

MODELING AND DEVELOPMENT OF A MULTI-AGENT SPACE FOR THE SECONDARY SCHOOL

I. K. Krasteva¹, T. A. Glushkova¹, S. N. Stoyanov¹

¹ *University of Plovdiv "Paisii Hilendarski", Bulgaria*
4000 Plovdiv, 24 Tzar Asen St., Bulgaria

Abstract

One of the guiding principles of the Fourth Industrial Revolution is the need for lifelong learning. This determines the growing role of intelligent educational systems to provide the necessary learning resources and services to users at any time and any place. This article presents the modeling and development of an intelligent multi-agent learning environment for the secondary school, developed by a team of the DeLC laboratory at the University of Plovdiv "Paisii Hilendarski", Bulgaria. The learners are placed at the focus of the environment by personal assistants supporting work with the environment.

Keywords: cyber-physical systems, cyber-physical-social space, virtual-physical space (ViPS), IoT, blockchain.

DOI: 10.32517/0234-0453-2020-35-4-53-62

For citation:

Krasteva I. K., Glushkova T. A., Stoyanov S. N. Modeling and development of a multi-agent space for the secondary school. *Informatika i obrazovanie — Informatics and Education*, 2020, no. 4, p. 53–62.

Received: February 27, 2020.

Accepted: April 21, 2020.

Acknowledgments

This research was supported by the Fund Scientific research, the University of Plovdiv "Paisii Hilendarski", contract number MY19-ΠΦ-023.

We acknowledge the provided access to the e-infrastructure of the Centre for Advanced Computing and Data Processing, with the financial support by the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014–2020) and co-financed by the European Union through the European structural and Investment funds.

About the authors

Irina Krasimirova Krasteva, Ph.D. student at the Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", Bulgaria; irina_krasteva@uni-plovdiv.bg; ORCID: 0000-0003-0842-8143

Todorika Atanasova Glushkova, Ph.D., Associate Professor at the Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", Bulgaria; glushkova@uni-plovdiv.bg; ORCID: 0000-0002-6243-9364

Stanimir Nedyalkov Stoyanov, Ph.D., Professor at the Faculty of Mathematics and Informatics, University of Plovdiv "Paisii Hilendarski", Bulgaria; stani@uni-plovdiv.net; ORCID: 0000-0002-3854-4260

1. Introduction

The Fourth Industrial Revolution is based on the rapid advancement of digital technologies, which are becoming increasingly complex and integrated, launching a world where virtual and physical systems collaborate in a flexible and global way [1]. In the last few years, Artificial Intelligence has made impressive progress, driven by ever-increasing rates of computing power. These changes are already being made in many aspects of public life, including education.

Modern development is defining the increasingly important role of pervasive computing and Internet of Things (IoT). Cyber-physical systems (CPS) based on them realize the collaboration and coordination between the virtual and the physical world [2]. CPS is identified as a key area of research that is expected to have a significant technical, economic, and social impact on the way we live and interact with the world around us. As the social factor becomes increasingly important in the implementation of such systems, the concept of cyber-physical and social systems (CPSS) is defined in [3]. The creation of these systems is largely

due to the unprecedented expansion of cyberspace and its profound impact on how people behave and interact. With the introduction of social elements in CPS, new scientific and practical challenges are emerging to build appropriate infrastructures and systems. At the same time, cyberspace is expected to stimulate the emergence of new methods and systems to solve complex and challenging problems in the real world that have not yet found their effective solutions.

The idea of using intelligent agents to assist people in their daily lives, work, and training [4] is not new, but the rapid development of artificial intelligence over the past few years has created the preconditions for building intelligent systems that can autonomously perform tasks on behalf of the users and in their favor. Nowadays, these assistants are usually deployed on mobile devices, can use social networks as a resource, and can be self-educated and improved.

One of the guiding principles of the Fourth Industrial Revolution is the need to educate people throughout their lives. This defines the growing role of smart learning systems to provide the necessary learning resources and services to consumers at any time and

place. These learners should be placed at the center of the educational space and communicate with it dynamically through their personal assistants.

Experience in the development of personal assistants worldwide is increasing. In the United States, PAL (Personalized Assistant that Learns) [5] is one of the first research programs in the field of cognitive systems, the purpose of which is to radically improve the way people interact with digital devices. One of the most interesting systems developed in PAL is the Cognitive Assistant CALO (Cognitive Assistant that Learns and Organizes) [6]. In Europe, in the Horizon 2020 program, the COMPANIONS Intelligent Assistants Development Project [7] aims to change the traditionally perceived way of communicating between humans and computers. The virtual “Companion” is an agent that interacts for long periods of time with the user while “creating” friendly relationships and “exploring” the person’s preferences and desires. Corporate developments for personal assistants to help people in their daily lives include, for example, Siri [8], Microsoft Cortana [9], Google Now [10], Google Assistant, Google Home, etc.

Over the last decade, the DeLC (Distributed eLearning Center) team at the University of Plovdiv has been working to create systems for e-learning and distance training for school and university learners. The first version of the DeLC system [11] involves the development of clusters of eLearning portals that share their resources and provide appropriate services to different groups of users — students, teachers, lecturers, administration, etc. The Virtual Education Space (VES) is the successor to DeLC. VES maintains the internationally accepted e-learning standards SCORM 2004 and QTI 2.1, taking into account the current trend in the development of Internet technologies [12]. Although

VES largely satisfies the requirements for conducting e-learning and distance learning, we have found that the integration between the virtual and the physical world, in which the real learning process de facto takes place, is not satisfactory. Recently, the Internet has been moving from a computer network to the Internet of Things, i. e. towards closer integration with the physical world. This identified the need to develop a new cyber-physical e-learning system, centered on consumers through their personal assistants.

2. Virtual Physical Space ViPS

The Virtual physical space (ViPS) is the ecosystem of the Internet of Things [13]. The ViPS architecture is being built as a reference architecture that can be adapted for different CPSS applications, as well as in various application areas such as smart agriculture, tourism, smart cities, and also in education [14]. The essential aspects of ViPS are the following:

- Consumers are the focus of attention;
- Physical “things” are virtualized;
- Integration between the virtual and the physical worlds.

The ViPS architecture presented in Figure 1 reflects and presents an identical model of the real physical world in the digital world. Since ViPS is essentially a CPSS environment, the user is placed in the center of the space through his / her personal assistant. ViPS provides virtualization of real objects. Modeling “things” that provide a link between the physical world and the digital world of the Internet requires that factors such as events, time, space, and location be taken into account.

Two subspaces are formed in the architecture of the space. The first one is the Analytical Subspace which

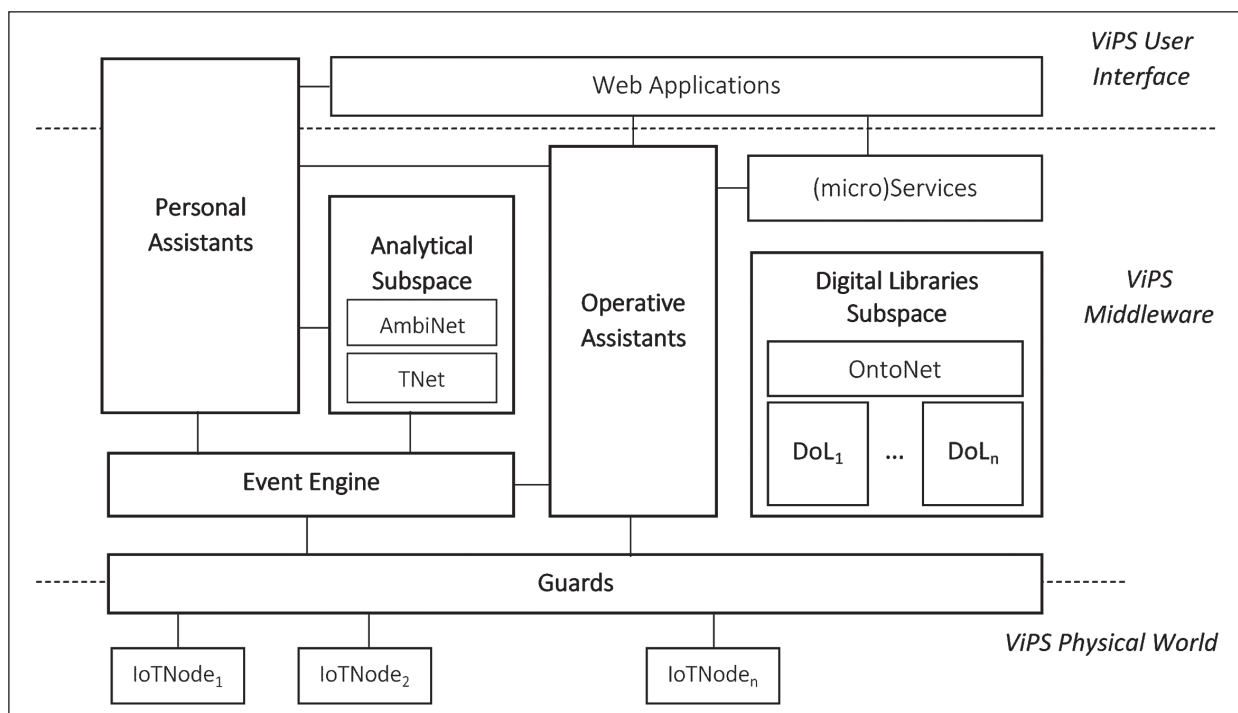


Fig. 1. Reference ViPS architecture

provides the preparation of domain-specific analyses, supported by three components for modeling:

- **AmbiNet** which presents the spatial aspects of “things” and events that are modeled as ambi-ents. AmbiNet is based on the official CCA (Calculus of Context Ambients) specification, which provides appropriate designations and modeling tools [15].
- **TNet** provides an opportunity to represent and work with the temporal aspects of “things”. It is based on the official Interval Temporal Logics specification [16].
- **ENet** — it models different types of events and their arguments such as identification, conditions for occurrence, and termination. It is essential for our model to distinguish three types of events — basic, systemic, and domain specific. Domain-specific events are realized as intelligent agents and they have a proactive behavior, that is, when an event occurs, the agent is dynamically generated to represent it and send a message to the other interested intelligent agents in the space [17].

The second subspace of ViPS are the Digital Libraries. They are implemented as open digital storage. Basic knowledge in various subject areas is structured and stored in separate libraries, which are serviced by specialized operating assistants. The OntoNet component is an ontology hierarchy that represents the essential characteristics and relationships between concepts in the subject area.

Agent-oriented approaches are particularly suited to building CPSS applications. The main components of ViPS are assistants, implemented as rational BDI agents, which consist of two main components:

- **Deliberation** — as a process by which an agent forms its current goal or objectives.
- **Means-Ends Reasoning** which prepares a plan for the achievement of the established goal.

By itself, the agent is a software component with a proactive, context-sensitive, and intelligent behavior that works taking into account the dynamics of the environment but is not suitable for providing business functionality. Services are a good solution for providing functionality, but they are static, inactive, and cannot be separate components in space. Therefore, agents include appropriate interfaces to services in their internal architecture. In this way, space functions as an ecosystem open to expanding and upgrading with new services. Three types of assistants have been implemented:

- **Personal assistants (PAs)** that are at the center of the system and help users work with the specific application. They must perform two main tasks: assisting the user in their daily activities and prevention.
- **Operational Assistants (OAs)** — intelligent agents located on the system server. They maintain the access to storage and services.
- **Guard Assistants (GA)** — they provide an interface between the physical and the virtual

worlds. GA transfer data from the physical world represented by the so-called IoT nodes to the virtual one and back. IoT nodes typically include different types of sensors, controllers, and actuators. Communication between the guards and IoT nodes is usually accomplished through the Internet or other communication such as the Lora technology.

Due to the nature of ViPS as a CPSS system, users need to be the center of attention. Because the expected complexity of the system under development is high, a Genetic Personal Assistant (GPA) is being developed that creates a specific personal assistant for each new user of the space. The GPA manages, stores, and restores the personal assistant versions that it has created in the past.

The Personal Assistant is based on the BDI architecture, which tries to mimic the way people think. The BDI architecture originated when working on the Stanford Research Institute’s Rational Agency project in the 1980s. It was developed by the philosopher Michael Bratman [18]. The agents have the following mental states:

- **Beliefs** — this is the information that an agent has about the world.
- **Desires** — all the possible things that the agent can do, i. e., these are potential factors for action.
- **Intentions** — these are things that the agent has decided to do. They can be goals or options considered for action.

The practical reasoning of the ViPS assistants consists of two main actions — the Deliberation process, through which the agent forms its current goal, and the Means-Ends Reasoning, which prepares a plan for achieving the goal. Generating a comprehensive plan to achieve a goal in the presence of different actions that will be applied to the space is not the most appropriate development option, as any action on the environment will change it. A more appropriate option is to develop partial plan collections, by which the agent can choose the sequence of plans to perform according to the purpose and environmental change in real time. This makes the system much more flexible. This is the approach that is implemented through the AgentSpeak multi-agent development language as well as in a specific implementation called Jason [19].

As we have already noted, each PA has two main tasks to perform:

- **Assistance** — operational assistance to the consumer in his/her daily activities;
- **Prevention** — helping to carry out the necessary pre-emptive actions and prevention. Prevention aims to provide optimal conditions for the user to participate in the upcoming event, including those related to his/her movement in the physical space in terms of time.

Any action on the environment would lead to a change in the agent’s beliefs and from there to a change in its intentions so he can go back to the reflection phase and change the plan to achieve the goal. Its beliefs may also be altered by communication with another agent.

3. Application of blockchain technologies in ViPS

Blockchain technologies are able to achieve integrity and trust in a distributed peer-to-peer (P2P) system, which consists of an unknown number of peers with unknown reliability. The P2P architecture consists of nodes with the same functionality, capabilities, and responsibilities. The nodes cooperate in a common communication environment to achieve a specific goal, without central coordination and a control component. A key role in building and maintaining the blockchain is to use cryptographic and security technologies to achieve integrity. The blockchain technology is based on a distributed Ledger system that maintains proprietary information and stores the entire history of transactions in the chain. Each node has its own copy of the Ledger registry, allowing the individual nodes to collectively and sequentially establish ownership through the blockchain algorithm [20].

The blockchain uses a public-private approach to asymmetric cryptography, which is the basis for identifying users, transferring ownership, and protecting them from unauthorized access to the system. The purpose of the blockchain is to store huge amounts of data and keep them unchanged after they are created. It can detect any changes very quickly and easily in case somebody attempts to manipulate them [21]. According to their ownership, blockchains can be classified as public and private. Public blockchains are open source and do not require activation rights. On the other hand, private blockchains are owned by an individual organization [22].

All these features of the blockchain technology enable it to be deployed in the implementation of various services and functionalities in ViPS. Security assurance and rapid identification of data manipulation attempts can be successfully implemented in the development of a school e-Diary — a system for tracking the movement of school documents with factory numbering (such as diplomas and certificates), building secure systems in agriculture and tourism.

4. Adaptation of ViPS in education

BLISS is an adaptation of the ViPS reference architecture to support the secondary school learning process. BLISS is implemented as a multi-agent system, the core of which are personal assistants which interact with each other. The agent environment consists of two parts – an event-driven BLISS server and a blockchain-based school diary. New types of specific personal assistants have been developed at BLISS as a GPA instance. The app also includes new components atypical for ViPS. Such a component is the blockchain representing the electronic school diary. In the development of BLISS, we have defined the following personal assistants:

- **PAStudent.** A personal assistant to assist students in the fulfillment of their daily duties in accordance with the approved syllabus and ongoing

learning activities. The PA informs the student of all upcoming events such as exams, lessons, study sessions, counseling, and more. The assistant monitors the student and reminds him/her what they need to prepare before the upcoming event. For example, if a notification about an upcoming event (Math exam) $L1 = \langle \text{Math, Exam, attr (02/20/2020, time (10: 00,10: 45))} \rangle$ is received, the PAStudent should not only inform the relevant user, but also be able to take pre-emptive action according to the prevention interval laid down in its knowledge base (e. g. 15 days). It must draw up a plan for assisting the student to participate successfully in the event. In this case, it can provide him/her with the necessary resources for self-study, remind him/her periodically and assist him/her in the learning process. This assistant is able to prepare analyses for the results of the student's participation in the learning process as well as to communicate and interact with other assistants in the learning space.

- The **PATeacher** is intended for teachers. It can also be a reminder of upcoming events and the necessary preparation for successful participation in these events. However, its primary function is to assist teachers in tracking and analyzing their students' participation, progress, and performance. Analyses can be used to improve the learning activities. In addition, the personal assistant provides services and information related to the electronic school diary.
- The **PAParent** is an assistant which provides parents with information about their child's involvement and progress in school. It provides information on teacher assessments, school events, messages, and notes. Owing to the constant internal communication between the agents and the analysis of the information received from the student's personal assistant, the parent can be informed of any changes in the student's behavior and status.
- **PAPrincipal.** The purpose of this assistant is to assist the school principal in the effective management of the school institution. This assistant is intended to assist in the planning, implementation, and control of the learning process. Of particular relevance is the role of this assistant in the work of the principal with the school's e-Diary. On account of the two-way communication with the SchoolDiary, the Specialist Assistant SA provides the basic functionality of this important BLISS component.

The environments of the various assistants are the BLISS server and the e-SchoolDiary. In the BLISS server, all information objects (such as schedules, lessons, exams, consultations, self-preparation, and meetings) are presented as domain-specific events. The server stores these events and controls access to them. Any changes to the information on the server are automatically perceived by all "interested" assistants.


```

// SA's initial belief was that the register (Ledger) was empty
ledger(new_data,0).
// The initial objective of the SA is to:
!get(data_ledger).
// A plan to inform the PPrincipal if no new data is available
+!get(data_ledger): not has(ledger, new_data)
<- .send(PPrincipal, askif, data_ledger).

```

Fig. 2. SA implementation in Jason environment

The SchoolDiary is also part of the assistants' environment. Since our wish is to record content in the school diary that is sensitive to change, such as student assessments, we use the blockchain technology. The other less critical information such as dates for parent meetings and notes on absences and student behavior is stored in a dedicated Data Module. Because of our specifics for the implementation of the school diary, we use a private blockchain. Nodes in the system are all teachers and the headmaster of the respective school. To become a node in the system, each teacher sends a request to the headmaster. Once access is granted, the system provides a public and a private key to the respective teacher through which he or she can check and sign the transactions. The students are not nodes in the system, which is why the school principal is the recipient of the transactions. At the end of each school day, he or she has the right and obligation to validate all transactions by signing them using his or her private key. Once the transactions are signed, the blockchain checks them for formal and semantic correctness and authorization. Only correct transactions are completed in a block at the end of the day and validated in the blockchain. Once the block is validated, all ledgers in the system are updated with the new data. Depending on the number of transactions for the day, the blocks have different sizes. Ledgers maintain property information and the history of transactions that have ever occurred. The BLISS server, where the DM and personal assistants are located, is also a blockchain node, that is, it stores the registry (Ledger) of that node. The Special Operations Assistant SA provides communication between the BLISS server, the DM, and the various PA groups. It aims to respond to changes in the environment of the e-SchoolDiary. In the event of a change in the Ledger on the server, the SA records the information in the DM, which is accessible to all personal assistants. It then informs all assistants about this change. The SA environment consists of two modules — the Ledger and the DM, while the environment of all other personal assistants is the DM and the BLISS server, managed by events such as teacher consultations, exams, online discussion, meetings, messages, etc.

Communication between agents in a multi-agent system is based on the speech act theory – Austin [23] and Searle [24]. It is based on the principle that language is an action. That means that the rational agent makes an attempt to change the state of the world in the same way as it performs a physical action on the

environment. Speech actions by one agent may aim to change the beliefs, desires, and intentions of another agent. In the Jason environment there are three sources of agent information in a multi-agent system:

- **Sensory information** — The agent acquires certain beliefs through a sense of its environment: annotation — source (percept).
- **Mental notes** — Beliefs of this type are added to the knowledge base of the agent itself and their purpose is to use it later, when necessary: annotation — source (self).
- **Communication with other agents** — This is available information that an agent has acquired from another agent, the name of the agent which sent the information, and what is recorded.

Let us look at the following example scenario: If at the end of the day there is no change in the registry (Ledger), the SA, which is proactive, sends a message to the PPrincipal Director's Assistant informing him/her and asking if the Director has failed to sign the transactions for the day. An exemplary arrangement of Jason for SA implementation is presented in Figure 2.

The personal assistant PPrincipal keeps track of the school principal's to-do schedule as well as his/her calendar. Upon receiving this message, the PPrincipal has different action scenarios and will accordingly select the most appropriate plan to meet the objective, depending on the context.

1. The day is a holiday. In this case, the PPrincipal checks that the day is not a working day and if it believes so, returns a message to the SA, informing it that there will be no entries on that day.
2. The day is working day, but no transactions have been confirmed that day. If the day is over, the PPrincipal prompts the Director to sign the transactions and then returns a message to the SA to wait for the Ledger to be updated. After one hour, if there is no change in the SA register, it again sends a message to the PPrincipal. The assistant asks the director and if he/she replies that there are no transactions in the blockchain that day, it returns a message to the SA not to wait any longer.
3. The day is a working day, but there are transactions waiting to be signed. The scenario is similar to the second one, but if the PPrincipal does not receive a new message from the SA within an hour, it believes that the school principal has already signed the transactions in the blockchain.

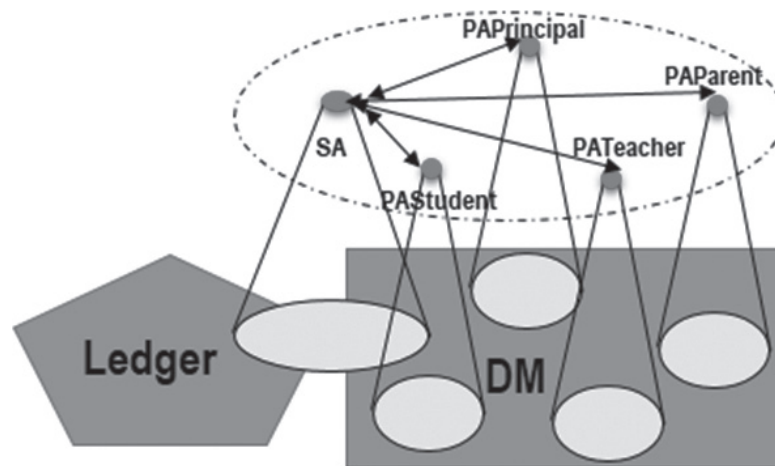


Fig. 3. Communication between agents in the e-School Diary

A very important part of building a multi-agent system is the communication between assistants. By definition, based on these messages, assistants can change their perception of the world and in the next stage of reflection, they can choose another plan to achieve the goal. The communication between them will look like this, and for the sake of clarity, we will only consider the interaction of the SA with the other personal assistants in the system. An important clarification is that the environment that agents observe can overlap (Figure 3).

An essential feature of BLISS is that although the blockchain is decentralized, the system has a central element for communication and control. Thus, if there is a technical or communication problem with the server, the system will stop working. For this reason, we think it is appropriate to maintain an up-to-date backup in the cloud ViPS space. After the SA has kept up-to-date information on student assessments in the DM, it sends messages to all interested personal assistants in the BLISS space. Each personal assistant, in turn, informs the particular user and records the grades in the student's or teacher's notebooks as well as in their knowledge base. The following Activity UML Diagram shows the communication between the main participants in the scenario presented (Figure 4).

5. Modeling, testing and verification

To model these scenarios, we will use the Ambient capabilities of the ViPS Analytical Subspace and the Calculus of Context-Aware Ambients (CCA) formalism [15]. The syntax and formal semantics of CCA enable modeling in a context-aware dynamic environment. The CCA ambient is the identity used to describe an object or a component (process, device, person, location, etc.). The ambient has a location, boundaries, and a name, by which it is identified in the space. It may contain other ambients within itself, thus allowing the creation of ambient's hierarchies. There are three possible relationships between two

ambients: parent, child, and relative. Communication is effected by sending and receiving messages. In CCA notation, when two objects exchange a message, “::” symbols are used to interact with relatives; “↑” and “↓” for parent-child interaction; “<>” to send a message, and “()” to receive a message. Ambients are mobile and can be moved in and out of other ambients in the hierarchy. CCA distinguishes four syntactic categories: location α , capabilities M, processes P, and context expressions k.

We will use the following ambients to model the scenario under consideration:

Name	Description
SA	Specialist Assistant
PAPr	PAPrincipal
PAS	PAStudent
PAP	PA Parent
PAT	PA Teacher
HL	Hyper Ledger
DM	Data Module
SC	Schedule
SD	School Diary

The processes of the main CCA-ambients in the scenario are presented in Figure 5.

To test and verify the model, we use the ccaPL language interpreter [23], which implements the software implementation of the CCA notation. The monitoring of the processes in the testing and verification mode makes it possible to evaluate the interactions between the ambients and to detect at this stage possible problems, overloads, and discrepancies before the actual start of these processes. Figure 6 depicts some of the interactions in simulation mode.

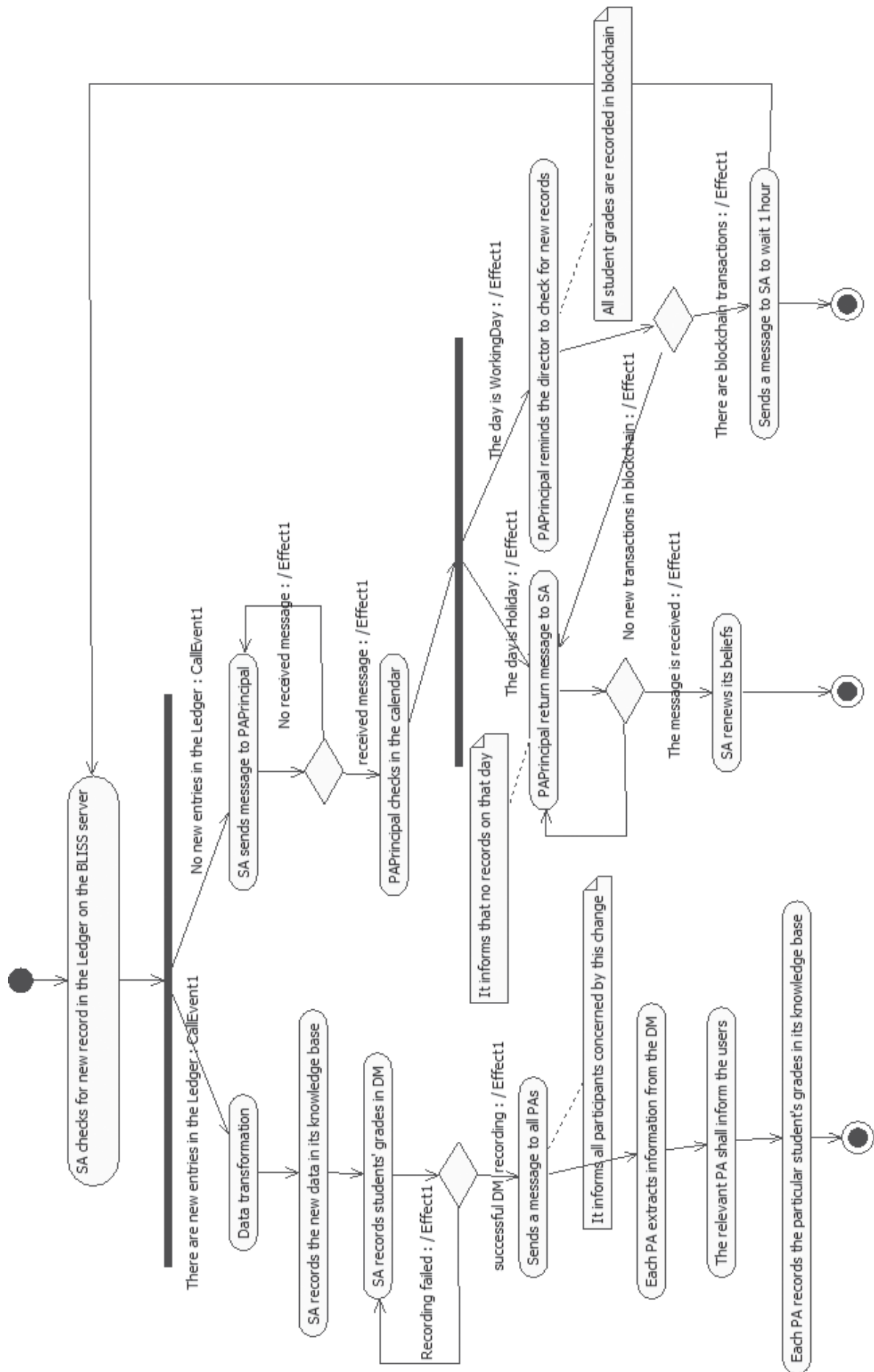


Fig. 4. UML Activity Diagram of the described scenario

$$P_{SA} \cong \left(\begin{array}{l} HL :: (There_are_no_transactions_today).PAPr :: < check_for_new_transactions > .0 | \\ PAPr :: (Today_is_no_WorkingDay).PAPr :: < OK > .0 | \\ PAPr :: (Wait_one_Hour).PAPr :: < OK > .0 | \\ HL :: (There_is_new_transaction).DM :: < save_records > .0 | \\ DM :: (OK).PAS :: < There_is_new_data_today > . \\ PAP :: < There_is_new_data_today > .PAT :: < There_is_new_data_today > .0 \end{array} \right)$$

$$P_{PAPr} \cong \left(\begin{array}{l} SA :: (check_for_new_transactions).DM :: < is_working_day_today > .0 | \\ DM :: (day). \\ \left((day = WorkingDay)?SA < Wait_one_Hour > .0 | \right. \\ \left. (day = Holiday)?PAPr :: (Today_is_no_WorkingDay).0 \right) | \\ SA :: (OK).0 | \\ HL :: < are_there_any_transactions > .0 | \\ HL :: (reply).0 | \\ \left((reply = Yes)?HL :: < confirmation_transaction > .0 | \right. \\ \left. (reply = No)?SA :: < today_is_no_transactions > .0 \right) \end{array} \right)$$

$$P_{DM} \cong \left(\begin{array}{l} PAPr :: (is_working_day_today).SC \downarrow < check > .0 | \\ SC \downarrow (day).PAPr :: < day > .0 | \\ SA :: (save_records).SD \downarrow < save_data > .0 | \\ SD \downarrow (OK).SA :: < OK > .0 | \\ PAS :: (stu_id,getdata).PAS :: < stu_id,newdata > .0 | \\ PAT :: (teach_id,getdata).PAT :: < teach_id,newdata > .0 | \\ PAP :: (parent_id,getdata).PAP :: < parent_id,newdata > .0 \end{array} \right)$$

$$P_{SC} \cong (DM \uparrow (check).DM \uparrow < day > .0)$$

$$P_{SD} \cong \left(\begin{array}{l} DM \uparrow (save_data).SA :: < OK > .0 | \\ DM \uparrow (stu_id,getdata).DM \uparrow :: < stu_id,newdata > .0 | \\ DM \uparrow (teacher_id,getdata).DM \uparrow :: < teacher_id,newdata > .0 | \\ DM \uparrow (parent_id,getdata).DM \uparrow :: < parent_id,newdata > .0 \end{array} \right)$$

$$P_{HL} \cong \left(\begin{array}{l} SA :: < There_are_no_transactions_today > .0 | \\ PAPr :: (are_there_any_transactions).PAPr :: < reply > .0 | \\ PAPr :: (confirmation_transactions).SA :: < There_is_new_transaction > .0 \end{array} \right)$$

$$P_{PAS} \cong \left(\begin{array}{l} SA :: (There_is_new_data_today).DM :: < stu_id,getdata > .0 | \\ DM :: (stu_id,newdata).0 \end{array} \right)$$

$$P_{PAT} \cong \left(\begin{array}{l} SA :: (There_is_new_data_today).DM :: < teacher_id,getdata > .0 | \\ DM :: (teacher_id,newdata).0 \end{array} \right)$$

$$P_{PAP} \cong \left(\begin{array}{l} SA :: (There_is_new_data_today).DM :: < parent_id,getdata > .0 | \\ DM :: (parent_id,newdata).0 \end{array} \right)$$

Fig. 5. The processes of the main CCA-ambients in the scenario


```

Administrator: Command Prompt
CCA Parser Version 2.0: Reading from file bliss.cca . . .
CCA Parser Version 2.0: CCA program parsed successfully.

--> {Sibling to sibling: HL ===(There_are_no_transactions_today)===> SA}
--> {Sibling to sibling: SA ===(check_for_new_transactions)===> PAPr}
--> {Sibling to sibling: PAPr ===(are_there_any_transactions)===> HL}
--> {Sibling to sibling: PAPr ===(is_working_day_today)===> DM}
--> {Sibling to sibling: DM ===(WorkingDay)===> PAPr}
--> {Sibling to sibling: PAPr ===(Wait_one_our)===> SA}
--> {Sibling to sibling: SA ===(OK)===> PAPr}
--> {Sibling to sibling: HL ===(Yes)===> PAPr}
--> {Sibling to sibling: PAPr ===(confirmation_transaction)===> HL}
--> {Sibling to sibling: HL ===(There_is_new_transaction)===> SA}
--> {Sibling to sibling: SA ===(save_records)===> DM}
--> {Sibling to sibling: DM ===(OK)===> SA}
--> {Sibling to sibling: SA ===(There_is_new_data_today)===> PAS}
--> {Sibling to sibling: SA ===(There_is_new_data_today)===> PAT}
--> {Sibling to sibling: SA ===(There_is_new_data_today)===> PAP}
--> {Sibling to sibling: PAS ===(stu_id,getdata)===> DM}
--> {Sibling to sibling: DM ===(stu_id,newdata)===> PAS}
--> {Sibling to sibling: PAT ===(teach_id,getdata)===> DM}
--> {Sibling to sibling: DM ===(teach_id,newdata)===> PAT}
--> {Sibling to sibling: PAP ===(parent_id,getdata)===> DM}
--> {Sibling to sibling: DM ===(parent_id,newdata)===> PAP}

```

Fig. 6. Testing and verification of interactions in simulation mode

6. Conclusions

The BLISS space as a ViPS adaptation for secondary school education presented in this article is currently in test mode at the experimental school — Hristo Smirnenki Secondary School in Brezovo. The BLISS server and the PASTudent assistant are fully implemented and the other Personal Assistants are in development and initial testing. In the last year, more than 50 students in lifelong learning mode have used their personal assistants to access the educational platform and the results are very good. For these types of students who are adults and who live long distances from school, their personal assistant is a great help in providing information and supporting their learning. We have created a test implementation of a blockchain-based electronic SchoolDiary. Work continues to create a specialist SA assistant with the functionalities described in the article.

References

1. Schwab K. The Fourth Industrial Revolution. Currency, 2017. 192 p.
2. Raganathan R., Lee I., Sha L., Stankovic J. Cyber-physical systems: The next computing revolution. *DAC '10: Proc. 47th Design Automation Conf.* NYC, Association for Computing Machinery, 2010, p. 731–736. DOI: 10.1145/1837274.1837461
3. Wang F.-Y. The emergence of intelligent enterprises: From CPS to CPSS. *IEEE Intelligent Systems*, 2010, vol. 25, is. 4, p. 85–88. DOI: 10.1109/MIS.2010.104
4. Panchanathan S., Chakraborty S., McDaniel T. Social interaction assistant: A person-centered approach to enrich social interactions for individuals with visual impairments. *IEEE Journal of Selected Topics in Signal Processing*, 2016, vol. 10, is. 5, p. 942–951. DOI: 10.1109/JSTSP.2016.2543681
5. A PAL. The PAL Framework. Available at: <https://pal.sri.com/>
6. CALO — Cognitive Assistant that Learns and Organizes. Available at: <http://www.ai.sri.com/project/CALO>
7. Companions-project. Available at: <http://www.companions-project.org/>
8. Apple Siri. Available at: <https://www.apple.com/siri/>
9. Microsoft Cortana. Available at: <https://www.microsoft.com/en-us/cortana>
10. Hill S. How to get the most out of Google Now. *Digital Trends*, 2016. Available at: <https://www.digitaltrends.com/mobile/what-is-google-now/>
11. Stoyanov S., Popchev I., Doychev E., Mitev D., Valkanov V., Stoyanova-Doycheva A., Valkanova V., Minov I. DeLC educational portal. *Cybernetics and Information Technologies*, 2010, vol. 10, no. 3, p. 49–69. Available at: http://www.cit.iit.bas.bg/CIT_2010/v10-3/49-69.pdf
12. Gramatova K., Stoyanov S., Doychev E., Valkanov V. Integration of eTesting in an IoT eLearning ecosystem: Virtual eLearning Space. *BCI '15: Proc. 7th Balkan Conf. on Informatics Conf.* NYC, Association for Computing Machinery, 2015, p 1–8. DOI: 10.1145/2801081.2801086
13. Stoyanov S., Glushkova T., Stoyanova-Doycheva A., Doychev E. Virtual Physical Space — an architecture supporting internet of things applications. *2018 20th Int. Symposium on Electrical Apparatus and Technologies (SIELA)*. Bourgas, IEEE, 2018. DOI: 10.1109/SIELA.2018.8447156
14. Stoyanov S., Glushkova T., Doychev E. Cyber-physical-social systems and applications. LAMBERT Academic Publishing, 2019. 100 p.
15. Siewe F., Zedan H., Cau A. The calculus of context-aware ambient. *Journal of Computer and System Sciences*, 2011, vol. 77, is. 4, p. 597–620. DOI: 10.1016/j.jcss.2010.02.003
16. Moszkowski B. C. Compositional reasoning using interval temporal logic and tempora. *Compositionality: the significant difference. Proc. COMPOS 1997*. Berlin, Springer Heidelberg, 1998, p. 439–464. DOI: 10.1007/3-540-49213-5_17
17. Guglev Z., Stoyanov S. Hybrid approach for manipulation of events in the virtual referent space. *Компютърни*

науки и коммуникации — *Computer Science and Communications*, 2018, vol 7, no. 1, p. 79–85. Available at: <https://csc.bfu.bg/index.php/CSC/article/view/210>

18. *Bratman M. E.* Intention, plans, and practical reason. Cambridge, Harvard University Press, 1987.

19. *Bordini R. H., Hübner J. F., Wooldridge M.* Programming multi-agent systems in AgentSpeak using Jason. Wiley-Interscience, 2007. 292 p. Available at: <https://www.wiley.com/en-us/Programming+Multi+Agent+Systems+in+AgentSpeak+using+Jason-p-9780470061831>

20. *Drescher D.* Blockchain basics. A non-technical introduction in 25 steps. Berkeley, Apress, 2017. 255 p. DOI: 10.1007/978-1-4842-2604-9

21. *Zyskind G., Nathan O., Pentland A.* Decentralizing privacy: using blockchain to protect personal data. *2015 IEEE*

Security and Privacy Workshops. IEEE, 2015. DOI: 10.1109/SPW.2015.27

22. *Filipova N.* Blokcheyn: Vazmozhnost za novi biznes modeli [Blockchain: Opportunity for new business models]. *Business Management*, 2018, is. 2, p. 75–92. (In Bulgarian.) Available at: https://EconPapers.repec.org/RePEc:dat:bmn_gmt:y:2018:i:2:p:75-92

23. *Austin J. L.* How to do things with words. Oxford, Oxford University Press, 1962.

24. *Searle J. R.* Speech acts: an essay in the philosophy of language. Cambridge, Cambridge University Press, 1970. 203 p.

25. *Al-Sammarraie M. H.* Policy-based approach for context-aware systems. Leicester (UK), De Montfort University, 2011. (PhD Thesis)

МОДЕЛИРОВАНИЕ И РАЗРАБОТКА МУЛЬТИАГЕНТНОГО ПРОСТРАНСТВА ДЛЯ СРЕДНЕЙ ШКОЛЫ

И. К. Крастева¹, Т. А. Глушкова¹, С. Н. Стоянов¹

¹ *Пловдивский университет «Паисий Хилендарски», Болгария*
4000 Болгария, Пловдив, ул. Цар Асен, 24

Аннотация

Один из основополагающих принципов Четвертой промышленной революции — необходимость обучения на протяжении всей жизни. Это определяет растущую роль интеллектуальных образовательных систем в предоставлении необходимых учебных ресурсов и услуг пользователям в любое время и в любом месте. В данной статье рассматриваются моделирование и разработка интеллектуальной мультиагентной образовательной среды для средней школы, осуществляемые командой лаборатории Центра электронного обучения Пловдивского университета «Паисий Хилендарски», Болгария. Учащиеся находятся в центре внимания среды благодаря личным помощникам, поддерживающим работу со средой.

Ключевые слова: кибер-физические системы, кибер-физическое социальное пространство, виртуально-физическое пространство (ViPS), интернет вещей, блокчейн.

DOI: 10.32517/0234-0453-2020-35-4-53-62

Для цитирования:

Крастева И. К., Глушкова Т. А., Стоянов С. Н. Моделирование и разработка мультиагентного пространства для средней школы // Информатика и образование. 2020. № 4. С. 53–62. (На англ.)

Статья поступила в редакцию: 27 февраля 2020 года.

Статья принята к печати: 21 апреля 2020 года.

Финансирование и благодарности

Работа выполнена при поддержке Фонда научных исследований Пловдивского университета «Паисий Хилендарски», контракт № МУ19-ПФ-023.

Авторы признательны за предоставленный доступ к электронной инфраструктуре Центра передовых вычислений и обработки данных при финансовой поддержке гранта № BG05M2OP001-1.001-0003, финансируемого операционной программой «Наука и образование для интеллектуального роста» (2014–2020 годы) и софинансируемого Европейским союзом через европейские структурные и инвестиционные фонды.

Сведения об авторах

Крастева Ирина Красимилова, аспирант факультета математики и информатики, Пловдивский университет «Паисий Хилендарски», Болгария; irina_krasteva@uni-plovdiv.bg; ORCID: 0000-0003-0842-8143

Глушкова Тодорка Атанасова, Ph.D., доцент факультета математики и информатики, Пловдивский университет «Паисий Хилендарски», Болгария; glushkova@uni-plovdiv.bg; ORCID: 0000-0002-6243-9364

Стоянов Станимир Недялков, Ph.D., профессор факультета математики и информатики, Пловдивский университет «Паисий Хилендарски», Болгария; stani@uni-plovdiv.net; ORCID: 0000-0002-3854-4260