създадени всички необходими амбиенти, съобщения и е генериран ССА моделът, може да се премине към последната стъпка – стартиране на ССА сценария. Това се случва с помощта на вече разработения интерпретатор на програмния език „*ssaPL*“, а в текущата версия на ССА редактора сценарият може да бъде стартиран с помощта на командния ред от меню „*File*“/„*Open Terminal*“, или чрез клавишна комбинация „*CTRL + T*“ (Фигура 6).



Фигура 6. CCA файл и резултат от изпълнение на модела

За реализация на предварителен анализ на разработения модел е необходимо да се разработи специален **Модул за анализ**. Този модул е част от Компонента за моделиране и се интегрира със CCA редактора, получавайки информация от него и предоставяйки набор от статистики, които могат да бъдат използвани веднага след реализирането на сценария. Те помагат на моделиращия да анализира разработения CCA модел. На база резултатите от извършения анализ могат да се предприемат допълнителни действия и промяна на сценария с цел оптимизация и бързодействие, което би довело до по-добро функциониране на системите, които ще бъдат разработени на база този CCA модел.

Оптимизацията на CCA сценарии може да бъде извършена в няколко направления: **използвани амбиенти** като брой и тип; **текущ статус на амбиентите**; **неоптимална комуникация**; **неефективни маршрути** или **планове** и т.н. (Фигура 7)



Фигура 7. Статистика чрез Модула за анализ

Глава 4.Тестване на компонента за моделиране

Следващата важна стъпка е тестването и верификацията на системата, както и анализ на постигнатите резултати при работата с компонента за ССА моделиране. За целта, се разработиха редица тестови ССА модели и сценарии в различни приложни области, което би позволило откриване на максимално количество проблеми и своевременното им отстраняване. Фазата на анализ би донесла по-детайлна информация за поведението на системата, поради факта, че тя е подложена на тестване в разнообразни условия и режими на работа.

4.1. ССА моделиране в земеделието

Използването на водните ресурси в земеделието е проблем, които има спешна нужда от оптимизация и ефективност при употребата му. За целта, са разработени множество автоматизирани системи за напояване, които малко или много допринасят за редуцирането на несъобразеното използване на вода, но само това не е достатъчно. Разработката на **предварителни модели** за това как, колко често и по какъв начин ще се използват водните ресурси, дава възможност на хората да открият и идентифицират потенциални проблеми още в зародиш. Това спестява време и ресурси, както и допринася за ефективното използване и намаляване на количеството вода, което е необходимо за напояване.

За да се докаже, че подобни предварителни модели могат да бъдат разработени и използвани, се моделира сценарий свързан със земеделие, който включва и процеса по напояване, в термините на контекстно-ориентираното моделиране. Моделиран е примерен сценария, според който Фермер, използвайки своя персонален асистент, комуникира със земеделския оперативен център, за да идентифицира дали даден земеделски парцел се нуждае от напояване. Оперативният център взаимодейства с интелигентните обекти около себе си, сред които са метеорологична станция, дрон, интелигентен трактор, помпена станция и интелигентна напоителна система, с цел да получи информация, подsigури и предостави възможност за стартиране на напоителен процес по най-оптимален и ефективен начин

Реализацията, тестването и верифицирането на конкретен ССА сценария преминава през описаните в предходната част стъпки. Генерираният от ССА редактора код се стартира от стандартизирания ssaPL интерпретатор, чрез използването на имплементираните в редактора ssaParser и Manager. Резултатът от работата на интерпретатора е коректен и идентичен с резултатът получен при ръчно създаване на ssaPL кода за същия сценарий. Чрез Модула за анализ се определиха направленията за оптимизиране на този сценарий в термините на разглежданата приложна област (*земеделие*).

4.2. ССА моделиране в образованието

Виртуалното образователно пространство (VES) се разработва като адаптация на ViPS в областта на образованието [8]. Подобряването на средата и трансформирането ѝ като киберфизическо пространство разширява възможностите за адаптиране и персонализиране на съдържанието и услугите, особено за хората с увреждания. Разработката на VES като CPSS система е комплексна задача. Разработката на **предварителни модели** и сценарии за това как и по какъв начин да се предаде и поднесе учебният материал на обучаващите се, би дало възможност на преподавателите да открият и идентифицират потенциални проблеми още в зародиш. Това спестява време и ресурси, както и допринася за постигане на по-ефективен учебен процес.

За да се докаже, че подобни предварителни модели могат да бъдат разработени и използвани, се моделира сценарий свързан с отдалеченото и електронно обучение, който включва процеса по обучение на студенти. За целите на тестване, разгледахме следният примерен сценарий, който е реализиран благодарение на компонента за ССА моделиране. Студент, чрез своя персонален асистент комуникира с образователната среда на Пловдивски университет „Паисий Хилендарски“, както и с редица други единици (*амбиенти*), които са част от нея (*семинарна зала, библиотека и други*), за да получи необходимата информация, така че да бъде възможно ползотворното участие на студента в провеждащото се обучение. Като част от него, студентът получава редица материали като лекции, книги, учебници, статии и други ресурси, до които има достъп [11]**Error! Reference source not found.**

Реализацията, тестването и верифицирането на гореописания ССА сценарий преминава през описаните стъпки. Избира се домейна, създават се амбиентите, описва се евентуалната комуникация между тях и се генерира ССА модела. По време на този процес всички амбиенти и съобщения се достъпват, обработват и преобразуват, като крайният резултат се записва във файл, който се съхранява в директория предназначена за генерираните ССА модели. Генерираният файл може да се визуализира в ССА редактора. Важно е да се отбележи, че ако вече е генериран ССА файл, но впоследствие са създадени допълнителен набор от амбиенти и/или съобщения, то моделът трябва да се генерира отново и промените ще се отразят. След като вече е генериран ССА файлът и моделиращият го е прегледал и/или редактирал, то може да се премине към последната стъпка от реализацията на сценария, а именно стартиране, тестване и верификация на ССА модела, използвайки интерпретатора на езика за програмиране „*ccaPL*“.

Последната стъпка, която трябва да се предприеме, след като вече е наличен разработен ССА модел, е **анализ на постигнатия резултат**. Това е възможно благодарение на разнообразния набор от **статистики за амбиенти и съобщения**, които са реализирани с помощта на **Модула за анализ**.

Резултатите от получените статистически обработват определиха необходимостта от допълнителни промени с цел оптимизирането на сценария.

4.3. ССА моделиране в интегрирани приложни области

Дигитализацията, енергийната ефективност и „зелените“ решения водят до по-добра градска среда, условия за живот и стимулират растежа на икономиките на глобално и регионално ниво. Трябва да се вземе предвид, че тази трансформация и преминаване към интелигентни решения не е никак евтина от финансова гледна точка. Необходимо е да се обмисли много добре всяка стъпка при реализацията на интелигентна система. Разработката на **предварителни модели** за това как и по какъв начин ще функционира системата, дава възможност на хората да открият и идентифицират потенциални проблеми още в зародиш. Това спестява време и ресурси, както и допринася за ефективната реализация на интелигентните системи.

За да се докаже, че подобни предварителни модели могат да бъдат разработени и използвани, се моделира сценарий на тема „**интелигентен град**“ с използването на компонента за ССА моделиране.

Трябва да се отбележи, че реализация на подобен сценарий изисква използването и интеграцията на повече от една приложна област, затова ще разгледаме сценарий в термините на интегрираните приложни области. Нека разгледаме следният примерен сценарий, с цел да бъде възможна демонстрацията на процеса по моделиране, симулация, проверка и тестване на сценарий, който е разработен в термините на интегрираните приложни области – „*умен град*“ и „*умно образование*“.

Да разгледаме следния сценарий: Студент, чрез своя персонален асистент комуникира с образователната среда на Пловдивски университет „Паисий Хилендарски“, за да получи информация, дали днес трябва да посети лекции. След взаимодействие с оперативния център на публичния градски транспорт и на такситата, персоналният асистент на студента не успява да намери подходящ транспорт, което означава, че студентът няма възможност лесно и бързо да се придвижи и няма как да присъства на лекциите, които са предвидени за днес. За щастие на студента, университетът изпраща лекциите, а студентът ги получава благодарение на виртуалната образователна среда. Освен това, той получава допълнителна информация за точната дата и час на изпитите си и по какъв начин трябва да вземе участие в тях (*отдалечено или присъствено*) [6].

Реализацията, тестването и верифицирането на гореописания ССА сценарий изисква работа в няколко приложни области. Първоначално, потребителят, който работи със ССА редактора, трябва да избере приложните области (*domains*), в термините, на които ще бъде реализиран въпросният ССА модел/сценарий. Това е необходимо, с цел да бъдат извлечени (*от Data модула*) и визуализирани (*в секцията „Ambients Library“ в ССА редактора*)

всички амбиенти, които се асоциират с приложните области, които ще бъдат използвани. Като втора стъпка, потребителят трябва да създаде всички амбиенти от различните приложни области, които са му необходими за реализацията на ССА модела. Това действие се извършва, посредством избор на тип амбиент от библиотеката с амбиенти, а ако амбиент от желания тип не е наличен, то той може да бъде създаден и съхранен в базата данни. След това, моделиращият трябва да въведе информация за наименование на амбиент, тип, физическа локация и координати, дали е статичен и не/активен, както и ниво на йерархия в амбиентната структура. Като следваща стъпка от реализацията на сценария, трябва да се осъществи комуникация между вече създадените амбиенти. Това се случва посредством създаване на съобщения чрез разработените шаблони.

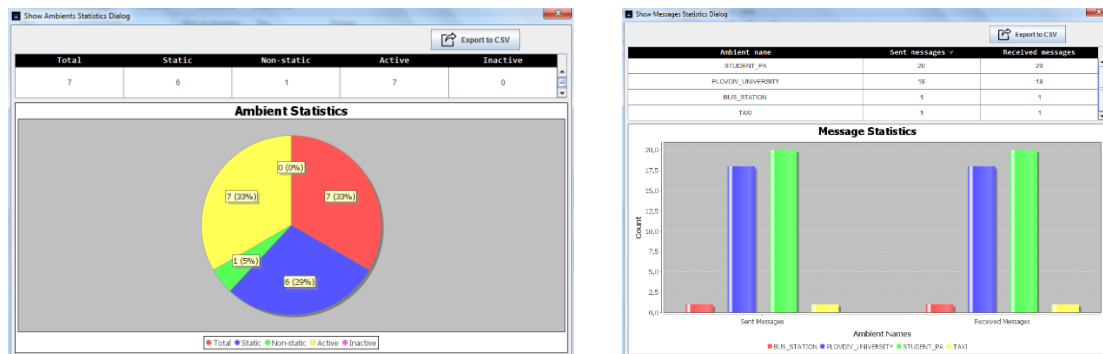
Веднъж след като са създадени всички необходими амбиенти и съобщения за разработвания сценарий, то би могло да се премине **генериране на ССА модел**, при което всички амбиенти и съобщения се достъпват, обработват и преобразуват. Крайният резултат се записва във файл с разширение „*.cca“. Изпълнението на генерирания ссаPL код се реализира чрез добавените функционалности на втората версия на прототипния ССА редактор.

Резултатите от тестването на работата на ССА редактора и при този тестов сценарий в интегрирани домейни също потвърди очакванията за коректно изпълнение (Фигура 8).

```
{Sibling to sibling: STUDENT_PA ===(Hello)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Hello,Hello_student)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(What_is_my_schedule_for_the_current_semester)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(What_is_my_schedule_for_the_current_semester,You_can_find_it_in_the_web_site)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Are_there_any_lectures_that_I_need_to_attend_today)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Are_there_any_lectures_that_I_need_to_attend_today,Yes_you_have)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Thank_you_for_the_information)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Thank_you_for_the_information,You_are_welcome)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Are_there_any_busses_available_soon)====> BUS_STATION}
{Sibling to sibling: BUS_STATION ===(Are_there_any_busses_available_soon,Unfortunately_no)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Are_there_any_free_taxi_cars_soon)====> TAXI}
{Sibling to sibling: TAXI ===(Are_there_any_free_taxi_cars_soon,No_the_taxi_cars_are_busy_now)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(I_am_not_able_to_attend_the_lectures_today)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(I_am_not_able_to_attend_the_lectures_today,Okay_it_is_not_an_issue)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Can_you_give_me_the_lectures_from_today)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Can_you_give_me_the_lectures_from_today,Sending_lectures)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Thank_you)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Thank_you,No_worries)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(When_is_my_exam)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(When_is_my_exam,At_the_end_of_the_semester)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Should_the_exam_be_online)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Should_the_exam_be_online,Yes_because_of_the_pandemic_the_exam_will_be_online)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Can_you_give_some_examples_for_the_exam)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Can_you_give_some_examples_for_the_exam,NO)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Okay_thank_you_for_the_whole_information_that_you_gave_me)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Okay_thank_you_for_the_whole_information_that_you_gave_me,No_worries_see_you_soon)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(See_you_on_the_exam)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(See_you_on_the_exam,BYE)====> STUDENT_PA}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Hello_student_you_have_an_exam_tomorrow)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Hello_student_you_have_an_exam_tomorrow,Thanks_for_the_information)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Sending_the_exam)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Sending_the_exam,I_am_done_here_are_my_answers_of_the_questions)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: STUDENT_PA ===(What_is_my_grade)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(What_is_my_grade,You_have_POOR_2)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(What_is_the_next_date_of_the_exam)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(What_is_the_next_date_of_the_exam,You_can_check_that_into_the_web_site_of_the_university)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Thanks_for_the_information)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Thanks_for_the_information,You_are_welcome)====> STUDENT_PA}
{Sibling to sibling: STUDENT_PA ===(Bye)====> PLOVDIV_UNIVERSITY}
{Sibling to sibling: PLOVDIV_UNIVERSITY ===(Bye,See_you_soon)====> STUDENT_PA}
```

Фигура 8. Резултат от ССА модел в интегрирани приложни области

В този тестов случай също се използва **Модула за анализ** за оптимизиране на създадените амбиенти, както и на взаимодействията (*комуникация и размяна на съобщения*) между тях (Фигура 9).



Фигура 9. Статистика за амбиенти в интегрирани приложни области

На база направеният анализ на ССА модела, може да се оптимизира сценарият и да се създаде библиотека от предефинирани планове за работа на интелигентните агенти. Впоследствие, даден план може да се използва наготово от библиотеката, вместо тепърва да се генерира и да се взима решение за бъдещи действия.

АВТОРСКА СПРАВКА ЗА РЕЗУЛТАТИТЕ В ДИСЕРТАЦИОННИЯ ТРУД

В дисертацията са изследвани подходи за създаване на концептуален модел и прототип на компонент за контекстно-зависимо моделиране на процеси, услуги и сценарии във виртуално-физическо пространство. Следвайки философията на референтната ViPS архитектура се предложи разширяване на аналитичното подпространство с допълнителни модули, които да осигурят процеса на предварително моделиране, тестване и анализ на базовите сценарии за разработваното CPSS-базирано пространство в контекста на един или няколко домейна. За целта първо се разглеждат характерните особености на кибер-физическите системи и нуждата от предварителен процес на моделиране. Обоснова се необходимостта от разработка на визуален ССА редактор, който да подпомага разработчиците и анализаторите в процеса на предварително моделиране.

След анализиране на структурата и взаимодействията на компонентите в референтната ViPS архитектура, се разработи концепция, модел и архитектура на компонент за контекстно-чувствително ССА-моделиране. Разработи се първа прототипна версия на ССА редактора, както и на някои от останалите компоненти на разработвания модул. След направени тествания се установи, че е необходимо добавяне на функционалности, позволяващи моделиране в няколко интегрирани домейна, което наложи разширяване функционалностите на компонентите в модула. Поради необходимостта от обработка и анализ на моделираните взаимодействия между амбиентите се разработи отделен Модул за анализ. Създадената втора прототипна версия на ССА редактора са интегрира в компонента за моделиране. Проведеха се апробирания на разработения компонент при моделирането на конкретни сценарии в разработваните адаптации на ViPS-архитектурата в областта на образованието и интелигентното земеделие, както и при работа в контекста на интегрирани домейни в областта на обучение и туризъм в интелигентен град. В следствие на извършените експерименти са направени заключение за реалната приложимост на компонента за ССА моделиране и се оформиха идеи за бъдеща работа.

Основни задачи, решени в дисертационното изследване

Задача 1: *Разработване на модел и архитектура на компонент за контекстно-чувствително ССА-моделиране във виртуално-физическо пространство.* След направения анализ и извеждането на необходимостта от разработка на отделен компонент за контекстно-чувствително ССА-моделиране се предложи концептуален модел за формализиране и обработка на динамично променящата се контекстна информация и отразяването ѝ в текущата амбиентна структура на моделирания сценарий. Предложи се

структура на отделен компонент за моделиране в рамките на референтната ViPS архитектура.

Задача 2: *Разработка на прототип на визуален ССА редактор за предварително моделиране на процеси и сценарии във виртуално-физическото пространство.* Въз основа на концептуалната рамка на компонента за моделиране се проучи възможността и се създаде прототипна версия на визуалния ССА редактор. Първата и основна задача при неговата разработка беше да се осигури механизъм за вграждане на стандартизирания сsaPL интерпретатор в общата архитектура на ССА редактора, което да верифицира и установи коректността на моделираните сценарии в термините и формалната семантика на ССА. Успоредно с това се разработиха първите версии на останалите елементи на компонента за моделиране – Data модула и Модула за анализ.

Задача 3: *Интегриране на прототипната реализация на ССА редактора в компонента за моделиране.* Следващата задача изискваше тестване на взаимодействието на ССА редактора с останалите компоненти на разработвания модул. Създадох се подобрени версии на Модула за анализ, както и на Data модула. Тази интеграция наложи разработката на втора прототипна версия на ССА редактора, която да позволява взаимодействие на амбиентите в повече от един домейн.

Задача 4: *Провеждане на експерименти с използването на Компонента за моделиране при моделирането на процеси в контекста на един домейн и в контекста на интегрирани домейни.* След разработката на текущата версия на компонента за ССА моделиране се проведеха експерименти за моделиране на базови сценарии в две кибер-физически и социални пространства, разработвани въз основа на базовата ViPS архитектура. Първият експеримент е свързан с моделиране на сценария „Напояване в система за интелигентно земеделие“, а вторият – с „Обучение на студенти във Виртуално-образователно пространство“. Тестването на работата на компонента в условията на интегрирани домейни е извършено в условията на „умен град“, в който се реализират услуги на потребители, свързани с обучение и транспорт.

Предимства на създадения компонент за ССА моделиране

Изграждането на кибер-физическо пространство е твърде комплексна, времеотнемаща и финансово-рискова задача. Представеният прототип на компонент за ССА моделиране дава възможност на разработчиците да създават и тестват проектираните сценарии в етапа преди началото на тяхната разработка във визуална и интуитивна среда. Разработено е ръководство за потребителя за използването на ССА Редактора. Предоставените допълнителни възможности за моделиране в интегрирани домейни и за анализ на взаимодействията между участващите идентичности (*амбиенти*), осигуряват възможности за предварително коригиране и оптимизиране на

моделираните сценарии и процеси. Вграденят в компонента за моделиране стандартизиран ссаPL интерпретатор гарантира коректността на моделите в термините на формалната ССА семантика.

Бъдещо развитие

На този етап на разработка на Компонента за ССА моделиране визуализацията на моделирания сценарий се реализира основно в конзолен режим. Планира се разработка на приложение с графична анимирана среда (*аниматор*), чрез което да се визуализират всички амбиенти, взаимовръзки и комуникация между тях по интерактивен начин в интерактивен режим.

Използването на контекстните изрази в ССА моделирането дава възможност за „улавянето“ на контекста и реализирането на определени възможности, свързани с поведението на участващите в моделираните процеси BDI агенти. Планира се реинженеринг на интеграцията между компонента за ССА моделиране и AmbiNet Хранилището с цел трансформиране на съхранения моделиран и оптимизиран сценарий в плановете на участващите BDI-агенти.

Необходима е по-тясна интеграция на компонента за ССА моделиране с останалите елементи на аналитичното подпространство на ViPS. Разработката на компонент за моделиране и управление на темпоралните аспекти на „нещата“ и изграждането на динамично взаимодействие с представения компонент би могъл да осигури по-пълнен и цялостен поглед върху моделираните сценарии, процеси и услуги.

Приноси

Научни:

1. Създаване на концептуален модел на компонент за контекстно ССА моделиране във виртуално-физическо пространство;

Научно-приложни:

2. Създаване на структура на компонент за ССА моделиране, реализиращ концептуалния модел;

3. Реализирана модификация на съществуващите компоненти, поддържащи работата на стандартизирания ссаPL интерпретатор и вграждането им в Компонента за моделиране.

4. Създаване на концептуална рамка на визуален ССА Редактор и свързаните с него модули за реализация на компонента за ССА моделиране;

Приложни:

5. Разработка на Компонент за моделиране и визуален ССА Редактор въз основа на създадения концептуален модел;

6. Приложение на Компонента за моделиране и визуалния ССА Редактор за моделиране на сценарии, както в отделни, така и в интегрирани домейни.

Връзката между приносите, целите, задачите, мястото на описание в дисертационния труд и свързаните публикации са дадени в следната таблица.

Принос	Задача	Глава, Секция	Публикации
1	1	II.2.2, II.2.3	2
2	1	II.2.4	1, 2
3	3	III.3.4	2
4	2	III.3.2, III.3.3	2
5	3	III.3.4	1,3
6	4	IV.1, IV.2, IV.3	1, 2, 3, 4

Апробации

Участие в международен научно-изследователски проект: *КП06-III-КИТАЙ/1*, „Разработване на многосервизна облачнобазирана IoT оперативна платформа с общо предназначение (EMULSION)“, Фонд научни изследвания при МОН, 2019/2022.

Резултатите от работата по дисертационното изследване са докладвани на следните научни национални и международни конференции:

1. *Пространствените аспекти при моделирането на интелигентен град Ловеч*, Национална научна конференция TechCo'21, 2-4.07.21, Ловеч, България.
2. *Ambient-oriented modeling in virtual-physical space with integrated domains*, Anniversary International Scientific Conference REMIA'2021, October 22-24, 2021, Plovdiv, Bulgaria
3. *Ambient-oriented cca modeling in agriculture*. International Conference ICAI'21, 30 Sept.-2 Oct. 2021, Bulgaria, Varna
4. *Персонален туристически екскурзовод в интегрирани домейни*. V-та национална научна конференция с международно участие TechCo-Lovech 2021, Ловеч, България
5. *Virtualization, processing, and standardization of knowledge about Bulgarian cultural, historical and natural heritage for use in school*

education. IEEE International conference DIPP'22, Burgas, 24-26 September 2022.

ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. Тодорка Глушкова, **Константин Русев**, Йордан Тодоров, Станимир Стоянов. ПРОСТРАНСТВЕНИТЕ АСПЕКТИ ПРИ МОДЕЛИРАНЕТО НА ИНТЕЛИГЕНТЕН ГРАД ЛОВЕЧ. *V-та НАЦИОНАЛНА НАУЧНА КОНФЕРЕНЦИЯ С МЕЖДУНАРОДНО УЧАСТИЕ TechCo-Lovech 2021*, 117-122
2. Glushkova, T., Stoyanov, St., **Rusev, K.**, Krasteva, I., Moraliyska, N. (2021). Ambient-oriented CCA Modeling in Agriculture. *2021 International Conference Automatics and Informatics (ICAI'21)*, 30 Sept.-2 Oct. 2021, Bulgaria, Varna, 310-313, ISBN: 978-1-6654-2662-6, doi: 10.1109/ICAI52893.2021.9639591, **SCOPUS**.
3. Glushkova, T., **Rusev, K.**, Stoyanov, St. (2021). AMBIENT-ORIENTED MODELING IN VIRTUAL-PHYSICAL SPACE WITH INTEGRATED DOMAINS. *Anniversary International Scientific Conference RESEARCH AND EDUCATION IN MATHEMATICS, INFORMATICS AND THEIR APPLICATIONS – REMIA'2021*, October 22-24, 2021, Bulgaria, Plovdiv, 99-108
4. Grancharova-Hristova, M., Moraliyska, N., **Rusev, K.**, Ivanova, V., Tabakova-Komsalova, V. (2021). Application of ontologies and digital libraries in school education. *Informatics and Education*, 10 (329), 15-20. doi:10.32517/0234-0453-2021-36-10-15-20

Библиография

- [1] Bugliesi, M., Castagna G. and Crafa, S., Access Control for Mobile Agents: The Calculus of Boxed Ambients, *ACM Trans. on Programming Languages and Systems*, 2004, 26(1), 57–124.
- [2] Cardelli, L. and Gordon, A., *Mobile Ambients*, Theoretical Computer Science, Elsevier Science (2000), 240177–213.
- [3] Cau, A., Moszkowski, B., & Zedan, H. (2006). Interval temporal logic. URL: <http://www.cms.dmu.ac.uk/~cau/itlhomepage/itlhomepage.html>.
- [4] Dey, A. K., "Understanding and Using Context", *Personal and Ubiquitous Computing Journal*, vol. 5, no. 1, pp. 4-7, 2001.
- [5] Glushkova, T., M. Miteva, A. Stoyanova-Doycheva, V. Ivanova, S. Stoyanov, Implementation of a Personal Internet of Thing Tourist Guide, *American Journal of Computation, Communication and Control*, issue 1/2018, Pages: 39-51, ISSN: 2375-3943, <http://www.aascit.org/journal/ajccc>
- [6] Glushkova, T., Rusev, K., Stoyanov, St. (2021). Ambient-Oriented Modeling in Virtual-Physical Space with Integrated Domains. *Anniversary International Scientific Conference "Research and Education in Mathematics, Informatics and their Applications" – REMIA '2021*, October 22-24, 2021, Bulgaria, Plovdiv, 99-108.
- [7] Glushkova, T., Stoyanov, S., Ivanova, V., Krasteva, I., AN IDEA FOR CONDUCTING THE EDUCATIONAL PROCESS IN A VIRTUAL-PHYSICAL SPACE, *AIP Conference Proceedings* 2333, 050006 (2021); <https://doi.org/10.1063/5.0041727> AIP
- [8] Glushkova, T., Stoyanov, S., Stoyanova-Doycheva, A., Internet of Things ecosystem supporting e-Learning, Monograph: E-learning and Smart Learning Environment for the Preparation of New Generation Specialists, E. Smyrnowa-Trybulska (ed.). Vol. 10 (2018) Katowice-Cieszyn: Studio Noa for University of Silesia ISSN: 2451-3644 (print edition) ISSN 2451-3652 (digital edition) ISBN: 978-83-66055-05-6, p. 409-427, 2018.
- [9] Glushkova, T., Stoyanov, St., Stoyanova-Doycheva, A., Ivanova, V., Doukovska, L. AmbiNet – An Environment for Ambient-Oriented Modeling. *International Journal of Computing*, 30 Sept. 2019, 18(3), 331-340, ISSN: 1727-6209.
- [10] Glushkova, T., Stoyanova-Doycheva, A., Ivanova, V., Stoyanov, S., Doychev, E., Cyber-Physical- Social Systems and Applications, Part 2: Applications, LAP LAMBERT Academic Publishing, pages 164, 978-620-0-49831-1, 2019.
- [11] Glushkova, T., Stoyanov, S., Popchev, I., Cherescharov, S., Ambient-oriented modelling in a virtual educational space, *Comptes Rendus de L'Academie Bulgare des Sciences*, Vol 71, No3, pp.398-406, 2018, DOI: 10.7546/CRABS.2018.03.13, ISSN: 2367-5535, <http://www.proceedings.bas.bg/>
- [12] HORIZON 2020 Work Programme 2018-2020 "Leadership in enabling and industrial technologies/Information and Communication Technologies" (draft), 2 February 2017.
- [13] Jamwal, A., Agrawal, R., Sharma, M., Giallanza, A. (2021). Industry 4.0 technologies for manufacturing sustainability: a systematic review and future research directions. *Applied Sciences*, 11(12), 572.
- [14] Milner, R., Parrow J. and D. Walker, A calculus of mobile processes, Parts1–2, *Inform. and Comput.* (1992)100(1) 1–77.

- [15] National Science Foundation (US), <https://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm>, посетен на 31.07.2022.
- [16] Shinde, A. V. (2021). Analysis of Context Aware Computing Systems in Internet of Things. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(12), 2908-2914.
- [17] Siewe, F., Cau, A. and Zedan, H., CCA: a Calculus of Context-Aware Ambients, Conference: Advanced Information Networking and Applications Workshops, 2009, WAINA'09. International Conference, DOI:10.1109/WAINA.2009.23
- [18] Stoyanov S., Glushkova T., Popchev I., Doukovska L. (2021) Virtualization of Things in a Smart Agriculture Space. In: Sgurev V., Jotsov V., Kacprzyk J. (eds) Advances in Intelligent Systems Research and Innovation. Studies in Systems, Decision and Control, vol 379. Springer, Cham. https://doi.org/10.1007/978-3-030-78124-8_16, <https://www.scimagojr.com/journalsearch.php?q=21100828949&tip=sid&clean=0>, ISBN 978-3-030-78123-1.
- [19] Stoyanov, S., A. Stoyanova-Doycheva, T. Glushkova, E. Doychev, Virtual Physical Space – An Architecture Supporting Internet of Things Applications, 20th International Symposium on Electrical Apparatus and Technologies (SIELA), 3-6 June 2018, DOI: 10.1109/SIELA.2018.8447156, <https://ieeexplore.ieee.org/abstract/document/8447156>
- [20] Stoyanov, S., Glushkova, T., Doychev (2019), E., Cyber-Physical-Social Systems and Applications - Part1, LAP LAMBERT Academic Publishing, 2019, ISBN: 978-620-0-31825-1.
- [21] Stoyanov, S., Stoyanova-Doycheva, A., Ivanova, V., Tabakova-Komsalova, V., Monov, V., & Radeva, Z. (2021). An Event Model for Smart Agriculture. In 2021 International Conference Automatics and Informatics (ICAI) (pp. 314-317). IEEE.
- [22] Todorov, J., Krasteva, I., Ivanova V. and E. Doychev, "BLISS-A CPSS-like Application for Lifelong Learning" 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), 2019, pp. 1-5, doi: 10.1109/INISTA.2019.8778363.
- [23] Valkanov, V., Stojanova-Doycheva, A., Doychev, E., Stojanov, S., Popchev, I., & Radeva, I. 2014. AjTempura - First Software Prototype of C3A Model. IEEE Conf. on Intelligent Systems.
- [24] Wang, F., "The emergence of intelligent enterprises: From CPS to CPSS," IEEE Intell. Syst., vol. 25, no. 4, pp. 85–88, Jul./Aug. 2010.
- [25] Zhang, J. J., Wang, F. Y., Wang, X., Xiong, G., Zhu, F., Lv, Y., Lee, Y. (2018). Cyber-physical-social systems: The state of the art and perspectives. IEEE Transactions on Computational Social Systems, 5(3), 829-840.