

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

ИЗГРАЖДАНЕ НА ПЕРСОНАЛНИ АСИСТЕНТИ ВЪВ ВИРТУАЛНО ОБРАЗОВАТЕЛНО 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: Йордан Георгиев Тодоров
Научни ръководители: проф. д-р Станимир Недялков Стоянов
доц. д-р Олга Добрева Рахнева

Пловдив, 2020 г.

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 24.09.2020 г.

Дисертационният труд съдържа 127 страници. Библиографията включва 112 източника. Броят на авторските публикации е 4.

Защитата на дисертационния труд ще се състои на
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: Йордан Георгиев Тодоров

Заглавие: Изграждане на персонални асистенти във виртуално образователно пространство

Пловдив, 2020 г.

Съдържание

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД	4
Цели и задачи на дисертационния труд	5
Методика на изследването.....	6
Структура на дисертационния труд.....	6
КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД.....	8
Глава 1. Състояние на изследвания проблем.....	8
Глава 2. Виртуално-физическо пространство.....	8
Глава 3. LISSA – Learning Intelligent Student Assistant	11
Идея	12
Архитектура.....	12
Прототип	14
Глава 4. Референтен персонален асистент	16
Обща характеристика на РПА	17
Модел на РПА	18
Обща характеристика на архитектурата.....	24
Глава 5. Среда BLISS	25
Клиентска част на BLISS	26
Прототип на PAStudent.....	27
АВТОРСКА СПРАВКА ЗА РЕЗУЛТАТИТЕ В ДИСЕРТАЦИОННИЯ ТРУД ...	28
ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД.....	30
БИБЛИОГРАФИЯ.....	31

ОБЩА ХАРАКТЕРИСТИКА НА ДИСЕРТАЦИОННИЯ ТРУД

През последните години научният интерес към създаване и практическата използваемост на персонални асистенти за различни групи потребители нараства значително. Освен най-известните системи с отворен код (като Moodle [10]) и корпоративните системи (като Microsoft Classroom [9]), много университети развиват свои собствени системи за електронно обучение. В съответствие с тази тенденция, във Факултета по математика и информатика към Пловдивски университет „Паисий Хилендарски“ беше реализиран проект „Разпределен център за електронно обучение“ (DeLC) с цел изграждане на инфраструктура за предоставяне на електронни образователни услуги и учебно съдържание [17]. Архитектурата на DeLC е проектирана като мрежа, която се състои от отделни възли, наречени eLearning Nodes. Възлите моделират реални единици (лаборатории, отдели, факултети, колежи и университети), които предлагат пълен или частичен образователен цикъл. Всеки възел на eLearning е автономен хост на набор от електронни услуги [15]. Възможностите за електронно обучение могат да бъдат изолирани или интегрирани в по-сложни виртуални структури, наречени клъстери. Дистанционното активиране и интеграция на eService е възможно само в рамките на един клъстер. В мрежовия модел можем лесно да създаваме нови клъстери, да реорганизираме или премахваме съществуващи клъстери, защото реорганизацията се извършва на виртуално ниво и не засяга реалната организация. DeLC осигурява мобилен достъп до услуги и съдържание чрез разширена локална мрежа, наречена InfoStations [16].

Един сериозен недостатък е, че DeLC, реализиран като виртуална среда, не отчита физическия свят, в който се осъществява учебния процес. Изграждането на инфраструктура, в която виртуалният свят е интегриран по естествен начин в заобикалящата го физическа среда, би отворил нови възможности за предоставяне на образователни услуги и учебно съдържание в персонализиран и контекстуално осъзнат начин. За постигането на тази интеграция се започна трансформация на DeLC в нова, кибер-физическа инфраструктура, известна като Virtual Education Space (VES). VES е разработена като „Интернет на нещата“ екосистема, състояща се от

автономни интелигентни компоненти, които показват контекстно-ориентирано поведение [13]. Освен това пространството се подсилва от подходи, използващи семантични модели, главно под формата на онтологии.

При работа със сложни системи като VES е препоръчително потребителите да бъдат поддържани от специализирани интерфейсни компоненти по интуитивен и интелигентен начин. Едно възможно решение е използването на персонални асистенти. В архитектурата на пространството персоналните асистенти играят важна роля.

Като следваща стъпка в еволюцията на системата е прерастването ѝ във Virtual Physical Space (ViPS) като референтна инфраструктура за разработване на CPSS подобни приложения. Така базов компонент е референтен персонален асистент, който може да се адаптира за различни приложни области.

Цели и задачи на дисертационния труд

Основната цел на дисертацията е да се разработят персонални асистенти за работа във виртуално образователно пространство. За постигането на тази цел са определени следните задачи за решаване:

Задача 1: Разработка на персонален асистент LISSA за подпомагане на студентите в Пловдивски университет „Паисий Хилендарски“.

Задача 2: Разработване на концепция, модел и архитектура на референтен персонален асистент (РПА).

Задача 3: Реализиране на прототип на РПА.

Задача 4: Интегриране на прототипната реализация на РПА в реална система за електронно обучение.

Задача 5: Провеждане на експерименти с прототипите.

Методика на изследването

За постигане на целта на дисертацията и в съответствие със задачите е предложена методика за провеждане на изследването, която включва следните стъпки:

- Разработване на първа версия на прототип на персонален асистент (ПА) с ограничени възможности за проверка на реализируемостта и логическата непротиворечивост на предложената концепция и архитектура – ПА ще подпомага студенти в техните учебни задължения. Прототипът се интегрира в DeLC 2.0 и се провеждат различни експерименти с него.
- Разработване на концепция, модел и архитектура на референтен ПА (РПА) – на основата на анализа и оценката на резултатите от работата на първата версия на прототипа се предлага референтен ПА. Реализира се прототип на РПА.
- Изграждане на среда за адаптиране на РПА за конкретна приложна област – прототипът на РПА се интегрира в реално приложение.
- Провеждане на експерименти с адаптираната версия на РПА.
- РПА се разширява с възможности за опериране в различни среди.

Структура на дисертационния труд

Увод – В уводната част на дисертационния труд 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 – разглеждат се различни известни разработки на персонални асистенти, които съществуват към момента. Обръща се специално внимание, както на персонални асистенти за общо ползване за ежедневни цели, така и на специализирани агенти, които да подпомагат хора в специфична сфера или хора със специфични нужди.

Глава 2. Virtual Physical Space – подробно разглежда ViPS пространството и всички подсистеми, които му принадлежат.

Глава 3. LISSA – Learning Intelligent Student Assistant – разглежда реализирания персонален асистент, неговите възможности и функционалности, както и различните итерация, през които е преминало, по време на разработката. Представени са също първите две версии на реализиран прототип на агента.

Глава 4. Референтен персонален асистент – Въз основа на придобития опит при разработка на системата LISSA е представен идеен теоритичен модел на референтен персонален асистент, който е подробно описан посредством примери и псевдокод.

Глава 5. Среда BLISS – показва разработката и реализацията на система BLISS (Brezovo's Learning Intelligent School System), която имплементира представения теоритичен модел на персонален асистент. Разгледани са получените резултати и извършените корекции по време на експлоатация на системата в реална работна среда.

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

КРАТКО СЪДЪРЖАНИЕ НА ДИСЕРТАЦИОННИЯ ТРУД

Глава 1. Състояние на изследвания проблем

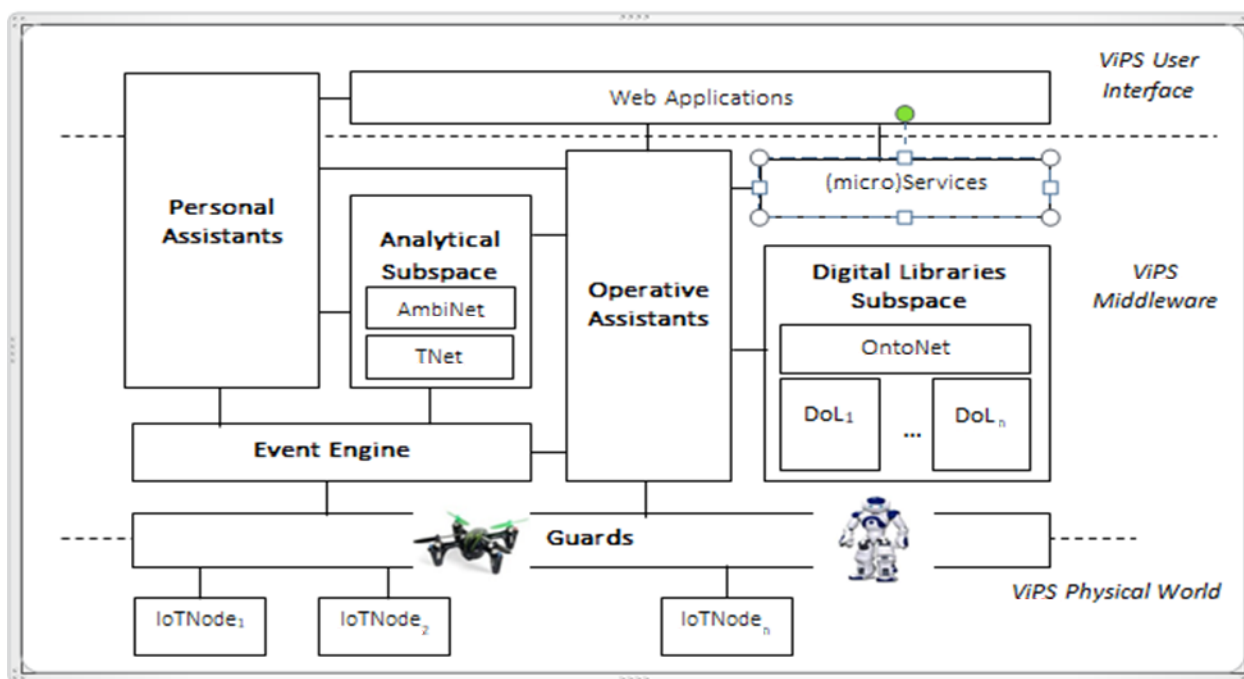
Въведена през 90-те години, концепцията за използване на персонални асистенти за подпомагане на хората в ежедневните им бизнес и лични дела се развива от много години. Както е показано в [11], технологии от областта на изкуствения интелект (ИИ) представят възможности за изграждане на интелигентни машини, които самостоятелно изпълняват задачи от името на потребителя. В [19] откриваме, че приложенията на агентите се разделят на две основни групи – разпределени системи и лични софтуерни асистенти, където агентите играят ролята на проактивни помощници на потребителите при работа с някои приложения. Персонални асистенти могат да помогнат, както за ежедневното, така и за дългосрочното управление, изпълнението и контрола на различни видове задачи. Днес тези асистенти обикновено се ползват на мобилни устройства и могат да използват ресурсите на социалните мрежи и да се самообучават посредством комуникацията с потребителя. Две полезни статии, първата за използването на ПА в контекста на интернет на нещата [6] и втората – на Интелигентните педагогически агенти (Intelligent Pedagogical Agents) (IPA) за персонализирано учене и повишаване на мотивацията на студентите [7], показват развиващата се област на персоналните асистенти.

Глава 2. Виртуално-физическо пространство

Virtual Physical Space (ViPS) се изгражда като референтна архитектура, интегрираща виртуалния и физическия светове [14], която може да бъде адаптирана за различни CPSS-like applications. В този смисъл съществени аспекти на ViPS са следните:

- Потребителите са във фокуса на вниманието.
- Виртуализация на физическите „неща“.
- Интеграция на виртуалния и физическия светове.

В съответствие със CPSS [5], ViPS архитектурата (Фигура 1) осигурява виртуализация на реални обекти, които могат да бъдат адаптирани за конкретен домейн. В контекста на софтуерните технологии това означава създаване на дигитализирани версии на физически реални обекти, които могат да бъдат определени и цифрово интерпретирани. В този аспект архитектурата отразява и представя в дигиталния свят по същество идентичен модел на реалния физически свят, в който процеси, потребители и познания за интересувашата ни област, както и взаимодействието между тях, се реализират в динамичен, персонализиран и контекстно-съобразен начин.



Фигура 1. Обща архитектура на ViPS

На практика, виртуализацията на „нещата“ се поддържа от мидълуеъра на ViPS. Моделирането на „нещата“ взема под внимание фактори като събития, време, пространство и местоположение. Виртуализацията на „нещата“ основно се подпомага от аналитичното подпространство. Освен това то осигурява средства за подготовка на специфични за дадена област анализи. ENet моделира различни видове събития и техните аргументи (идентификация, условия за възникване и завършване), представителни за интересувашата ни проблемна област. TNet предоставя възможност за представяне и работа с времевите аспекти на

„нещата“. В AmbiNet могат да бъдат моделирани като „амбиенти“ пространствените характеристики на „нещата“ [18]. Работата на тези компоненти се подкрепя от специализирани оперативни асистенти. За да могат компонентите на аналитичното подпространство да правят изводи и заключения, както и да интерпретират моделите, те използват фонові знания и документи за конкретната приложна област, съхранявани в подпространството на дигиталните библиотеки. Библиотеките са реализирани като отворени цифрови хранилища. Компонентът OntoNet е йерархия от онтологии, която представлява съществените характеристики и отношения на интересуващите ни „неща“.

Поради естеството (потребителите са в центъра на вниманието), а също така и поради очакваната комплексност и сложност на една CPSS екосистема, за потребителите се предлагат специални интуитивни, разбираеми и лесни за използване средства, обикновено наричани персонални асистенти или помощници. Достъпът до ViPS е контролиран и персонализиран. Персоналните асистенти, работещи от името на потребителите и запознати с техните нужди, ще подготвят сценарии за изпълнение на заявките на потребителите и ще управляват и контролират тяхното изпълнение чрез взаимодействие с мидълуеъра на ViPS. Освен това, персоналните асистенти трябва да могат да работят и в режим на „изпреварващо действие“ (превенция), включително да поддържат система за „ранно и навременно предупреждение“.

Оперативните асистенти, реализирани като рационални интелигентни агенти, осигуряват достъп до ресурсите на двете подпространства. Освен това, те осъществяват взаимодействие с персоналните асистенти и уеб приложения. Те са архитектурни компоненти, подходящи за осигуряване на необходимата динамика, гъвкавост и интелигентност на ViPS. Въпреки това, те не са подходящи за предоставяне на необходимата бизнес функционалност в пространството. Поради тази причина асистентите работят в тясно сътрудничество с услугите и/или микро-услугите. Освен това те взаимодействат с гардовете за доставка на данни от физическия свят.

Гардовете функционират като интелигентен интерфейс между виртуалния и физическия свят. Те предоставят данни за състоянието на физическия свят, прехвърлен във виртуалната среда на пространството (двете подпространства). В архитектурата на гардовете са интегрирани множество IoT възли, които осигуряват достъп до сензори и изпълнители на „нещата“, намиращи се във физическия свят. Наборите на сензорите – изпълнителните механизми са конфигурирани в съответствие с приложението. Комуникацията в охранителната система работи като комбинация от лична мрежа (например LoRa) и Интернет.

Публичните информационни ресурси на пространството са достъпни чрез подходящи уеб приложения, които обикновено се прилагат специално за конкретния домейн [1].

ViPS се разработва като референтна архитектура, която може да бъде адаптирана за различни CPSS-like applications. Следваме един общ подход за адаптиране на ViPS, включващ следните две стъпки:

- Адаптираме обикновено не цялата архитектура, а отделни нейни компоненти за конкретната приложна област.
- Създадените компоненти за конкретното приложение се архивират и съхраняват в отделни приложни библиотеки, които стават част от ViPS. Така ViPS постепенно, с всяко ново приложение се разширява с нови компоненти.

Глава 3. LISSA – Learning Intelligent Student Assistant

За постигане на първата задача от дисертацията е реализиран персонален асистент за подпомагане на учебния процес на студентите във Факултета по математика и информатика. Основната цел на персоналния асистент в началото на неговата разработка целеше осигуряване на достъп на студентите до виртуалното образователно пространство. По време на работата върху реализацията на асистента, се разработиха различни версии и постепенно функционалностите и задачите, които изпълнява LISSA се промениха. Една от последните итерации на системата позволи добавяне на сензорни мрежи, което позволи LISSA да се разглежда като Интернет на

Нещата персонален асистент. В тази глава ще разгледаме обобщено архитектурата, жизнения цикъл, както и прототипа на системата, който е създаден.

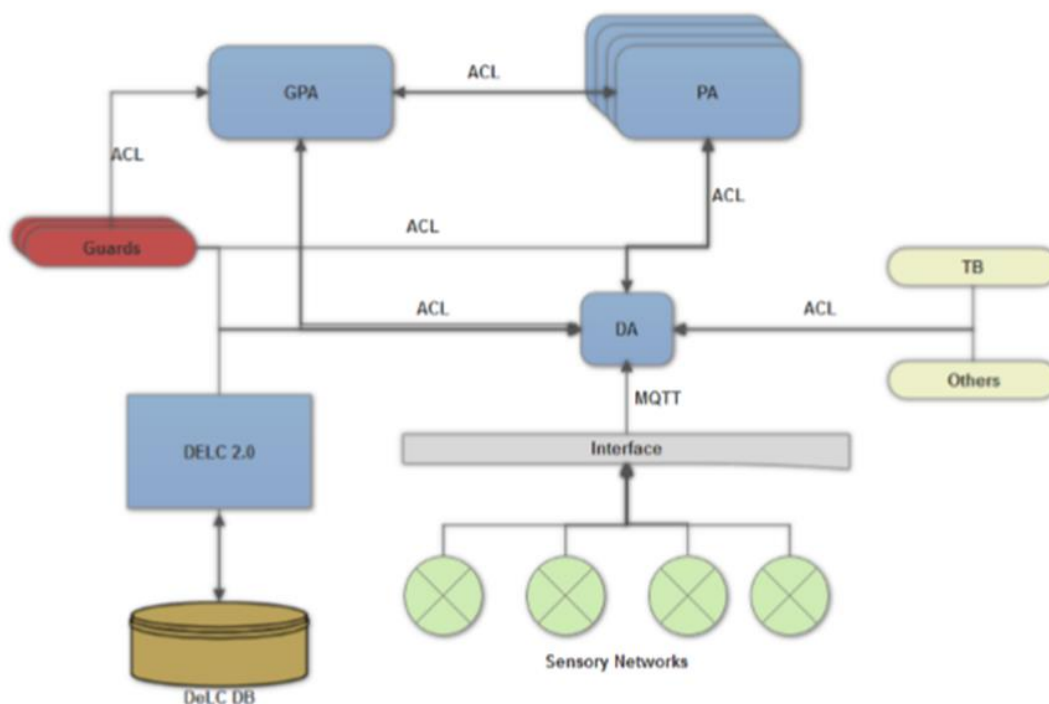
Идея

Първоначалната идея за реализацията на LISSA, беше да се реализира агент, който да напомня на студентите за приближаващи дати за изпити. В процес на работа започнаха да се появяват възможности за добавяне на нови функционалности и възможности към първоначалната идея и LISSA прерасна от самостоятелен агент в мулти-агентна система, позволяващи различни функционалности.

В последната си итерация системата се модифицира отново, като започна да взима по централно място в образователното пространство. За да се позволи по-голяма гъвкавост на системата, тя се предефинира като Интернет на Нещата персонален асистент, който позволява голям обем от функционалности, които могат да бъдат използвани от студентите. За да може допълнително да бъде разширявана, в системата се включи възможност за лесно добавяне на нови сензорни мрежи, които тя да използва за своите услуги. Това дава възможност LISSA да предлага различни услуги спрямо мрежите, с които разполага мястото, на което се използва.

Архитектура

Архитектурата на LISSA е разработена в съответствие с предложения жизнен цикъл (Фигура 2), който включва множество инстанции на Персонален Асистент (PA), Generic Personal Assistant (GPA) и Generic Dispatcher Agent (GDA). Системата може да комуникира с guard агентите, от които получава важна информация, която се използва при взимането на решение и определянето на цели.



Фигура 2. Архитектура на LISSA

Разполага и с комуникация към DELC 2.0, който от своя страна може да предостави информация относно студентите, намиращи се в прилежащата му база от данни. Всеки един от външните агенти за системата може да изпраща съобщения към нея, като те минават през GDA агента, който определя важността на съобщението и го препраща на агента, за който се отнася. Системата е устроена така, че да може да се закачват набор от сензорни мрежи (тоест функционалностите на системата могат да бъдат лесно разширявани посредством добавянето на нови сензорни мрежи, представящи нов вид информация). Информацията от сензорите се изпраща през интерфейс за комуникация до GDA агента.

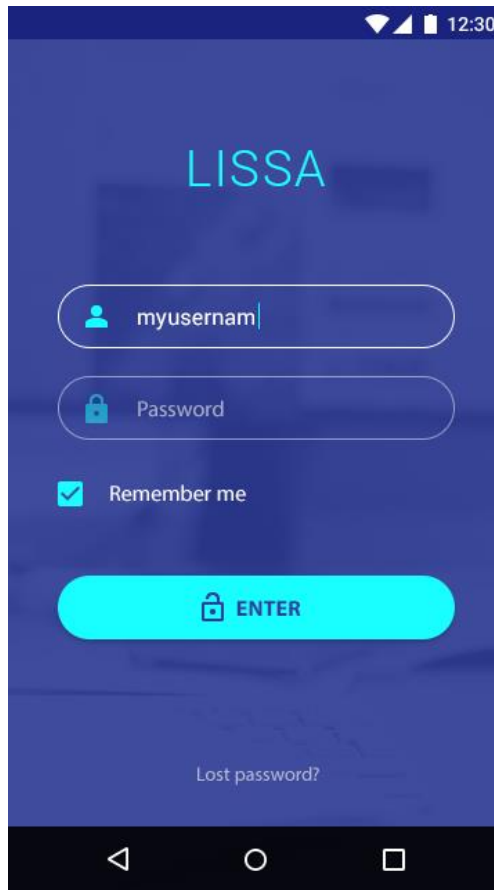
Прототип

За тестване на част от системата се създаде прототип на персоналния асистент на LISSA. Избрана е Android операционна система за работа на прототипа по две причини:

- Android е най-широко използваната операционна система за мобилни устройства и може да се използва на голяма група от различни устройства, като например: мобилни телефони, таблети, смарт часовници, телевизори и др.
- Втората важна причина е, че използва езика за програмиране на JAVA, което ще позволи да се запази хомогенността на системата.

Прототипът е изграден от рационални BDI агенти. За реализацията се използва рамката JADDEX [4, 12] като разширение на JADE [2, 3], която позволява описването на менталните състояния. Прототипът силно взаимодейства с портала DeLC 2.0, като специална входна точка в пространството. DeLC 2.0 се реализира като динамично уеб приложение, разпределено в две основни области – образователен портал, работещ като специализиран потребителски интерфейс и сървър. И двете области комуникират чрез чисти HTTP заявки, RESTful услуги и Web Sockets.

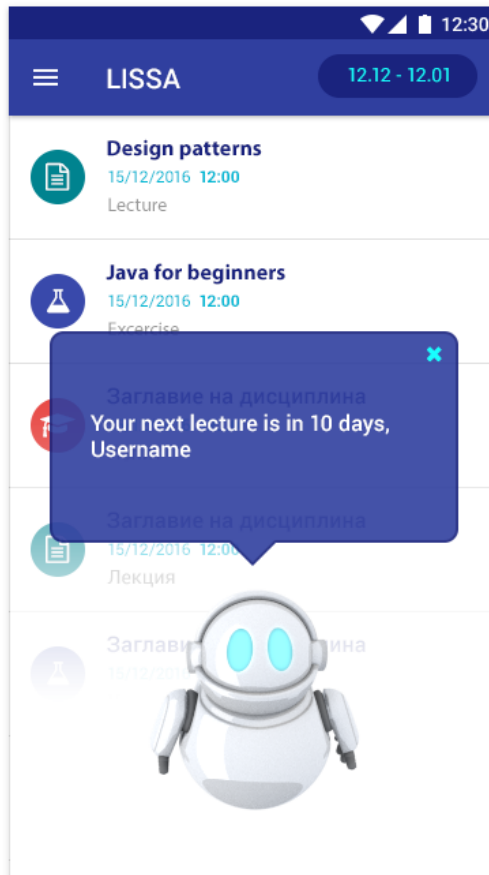
На Фигура 3 се вижда началния екран на прототипа, който позволява вход или регистрация в системата. Ако потребителят е в режим на регистрация, GPA, във взаимодействие с DeLC 2.0, доставя данни за създаване на профил и желанието на личния асистент.



Фигура 3. Входен екран на прототипа LISSA

В нормален работен режим на екрана се визуализира аватара на LISSA, а над него е списък с чакащи събития (Фигура 4). Най-новите събития са най-отгоре. За всяко събитие има име, дата и час на събитието, а под него е видът на събитието. В горния десен ъгъл се вижда диапазон, който се използва за преглед на събития и зависи от видовете събития в календара. За обхват се използва събитието с най-ранно предизвестие. Когато е необходимо напомняне, помощникът едновременно произнася и показва съобщението на екрана.

Нормалният работен процес на прототипа преминава през BDI архитектурата, както е описано в предишната глава, като се използва датата и времето за фактор в околната среда, а местоположението на потребителя се намира с помощта на GPS сензора.



Фигура 4. Главен екран на LISSA

Глава 4. Референтен персонален асистент

На основата на анализа на резултатите от работата с LISSA, в тази глава ще бъде предложен референтен персонален асистент (РПА). ViPS е изградено на принципите на CPSS, където подпомагането на потребителите е първостепенна задача – по тази причина РПА става централен компонент в архитектурата на пространството.

Обща характеристика на РПА

Най-общо един РПА, опериращ в конкретна адаптация на ViPS, трябва да изпълнява две основни задачи, които ще наречем:

- **Подпомагане** – оперативно подпомагане на потребителя в неговата ежедневна дейност.
- **Превенция** – включва различни изпреварващи действия, за да се осигурят необходими, сигурни или оптимални условия за потребителя да участва (да се възползва от) в предстоящо събитие, състояние или ситуация. В този смисъл РПА трябва да се разглежда също като система за ранно предупреждение.

При изпълнение и на двете задачи един РПА трябва да е в състояние да отчита пространствените и времевите аспекти на събитията, състоянията и ситуацияите, в които може да попадне подпомаганият потребител. РПА трябва да оперират изключително във виртуалното пространство, отчитайки физическия свят, в който пребивава потребителят. РПА трябва да следва философията на интегриране на виртуалния и физическия светове на ViPS. Също, при изпълнение на задачите си конкретните инстанции трябва да взаимодействат тясно с останалите компоненти на ViPS, особено с аналитичното подпространство.

РПА ще бъде реализиран като рационален агент. Най-общо, един рационален агент се опитва да намери добри решения за това какво трябва да прави. По-конкретно, един рационален агент решава да извършва действия, които са в негов най-добър интерес, отчитайки представата (вярата), която той има за света. Тези агенти обикновено са разположени в една непрекъснато променяща се физическа или виртуална среда с изключително богатство, сложност и разнообразие. В такава среда плановете на рационалните агенти са постоянно подложени на осуетяване поради различни причини – природни явления и феномени, умишлена или неумишлена, злонамерена или добронамерена намеса. И въпреки това, вземайки решения, агентите се опитват да се справят с капризите на заобикалящата ги среда. Освен това, те са в състояние да взаимодействат със себеподобните си (дори, когато са в конфликт с тях), да се справят с несигурностите в света, с неуспешните

действия и планове, както и със сложността на средата. С други думи, искаме да създадем рационални агенти, които са въплътени в някаква среда и които обитават и действат върху тази среда по начин, аналогичен на този на хората. В идеалния случай бихме искали да разработваме агенти, които са добри при вземането на решения и да действат спрямо тях. За съжаление, подобно на много други проблеми в изкуствения интелект, е трудно да се намерят ефективни и добри решения за реални среди, които са достатъчно сложни и бързо променящи се. Въпреки това си заслужават усилията за разработване на рационални агенти.

Модел на РПА

Моделът е основа за разработването на архитектурата на РПА. Теоретична негова основа е BDI моделът, комбиниран с концепцията за събития (събитийен модел) на ViPS, представена в предишната глава на дисертацията. Основна характеристика на РПА модела е, че менталните състояния (вяра, желания, цели) имат унифицирано представяне във формата на събития в съответствие със събитийния модел на ViPS. За да внесем по-голяма яснота в изложението, моделът на РПА ще бъде представен на отделни стъпки.

Първа стъпка (разширение концепцията за събития на ViPS). Оперирането на РПА не може да се моделира директно чрез концепцията за събития на ViPS. Необходимо е разширение на тази концепция, което ще бъде представено тук. Ще въведем следните понятия:

- Множество на събитията: $E = \{e_i \mid e_i \text{ събития, специфицирани за конкретна приложна област, } i = 1, \dots, n\}$. Множеството E се декомпозира на три еквивалентни класа както следва: $E = E_1 \cup E_2 \cup E_3$, $E_i \cap E_k = \emptyset$ за $i \neq k$, $1 \leq i, k \leq 3$. За подобряване на четемостта и показване връзката със събитийния модел на ViPS ще приемем следните означения: $E_1 = E_{\text{base}}$, $E_2 = E_{\text{system}}$, $E_3 = E_{\text{domain}}$.
- Характерна точка на събитие: $C T_e$ означаваме характерна точка, в която настъпва събитието $e \in E$. Най-общо характерната точка може да бъде момент във времето, състояние или нещо друго, характерно за конкретното приложение.

- Превенционен интервал: с ϵ_e означаваме един интервал, предхождащ T_e . В общия случай превенционният интервал може да бъде период от време, поредица от събития или нещо друго, подпомагащо превенцията на РПА за конкретното приложение.
- Събитиеен интервал: с Δ_e означаваме продължителността на събитието $e \in E$.

Всички тези дефиниции могат да бъдат адаптирани за използване в зависимост от актуалната приложна област. Също така е възможно $\epsilon = \Delta = 0$; напр. $\Delta = 0$ при мигновени събития или $\epsilon = 0$, ако интервалът, предхождащ събитието, не се отнася до ПА, т.е. не засяга превантивните или предстоящи действия. Например, в електронното обучение може да има споразумение, според което различните събития ще бъдат различни – напр. изпита $\epsilon = 2$ седмици, лекция $\epsilon = 2$ дни, упражнение $\epsilon = 3$ дни.

Втора стъпка (интервал на действие). Сега сме в състояние да дефинираме интервал на действие (interval of action) на РПА като $intacs(T_e, \epsilon_e, \Delta_e)$, където T_e е характерна точка, ϵ_e е превенционен интервал и Δ_e е събитиеен интервал.

Теоретично, оперирането на един РПА може да се разглежда като цикъл, който управлява контролен поток, състоящ се от един или последователност от $intacs$.

Тук е необходимо да направим някои уточнения. В общия случай, в интервала преди събитието Δ_e РПА не може да разчита на Event Engine (EE), понеже EE се задейства, когато възникне определено събитие. За да се гарантира изпълнението на стратегията за „изпреварващо действие“, една възможност е РПА да взаимодейства и да бъде подпомаган от аналитичното подпространство и по-специално от AmbiNet и TNet компонентите. Нашата идея е за интервала ϵ да бъде управляван от CCA и AjTempura. По отношение на AjTempura е мислимо:

На ϵ да се присвои един ITL интервал, като по този начин AjTempura може да се включи в оперирането на ViPS;

Необходимите дейности по време на този интервал се управляват от AjTempura, т.е. AjTempura става проактивна (тя се реализира като система с множество агенти).

Трета стъпка (Personal Schedule Table). Основна структура за представяне на жизнения цикъл на РПА. PST е абстрактна структура, дефинирана както следва: $PST = (index, option, epsilon, delta)$, където:

- *index* – входни точки в структурата. Индексът еднозначно идентифицира всеки запис в таблицата. Като индекси са допустими *V* на РПА (базови събития според класификацията на CM), т.е. $index = \{date, time, location\}$.
- *option* – това са същинските записи в таблицата. Записите са домейн-събития според класификацията на CM. За приложната област електронно обучение такива могат да бъдат напр., $option = \{lecture, exam, lab, \dots\}$.
- *epsilon* – превенционен интервал на съответната опция. Определя интервала за превенция на кореспондиращото домейн-събитие. *epsilon* играе ролята на своеобразен метрику за определяне на намеренията (целите) на РПА в режим на превенция.
- *delta* – събитиен интервал на кореспондиращата опция. Определя интервала за управление, контрол и подпомагане във времето на протичане на кореспондиращото домейн-събитие. *delta* е своеобразен метрику за определяне на намеренията (целите) на РПА в режим на управление на събитието.

За управление и работа с PST са специфицирани няколко групи оператори, които ще бъдат представени по-долу. Всички специфицирани оператори са изключително право на РПА.

Инициализация на PST: `initialize(method)` – инициализация на PST посредством посочения метод.

Insert-оператори: тези оператори се използват за въвеждане на записи в таблицата:

- `insert(entry-point)` – създава се нов вход в PST (резервира се времеви сегмент), без да се специфицира кореспондиращата опция.
- `insert(entry-point,option)` – в указания времеви сегмент (първи параметър) се записва дадената опция (втори параметър).
- `insert(entry-point,epsilon)` – въвежда се кореспондиращият превенционен интервал.

Изтриване на запис от таблицата: `delete(entry-point)` – изтрива записа със зададения като параметър индекс.

Update-оператори: използват се за актуализиране на записите в PST. Не се допуска актуализация на опциите. При такава необходимост се създава нов запис в таблицата, представящ актуализираната опция, а невалидната опция се изтрива. Също така индексът на един запис в таблицата не може да бъде променян.

- `update(entry-point,epsilon)` – актуализира кореспондиращия превенционен интервал.
- `update(entry-point,delta)` – актуализира кореспондиращия събитийен интервал.

Оператори активатори/деактиватори: също така могат да се нарекат тригери. Тук е необходимо малко по-подробно обяснение. Стойностите на двата интервала (`epsilon` и `delta`) могат да бъдат:

- Очаквани – това са предварителни стойности за двата интервала. Очакваните стойности се задават при създаване или актуализиране на кореспондиращия запис в таблицата. Тези стойности се определят в зависимост от предишни наблюдения, статистика или някой друг метод, приложим за конкретната проблемна област. В действителност обаче, може да се окаже, че в дадения случай тези стойности не са съвсем точни.

- Актуални – точни реални стойности, които събитийният процесор докладва на РПА. При получаване на съобщение от СП, РПА отменя очакваните стойности и активира съответния оператор.

РПА разполага със следните оператори активатори/деактиватори:

- epsilon-activator() – активатор на epsilon интервала;
- delta-activator() – активатор на delta интервала;
- epsilon-terminator() – деактиватор на epsilon интервала;
- delta-terminator() – деактиватор на delta интервала.

За конкретните инстанции (екземпляри) на РПА, структурата PST може да има различни програмно-технически имплементации. Предполагаме, че при генериране на конкретна инстанция РПА ще предоставя необходими средства за първоначална инициализация на PST. Съществуват различни възможности, като напр.:

- РПА да осъществи достъп до вътрешната (in-house) информационна система на приложението за извличане на необходимата информация за инициализиране на PST.
- РПА да предлага средства за „ръчно“ инициализиране на PST, като напр. подходящи входни екрани, редактори на събития и PST съответно.

Четвърта стъпка (BDI модел на РПА). РПА използва три ментални състояния, дефинирани както следва:

- Beliefs (B) – **B** моделират „представата“ на агентите за заобикалящата ги околна среда. В настоящия теоретичен модел **B** основно са базови ($\{date, time, location\}$) събития. В определени случаи могат да бъдат също системни събития. Т.е. дефинираме $B \subseteq E_{base} \cup E_{system}$.
- Desires (D) – **D** моделират опциите, които има един агент във всеки един момент от жизнения му цикъл. Понеже трябва да кореспондират с ежедневните дейности на подпомагания от

персоналния асистент потребител, в модела те се представят посредством domain-събитията. Т.е. можем да ги дефинираме като $D \subseteq E_{\text{domain}}$.

- Intentions (I) – във всеки един момент РПА има актуална специфицирана цел (I). В зависимост от начина на опериране на РПА тя се селектира от множеството на възможните опции (D). Допускаме, че в общия случай асистентът може да има повече от една актуална цел. Така целите на агентите дефинираме като $I \subseteq D$.

Функция dra. Действията на РПА във фазата „Обмисляне“ могат да се представят най-общо посредством функция, наречена dra (deliberation of personal assistant). Функцията има следния вид: $\text{dra}: 2B \times 2D \rightarrow 2I$. В зависимост от актуалната вяра и възможните опции на РПА като резултат функцията връща актуални цели, които трябва да бъдат постигнати в следващата фаза „Планиране“. В общия си вид функцията dra е детерминирана, но в конкретни случаи е възможна и детерминиран неин вариант, т.е. $\text{dra}: 2B \times 2D \rightarrow I$.

Пета стъпка (Структура Profile). За персонализация на конкретни адаптации на РПА се поддържа специална структура Profile. В тази структура се съхраняват параметри, характеризиращи конкретния потребител на персоналния асистент. Обикновено Profile ще включва следните компоненти:

- Identification – данни за еднозначна идентификация на потребителя;
- Accommodation – данни за организацията, с която е свързан потребителя;
- Group – група, към която принадлежи потребителя, спрямо използваната класификация на потребителите.

Структурата може да бъде разширявана с допълнителни компоненти в зависимост от конкретното приложение.

Ще маркираме общата стратегия за адаптиране на РПА. В предлаганата версия на модела на РПА стратегията включва две йерархични нива:

- Адаптация на РПА за конкретна приложна област – за целта се използват домейн-събитията, характеризиращи интересуващата ни приложна област.
- Персонализиране на РПА за конкретния потребител – адаптираният вече за конкретната приложна РПА може да бъде персонализиран за конкретна група потребители или конкретен потребител. За целта са необходими структури като Profile, които да съхраняват необходимите за персонализацията параметри. Към Profile структурата могат да се специфицират подходящи операции за поддържане актуалността на тази характеристична информация. Също е възможно да се съхранява значеща история.

Обща характеристика на архитектурата

На основата на предложения модел се предлага абстрактна архитектура на РПА. Основно за тази архитектура е, че РПА ще бъде разработен съответно представения VDI модел и кореспондиращия му референтен жизнен цикъл (познат като agent control loop), който е описан детайлно в [8]. Контролният цикъл отразява принципите и основните характеристики на практически разсъждаващи рационални агенти с ограничени ресурси. С ограничени ресурси означава, че изграждайки архитектурата е необходимо да се отчита, че дейностите на рационалния агент са изчислителни процеси, ресурсите, за които са ограничени (памет, изчислително време, време за изчакване). Независимо от нивото на детайлизация (както е представен в [8]) agent control loop включва две базови действия, наречени разсъждение и планиране.

Разсъждение (Deliberation). Тази дейност се базира на менталните състояния на рационалните агенти. Накратко, разсъжденията трябва да определят актуалната цел (намерение) на един рационален агент, като се отчитат неговите желания (опции) и вярата му за света.

Планиране (Means-Ends-Reasoning). Разбира се, изборът на намерение за постигане е само част от цялостния проблем с практически разсъждения – агентът все още трябва да определи как да постигне намерението чрез разсъждения за крайни средства.

Глава 5. Среда BLISS

РПА, предложен в предходната глава, е адаптиран в среда за електронно обучение, наречена BLISS (Brezovo`s Learning School System). Средата подпомага провеждане на свободна форма на обучение в средното училище. Свободна форма за обучение се прилага основно за хора, които желаят да завършат средното си образование, но не са в състояние да посещават училище, т.е. в редовна форма за обучение. Обикновено тези хора не са в ученическа възраст, работят, имат семейства и деца. От една страна, за всеки от тях е необходимо да се изготви отделна учебна програма, която се изпълнява в различни времеви периоди. От друга страна, учителите трябва да провеждат този вид обучение заедно с ангажиментите си в редовната (стандартната) форма, което ги затруднява допълнително. Един от най-големите проблеми е проследяването, контрола и изпълнението на всяка една отделна учебна програма.

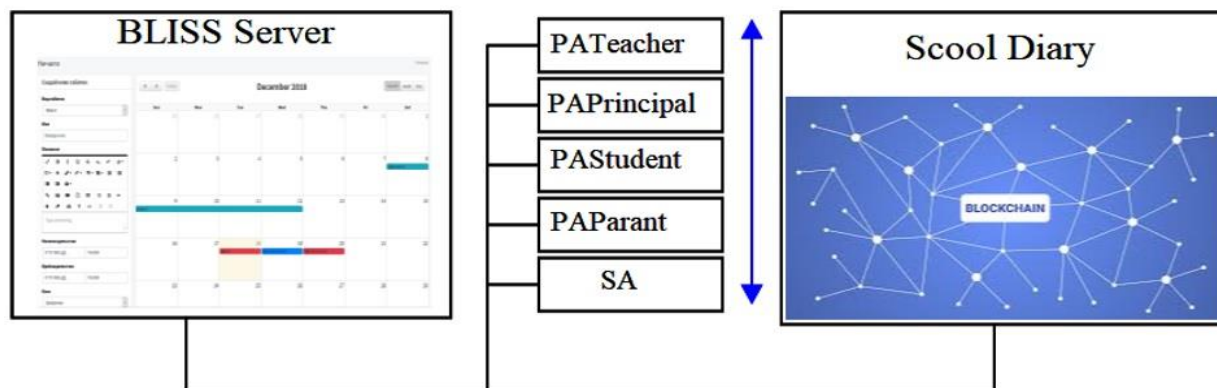
Анализът на представената по-горе ситуация показва, че е целесъобразно приложение на един агентно-ориентиран подход за решаване на този проблем. РПА може да се адаптира така, че в средата, всеки отделен ученик е представен от собствен персонален асистент. От клиентската страна на BLISS асистентът е персонализиран така, че познавайки конкретната учебна програма и в съответствие с нея той подготвя персонален план за подготовка на ученика, води диалог с ученика, прави предупреждения. Персоналният асистент взаимодейства с оперативните агенти на сървъра на средата, за да получи информация, засягаща конкретния ученик. Такива данни могат да бъдат оценките на ученика, информация за консултации, различни промени и актуализации на учебния график и други такива.

Клиентска част на BLISS

Компонентите, които са адаптирани за разработване на BLISS са GPA и ENet от ViPS. Разработени са нови видове специфични персонални асистенти, предназначени за различни групи потребители на пространството. Освен това, приложението включва нови компоненти, неизвестни на ViPS. Пример за такъв компонент е блокчейн, представящ училищния дневник.

На Фигура 5 може да се види общата архитектура на BLISS, която е разделена на три слоя:

- agent layer – в този слой принадлежат всички инстанции на агентите от пространството;
- server layer – той е съставен от администраторски панел включващ RESTful услуги, които се използват за комуникация между различните агенти и слоеве на системата;
- blockchain layer – се намира blockchain верига, която се използва за водене на дневника на учениците.



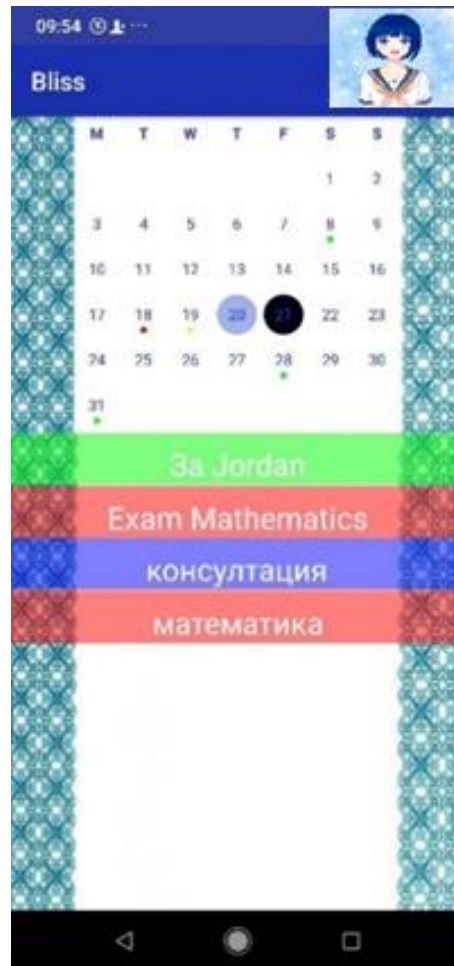
Фигура 5. Архитектура на BLISS

Асистентите комуникират помежду си чрез ACL съобщения. Всяка от тях има връзка с околната среда, посредством набора от сензори принадлежащи на устройството, на което е стартиран агента и има връзка към сървърната част и към училищния дневник.

В средата BLISS се разработват също персонални асистенти, подпомагащи работата на ангажираните със свободната форма на обучение учители и директорът.

Прототип на PASTudent

На Фигура 6 е показан главния екран на разработения прототип на PASTudent. Той държи информиран ученика за всички предстоящите събития, по логиката описана по горе в дисертацията. Освен ранното предупреждение, асистентът визуализира и целия календар с предстоящи събития. Всеки ден, в който има събитие, което е от значение за конкретния ученик е маркиран с различен цвят на календара. Различните цветове отговарят на различни типове събития.



Фигура 6. Прототип PASTudent – главен екран

АВТОРСКА СПРАВКА ЗА РЕЗУЛТАТИТЕ В ДИСЕРТАЦИОННИЯ ТРУД

Основните резултати, представени в дисертационния труд могат да бъдат обобщени, както следва (Таблица 1):

1. Предложена е референтна архитектура на персонален асистент. Основна характеристика на нашата архитектура е, че в менталния компонент на агента (етап на обмисляне) е намерено единно унифицирано представяне – вярата, желанията и намеренията (целите) на агента се представят като събития. Постигнатата хомогенност дава очаквания за определени предимства за реализацията и оперирането на агентите, като напр. подобряване на поведението на агента по време на изпълнение (run-time), опростяване на интерпретацията и трансформациите между менталните състояния. Друга характеристика е, че за разлика от описаните в реферираниите публикации архитектури, в чийто жизнен цикъл се игнорира (или се обръща малко внимание) на фазата инициализация (теоретично агентите работят в режим 24/7), в жизнения цикъл на нашия референтен агент е включен като отделен начален етап инициализацията на агента – според нас, съществен от технологична гледна точка. Съществено за инициализацията е взаимодействие с наследена (in-house) система на конкретната приложна област. Посредством това взаимодействие се инициализират менталните състояния на агента. Опитът ни показва, че за референтна архитектура, която трябва да се адаптира за различни приложни области, това е съществено предимство. Третата характерна особеност е различаване на два режима на работа на персоналните асистенти – превантивен и оперативен, които се поддържат директно от архитектурата.

2. Реализиран е първи прототип на персонален асистент. Първият прототип, наречен LISSA, подпомага студенти по време на учебен процес. Неговата цел (в съответствие с методологията на изследването) е да провери доколко първоначално предложената архитектура може да бъде реализирана с очакваното поведение на асистента. Резултатите от анализа на прототипа са отчетени при изготвяне на окончателния вариант на

референтната архитектура. Прототипът е тества с избрана група студенти от ФМИ на Пловдивски университет „Паисий Хилендарски“.

3. Реализиран е втори прототип на персонален асистент. Целта на втория прототип е да демонстрира възможностите на референтната архитектура при адаптация за конкретна приложна област. Прототипът е ядро на средата за подпомагане на обучението по индивидуални програми в средното училище, наречена BLISS. Използването на системата потвърждава нашето убеждение за важността на предоставяне на възможности за интеграция с наследени системи – в случая, с училищната информационна система. Също, възможността за взаимодействие с нови разработки – в случая, с ново изградена блокова верига.

Таблица 1

Резултат	Задача	Глава	Публикация
1	2	4	3
2	1	3	1, 2
3	3	5	4
	4		
	5		

ПУБЛИКАЦИИ ПО ДИСЕРТАЦИОННИЯ ТРУД

1. Тодоров Й., Даскалов Б., Стоянов С., Вълканов В., „Персонален асистент за подпомагане на студента“, юбилейна научна конференция с международно участие "Новата идея в образованието", Бургас 20-21 юни, 2016 Бургас.
2. Todorov J., V. Valkanov, I. Popchev, “Intelligent personal assistant for aiding students”, International Conference AUTOMATICS AND INFORMATICS`2017, Bulgaria, Sofia, October 4-6 2017, pp. 283-286, ISSN 1313-1850, CD ISSN 1313-1869
3. Й.Тодоров, „Генетичен IoT агент“ стр. 85 - 90 Национална научна конференция "Образование и наука - за личностно и обществено развитие" Пловдивски университет филиал Смолян 27-28 октомври 2017 гр. Смолян
4. Йордан Тодоров, Ирина Кръстева, ПЕРСОНАЛЕН АСИСТЕНТ ЗА ПОДПОМАГАНЕ НА УЧЕБНИЯ ПРОЦЕС В СРЕДНИТЕ УЧИЛИЩА, НАУЧНА КОНФЕРЕНЦИЯ TechCo–Lovech 2019, 217 - 222

БИБЛИОГРАФИЯ

- [1] Stoyanova-Doycheva, E. Doychev, S. Stoyanov, Digital Library in Virtual Education Space, The Journal of Applied Science, Applied Science University, Kongdom of Bahrain, Vol, 1(1), 2016, ISSN 1764-2210, 56-67.
- [2] Bellifemine, Fabio Luigi, Giovanni Caire, and Dominic Greenwood. Developing multi-agent systems with JADE. Vol. 7. John Wiley & Sons, 2007.
- [3] Bellifemine, Fabio, Agostino Poggi, and Giovanni Rimassa. "JADE: a FIPA2000 compliant agent development environment." Proceedings of the fifth international conference on Autonomous agents. 2001.
- [4] Braubach, Lars, Winfried Lamersdorf, and Alexander Pokahr. "Jadex: Implementing a bdi-infrastructure for jade agents." (2003).
- [5] F.-Y. Wang, "The emergence of intelligent enterprises: From CPS to CPSS," IEEE Intell. Syst., vol. 25, no. 4, pp. 85–88, Jul./Aug. 2010.
- [6] J. Santos, J. Rodrigues, J. Casal, K. Saleem, V. Denisov, Intelligent Personal Assistants Based on Internet of Things Approaches IEEE Systems Journal, Vol. PP, Issue 99, 2015, 1-10.
- [7] M. Soliman, C. Guetl, Intelligent Pedagogical Agents in Immersive Virtual Learning Environments: A Review, The 33th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2010), May 24 - 28, 2010, Opatija, Croatia, 827-832.
- [8] M. Wooldridge, Reasoning about Rational Agents, The MIT Press, Cambridge, Massachusetts, London, England, 2000.
- [9] Microsoft Classroom, (2019, May) , <https://classroom.microsoft.com/>.
- [10] Moodle, (2019, May), <https://moodle.com/>.
- [11] P. Maes, "Agents that reduce work and information overload," Communications of the ACM, Vol. 37, No. 7, July 1994, 30-40.
- [12] Pokahr, Alexander, Lars Braubach, and Winfried Lamersdorf. "Jadex: A BDI reasoning engine." Multi-agent programming. Springer, Boston, MA, 2005. 149-174.

- [13] S. Stoyanov, A Virtual Space Supporting eLearning, Proceedings of the Forty Fifth Spring Conference of the Union of Bulgarian Mathematicians Pleven, April 6–10, 2016, 72-82.
- [14] S. Stoyanov, D. Rusev, Integration of virtual and physical worlds in ViPS, Big Data, Knowledge and Control System Engineering (BdKCSE'19), IEEE Conference, 21-22 November, 2019, Sofia.
- [15] S. Stoyanov, H. Zedan, E. Doychev, V. Valkanov, I. Popchev, G. Cholakov and M. Sandalski, Intelligent Distributed eLearning Architecture, V. M. Koleshko (Ed.), Intelligent Systems, InTech, March, 2012, 978-953-51-0054-6, Hard cover, 366 pages, pp. 185-218.
- [16] S. Stoyanov, I. Ganchev, M. O'Droma, H. Zedan, D. Meere, V. Valkanova, Semantic Multi-Agent mLearning System, A. Elci, M. T. Kone, M. A. Orgun (Eds.): “Semantic Agent Systems: Foundations and Applications”, Book Series: Studies in Computational Intelligence, Vol. 344, Springer Verlag,2011, ISBN: 978-3-642-18307-2.
- [17] Stoyanov, S. et al., From CBT to e-Learning, Journal Information Technologies and Control, 3(4), 2005, 2-10.
- [18] T. Glushkova, S. Stoyanov, I. Popchev, S. Cheresharov, Ambient-Oriented Modeling in a Virtual Educational Space, Comptes rendus de l'Académie bulgare des Sciences, Tome 71, No 3/2018, 398-406.
- [19] Wooldridge, M., An Introduction to MultiAgent Systems. Wiley, 2009.