Пловдивски университет „Паисий Хилендарски“

Факултет по математика и информатика

Катедра „Компютърни системи“

Изследвания за създаване на виртуален оператор в инфраструктура за интелигентно земеделие

АВТОРЕФЕРАТ

на дисертационен труд за присъждане на образователна и научна степен “доктор“
в област на висше образование 4. Природни науки, математика и информатика;
професионално направление 4.6. Информатика и компютърни науки;
докторска програма Информатика

Редовен докторант: Иван Станимиров Стоянов
Научен ръководител: проф. д-р Ася Стоянова-Дойчева

Пловдив, 2023

УВОД

Дисертацията е посветена на интелигентното земеделие - една тематика, ставаща все по-интересен обект на теоретични изследвания и практически разработки. В България перспективите за интелигентно земеделие основно са формулирани в Стратегия за развитието на изкуствения интелект в България до 2030 г., както и в двете Национални научни програми (ННП) „Интелигентно растениевъдство“ и „Интелигентно животновъдство“.

Основната цел на дисертацията е да се разработи персонален асистент (ПА), подпомагащ земеделски стопани и земеделски специалисти, работещи в условията на интелигентно земеделие. ПА трябва да оперира като ядро на платформата ЗЕМЕЛА, която е в процес на изграждане. За постигане на целта са формулирани следните задачи:

- Да се актуализира събитийния модел и се предложи нова негова версия.
- Да се актуализира архитектурата на платформата ЗЕМЕЛА.
- Създаване концепция, модел, референтна архитектура и жизнен цикъл на персонален асистент за земеделски стопани.
- Прототипна реализация на персоналния асистент.

Изследването, проведено по време на докторантурата е част от работната програма на ННП „Интелигентно растениевъдство“ [1]. Целта на тази програма е да стимулира целенасочени научни и приложни изследвания за приложение на изкуствения интелект в земеделието. Очаква се изкуственият интелект съществено да допринесе за развиване на модерно, ефективно, базирано на знание, селско стопанство, с което да се повиши качеството на храната и да се опазят природните ресурси.

В съответствие с целта и задачите е предложена подходяща методика за провеждане на изследването. Дисертацията се състои от увод, пет глави и заключение.

ПЪРВА ГЛАВА: СЪСТОЯНИЕ НА ПРОБЛЕМАТИКАТА

В съответствие с темата на дисертацията следните представляват интерес:

- Кибер-физически и кибер-физически-социални системи;
- Референтната архитектура ViPS;
- Платформи за прецизно и интелигентно земеделие;
- Персонални асистенти.

През последните десетилетия, с големите пробиви в науката, технологиите и инженерството, хората все по-интензивно взаимодействат с околния физически свят. Съвременните технологии кибер-физични системи [2] и Интернет на нещата [3] допринасят значително за еволюция в компютърната взаимосвързаност. Те интегрират динамиката на физическите процеси с тези на софтуера и комуникацията, като осигуряват абстракции и техники за моделиране, проектиране и анализ на комплексни интегрирани системи.

Проведеното по време на дисертацията изследване е тясно свързано с референтната архитектура, наречена Виртуално-физическо пространство (Virtual Physical Space, ViPS). Започналата преди повече от двадесет години разработка минава през различни етапи. Три етапа могат да бъдат отбелязани, през които бяха разработени архитектурите Distributed eLearning Center (DeLC), Virtual Education Space (VES) и Virtual-Physical Space (ViPS). DeLC [?] е среда за доставка на образователни услуги и електронно съдържание за различни форми на електронно обучение. Архитектурата на DeLC е разпределена и се моделира като мрежа, състояща се от отделни възли, наречени eLearning Nodes (eLNs). Най-пълно описание на DeLC може да бъде намерено в [5]. VES е инфраструктура, поддържаща електронно обучение, в която потребители, време, местоположение, автономност и контекстна зависимост са основополагащи и която осигурява унифицирана обработка и интерпретация на информацията, идваща както от виртуалната околна среда, така и от физическия свят [6]. ViPS се изгражда като референтна архитектура, интегрираща виртуалния и физическия светове [7], която може да бъде адаптирана за различни CPSS подобни приложения. В този смисъл съществени аспекти на ViPS са следните: потребителите са във фокуса на вниманието, виртуализация на физическите „неща“ и интеграция на виртуалния и физическия светове.

Във всички региони на света се полагат все повече усилия за дигитализиране на селскостопанския сектор и превръщането му в така нареченото интелигентно земеделие. В момента се разработват огромен брой разнообразни проекти в областта на интелигентното земеделие. Проектът FaST (Farm Sustainability Tool) [8] има за цел да разработи селскостопанска платформа, предоставяща дигитални услуги, за да помогне на европейските фермери и агенции да подобрят съответните си възможности в множество селскостопански, екологични и фокусирани върху устойчивостта дейности. Платформата Farm21 [9] предоставя полеви решения, които позволяват последователен и актуален достъп от сензори, почва, култури, време и сателити. Регионалният офис на ФАО за Африка разработва регионална техническа платформа [10] в подкрепа на общност от практики от заинтересовани страни и партньори за споделяне на знания, най-добри практики и научени уроци относно общи селскостопански политики и практики и за насърчаване на съвместни усилия.

В последните години изследванията и практическите разработки, свързани с интелигентен персонален асистент (ИПА) нарастват значително. ИПА могат да бъдат класифицирани в две големи групи: с общо предназначение и със специализирано предназначение. Актуалното и най-високо технологично ниво на разработване на ИПА е постигнато с Generative Pre-trained Transformer-4 (GPT-4) [11].

От прегледа на състоянието на проблематиката са направени следните изводи, които направляваха провеждането на изследването, резултатите от което са представени в дисертационния труд.

- Платформите са с глобално предназначение. Те целят решаване на голямо разнообразие от задачи в областта на прецизното земеделие в по-глобален мащаб. Тези платформи се подкрепят от сериозни институции и се радват на мощно финансиране и свързани с други значими проекти (напр. Коперник, Галилей), от които получават нужната информация.
- Представената в тази дисертация платформа ЗЕМЕЛА е с *регионално предназначение*. Нашата платформа е насочена предимно към малки и средни земеделски стопани. Тя използва данни предимно от стационарни сензорни мрежи, разположени в малки открити блокове или оранжерии. Идеята е да се използва спецификата на локалния регион за получаване на оптимални резултати. Функционирането на платформата се осъществява от личен асистент, използващ събитиен модел и специфично представяне на специализирани знания.

- Изследванията за създаване на персонални асистенти стават все по-актуални и значими в последните години. Определени персонални асистенти с общо предназначение се използват за решаване на специализирани задачи, използвайки мощните си езикови процесори и възможностите за предварително трениране. В разгледаните платформи за интелигентно земеделие преобладават архитектури, ориентирани към услуги. Агентно-ориентираният подход е слабо застъпен.
- Въпреки появата на все повече и все по-добре работещи интелигентни персонални асистенти с общо предназначение, персоналните асистенти със специализирано предназначение остава все още малко изследвана научна област. Персоналният агент, който е резултат от проведеното по време на докторантурата изследване, е интелигентен асистент със специално предназначение – за подпомагане на земеделски стопани, работещи в условията на интелигентно земеделие. Предлагаме един по-различен подход – не се разчита на мощен езиков процесор, а по-скоро предложеният персонален асистент е ориентиран към използване на хранилища от специализирани знания, представени като семантични мрежи (онтологии), правила и фрейми. Освен това, функционирането на персоналния асистент се поддържа от събитиен модел.
- Ориентацията към кибер-физически системи в различни приложни области, включително и интелигентно земеделие, става все по-очевидна. Всички разгледани платформи използват различни типове сензори, дронове, роботи и спътници за събиране на информация за физическия свят.
- Платформата ЗЕМЕЛА се разработва като виртуално-физическо-социално пространство – пространство понеже персоналните асистенти, имплементирани като интелигентни агенти, оперират като централен компонент, управляващ и контролиращ цялостната работа на платформата. ЗЕМЕЛА е адаптация за интелигентно замеделие на референтната архитектура ViPS.

Още един съществен извод: за успешната реализация на планирания персонален асистент е необходим задълбочен преглед и анализ на съществуващото състояние на платформата ЗЕМЕЛА и събитийния модел.

ВТОРА ГЛАВА: СЪБИТИЕН МОДЕЛ

Платформата ЗЕМЕЛА се разработва като виртуално-физическо пространство. Съществено предизвикателство е виртуализацията на физическите обекти, която е един многоаспектен процес. Освен характеристиките, произтичащи от естеството на обектите, е необходимо да се отчитат също техните времеви, пространствени и събитийни (т.е. тяхното въвличане в различни случвания) аспекти. За целта е необходима теоретична основа, която да изпълнява ролата на специфична рамка, в границите на която да се имплементира платформата. Търсим опростен теоретичен модел, който да може реално да бъде търсената рамка. В резултат на наши предварителни проучвания и анализи се формира тезата, че върху понятието „събитие“ може да се предложи такъв модел. Настоящата дисертация продължава също изследванията за доказване на тази теза. По тази причина беше разработен така нареченият *събитиен модел*.

Събитията са основен механизъм за активиране на сценарии и действия, както и за синхронизиране на работата на активните компоненти в платформата ЗЕМЕЛА. Едно **събитие** е нещо, което се случва или приемаме за случващо се на определено място и в даден момент (или интервал) от време и което оказва влияние върху работата на платформата. В ZEMELA, тъй като събитията могат да се случват както във физическия, така и във виртуалния свят, се стремим да предложим общо абстрактно (символно) представяне на събитията. Тези представяния се съхраняват и управляват във виртуалния свят (т.е. събитията във физическия свят са виртуализирани), но могат да предизвикват ефекти във физическия свят. Намирането на подходяща презентация е предизвикателство поради голямото разнообразие от събития с техните специфики за интересуващата ни област.

Повечето от представените по-горе събитийни модели предлагат сложни структури за представяне и моделиране на събитията. Донякъде това е обяснимо – както беше подчертано събитията имат много аспектно естество. Нашият събитиен модел може да се характеризира накратко, както следва:

- Опростено представяне на събития;
- Йерархия на събитията;
- Представяне на сложни събития чрез рекурсивни дефиниции;
- Проактивност.

Събитийният модел предоставя обща единна схема, изпълняваща ролята на теоретична основа за създаване на платформата ЗЕМЕЛА. Разработването на платформата се извършва на следните нива:

- Най-абстрактното ниво е събитийният модел.
- Второто ниво е представително ниво – представяне на събитията като знания с общ смисъл (правила, фрейми, онтологии).
- Програмно-техническо ниво – програмната реализация на събитията и тяхната обработка.

Класификациите играят много съществена роля в нашия събитиен модел. В събитиния модел са дефинирани две основни класификации. Първата използва йерархичен подход, при който събитията от по-високо ниво се третират като предварително определени от събития от по-ниско ниво. По този начин нашият събитиен модел разграничава следните три нива на събития:

- *Базови събития* – обикновено това са атомарни събития, т.е. събития, които разглеждаме като неделими. Тези събития представяме като двойка {идентификатор_събитие : стойност_събитие}. Следните базови събития са от особен интерс в модела: време (дата, час, минута), местоположение, (екстремални) стойности на сензори. Този вид събития заемат най-ниското място в йерархията. Базовите събития могат да инициират различни събития от по-високо ниво в платформата и като такива могат да обслужват други събития. В зависимост от особеностите на поддържания домейн, могат да бъдат специфицирани допълнителни базови събития.
- *Системни събития* – този тип събития представят различни събития в системната инфраструктура на платформата, като например генериране/премахване на динамични компоненти и изпращане/получаване на съобщения. Системните събития обикновено се използват за управление на взаимовръзката между ПА и другите компоненти на платформата. Тези събития се намират в средното ниво на йерархията.
- *Домейн-събития* – те представят събития, типични за интересуващата ни приложна област. Тези събития обикновено имат сложна структура. За всяка приложна област съществува голямо разнообразие от домейн-събития.

В нашия модел тези събития играят изключително важна роля.

Втората класификация е свързана с това как протичат събитията във времето. В този смисъл разграничаваме две големи групи събития, които сме назовали:

- *Планови* – тези събития се очакват и могат да се появят в определен момент (или интервал) във времето. В нормален режим работата на ПА е направлявана от този вид събития.
- *Инцидентни* – тези събитията възникват случайно в определено време или интервал от време. Инцидентните събития обикновено се идентифицират от Event Engine.

Между двете класификации съществува тясна зависимост (връзка). За да я разкрием ще разгледаме по-подробно базовите събития на модела:

- *Time* – най-общо времето може да бъде реално или относително, представено като часове, дни, месеци или години. Това събитие е основополагащо за работа с темпоралните аспекти на моделираните физически обекти, процеси или сценарии. Този тип събития обикновено са съставна част на плановите домейн-събития.
- *Location* – най-общо това базово събитие се използва за представяне (моделиране) на пространствените аспекти на интересуващите ни физически обекти, процеси или сценарии. Евентуално може да бъде съставна част на двата типа домейн-събития.
- *Extreme value* – този тип базови събития служат за идентифициране на неочаквано възникнали домей-събития. Като такива те обикновено са съставна част на инцидентните домейн-събития.

Формално едно събитие се представя като e = <eid, etype, epar> със следните елементи: eid - идентификатор на събитието, etype – тип на събитието, epar - списък с атрибути на събитие, който може да бъде празен.

В събитийния модел са въведени различни групи оператори върху домейн-събитията. Използването на подходящи оператори усилва формализма с възможност за специфициране на събитийни изрази. От една страна, събитийните изрази могат формално да моделират сложни и комплексни ситуации, свързани със събитията. От друга страна, можем да имплементираме интерпретатори на тези изрази, които ще

улесняват управлението на платформата. Основните оператори са следните:

- $e_i \uparrow e_j$ – възникването на e_i предизвиква случването на e_j.
- $e_i \downarrow e_j$ – възникването на e_i прекратява събитието e_j.
- accu(e) – акумулиране на събития.

За да можем да използваме модела на събитието като основа за оперативното управление на платформата изискваме и двата оператора да отговарят на следните изисквания:

- *Некомутативност* – $e_i \uparrow e_j \neq e_j \uparrow e_i$, т.е. ако събитието e_i предизвиква случването на събитие e_j, от това не трябва да следва, че e_j може да предизвика случването на e_i. Аналогично за оператора $\downarrow$.
- *Ляво-асоциативност за* $\uparrow$ – от досегашния анализ изглежда, че операторът $\uparrow$ е ляво-асоциативен, т.е. $e_i \uparrow e_k \uparrow e_j$ може да се интерпретира като $(e_i \uparrow e_k)\uparrow e_j$ или събитие e_i предизвика възникване на събитие e_k, което от своя страна води до възникване на събитие e_j.
- *Дясно-асоциативност за* $\downarrow$ – по подобен начин изглежда, че операторът $\downarrow$ е дясно-асоциативен, т.е. изразът $e_i \downarrow e_k \downarrow e_j$ трябва да се интерпретира като $e_i \downarrow (e_k \downarrow e_j)$ или събитие e_k трябва първо да прекрати събитие e_j преди то да бъде прекратено от събитие e_i.
- *Транзитивност* – домейн-събитията са транзитивни.

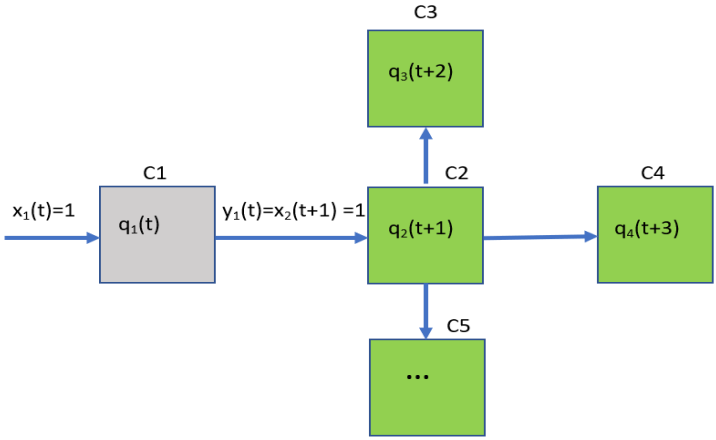
В последната година и половина, въпреки усилията и различните опити за имплементиране, нямаме подходящо средство за идентифициране и локализиране на събития. Оказва се, че идентифицирането на домейн-събитията (особено инцидентните) е сериозно предизвикателство. Причините за това са от различно естество, като напр. голямо разнообразие на домейн-събития, различни източници за получаване на стойностите на характеризиращите едно домейн-събитие параметри се получават от различни източници, факторът време.

Обобщавайки актуалното състояние в дисертацията се предлага следното решение: **идентификацията да стане част от събитийния модел**. В събитийния модел да се предложи *референтна абстрактна събитийна машина* (Abstract Event Engine). В платформата да се поддържат различни имплементации на абстрактната събитийна машина, в зависимост от естеството на конкретното домейн-събитие и от необходимите източници на данни. Като част от събитийния модел

търсим формално представяне на абстрактната Event Engine. В тази точка ще разгледаваме възможността за използване на *клетъчен автомат* за моделиране на машината. Един клетъчен автомат може да се разглежда като абстракция (идеализация) на физическо явление, в което пространството и времето са дискретизирани, т.е. множеството от състояния на автомата е дискретно и крайно [12]. Отделните компоненти на един такъв автомат, наречени *клет*ки, са идентични с някакъв изчислителен апарат. Често, за представяне на автомата се използва едномерна, двумерна или многомерна мрежа, като отделните клетки са свързани по единен начин помежду си. Клетките, които влияят на дадена конкретна клетка, се наричат *съседство* на клетката.

Ще демонстрираме идеята за моделиране на поведението на абстрактната събитийна машина (AEE) с пример. Значимите инциденти са болестите по културите. Една от най-разпространените и опасни болести по доматите е картофената мана. Първите признаци на заболяването се забелязват по най-долните листа, които са в контакт с почвата. Появяват се кафяви петна, които нарастват и листата изсъхват. При благоприятни условия болестта се разпространява много бързо по листните дръжки, стъблата и накрая обхваща цялото растение. По зелените плодове от страната на дръжката се вижда ръждивокафяво петно с неправилна форма. Това заболяване може да се моделира като домейн-събитие, настъпващо във времето и представено като клетъчен автомат, в който всяка отделна клетка моделира различна част от растението. На Фиг. 1. е даден примерен клетъчен автомат за болестта, където използваме следните означения:

- Клетки на модела: C1 (най-ниските листа), C2 (листа на стъблото), C3 (стъблата), C4 (зелени плодове), и т.н.
- t – времеви момент;
- q – състояние на клетката в момент t;
- x – вход в момент t;
- $\delta(q,x)$ – състояние на клетката в момент t+1 (функция на прехода);
- $y = \lambda(q,x)$, изход в момент t, λ (изходна функция).



Фигура 1. Клетъчен автомат, моделираш болестта „картофена мана“ при доматите

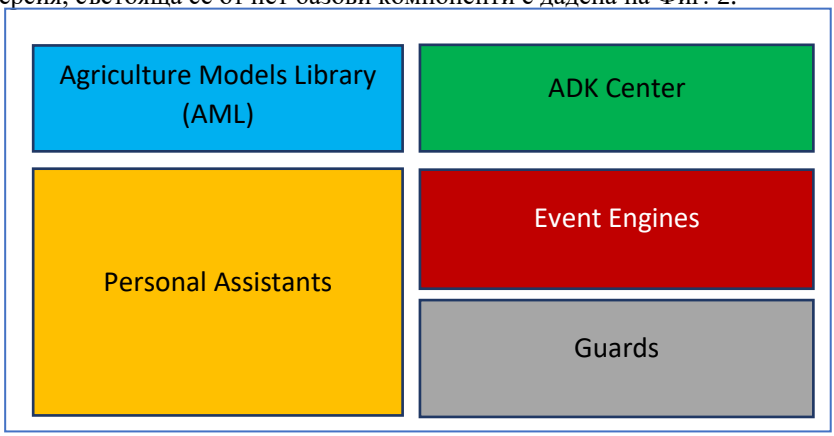
Могат да бъдат определени различни правила за клетъчния автомат, които да направляват работата му, като например следните:

- Когато се забележи появата на кафяви петна върху най-долните листа, които са в контакт с почвата, тогава C1 "оживява", което означава, че t е началното време, x1(t) = 1 и вътрешното състояние на автомата става q1(t).
- Състоянието на предишната клетка (напр. най-долните листа), представено като изход y1(t) = 1, се предава като вход (часовникът преминава към следващата стъпка) към съседната клетка (напр. листата на стъблото) като y1(t) = x2(t+1) = 1 .

Така предложеният модел на АЕЕ може да се вгради във вярата на ПА.

ТРЕТА ГЛАВА: ПЛАТФОРМАТА ЗЕМЕЛА

Предназначение на платформата е да подпомага разработването на приложения за интелигентно земеделие. ЗЕМЕЛА е адаптация на ViPS за конкретна приложна област – в случая интелигентно земеделие. Като такава тя се изгражда като кибер-физическо социално пространство [13]. Платформата ЗЕМЕЛА е в процес на непрекъснато развитие. Предишни версии на платформата са представени в [14]. Основавайки се на извършения по време на докторантурата анализ тук ще представим накратко актуалната версия на платформата. Целта е да се предложи опростена архитектура, реализиран прототип на която да работи в условията на пилотния проект за района на Пловдив. Предлаганата нова версия, състояща се от пет базови компоненти е дадена на Фиг. 2.



Фигура 2. Нова версия на архитектурата на платформата ЗЕМЕЛА

Персонални асистенти (Personal Assistants). Те оперират като основен оперативен механизъм, осигуряващ функционирането на платформата. Основната задача на ПА е да следят за нормалното протичане на наблюдаваните селскостопански процеси и сценарии. ПА са представени детайлно в следващите две глави на дисертацията.

Събитийни машини (Event Engines). Отговорността на този тип модули, наречени Event Engines, е основно да подпомагат ПА за идентифициране основно на инцидентни домейн-събития. Откриването на такива събития са основен механизъм за откриване на аномалии във вегетацията на земеделските култури. Теоретичната основа за

реализиранена Event Engines е събитийният модел, представен в предишната глава на дисертацията. В новата версия на платформата концепцията за събитийни машини е напълно променена. Специфицирането на референтна абстрактна машина като част от самия модел, предоставя възможност за различни имплементации и с това съществуването на различни Event Engines. Това доставя много по-голяма гъвкавост и вариативност за идентифициране на аномалии – основополагаща дейност за платформата ЗЕМЕЛА. Модулите, изпълняващи ролята на събитийни машини могат да бъдат от различен вид, включително и такива, за които разпознаването на инцидентни домейн-събития може да се комбинира с други функции или да се получи като страничен ефект – например разработваните експертни системи за ранно откриване на заболявания по земеделските култури.

Център за данни и знания в земеделието (ADK Center). Този център действа като централно разпределено хранилище на знания и данни на платформата. Състои се от две основни части. Първата част е разпределена база от знания, в която се съхраняват и управляват теоретични знания за селското стопанство. Съхраняват се знания за така наречените „природни артефакти“ – това са селскостопанските култури и факторите, влияещи върху тяхната растителност като почва, въздух и вода. Освен това се съхраняват знания за „изкуствени артефакти“ – това са напоителни системи, сензорни мрежи, вентилационни системи и др. Третият тип съхранено знание е за земеделски дейности и събития, случващи се в сценарии и настройки на земеделието. По принцип знанията се съхраняват във взаимно интегрирани онтологии, фрейми и правила. Някои от внедряваните в момента основни онтологии за селското стопанство са представени в [15]. Базите от знания, моделиращи селскостопански знания под формата на правила и рамки, са описани в [16]. Втората част се състои от релационни бази данни и NoSQL за съхраняване на различни данни за земеделските култури и процесите, протичащи в тях. Данните, необходими за обслужване на платформата, могат да бъдат разглеждани от различни аспекти. Те могат да бъдат структурирани, полуструктурирани и неструктурирани. Втори аспект може да бъде честотата на промените – в този смисъл те могат да бъдат статични и динамични. Трети аспект е естеството на представяните обекти – в този смисъл можем да различаваме естествени обекти и изкуствени обекти. Отчитайки това разнообразие от фактори е съществено разработването на подходящ модел на данните. В новата версия на системата концепцията за ADK Center също е променена. Освен интеграцията на ViSMod модула в новата структура на

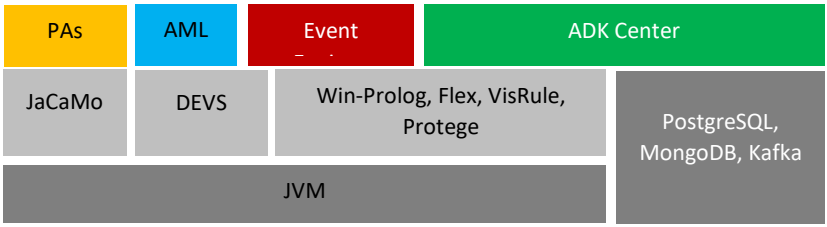
хранилището тук ще представим първоначалния идея за изграждане на единен модел на данните, които ще се съхраняват в ADK Center. Такъв модел напълно отсъства в предишната версия на платформата ЗЕМЕЛА. Условно моделът ще разполага разпределената база данни в две плоскости – нека ги означим като „вертикална“ и „хоризонтална“. Във „вертикалната“ плоскост ще разполагат базите данни („естествени бази данни”), съхраняващи данни за естествените обекти и фактори, като земеделски култури, въздух, почви и води. В хоризонталната плоскост ще се изграждат така наречените „изкуствени бази данни” за съхранение на данни за изкуствени артефакти като напоителни инсталации, сензорни мрежи, вентилационни инсталации и др. Особеност на нашия модел на данни е, че той трябва информационно да осигурява виртуално-физическо пространство и свързаната с това необходимост от виртуализация на физическите обекти [17]. За целта допълнително е необходимо съхраняване и управление на пространствените и времевите характеристики на физическите обекти.

Библиотека с модели на селското стопанство (Agriculture Models Library). В допълнение към земеделските знания и данни, платформата поддържа библиотека с модели, наречена Agriculture Models Library (AML). Моделирането на различни селскостопански сценарии и процеси би подпомогнало работата на ПА по един ефективен начин. Така например, когато взема определено решение, ПА може да вземе предвид резултатите, получени от обработката на определен модел. Анализирайки резултатите от последните две години (в предишната версия на платформата беше предвидено използването на различни моделиращи подходи и системи) в новата версия на тази библиотека предлагаме водещият подход за моделиране да се основава на спецификацията на системата за дискретни събития DEVS. DEVS (Discrete Event Specification) е общ формализъм за моделиране и симулация, посветен на системата за динамични дискретни събития. Резултатите от експеримент за използваемостта на този подход за моделиране на процеси, свързани с напояване са представени в [18].

Гардова система (Guards). Една от функциите на гардовете е да действат като интерфейс между виртуалния и физическия свят. Те могат да подпомагат обмена на информация (и при необходимост нейното трансформиране) между двата свята. Физическият свят в платформата се възприема като мрежа от IoT възли, която е в състояние да събира актуална информация за „ситуацията на място“, т.е. сензорни данни от наблюдаваните отворени (например блокове, парцели) и затворени селскостопански площи (например оранжерии).

След представяне на отделните модули, изграждащи архитектурата на платформата, вторият съществен аспект е взаимодействието между тях. За взаимодействие между компонентите на абстрактно ниво ще използваме A&A (Agents & Artifacts) концепцията. В съответствие с [19] A&A моделът разглежда средата на агентите като набор от артефакти, които агентите могат да създават, фокусират и разпореждат за преследване на своите цели.

За имплементиране на платформата ЗЕМЕЛА предлагаме една интегрирана технология, опериираща върху JVM (Java Virtual Machine) (Фиг. 3.).



Фигура 3. Интегрирана технология

За разработване на моделите в AML, както и за някои варианти на събитийните машини, предлагаме използване на подход, основаващ се на DEVS.

Компютърната конфигурация, където може да бъде разгърната платформата, се изградена на три нива: местно, регионално и национално. На местно ниво обикновено са сензорни мрежи за събиране на информация за състоянието на земеделските земи и растения. Регионалното ниво е сървърна конфигурация за събиране, съхранение и предварителна обработка на данни от динамични сензори. На това ниво се реализира регионален център за данни. Националното ниво ще бъде развито за агрегиране на данни и знания за изготвяне на статистики, анализи и прогнози от национално значение.

ZEMELA е софтуерна платформа, внедрена върху представената IoT тристепенна вертикална инфраструктура, която може да се използва за изграждане на интелигентни селскостопански приложения.

ЧЕТВЪРТА ГЛАВА: РЕФЕРЕНТЕН ПЕРСОНАЛЕН АСИСТЕНТ

Представеният в тази глава персонален асистент за подпомагане на земеделските стопани ще бъде със специално предназначение. ПА използва събитийния модел и ADK Center и е имплементиран като рационален практически разсъждаващ агент с BDI архитектура. Различните услуги, които ПА може да доставя на земеделските стопани се основават на една негова фундаментална способност – да идентифицира и локализира различни аномалии, основно свързани с вегетацията на селскостопанските култури. Аномалиите могат да имат различни форми и размери на проявление, включително като обекти, числови стойности, събития, модели и наблюдения.

В съответствие с [20] стандартният жизнен цикъл е предствен на Фиг. 4. Основната структура на процеса на вземане на решения е цикъл, в който агентът непрекъснато наблюдава света и актуализира убежденията. След това обмисля да реши какво намерение (цел) да постигне (обсъждането се извършва чрез първо определяне на наличните опции и след това чрез филтриране). В последствие използва метод за планиране за да намери план за постигане на приетото намерение (цел). Последната стъпка е изпълнение на плана.

```
// Първа стъпка: Инициализация
B ← consult(ADK_Center, plant); I ← beginning_vegetation(B) ;
vegetation(plant) ← true;
// Втора стъпка: Идентификация на аномалии
while vegetation(plant) do
   получаване на следващо възприятие ρ чрез guard(…) функция;
   B ← brf(B, ρ) ;
   if incidental_event_recognized(Event_Engine) then
      D_ie ← options(B, I) ;
      I_ie ← filter(B, D, I) ;
      π_ie ← plan(B, I, Ac) ;
   end-if
   if incidental_event_recognized(Event_Engine) then
      D_pe ← options(B, I) ;
      I_pe ← filter(B, D, I) ;
      π_pe ← plan(B, I, Ac) ;
   end-if
```

```
// Трета стъпка: Превеждане на избрания план за изпълнение
    execute(πie) or execute(πpe);
    vegetation(plant) ← false;
end-while
```

Фигура 1. Жизнен цикъл на стандартен ПА за интелигентно земеделие

Означенията, използвани в псевдокода на жизнения цикъл на ПА са: B (актуалната вяра), I, I_{ie}, I_{pe} (актуални цели), D, D_{ie}, D_{pe} (налични опции), π_{ie}, π_{pe} (планове за обработка на ицидентни, съответно на планове събития). Най-общо, за предложения тук жизнен цикъл за нашия ПА трябва да отбележим, че той моделира само един отделен вегетационен цикъл за избраното за наблюдение растение. Отделните стъпки на жизнения цикъл са представени както следва:

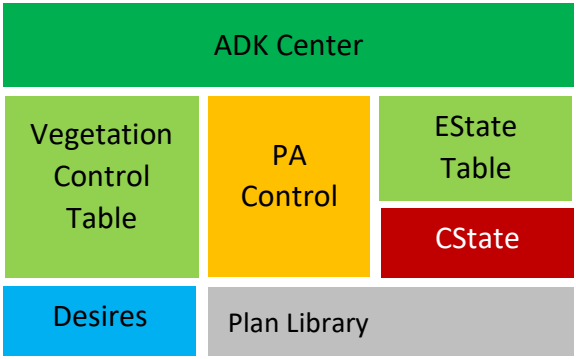
Инициализация – това е един много съществен етап в жизнения цикъл на ПА (в класическия жизнен цикъл просто се предполага, че съществува някаква начална вяра B_0). В този етап ПА се адаптира за конкретната земеделска култура, която ще бъде наблюдавана. ПА трябва да подготви вътрешните си структури, представящи неговата вяра. Необходимите за това знания и данни ПА трябва да извлече от ADK Center посредством подходящ интерфейс, означен в псевдокода с *consult()*. През този интерфейс ПА трябва да селектира необходимата му информация за конкретната земеделска култура *plant*. В този етап ПА трябва да инициализира също началната си цел – тя ще бъде планово домейн-събитие, моделиращо началния етап на вегетацията на наблюдаваното растение (*beginning_vegetation()*). Предполага се също, че вегетационният цикъл е започнал (*vegetation(plant)* ← *true*).

Идентификация на аномалии – в тази стъпка е представена работата на ПА за идентифициране на случващите се събития по време на протичане на един вегетационен цикъл на наблюдаваното растение (*vegetation(plant)* е логическа функция, която връща стойност true докато продължава вегетацията). Тук е важно да отбележим, че базирайки се на събитийния модел тази стъпка от жизнения цикъл трябва да се конфигурира в два аспекта - по отношение на плановите домейн-събития и съответно по отношение на инцидентните домейн-събития.

Превеждане на избрания план за изпълнение – поведението на ПА е двояко. Едното е поведение, направлявано от планови събития - в този случай ПА изпълнява планове, кореспондиращи с такъв тип събития

(*execute*(π_{pe})). Другото е поведение, направлявано от инцидентни събития - в този случай ПА изпълнява планове, кореспондиращи с такъв тип събития (*execute*(π_{ie})).

Базирайки се на жизнения цикъл, представен в предишната точка, тук ще представим архитектурата на ПА (Фиг.5.). ПА е централен компонент на платформата, който отговаря за организацията, управлението и контрола на сценариите и процесите, поддържани от ЗЕМЕЛА.



Фигура 5. Архитектура на ПА

Вяра на ПА. Вярата на персоналния асистент (неговата представа за околната среда) има сложна структура. Както беше обяснено, ПА използва A&A концепцията, съответно която всички нейни елементи се разглеждат като артефакти. Освен стандартните компоненти на платформата ЗЕМЕЛА (ADK Center, AML, Event Engines, Guards), като част от вярата са включени три важни структури, подпомагащи оперативната работа на ПА. Основополагащата управляваща структура е *Vegetation Control Table (VCT)*, която се генерира при инициализация на асистента. Тази структура изпълнява ролята на своеобразно разписание на процеса на очакваната вегетация на наблюдаваната земеделска култура. Отделните записи моделират отделните фази на вегетацията под формата на планови домейн-събития. Въз основа на тази структура ПА се ориентира за текущата фаза на вегетация на растението. Втората структура, наречена *EState Table* съхранява записи за очакваното състояние на растението за всяко събитие. Записите в тази таблица кореспондират със записите от VCT. Третата структура, позната като *CState* представя реалното състояние на растението за

настъпилото събитие. Тази структура е различна от другите две. Тя съхранява само текущото реално състояние на наблюдаваното растение и се променя динамично с напредването на вегетация във времето. При адаптирането си за определена земеделска култура референтният ПА инициализира VCT и EState Table. ПА извлича необходимата информация от знанията и данните, съхранявани в ADK Center. Веднъж инициализирани, тези две структури остават статични (те не променят съдържанието си). EState се актуализира в зависимост от данните, измерени от сензорите и/или наблюдавани от фермерите. Това се прави от гардовете на платформата. Работейки с трите структури, ПА успява да открие аномалии във вегетацията на наблюдаваните селскостопански култури. За текущата фаза на вегетацията сравнява очакваното с действителното състояние и при констатирани отклонения прави заключение за наличие на аномалии.

Опции на ПА. Опциите (намеренията) моделират способностите на ПА да обработва различни ситуации, възникващи в процеса на неговата работа. Типичните опции могат да бъдат *проследяване(VCT), информиране, предупреждение, предотвратяване, реагиране на аномалия*.

Цели на ПА. Една постоянна цел на ПА е проследяването на VCT. При откриване на аномалия ПА актуализира постоянната цел с такава, която осигурява той да се задейства за да избере план за нейното отстраняване. В зависимост от конкретната ситуация ПА избира от наличните опции актуална цел за постигане, с която се ангажира (целта е избрана и ангажирана опция). Например, ако ПА открие разлика между запис в CState Table и EState, тогава той може да генерира цел *реагиране на аномалия* или *предупреждение* - в противен случай може да генерира цел *информиране*.

PA Control. Този компонент реализира жизнения цикъл, направляващ двете поведения на ПА. След уточняване на настоящата цел ПА трябва да подготви подходящ план за действие. В нормалния ход на вегетацията, той може просто да информира фермера за стойностите на желани от него параметри. Когато се открие аномалия, има различни възможности. ПА поддържа собствена вътрешна библиотека с планове, от която може да избере напълно готов или да адаптира възможен план. Друга възможност е ПА да активира компоненти от платформата, напр. AML (Agent Communication Language) за изпълнение на определен модел.

ПЕТА ГЛАВА: ПРОГРАМНА РЕАЛИЗАЦИЯ НА ПРОТОТИП НА ПЕРСОНАЛЕН АСИСТЕНТ

JaCaMo [21] е рамка за мулти-агентно програмиране, която съчетава три отделни технологии, всяка от които е може да се използва самостоятелно. Рамката е разработвана в продължение на години, като сегашната версия е стабилизирана и пълноценна. JaCaMo е комбинация от следните три технологии: Jason, Cartago и Moise.

Актуалната версия на прототипа, адаптирана за откриване аномалии при вегетацията на домати, е представена в тази точка. Използваните данни са от района на гр. Пловдив. При стартиране на Персоналния Асистент (ПА), неговите желания или цели са представени в JaCaMo по следния начин:

```
/* Initial goals */

!communicate_farmer.
!monitor_plant.
!search_anomalies.
```

Целите са да наблюдава развитието на растението, да търси аномалии в неговата вегетация и своевременно да комуникира с фермера, да го уведомява за резултатите от работата си, и съответно да получава, и изпълнява нови фермерски желания. За реализиране на комуникацията с фермера, ПА създава артефакт с име ‘’ArtName’’, от тип ‘’gui.UserGUI’’, с празен списък от параметри и с ID ‘’ArtID’’, и след това се фокусира върху него:

```
+!communicate_farmer
   <- makeArtifact(ArtName,"gui.UserGUI",[],ArtID);
      focus(ArtID).
```

Така, извършвайки ‘’actions’’ върху този артефакт, те ще се отразяват на интерфейса на фермера и съответно фокусирайки се върху артефакта, ще приема получените сигнали от фермера. Целта за следене вегетацията на растението има план за постигане включващ две подцели:

```
+!monitor_plant
   <- !!track_heat_units;
      !track_tomato_phases.
```

а именно следене на топлинните единици и съответно на тяхна база предполагаемата фаза от вегетацията на домата. Концепцията за топлинни единици или degree days (DD) е въведена за първи път от Даниел Пътнам през 1868 г., американски учен по земеделие, който работи върху начини за подобряване на добивите от култури. Той отбеляза, че растежът и развитието на културите са тясно свързани с температурата и предложи идеята за използване на натрупаната топлина (измерена в градусни дни) като начин за прогнозиране на времето на различните етапи на растеж. Концепцията на Putnam за градусови дни оттогава е усъвършенствана и разширена от други учени и сега се използва широко в селското стопанство за проследяване на растежа и развитието на голямо разнообразие от култури, включително домати. DD измерват количеството топлина, натрупано с течение на времето. Те могат да бъдат изчислени по много различни начини. Най-простият начин за изчисляване на натрупването на DD е методът на простата средна стойност:

```
DD = (Tmin + Tmax)/2 - Tbase
Tmin = минимална дневна температура
Tmax = максимална дневна температура
Tbase = долен праг на развитие
```

База за доматите е 10 °C. Специфичните degree days необходими за различните етапи на растеж на доматените култури, могат да варират в зависимост от сорта, датата на засаждане и метеорологичните условия. Въпреки това, като цяло, следните приблизителни стойности често се използват като ориентир за доматени култури в климат като този на град Пловдив:

```
Germination (Покълване): 60 DD (degree days)
Seedling stage (Фаза на разсаждане): 175 DD
Vegetative growth (Вегетативен растеж): 275 DD
Flowering (Цъфтеж): 700 DD
Fruit set (Формиране на плод): 900 DD
Maturity (Зрялост): 1500 DD
```

Долу са дадени плановете в Jason за изчисление на топлинните единици:

```
+!track_heat_units
   <- .time(H,M,S) ; H=0 ; ?temperature(X0) ; X0=Max ; X0=Min ; .wait(3600000) ; ?temperature(X1) ;
  if (X1>X0) { X1=Max; } elif (X1<X0) { X1=Min; } else { true; } .wait(3600000) ; ?temperature(X2) ;
  if (X2>Max) { X2=Max; } elif (X2<Min) { X2=Min; } else { true; } .wait(3600000) ; ?temperature(X3) ;
  if (X3>Max) { X3=Max; } elif (X3<Min) { X3=Min; } else { true; } .wait(3600000) ; ?temperature(X4) ;
  if (X4>Max) { X4=Max; } elif (X4<Min) { X4=Min; } else { true; } .wait(3600000) ; ?temperature(X5) ;
  if (X5>Max) { X5=Max; } elif (X5<Min) { X5=Min; } else { true; } .wait(3600000) ; ?temperature(X6) ;
  if (X6>Max) { X6=Max; } elif (X6<Min) { X6=Min; } else { true; } .wait(3600000) ; ?temperature(X7) ;
  if (X7>Max) { X7=Max; } elif (X7<Min) { X7=Min; } else { true; } .wait(3600000) ; ?temperature(X8) ;
  if (X8>Max) { X8=Max; } elif (X8<Min) { X8=Min; } else { true; } .wait(3600000) ; ?temperature(X9) ;
  if (X9>Max) { X9=Max; } elif (X9<Min) { X9=Min; } else { true; } .wait(3600000) ; ?temperature(X10);
  if (X10>Max) { X10=Max; } elif (X10<Min) { X10=Min; } else { true; } .wait(3600000) ; ?temperature(X11) ;
  if (X11>Max) { X11=Max; } elif (X11<Min) { X11=Min; } else { true; } .wait(3600000) ; ?temperature(X12) ;
  if (X12>Max) { X12=Max; } elif (X12<Min) { X12=Min; } else { true; } .wait(3600000) ; ?temperature(X13) ;
  if (X13>Max) { X13=Max; } elif (X13<Min) { X13=Min; } else { true; } .wait(3600000) ; ?temperature(X14) ;
  if (X14>Max) { X14=Max; } elif (X14<Min) { X14=Min; } else { true; } .wait(3600000) ; ?temperature(X15) ;
  if (X15>Max) { X15=Max; } elif (X15<Min) { X15=Min; } else { true; } .wait(3600000) ; ?temperature(X16) ;
  if (X16>Max) { X16=Max; } elif (X16<Min) { X16=Min; } else { true; } .wait(3600000) ; ?temperature(X17) ;
  if (X17>Max) { X17=Max; } elif (X17<Min) { X17=Min; } else { true; } .wait(3600000) ; ?temperature(X18) ;
  if (X18>Max) { X18=Max; } elif (X18<Min) { X18=Min; } else { true; } .wait(3600000) ; ?temperature(X19) ;
  if (X19>Max) { X19=Max; } elif (X19<Min) { X19=Min; } else { true; } .wait(3600000) ; ?temperature(X20) ;
  if (X20>Max) { X20=Max; } elif (X20<Min) { X20=Min; } else { true; } .wait(3600000) ; ?temperature(X21) ;
  if (X21>Max) { X21=Max; } elif (X21<Min) { X21=Min; } else { true; } .wait(3600000) ; ?temperature(X22) ;
  if (X22>Max) { X22=Max; } elif (X22<Min) { X22=Min; } else { true; } .wait(3600000) ; ?temperature(X23) ;
  if (X23>Max) { X23=Max; } elif (X23<Min) { X23=Min; } else { true; }
  if (Min<10) { Min=10; } else { true; }
  if (Max>35) { Max=35; } else { true; }
  (Max+Min)/2-10=D ; -+heat_units(D) ; !track_heat_units.

-!track_heat_units
   <- .wait(3600000) ; !track_heat_units.
```

Агентът проверява температурата на всеки час и в края на деня установява минималната и максималната температура за деня, и съответно изчислява ‘’degree days’’ по формулата за този ден и актуализира натрупаните топлинни единици от началото на работата си.

При достигане на необходимите топлинни единици и предполагаемото влизане на растението в следваща фаза, ПА уведомява фермера със съобщение, че се предполага дадена фаза да е достигната:

```
+!track_tomato_phases
   <- ?heat_units(D) ;
   if (D>=0 & D<60) {  .wait(3600000); !track_tomato_phases; }
   elif (D>=60 & D<175) { println("It is assumed that the 'Germination' phase has been reached!"); }
   elif (D>=175 & D<275) { println("It is assumed that the 'Seedling' stage phase has been reached!"); }
   elif (D>=275 & D<700) { println("It is assumed that the 'Vegetative growth' phase has been reached!"); }
   elif (D>=700 & D<900) { println("It is assumed that the 'Flowering' phase has been reached!"); }
   elif (D>=900 & D<1500) { println("It is assumed that the 'Fruit set' phase has been reached!"); }
   elif (D>=1500) { println("It is assumed that the 'Maturity' phase has been reached!"); }
   .drop_intention(track_tomato_phases);
   .wait(86400000); !track_tomato_phases.
```

Персоналният асистент използва създадения GUI артефакт и чрез ‘’external action’’ (в случая ‘’println’’), принтира съобщението върху фермерския интерфейс.

Освен за натрупване на топлинни единици, Персоналният Асистент следи и за екстремни температурни стойности при които вегетационният процес е застрашен да бъде необратимо прекъснат:

```
+temperature(T) : heat_units(D) & D>=60 & D<175 & T<10 | T>35
   <- println("Attention, critical temperature in the 'Germination' phase!").

+temperature(T) : heat_units(D) & D>=175 & D<275 & T<10 | T>29
   <- println("Attention, critical temperature in the 'Seedling' phase!").

+temperature(T) : heat_units(D) & D>=275 & D<700 & T<10 | T>35
   <- println("Attention, critical temperature in the 'Vegetative growth' phase!").

+temperature(T) : heat_units(D) & D>=700 & D<900 & T<13 | T>32
   <- println("Attention, critical temperature in the 'Flowering' phase!").

+temperature(T) : heat_units(D) & D>=900 & D<1500 & T<13 | T>32
   <- println("Attention, critical temperature in the 'Fruit set' phase!").

+temperature(T) : heat_units(D) & D>=1500 & T<10 | T>35
   <- println("Attention, critical temperature in the 'Maturity' phase!").
```

Минималната и маскималната критична температура варират в различните фази, минималната за доматите обичайно е около тяхната база или около 10 градуса по Целзий, докато максималната достига до 32-35 градуса. В случая при надвишаване на тези градуси или падане под минималната критична, Персоналният Асистент предупреждава фермера отново чрез GUI артефакта. За изпълнение на целта ‘’търсене на аномалии’’ имаме две подцели:

```
!search_anomalies
   <- !!init_VCT;
      !init_EState;
```

При стартирането си агента има за цел да инициализира VCT (Vegetation Control Table) и EState (Estimated State Table), или Таблицата за контрол на растителността, и таблицата с прогнозното състояние на растението.
Планът за инициализация на VCT е:

```
+!init_VCT
   <- !create_VCT;
      getInstances(tomato_stages);
      for (.member(M, instances)) {rules.add_rule}.
```

Първата формула за изпълнение в тялото на плана е създаване на VCT артефакт. След това имаме изпълнение на ‘’action’’ (external), който представлява метод от кода на java артефакта със същото име:

```java
public class VCT extends Artifact {

    private OWLOntology ontology;

    void init(String ontologyPath) {
        // Load ontology from file
        try {
            OWLOntologyManager manager = OWLManager.createOWLOntologyManager();
            ontology = manager.loadOntologyFromOntologyDocument(getArtifactURI().resolve(ontologyPath).toURL().openStream());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    @OPERATION
    List<String> getInstances(String concept) {
        List<String> instances = new ArrayList<String>();
        OWLClass cls = ontology.getOWLOntologyManager().getOWLDataFactory().getOWLClass(IRI.create(concept));
        NodeSet<OWLNamedIndividual> individuals = reasoner.getInstances(cls, false);
        for (OWLNamedIndividual ind : individuals.getFlattened()) {
            instances.add(ind.getIRI().toString());
        }
        return instances;
    }
}
```

Методът ‘’init’’ зарежда онтология от файл с дадения му път. Методът ‘’getInstances’’ приема име на ‘’concept’’ като вход и връща списък на инстанциите, които принадлежат към този консепт. Той прави това, като първо извлича съответния обект OWLClass от онтологията, използвайки неговия IRI, и след това използва обект OWLReasoner, за да получи NodeSet на индивидите, които принадлежат към този клас. Накрая добавя IRI на лицата към списък и го връща.

Последната формула от тялото на плана обхожда всеки елемент от списъка и прилага върху него ‘’internal action’’, който в случая записва като правило (rule) в базата на вярванията на агента всяка фаза от вегетацията на доматите, като вярването, че сме в тази фаза е налично когато дясната страна на правилото е вярно, или когато са достигнати необходимите стойности на топлинните единици взети като данни от онтологията. Долу е показана своеобразно получената таблица за контрол на растителността в базата на вярвания на агента:

```
/* Initial beliefs and rules */

temperature(X).
heat_units(D).

germination :- D>=60 & D<175.
seedling :- D>=175 & D<275.
vegetative_growth :- D>=275 & D<700.
flowering :- D>=700 & D<900.
fruit_set :- D>=900 & D<1500.
maturity :- D>=1500.
```

Съответно тази VCT таблица служи като разписание за агента. При всяко достигане на нова фаза, тези домейн събития предизвикват изпълнение на планове, които съм показал по-долу.

Както Таблицата за контрол на растителността, по същия начин се справяме с целта за инициализиране на Таблицата с прогнозното състояние на растението (Estate):

```
+!init_EState
    <- !create_EState;
       getInstances(stages_attributes);
       for (.member(M, instances)) {+M}.
```

Създаваме EState артефакт, чрез метода ‘’getInstances’’ извличаме от онтологията атрибутите на всяка вегетационна фаза и обхождаме списъка, като всеки елемент от него добавяме като вярване в базата на вярванията на агента. В случая с доматите, едно такова вярване ще изглежда по следния начин:

```
phase(height,stem_diameter,leaves_number,fruit_presence,fruit_color).
```

Всяка фаза има пет определения, които описват състоянието на растението намиращо се в нея. За доматите те са височина на растението, диаметърът на стъблото, броят на листата, наличието на плодове и цветът на плодовете.

В етапа на поникване доматеното растение все още не е излязло от почвата, така че няма никаква височина. По време на покълването семето абсорбира вода и хранителни вещества от почвата, и ембрионът в семето започва да расте, и да се развива. След като стръкът излезе от почвата, започва етапът на разсаждане и растението започва да расте на височина. Съответно първите два параметри (височина и диаметър на стъблото) липсват във фазата на покълване. Броят на листата в тази фаза са 1-2 котиледона (семенни листа), а параметрите наличие и цвят на плодовете са празни.

Във фазата на разсаждане височината на растението е около 10-15 сантиметра, диаметърът на стъблото между 1 и 3 милиметра, броят на листата от 2 до 4, и празни параметри за наличие, и цвят на плодовете.
Във фазата на вегетативен растеж височината на растението достига до 60-90 сантиметра, диаметърът на стъблото между 3 и 5 милиметра, броят на листата от 5 до 7, и празни параметри за наличие, и цвят на плодовете.

Във фазата на цъфтеж височината на растението достига до 90-120 сантиметра, диаметърът на стъблото между 8 и 12 милиметра, броят на листата от 10 до 15, и празни параметри за наличие, и цвят на плодовете.

Във фазата на формиране на плод височината на растението е между 90-120 сантиметра, диаметърът на стъблото между 15 и 20

милиметра (или дори повече), броят на листата от 10 до 15, и налични плодове със зелен цвят.

Във фазата на зрялост височината на растението е между 90-120 сантиметра, диаметърът на стъблото между 25 и 50 милиметра, броят на листата от 5 до 10, и налични плодове с червен, жълт, оранжев или дори лилав цвят.

Текущото състояние (CState) представя текущото състояние на растението. Тази структура се различава от VCT и EState защото съхранява само текущото реално състояние на растението и се променя динамично с напредването на растителността във времето. Постъпващите данни от сензорите и обработени от гардовете, CState се ъпдейтва в базата на вярванията на агента като това на EState, но с динамични стойности.

```
cstate(height,stem_diameter,leaves_number,fruit_presence,fruit_color).
```

За да открие аномалиите, Персоналният Асистент водейки се по разписанието, във всеки един момент сравнява очакваното състояние за фазата в която понастоящем се намира с динамичните данни за сегашното състояние на растението.

```
+germination
   <- .findall(X,germination(X),E);
      .findall(Y,cstate(Y),C);
      compare.lists(E,C);
      println(result).

+seedling
   <- .findall(X,seedling(X),E);
      .findall(Y,cstate(Y),C);
      compare.lists(E,C);
      println(result).

+vegetative_growth
   <- .findall(X,vegetative_growth(X),E);
      .findall(Y,cstate(Y),C);
      compare.lists(E,C);
      println(result).

+flowering
   <- .findall(X,flowering(X),E);
      .findall(Y,cstate(Y),C);
      compare.lists(E,C);
      println(result).

+fruit_set
   <- .findall(X,fruit_set(X),E);
      .findall(Y,cstate(Y),C);
      compare.lists(E,C);
      println(result).

+maturity
   <- .findall(X,maturity(X),E);
      .findall(Y,cstate(Y),C);
      compare.lists(E,C);
      println(result).
```

За всяка от фазите извлича параметрите от вярванията и ги вкарва в списък. Същото прави с параметрите от вярването за настоящото състояние на растението, сравнява двата списъка чрез ‘’internal action’’ написан на java и принтира резултатите на фермерския интерфейс.

ЗАКЛЮЧЕНИЕ

Резултатите от изследването, проведено в рамките на дисертацията, могат да бъдат обобщени в както следва.

Събитиен модел. Представена е нова версия на събитийния модел. Тя запазва основните постановки, представени в предишната версия. Новите неща са свързани с прецизиране на дефинициите на основните операции със събития по отношение на техните алгебрични свойства. Мотивирани са тяхната некомутативност, както лявата, съответно дясната им асоциативност. Неопределена остава за момента дистрибутивността. Въведена е съвсем нова концепция за абстрактна събитийна машина, вече като част от събитийния модел. Предлага се тя да се формализира като клетъчен автомат. Предстоят следващи задачи, свързани с практическата реализация на абстрактната машина.

Платформата ЗЕМЕЛА. Представена е нова версия на архитектурата на платформата ЗЕМЕЛА. В новата версия е предложена нова структура на хранилището за специализирани знания и данни ADK Center, интегрираща компонента ViSCoD от предишната версия. Опростена е структура на специализираната библиотека за модели AML. В новата версия се предлага използване основно на подхода DEVS. Анализите от получените до момента резултати показва, че той може да бъде резултатен за целите на платформата.

Персонален асистент. Разработена е референтна архитектура на персонален асистент, предназначен за подпомагане на земеделски стопани, работещи в условията на интелигентно земеделие. Персоналният асистент е със специализирано предназначение, базиран на специфичен събитиен модел и използващ хранилище със специализирани знания и данни. В представената в дисертацията версия персоналният асистент е адаптиран за интелигентно земеделие. Представена е също прототипна имплементация на асистента, реализирана посредством развойната рамка JaCaMo.

Съществуват различни възможности за бъдещи разработки и проекти, свързани с темата на дисертацията, като напр. разширяване функционалност на ПА, машинно учене в ПА, взаимодействие на ПА с технологични вериги, „сваляне надолу“ интелигентността на платформата, реинженеринг на архитектурата на платформата и преминаване към „чиста“ агентно-ориентирана версия, разгръщане на ПА върху различни устройства.

СПИСЪК НА ЦИТИРАНИТЕ В ДИСЕРТАЦИЯТА

ПУБЛИКАЦИИ

на Иван Станимиров Стоянов,
докторант към катедра „Компютърни системи“

1. S. Stoyanov, J.Todorov, I. Stoyanov, V. Tabakova-Komsalova, L. Dukovska, ZEMELA – An Intelligent Agriculture Platform, Big Data, Knowledge and Control Systems Engineering – BdKCSE'2021, 28–29 October 2021, Sofia, Bulgaria.
2. S. Stoyanov, I. Stoyanov, V. Tabakova-Komsalova, A. Dukovski, L. Doukovska, An Event-Based Platform Supporting Smart Agriculture Applications. 2022 IEEE 11th International Conference on Intelligent Systems (IS), Warsaw, Poland, 2022, pp. 1-5, doi: 10.1109/IS57118.2022.10019674.

БЛАГОДАРНОСТИ

Искам да изразя своята благодарност към личностите, без които този дисертационен труд не би бил възможен.

Огромна благодарност дължа на научния си ръководител, проф. д-р Ася Стоянова-Дойчева, за доверието в мен и всеотдайността във всяка ситуация, идеите и съветите, които оценявам изключително много.

Изказвам специална благодарност към чл. кор. проф. дн Любка Дуковска и гл. ас. д-р Венета Табакова-Комсалова за позитивната енергия и сърдечно отношение, ценните насоки и безрезервна помощ.

Благодаря на целия колектив, изпълняващ Националната научна програма „Интелигентно земеделие“, за помощта и приятелската творческа обстановка, в която беше проведено на научното изследване, представено в този дисертационен труд.

РЕФЕРИРАНА ЛИТЕРАТУРА

[1] Национална програма „Интелигентно растениевъдство 2021-2023“, 2020.

[2] W. Wolf. Cyber-physical systems. Computer, 42(3):88-89, March 2009.

[3] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. Comput. Netw., 54:2787-2805, Oct. 2010.

[4] S. Stoyanov, T. Glushkova, E. Doychev, A. Stoyanova-Doycheva, V. Ivanova, Cyber-Physical-Social Systems and Applications. Part I: Reference Architecture,, LAP LAMBERT Academic Publishing, 2019.

[5] Е. Дойчев, Среда за електронни образователни услуги, дисертация, Пловдивски университет „Паисий Хилендарски“, 2013.

[6] S.Stoyanov, V. Valkanov, I. Popchev, A. Stoyanova-Doycheva, E. Doychev, A Model of Context-Aware Agent Architecture, Compt. Rend. Acad. Bulg. Sci., Tome 67, № 4, 2014, pp. 487-496.

[7] S. Stoyanov, Context-Aware and Adaptable eLearning Systems, PhD Thesis, STRL, De Montfort University, Leicester, UK, 2012.

[8] FaST Platform, https://fastplatform.eu/.

[9] Farm21, https://www.farm21.com/farming-platform-to-farm-smarter/

[10] FAO, https://www.fao.org/platforms/common-practices/en.

[11] GPT-4, https://www.bbc.com/news/technology-64959346.

[12] B. P. Zeigler, A. Muzy, E. Kofman, Modeling and Simulation. Discrete Event and Iterative System. Computational Foundations, Elsevier, Academic Press, 2019.

[13] S. Stoyanov, J. Todorov, I. Stoyanov, V. Tabakova-Komsalova, L. Dukovska An Event-Based Platform Supporting Smart Agriculture Applications, ZEMELA – An Intelligent Agriculture Platform (BdKCSE2021).

[14] S. Stoyanov, I. Stoyanov, V. Tabakova-Komsalova, A. Dukovski, L. Doukovska, An Event-Based Platform Supporting Smart Agriculture Applications. 2022 IEEE 11th International Conference on Intelligent Systems (IS), Warsaw, Poland, 2022, pp. 1-5, doi: 10.1109/IS57118.2022.10019674.

[15] A. Stoyanova-Doycheva, E. Doychev, V. Ivanova, V. Valkanov, V. Tabakova-Komsalova, Event Ontology about Wheat Cultivation, AgriControl 2022The 7th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture, 14-16 September 2022, Munich, Germany.

[16] S. Stoyanov, V. Tabakova-Komsalova, L. Doukovska, I. Stoyanov & A. Dukovski, An Event-Based Platform Supporting Smart Agriculture

Applications, IEEE Intelligent Systems IS’22, Warsaw, Poland, October 12-14, 2022.
[17] К. Граматова, Изграждане на виртуално образователно пространство като екосистема в Интернет на нещата, дисертация, Пловдивски университет „Паисий Хилендарски“, 2017.
[18] P. Yochkova, V. Tabakova-Komsalova, S. Cherecharov, L. Doukovska, S. Stoyanov, DEVS Modeling of an Irrigation System, IEEE Intelligent Systems IS’22, Warsaw, Poland, October 12-14, 2022.
[19] Ricci A, Piunti M, Viroli M, Omicini A. 2009. Environment programming in CArtAgO. In: El Fallah Seghrouchni A, Dix J, Dastani M, Bordini R, eds. Multi-agent programming. Boston: Springer, 259–288 DOI 10.1007/978-0-387-89299-3_8.
[20] Иван Стоянов, Станимир Стоянов, Ирина Кръстева, Персонален туристически екскурзовод ‚Ловеч и региона‘, Международна научна конференция „Tech-Co” 2020, 17-18 юли, Ловеч (https://www.tugab.bg/images/tk-lovech/konferencia/Techko-Lovech-2020-w.pdf).
[21] Boissier, O., Bordini, R., Hübner, J., Ricci, A. 2020. Multi-Agent Oriented Programming Programming. Multi-Agent Systems Using JaCaMo, The MIT Press Cambridge, Massachusetts London, England.