

## ИГРОВО-БАЗИРАНО ОБУЧЕНИЕ В ИНТЕГРИРАНИ ДОМЕЙНИ С ПРИЛОЖЕНИЕ ЗА STEAM ЦЕНТРОВЕ

Стефани Паунова, Виляна Дойчевска, Валентин Йорданов, Станимир Стоянов  
Пловдивски университет „Паисий Хилендарски“, Пловдив

## GAME-BASED LEARNING IN INTEGRATED DOMAINS WITH AN APPLICATION FOR STEAM CENTERS

Stefani Paunova, Vilyana Doychevska, Valentin Yordanov, Stanimir Stoyanov  
Plovdiv University „Paisii Hilendarski“, Plovdiv

\* Авторите изказват благодарност към научния проект СП21-ФМИ-002 „ViSCoD – среда за моделиране на системи за интелигентно земеделие“ към Фонд «Научни изследвания» при Пловдивския университет „Паисий Хилендарски“ и към Национална научна програма „Интелигентно земеделие“ (2021-2024).

**Abstract:** The article presents a game-based learning environment that is being developed as a component of the ATOS platform (the platform is an article described for the same conference). The proposed environment is an adaptation of a component for serious games, developed within two projects – the National Research Program „Intelligent Agriculture (2021-2024)“ and the university project „ViSCoD – environment for modeling systems for intelligent agriculture“. The opportunities for game-based learning in integrated domains (in this case intelligent agriculture – STEAM center) are demonstrated with a version of the well-known game „Twenty question game“, adapted for the Bulgarian flora. The system is fully implemented in the logic programming language Prolog and thus can also be used to teach students in the discipline „Artificial Intelligence“.

**Keywords:** game-based learning, STEAM, intelligent agriculture, logic programming, Prolog.

**Увод.** От една страна, през последните десетилетия, с големите пробиви в науката, технологиите и инженерството, хората все по-интензивно взаимодействат с околния физически свят. Съвременните технологии като Кибер-физични системи (CPS) [1]) и Интернет на нещата (IoT) [2]) 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 [3]. Кибер-физическите системи тясно свързват и координират изчислителните (кибер) и физическите ресурси – т.е. системи, поддържащи тясна интеграция между изчисления, комуникация и контрол, взаимодействайки си със средата, в която са разположени. От друга страна, човечеството е стигнало до точката, в която социалната и човешката динамика трябва да се счита за неразделна част от всяка ефективно работеща кибер-физическа система. Така вграждането на социалното в CPS е напълно основателно и оправдано – така възниква понятието Cyber-Physical-Social System (CPSS) [4]. Тази промяна има и философско измерение, което привежда CPSS в съответствие с теорията за реалността на К. Попър, която твърди, че нашият универсум се състои от три взаимодействащи си свята – физически, ментален и изкуствен. Преди да съществува Интернет, нашето жизнено пространство беше ограничено до физическия свят, включващ предимно физическия и менталния светове. Днес, с наличието на кибер-пространство, в нашето жизнено пространство се включва и изкуствения свят. Изкуственият интелект, усилен с технологиите IoT, CPS и CPSS се превръща в новия фокус на граничните изследвания на световната индустрия, академичните среди и правителствените агенции. Така напр., националният научен фонд на САЩ определи CPS като ключова област на научните изследвания, очаквайки кибер-физични системи да окажат изключително техническо, икономическо и социално въздействие върху начина на живот в близко бъдеще [5]. През последното десетилетие основни финансови агенции по целия свят спонсорират изследователски проекти за изграждане на кибер-физични системи в различни области.

В последните години увеличаващ се научен и практически интерес представляват CPSS в сел-

ското стопанство. В България перспективите за интелигентно земеделие са формулирани в . Идеята за прецизно и интелигентно земеделие възниква поради усилията за получаване на биологично чисти и в същото време евтини селскостопански продукти, минимизиране на натоварването на околната среда с пестициди и ранно откриване на болести по културите. За да разрешат тези проблеми, хората разчитат особено на прилагането на съвременни информационни и комуникационни технологии. Особено подходящи са тези, които позволяват анализ и вземане на решения, отчитайки околния физически свят. Трите относително нови и взаимосвързани технологии – IoT (Internet-of-Things), CPS (Cyber-Physical Systems) и CPSS (Cyber-Physical-Social Systems) – предоставят разнообразие от немислими възможности за изграждане на инфраструктури за интелигентно и прецизно земеделие.

Във Факултета по математика и информатика на Пловдивския университет е разработена референтна архитектура ViPS (Virtual Physical Space), която може да бъде адаптирана за изграждане на CPSS приложения в различни области [6, 7]. Адаптирането на ViPS за интелигентно земеделие е предвидено в Националната научна програма (ННП) „Интелигентно земеделие (2021–2024)“. Една от целите на ННП е да предизвика интерес и привлича млади изследователи (студенти, ученици) към идеите на интелигентното земеделие. В изпълнение на тази задача ViPS се разширява с нов компонент за моделиране на сценарии и процеси в интелигентното земеделие, наречен ViSCoD. Едно от приложенията на ViSCoD е да се използва за игрово-базирано обучение на студенти по информатика и селско-стопански науки. Компонентът е имплементиран напълно на езика за логическо програмиране Пролог [8]. Пролог е език за програмиране основаващ се на логика от първи ред, който в последно време става все по-актуален за моделиране на системи с интелигентно поведение.

Освен за обучение на студенти ViSCoD ще се използва и за обучение на ученици като стане интегрална част на платформата ATOS [9]. ATOS се разработва за постигане на споделено използване учебните ресурси в университета и в средното училище (по-специално в STEAM центрове). Компонентът за игрово-базирано обучение е имплементиран напълно на езика за логическо програмиране Пролог. Използването му в средното училище може да се комбинира с обучението на ученици в дисциплината „Изкуствен интелект“ като разширение на примерите от учебника [10].

**Кратък преглед на ViSCoD.** Най-общо ViSCoD е среда за моделиране на процеси, сценарии, технически средства и системи, които могат да бъдат елементи от бъдеща инфраструктура за интелигентно земеделие. Средата ще доставя спецификации и поддържащи програмни средства за представяне и интерпретиране на реални процеси от интелигентното земеделие. Съпътстваща цел е средата да се използва за различни форми на електронно обучение (по-специално игрово-базирано обучение), основно в дисциплините изкуствен интелект, биология, селско стопанство. ViSCoD се разработва като система, използваща знания (експертна система) с архитектура, съставена от следните базови компоненти (Фиг. 1.):



Фигура1. Архитектура на ViSCoD

- База знания – разпределено хранилище на знанията за проблемната област в някакво подходящо представяне. Базата знания се състои от два базови компонента – библиотека „Модели“ и игротеката. Библиотеката съхранява знания, моделиращи различни сценарии и процеси от интелигентното

земеделие. Игротеката съхранява игри, които могат да се използват в игрово-базирано обучение.

- Машина за извод – аналогично на базата знания този модул се състои от две части. Симулаторът е вид машина за извод, която може да интерпретира съхраняваните в библиотеката модели и знания, както и външни знания. Геймърът е вид интерпретатор с възможности за активиране и управление на избрана игра.
- Обяснителен компонент – може да обяснява на потребителите начина, по който системата е достигнала до резултатите от заявките към системата или от проведената игра.
- Потребителски интерфейс – служи за връзка между потребителя и системата – обикновено за визуализация на резултатите от обработката на модела или от актуално провеждащата се игра.

ViSCoD се програмира напълно на езика за логическо програмиране Пролог. За представяне на знанията за интелигентното земеделие се използват различни форми, като напр. правила, фрейми и семантични мрежи. Също, използваме общи средства за инженеринг на знанията. Обикновено такива средства могат да бъдат разработване на така нареченото мета-ниво, т.е. обект на манипулация е структурата на базата знания, а не ефектът на прилагане на самите знания от базата знания. В този смисъл различаваме две нива на разглеждане – обектно ниво и мета-ниво.

Платформата ATOS. Идеята за изграждане на платформата ATOS възниква във връзка с изследванията на така наречените интегрирани домейни. Както беше споменато ViPS е адаптиран за различни приложения области. Пространството включва стандартни компоненти (обикновено това са формални системи за моделиране на пространствените, времевите и събитийните аспекти на физическите обекти), които се адаптират посредством специфични за домейна (приложната област) структури данни. Освен това, в определени случаи е наложително разработването на нови строго специфични за домейна компоненти. Новите компоненти се съхраняват в специализирани библиотеки, които стават съставна част на ViPS. Така, при всяка адаптация референтната архитектура се разширява и обогатява с нови компоненти. Същевременно, в определени случаи би било целесъобразно ViPS да реферира не един единствен, а повече домейни. Така възниква идеята за търсене на възможности за работа в интегрирани домейни. В един интегриран домейн особено интересна става ролята на персоналния асистент. Той може да бъде „споделения“ или „интегриращия“ компонент, предлагащ комплексни услуги, които от своя страна са композирани от специфични за различни домейни такива. Пример за персонален асистент, опериращ в интегриран домейн, може да бъде персонален туристически екскурзовод в условията на интелигентен град.

Идеята с разработването на ATOS е създаване на интегриран домейн за различни форми на електронно-базирано обучение (игрово-базирано, проектно-ориентирано, проблемно-ориентирано, интегриращо) между университет и STEAM училищни центрове.

Прототипната версия на технологичната платформа включва два базови компонента:

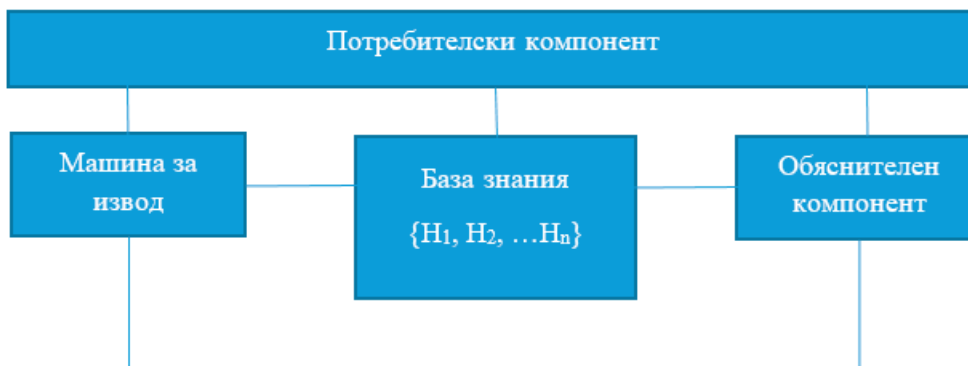
- Back-end – образува се от адаптацията на ViPS за електронно обучение в университет (по-специално в Пловдивския университет „Паисий Хилендарски“) и от средата за моделиране на интелигентно земеделие ViSCoD (по-специално игрово-базирания модул).
- Front-end – персонален асистент, разположен върху робот от трето поколение Robobo (<https://theroboboproject.com/en/>).

**Игрово-базирано обучение в ATOS.** Първата игра, която предлагаме, е версия на известната игра „Twenty question game“. Това е игра, която насърчава дедуктивни решение и креативност. Произхожда от САЩ през 19-ти век, но популярността и нараства през 40-те години на миналия век, когато започва масово да се играе по радио-викторини. В традиционната игра се избира един играч, който да отговаря. Този човек избира субект (обект), но не разкрива това на останалите. Всички останали играчи са питащи. Всеки от тях се редува да зададе въпрос, на който може да се отговори с просто „да“ или „не“. Във вариантите на играта могат да бъдат включени множество различни отговори, дори отговора „може би“ е позволено да бъде включен. Отговарящият отговаря поотделно на всеки въпрос. Лъжата не е разрешена в играта. Ако питащият отгатне верния отговор, той печели и става отговорът за следващия кръг. Ако 20 въпроса са зададени без правилно предположение, тогава отговорилият е успял да надхитри питащите и той печели рунда и има правото да задава нова дума за следващото предложение. Най-популярната версия на тази игра се казва „Животно, Растение, Минерал“. Като при този вариант на играта отговарящият казва към коя категория принадлежи дадената дума.

Как може да се използва играта в платформата ATOS? Възможни са различни сценарии. Тук ще представим два възможни сценария. В първия учениците могат да проверяват хипотеза за определен зеленчук, отговаряйки на въпросите на играта. В този сценарий университетският домейн изпълнява ролята на

„обучаващ“, а училищният домейн – на „обучаем“. Работейки в STEAM центъра (напр., зеленчукова градина) ученикът може да формулира определена хипотеза. Той иска да провери дали тази хипотеза е вярна. Персоналният асистент на ученика, взаимодействайки с оперативните асистенти на ATOS, които осъществяват достъп до игротеката на ViSCoD, инициализира диалог между ученика и ViSCoD. По време на диалога играта „Познай растението“ разпитва ученика за характерните особености на предложената от него хипотеза. Като резултат ViSCoD потвърждава или отхвърля предложената хипотеза.;

Архитектурата на версията на играта, поддържаща първия сценарий, е представена на Фиг. 2. Характерните особености на растенията са представени като отделни хипотези.



Фигура 2. Архитектура първа версия на играта

В машината за извод се представя определена хипотеза, която тя се опитва да потвърди или отхвърли като инициира диалог с потребителя, следвайки правилата за идентификация на растенията. В малкия сегмент от базата знания на играта (Фиг. 3.) са специфицирани две хипотези – едната е за „домат“, а втората за „пипер“. При стартиране на играта геймърът (машината за извод) последователно се опитва да потвърди определена хипотеза, използвайки съдържанието на базата знания и отговорите на играещия. Когато еднозначно се определи една хипотеза геймърът автоматично не опитва друга възможност, освен ако потребителят явно не търси алтернатива посредством ново активиране на процеса на търсене. За диалог с играещия е програмиран опростен потребителски интерфейс (част от него е показана също на Фиг. 3.).

```

% hypotheses to be tested
hypothesize(домат) :-
    домат,!.
hypothesize(пипер) :-
    пипер,!.
...
% plant identification rules
домат :-
    картофови,
    зеленчук,
    verify(плод_двугнезна_или_многогнезна_ягода),
    verify(тегло_1_2_до_500_гр).
пипер :-
    картофови,
    зеленчук,
    verify(първо_място_сред_зеленчуците_по_съдържание_витамин_с),
    verify(обикновено_лют).
...
% How to verify something
verify(S) :-
    (yes(S)-> true ;
    (no(S)-> fail ;
    ask(S))).
% how to ask questions
  
```

```
ask(Question) :-
  write('Има ли растението следните атрибути: '),
  write(Question),write('? '),
  read(Response),nl,
  ( (Response == yes ; Response == y)->
  assert(yes(Question));
  assert(no(Question)), fail).
```

*Фигура 3. Програмен код на първата версия на играта*

Следният пример демонстрира примерен рунд на играта:

?- go.

Има ли растението следните атрибути: типично\_етноботанически? **yes.**

Има ли растението следните атрибути: пет\_срастнати\_венчелистчета? |: **yes.**

Има ли растението следните атрибути: използвано\_за\_салата? |: **yes.**

Има ли растението следните атрибути: плод\_двугнездна\_или\_многогнездна\_ягода? |: **yes.**

Има ли растението следните атрибути: тегло\_1\_2\_до\_500\_гр? |: **yes.**

Предполагам, че растението е: **домат**

true.

Във втория сценарий ролите са разменени – училището е „обучаващ“, а университета е „обучаем“. В тази версия на играта учениците могат да обучава играта, провеждайки диалог с нея. В този случай се използва версията на играта, имплементираща разширяващи се дървовидни структури. Ако, според ученика играта не върне правилния отговор, за какво растение става въпрос в случая, геймърът динамично разширява дървото на извод, като замества положителния отговор с нов въпрос и съответните отговори. Така дървото на решенията расте и програмата става „по-интелигентна“.

Примерът (Фиг. 4.) започва с просто дърво с възел-въпрос и два клона, всеки сочещ към различно растение. Играта отгатва погрешно, че предполагаемото растение е „домат“. Тогава кореспондиращият лист на дървото се заменя с нов въпрос, предложен от играещия (обучаващия).

?- go.

Розов плод ли има? да.

Това е домати? |: не.

За какво растение мислите?

|: петуния.

Какъв да/не въпрос, който го различава?

|: 'За ядене?'.

За петуния отговорът е: |: не.

Нова игра? |: да.

Розов плод ли има? |: да.

За ядене? |: да.

Това е домати? |: да.

Нова игра? |: да.

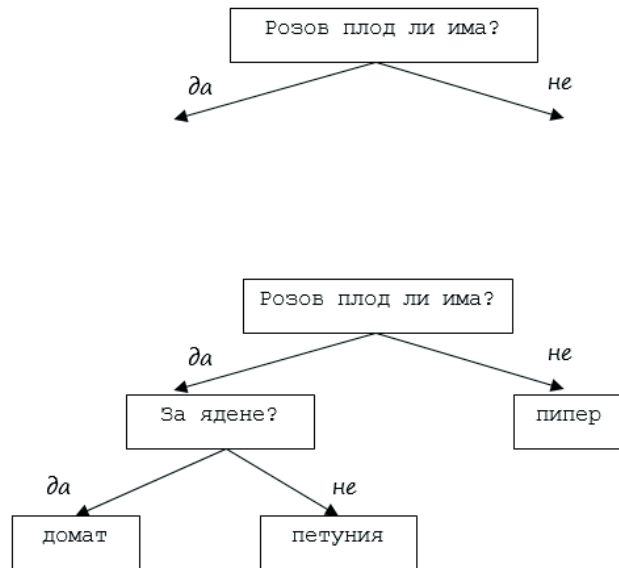
Розов плод ли има? |: да.

За ядене? |: не.

Това е петуния? |: да.

Нова игра? |: не.

**true.**



Фигура 4. Версия на играта с разширяващо се дърво

**Заклучение.** В статията е представена среда за игрово-базирано обучение, която може да се използва в платформата ATOS. Средата ViSCoD се използва като игротейка, разположена върху съвърнатата част на платформата. При интегриране на ViSCoD в ATOS трябва да се отчетат някои особености. Първият проблем е свързан с това, че ViSCoD е имплементирана напълно в Пролог, докато ATOS е агентно-ориентирана система, имплементирана на JaCaMo [11]. Тази интеграция не е сложна поради това, че менталният модул на агентите използва синтаксиса на Пролог.

Вторият проблем е свързан с усъвършенстване на потребителския интерфейс, така че да стане по-атрактивен за потребителите. Този проблем също е сравнително лесен за решаване, понеже използваната версия на Пролог поддържа двустранни интерфейси към езиците Java и C++.

В момента игротейката предлага само описаната в статията игра. В бъдеще е необходимо да се планира нейното разширяване с нови игри. Също би било смислено да се мисли за игри, в които едновременно да участват повече участници, включително и смесено ученици и студенти.

#### References:

1. W. Wolf. Cyber-physical systems. *Computer*, 42(3):88{89, March 2009.
2. L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. *Comput. Netw.*, 54:2787-2805, Oct. 2010.
3. Jianhua Shi, Jiafu Wan, Hehua Yan, and Hui Suo. A survey of cyberphysical systems. In *IEEE WCSP'11*, 1-6, 2011.
4. 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.
5. US National Science Foundation, „Cyber-Physical Systems (CPS),“ NSF 08-611, 2008, <http://www.nsf.gov/pubs/2008/nsf08611/nsf08611.htm>.
6. 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.
7. T. Glushkova, A. Stoyanova-Doycheva, V. Ivanova, S. Stoyanov, E. Doychev, *Cyber-Physical-Social Systems and Applications. Part II: Applications*, LAP LAMBERT Academic Publishing, 2019.
8. W. F. Clocksin, C. S. Mellish, *Programming in Prolog*. Berlin; New York: Springer-Verlag, 2003, ISBN 978-3-540-00678-7.
9. I. Krasteva, Y. Todorov, I. Stoyanov, *Inteligentno zemedelie v STEAM tsentrove*, XII Nauchno-prakticheski forum „Inovatsii v obuchenieto i poznavatelното razvitie“, 25 – 27.08.2021, gr. Burgas (prieta za pechat).
10. S. Stoyanov, T. Glushkova, R. Papancheva, *Izkustven intelekt. Predstavяnе na znaniyata chrez logika. Logichеско програмirane*, Izdatelstvo „Izkustva“, Sofia, 2021.
11. O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, *Multi-Agent Oriented Programming Programming. Multi-Agent Systems Using JaCaMo*, The MIT Press, Cambridge, Massachusetts, London, England, 2020.