INT. J. BIOAUTOMATION, 2019, **23**(3), 355-368 doi: 10.7546/ijba.2019.23.3.000590

Internet of Things Platform Supporting Mobility of Disabled Learners

Todorka Glushkova[1], Stanimir Stoyanov[1*], Ivan Popchev[2]

[1]*Faculty of Mathematics and Informatics*
Plovdiv University Paisii Hilendarski
236 Bulgaria Blvd., 4000 Plovdiv, Bulgaria
Emails: glushkova@uni-plovdiv.bg, stani@uni-plovdiv.bg

[2]*Faculty of Economics and Social Sciences*
Plovdiv University Paisii Hilendarski
24 Tsar Asen Str., 4000 Plovdiv, Bulgaria
Email: ipopchev@iit.bas.bg

[*]*Corresponding author*

Received: May 22, 2018 **Accepted: June 13, 2019**

Published: September 30, 2019

Abstract: *Cyber-Physical-Social Spaces (CPSS) and Internet of Things (IoT), two closely related concepts provide new yet unknown opportunities to support disabled people through information and communication technologies. This paper presents an IoT reference architecture known as Virtual Physical Space (ViPS) designed to support the mobility of people with disabilities. Furthermore, a modeling component, known as AmbiNet and hosted in ViPS, is described in more detail. The AmbiNet supports presentation of spatial aspects of physical "things" to be virtualized in an Internet of Things application. In AmbiNet, the "things" of interest are modeled as a network of interacting ambients. The environment is implemented on the top of the Calculus of Context-aware Ambients (CCA). The ambient-oriented modeling in AmbiNet is demonstrated by an example.*

Keywords: *Cyber-physical-social spaces, Internet of Things, Ambient-oriented modeling, Calculus of context-aware ambients, AmbiNet, Disabled learners.*

Introduction

Cyber-Physical Spaces (CPS) and Internet of Things (IoT) are closely related concepts. Despite certain differences, the integration of the virtual and physical worlds is common to both of them. By placing the person (the user) in the centre of such spaces, they become Cyber-Physical-Social Spaces (CPSS). We belief that from a software architecture point of view, a CPSS application has to include various components designed to provide effective support (effective help) to different user groups, taking into account changes in the physical environment. Effective software models for building a CPSS can be such as to support the creation of distributed, autonomous, contextually-informed, intelligent software.

CPSSs can be built for various domains including education. To support e-learning at the Faculty of Mathematics and Informatics at the University of Plovdiv, the environment Distributed eLearning Center (DeLC) has been used for years [24, 26]. Although DeLC was a successful project for applying information and communication technologies in education, one of its major drawbacks is the lack of close and natural integration of its virtual environment with the physical world where the real learning process takes place [22]. Taking into account the temporal and spatial characteristics of the physical world and the events that take place in it is particularly important in helping people (in our case learners) with different disabilities. CPSS and IoT paradigms reveal entirely new opportunities for taking into account the needs of disabled people, in our case disabled learners. For these reasons, in the recent years, the DeLC environment has been transformed into a Virtual Education Space (VES) that operates as an IoT ecosystem [20, 25].

Summing up the experience of constructing VES, we began developing a reference architecture known as Virtual Physical Space (ViPS) [21] that could be adapted to various CPSS-like applications such as a smart seaside city [23] or a smart tourist guide [6]. Implementing a CPSS application, a major challenge is the virtualization of "things" from the physical world, which are of interest to us. Moreover, account has to be taken of related events, time and spatial aspects. In ViPS, to present the spatial aspects of "things", we have decided to use an ambient-oriented modeling approach. A component for presenting the location and movement of objects of interest in the physical world was implemented adopting a formal ambient-oriented approach.

In this paper, we want to demonstrate the use of ViPS specifically for helping people with mobility problems. The rest of the paper is organized as follows: short reviews of ambients and ambient intelligence and personal assistants supporting disabled people are considered in Section 2, respectively Section 3, which is followed by an overall description of the ViPS architecture in Section 4. Section 5 addresses the architecture of AmbiNet. Section 6 demonstrates the application of ViPS to model a services supporting mobility of disabled people. Finally, Section 7 concludes the paper.

Ambients and ambient intelligence

According to [3], an ambient is an identity that can be characterized by boundary, occupancy and mobility, and it is possible to create hierarchies or networks of ambients. For modelling purposes, an ambient could be represented as a structure with the following elements: an identifier (name), a corresponding set of local agents, and a set of adjacent sub-ambients. Thus, there is the possibility of recursively building various more complex structures of ambients.

Ambient Intelligence (AmI) makes the daily information environment of users "more sensitive" to their problems and peculiarities by adding sensors for sensing and actuators for influencing the users' environment. All these devices are connected through a suitable computer network. As determined in [5], the main features of AmI technologies are: sensitivity, adaptability, transparency, universality, and intelligence. AmI is growing rapidly as a multidisciplinary area of scientific and practical interest. The basic idea behind AmI is that applications using real-time aggregated information and background data accumulated over time can make decisions for the benefit of users. In AmI systems, Artificial Intelligence plays a key role to provide flexibility, adaptability, predictability, and acceptable interface. In [16], intelligence is highlighted as a basic aspect, and an AmI system is defined as a digital environment as long as it proactively, but reasonably supports people in their daily lives. Typical examples are personal assistants that, depending on the situation, are able to provide proactive help or exercise restraint. Sensible requirements to them can be the ability to recognize the user, learn or know preferences and likes, and show empathy to the moods of users. A fundamental formalism, suitable for ambient-oriented modelling, is π-calculus, which represents a kind of process calculus [11].

Personal assistants supporting disabled people

Generated in the end of 20th century, the idea of using intelligent agents for assisting people in their daily business and personal activities has developed as an area of ever increasing scientific and practical interest. Artificial intelligence techniques have opened opportunities for constructing intelligent systems that can autonomously execute tasks on the behalf of the users, and in their benefit. Intelligent assistants can support both the operative daily management, as well as the long-term strategic management, execution and control of various types of tasks, for instance ones related to planning, reservation, shopping, payment, learning etc. Nowadays, these assistants are usually located on mobile devices, can use social networks as resource, and can self-learn. One of the developing trends is the use of robots as intelligent assistants.

In the United States, *PAL* (*Personalized Assistant that Learns*) [14] is one of the first multipurpose research program in the area of cognitive systems. The program aims at the radical improvement of the way computers interact with people. In Europe, in the frames of the Horizon 2020 Programme, the project for development of the intelligent assistants *COMPANIONS* [4] aims to change the traditionally perceived way people and computers interact. A virtual "*Companion*" is an agent that shares long periods of time with the user, while 'creating' a friendly relation and 'exploring' the person's preferences and desires. Corporate developments of personal assistants for supporting people in their daily routine, are for instance, *Siri* [19] (which is currently being migrated from a mobile application to a standalone device/gadget), *Microsoft Cortana* [10], *Google Now* [8], and many others.

In recent years, significant efforts have been paid for the development of intelligent assistants for support of disadvantaged people. Realizing the need for multimedia technologies for people with disabilities, the *Center for Cognitive Ubiquitous Computing* (*CUbiC*) in the University of Arizona has concentrated their efforts in solving this challenge. A sample project of the Center is dedicated to the design and implementation of a social interaction assistant for improving the interaction opportunities of people with visual impairments [15]. The aim of this assistant is improving the accessibility of the nonverbal social signals for blind people or people with impaired vision. It comprises a pair of glasses, equipped with a special camera. The input video stream is being analyzed with the help of algorithms for machine learning and computer vision, which allow retrieval of relevant non-verbal signals, and as a result, the user is being supplied with this information. Furthermore, there have been concentrated efforts towards the development of robots able to take care of elderly patients. The major challenge there is related to the personalization of these robots in order to answer the particular needs of the served people. Even robots trained with machine learning algorithms need rather long time to understand the preferences of elderly people. In [27], the aim was to present an intelligent agent that provides personalized care for elderly patients, implemented as an integrated service, comprising of a range of intelligent sub-services. The general architecture of the service contains three main components – virtual nurse, *Virtual Care Personalizer* (*VCP*) and *Care Template*. For the effective provision of the integrated service, *VCP* controls and generates the necessary personalization of the activities in a special "cloud" (*Care Cloud*). The occurrence of intelligent personal agents is an example of supply of new types of services, using electronic devices with a huge potential, exhibiting the achievements of Artificial Intelligence, as well as the (r)evolution of the information and communication technologies and the Internet.

In the projects for intelligent assistant development, there is a growing trend of combining results from the research in the area of IoT, robotics and machine learning. Blending the concept of intelligent assistant with the paradigm of IoT offers new opportunities for providing context-aware services to the end-users. In [17], a healthcare assistance scenario has been offered where a mobile gateway has been integrated into an intelligent assistant. Using this gateway, personal information about the physical status of the individual, obtained from a bodily sensor network, is transferred in real time to the assistant. Project *PAL* (*Personal Assistant for Healthy Lifestyle*) [13] envisages development of a *NAO*-based social robot, its mobile avatar and a scalable set of mobile applications in the area of healthcare, which use a shared knowledge base and inference method. *PAL* can serve as an assistant and teacher of kids, too.

ViPS in a Nutshell

Creating an IoT ecosystem addresses directly the notions of "things" and "intelligent objects". These are the main components that provide the connection with the real physical world, thus building a complete ecosystem. Each "thing" has its own characteristics, delivers different information and responds in a certain way depending on the type of sensors or actuators [2]. They communicate with each other through a particular communications network and can build complex hierarchical structures. Usually, the virtualization and digitization of any real physical object depends on the specific domain in which it is being considered. The ViPS architecture (Fig. 1) presents the virtualization of real-world objects and the ability of objects, processes, users, and knowledge of the domain to interact dynamically and contextually. Furthermore, it also takes into account events related to these objects as well as their temporal and spatial aspects. The ViPS consists of three logical layers which will be briefly presented below. In fact, the virtualization of physical objects is supported by the ViPS middleware. The modelling of things depends on their features such as the characteristic of events, time, and location. In the Digital Libraries Subspace is saved information about the domains of interest, for example, objects related to tourism, medical assistance, transport, eco-agriculture and aquaculture, etc. The current version is adapted to the DSpace standard enhanced with domain-depended ontologies. The analytical subspace prepares the necessary analytical information – reports, etc., which is processed by the following three components:

- **ENet** aims to model different types of events and their characteristics (identification, occurrence, and execution conditions), representative of the domain under consideration.
- **TNet** provides the opportunity to present and to deal with the time aspects of the "things", usually related to certain events and locations.
- **AmbiNet** models the spatial characteristics of "things" and events as a network of ambients.

The running and the management of various processes in these components are conducted by specialized interpreters based on formal specifications such as Interval Temporal Logics [12], Calculus of Context-aware Ambients [18], and Event Model [9]. The interpreters are implemented as operative assistants. The OntoNet is a hierarchical structure of ontologies that provide the knowledge of the specific elements of things in the domain of interest. It could also be seen as a meta-level of both subspaces, which supports intelligent search and the provision of data in a user-friendly manner.

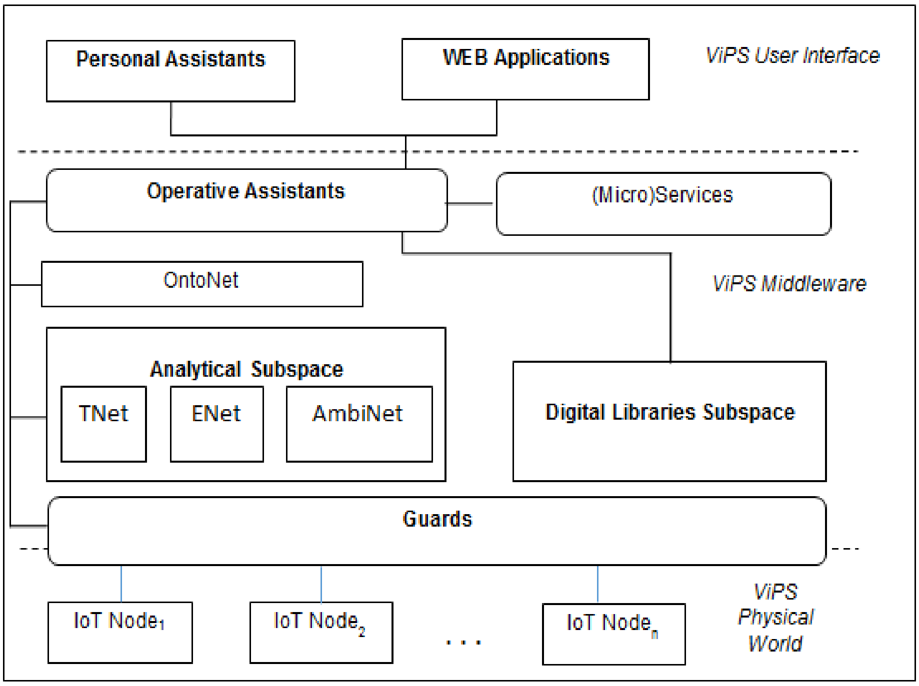

Fig. 1 Architecture of the ViPS

The operative assistants provide access to the resources of both subspaces and interact both with the users' personal assistant and the guard system. Assistants are architectural components suitable for providing the necessary dynamics, flexibility, and intelligence of the system but they are not suitable to deliver the necessary business functionality in the space. For this reason, the operative assistants work closely with services or micro-services implementing the needed business functionality. The guard system aims to provide a smart interface between the virtual world and the physical world. In ViPS, the real world is presented by many IoT nodes that access sensors and actuators of the "things". Communication is provided through a specialized interface using a combination of a private network (e.g. LoRa) and the Internet. The guards can detect and locate events that are related to the safety and security of individual users, and trigger appropriate emergency scenarios.

Access to the ViPS resources and services is usually implemented by the user's personal assistant. The personal assistant interacts with other space assistants and provides continuous help and information to the user in a dynamic mode.

AmbiNet

AmbiNet is a network structure modelling the physical world of interest (a city, area, district, or a campus). The network consists of separate building blocks known as "ambients". We use ambients for modelling of physical or virtual objects with their attributes, and spatial, temporal, and event characteristics. The network can be parameterized to the dynamically changing environment by means of data received from the IoT nodes. Additionally, it can be adapted to various domains such as a smart seaside city [23], a tourist guide [6], or education [7]. Electronic services could be provided as an upgrade layer, above the network structure itself, to support various user groups, especially disabled people, children, and patients. Examples of services are: inspecting the current environment (around the user) and alerting him/her about potential dangers; monitoring the quality of the air and alerting the user; navigating the user along routes that surround dangerous areas; monitoring patients' vital signs and alerting the hospital in case emergency medical assistance is needed; generating tourist routes.

AmbiNet is supported by the following run-time and development tools (Fig. 2.):

- AmbiNet ccaPL Interpreter – this is a run-time interpreter of the formal modeling language ccaPL based on the Calculus of Context-aware Ambients (CCA);
- AmbiNet Route Generator – used for routes generation on the network by user-defined criteria;
- AmbiNet Route Optimizer – applied to optimize routes depending on the instant state of the participating ambients;
- AmbiNet Editor – a visual modeling editor;
- AmbiNet Visualizer – used to present the results as a virtual reality.

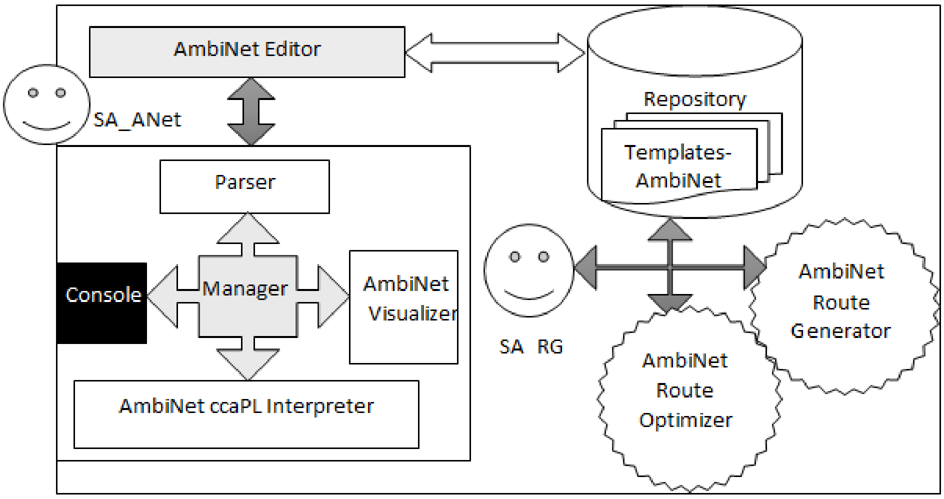

Fig. 2 AmbiNet architecture

Depending on their location, ambients can be static and mobile. Static ambients have a constant location in the physical world or in the modelled virtual reality. Examples are ambients modelling hospitals, schools, universities, museums, bus stations, rail stations, ports, bus stops, etc. Any such ambient may include a hierarchical structure of ambient-children, which can be static or mobile themselves; for example a university exposes a whole subambient structure of faculties, corps, lecture rooms, etc. Mobile ambients have a variable location, e.g. buses, people, ships, cars, ambulances, etc. They may also have a hierarchical sub-ambient structure and can move – "coming in" and "going out" from other static or mobile ambients. Typically, if they change their location, then all the "ambient-children" move as well.

In a CPSS, the real world is modeled through an ambient-oriented context-aware network. Especially in AmbiNet, for each individual context – a city, educational institution, tourist sites, etc., the AmbiNet Editor creates a structure of static ambients stored as a separate template in a dedicated repository. These templates are used not only by the AmbiNet interpreter but also by the AmbiNet Route Generator and the AmbiNet Route Optimizer to generate and optimize requested routes. AmbiNet is supported by specialized operative assistants providing various services in the space. SA_ANet is responsible for communicating with other assistants in ViPS. After receiving a service request, it organizes and manages the processes in AmbiNet, according to its conceptions and the generated plan. Since one of the most important services provided by AmbiNet is the generation and optimization of routes, it is necessary to use the SA_RG assistant to communicate with SA_ANet while being directly responsible for the provision of the service. Some of the assistants are represented in the next section in the demonstration of AmbiNet's for a particular service.

Supporting of disabled learners in ViPS

The syntax and formal semantics of the CCA extend the CBA with the following characteristics: the ability to perform processes only while ensuring the conditions in a particular context; treating abstract processes by applying context mechanisms; the CCA can fully implement the π-calcule modelling by providing much better opportunities for context-aware Ambient modelling.

CCA Ambients are defined as an identity that is used to describe an object or a component – a process, a device, a location, etc. Each aambient has a name and a boundary; it may contain other ambients within itself, and be included in another ambient. Between two ambients there are three possible relationships (parent, child, and sibling). Each ambient can communicate with other ambients around it; they can exchange messages with each other. The process of messaging is performed by using the handshaking process. In the CCA notation, when two ambients exchange a message, we use the "::" symbol to describe an interaction between sibling ambients; "↑" and "↓" are symbols for parent and child; "<>" means sending a message, and "()" – receiving a message. The ambient can be mobile, i.e. it can move within the surrounding environment. With the CCA, there are two ways to move: inwards (in) and outwards (out), which allow ambients to move from one location to another. The CCA can distinguish four syntax categories: location α, opportunities M, processes P, and context expressions k.

The student with mobility disabilities participates in ViPS with his smart IoT-wheelchair, which provides him/her a specific support environment. Upon entering in the education university physical space, the PA assistant starts a two-way communication process, providing the student both standard educational services and some specific ones connected with movement of wheelchair into the campus's physical environment. We will look at a specific service for students with mobility disabilities: ensuring a convenient route for moving the student's wheelchair to the study hall or lab. The student is introduced into the ViPS Space through his smart wheelchair and Personal Assistant situated into his mobile device.

For modeling purposes, an ambient can be represented as a structure with the following elements:

- Identifier (name) – a mandatory element that, in addition to identification, also serves to control access.
- Corresponding set of local agents (threads, processes) – these are computational processes that works directly in the ambient and in a sense control it.
- A set of sub-ambients, that also have identifiers, agents, and other sub-ambients. Thus, there is the possibility of recursively building different complex ambients' structures.

The wheelchair has a variety of physical sensors that collect information about changing environmental parameters and interact with the modeled cyber-physical space. We will consider the wheelchair (and the active area that it creates) as a separate Ambient, with internal Ambients: the student's mobile device and the personal assistant PA. When the wheelchair physically enters the university campus, an automatic identification process is started. After verification ViPS sends a response to the student's PA, containing a list of appropriate services for him/her. The guard assistants (GA) provide up-to-day information on the activity and performance of the important zones for the wheelchair. Upon leaving the university campus, the active area of the wheelchair stops interacting with the space and follows the student's automatic logout, which terminates the use of the provided services.

After recognizing the wheelchair (and the student in it), the system activates personal assistant (PA_i) of the student, which takes care of the delivery of all educational services and learning resources to the mobile device, like all other students. The specific services that the environment will provide to this student are mainly related to his/her mobility in the physical space of the university campus. Once the student receives the list of all services (sList) from his PA_i, the student understands at which floor and in which room the relevant training session or exam will be held. This information must be delivered at a certain time before the event begins so that the student with the wheelchair can move to the appropriate room in the university building. The wheelchair has to pass through a series of important points (zones) such as ramps, lifts, opening doors and more. Each of these important points is provided with a collection of sensors that dynamically provide actual information to GA.

Once the PA receives information about the upcoming event, it sends a message to ANet Ambient in the Analytical Subspace (AS ambient) asking for an appropriate route to be generated. ANet Ambient starts a bidirectional communication process with the corresponding GA for providing of up-to-date information from the physical world. After receiving the list of currently active zones from GA, ANet Ambient generates a list of appropriate routes and sends it to the student's PA. From the received information, the student chooses a route and sends it to the wheelchair (Cart ambient). When the wheelchair moves in the physical campus, GA tracks its location and, when it is close to some of the important zones, activates the sensors associated with opening the doors, providing a lift, etc. If in real time any of the important zones changes its status and becomes inactive, GA promptly informs PA, which expects the student to choose a new route.

Once the SA_RG assistant has received a request from the ANet to generate a route to the port, it chooses from the AmbiNet Repository an appropriate template of a virtual model of the university campus with stationary ambients – the study halls, computer labs etc. The SA_RG assistant requires from the GA additional information about the location of mobile ambients (lifts, ramps). Finally, taking into account the user's current location and using certain algorithm, it generates a list of possible routes (ListRoutes) and returns it to the ANet assistant. The route is a sequence of important zones (iz_1, iz_2, ...). The search for suitable routes can be realized through different algorithms – graph search, genetic and heuristic algorithms, neural networks, etc. Since the map with the location of all study halls and labs is known, one way to improve the search efficiency is to set a list with limited number of important zones (islands) through which the search is done with fewer steps. For example, if we are looking for a path between a wheelchair's current location and the destination, it is appropriate to limit the search to a few steps: first to the nearest object, then to some of the other ones and so on, to the destination. Intuitively, these particular positions are the Search Graph Islands, which are required to be a mandatory part of the route to move from a location to a destination. Once the islands are identified, we can break the common problem of searching into several simpler problems. This reduces the search space because, instead of dealing with one solving problem, a few simpler ones are considered. The search for a path between the current location and the destination by using this approach is realized using the following formally described algorithm:

- Identify a set of islands (important zones) $iz_1, iz_2, \ldots, iz_k$;
- Find paths from a location to iz_1, from iz_{j-1} to iz_j for each j, and from iz_k to a destination.

This algorithm does not guarantee the detection of an optimal route but it ensures the discovery of one or several possible routes (listRoutes). Optimality can be achieved both in terms of time and distance and the number of important zones for the wheelchair, such as escalators, elevators, ramps, etc. Therefore, a convenient route will be any possible route that satisfies the particular student's wishes and preferences, such as the most short time, the shortest distance, or the low-risk route. If we still want to provide the student with the optimal route, we need to additionally implement an optimization algorithm. The optimization methods for selecting the route with the lowest value are also different; one of them is Kruskal's algorithm. Let us look at the weighted graph $G(V, E)$ where V is the list of vertices and E is the list of weighted edges. The weight of each edge between the vertices i and j is $f(i, j)$. The goal is to find the smallest optimal value (the shortest time in our case) simultaneously with the route search. The algorithm includes the following main steps:

- We create sets, whereas in the i-th set we initially place only the i-th vertex of the graph;
- We sort all edges in ascending order of $f(i, j)$;
- We create a blank tree $T(V, E)$, which, after finishing the algorithm, will contain the tree with the fastest route;
- If the number of vertices is n, we add in succession (n – 1 times) the edge (i, j) from E to T so that the weight of $f(i, j)$ is the next smallest of the sorted list in step 2 and the vertices i and j are from different sets. After each added edge, we unify the sets in which the vertices i and j are located.

The complexity of the algorithm depends on several factors; the steps for initializing the sets are $\Theta(n)$. The pyramidal sorting in step 2 has complexity $\Theta(m\log_2 m)$, where m is the number of edges. The verification of whether the two vertices of the edge belong to one and the same set at each extension of the tree with a new edge, and with new peaks respectively, has the complexity of $\Theta(\log_2 n)$. Since the total number of steps is $n - 1$, the overall complexity of the algorithm is $\Theta(m\log_2 m + n\log_2 n)$.

The main ambients used in the CCA model of the described scenario are presented in the following Table 1.

Table 1. CCA Ambients in the simple scenario

Notation	Description	Notation	Description
PA	Personal assistant	ANet	Ambient net
AS	Analytical subspace	SA_RG	Specialist assistant for route generation
IoTN	IoT nodes	GA	Guard assistants

Let's imagine that the personal assistant PA received information about the lecture on Object-oriented Programming, which will take place in a 424 study hall at 10 o'clock. CCA processes of these ambients can be modeled as follows:

$$P_{PA} \triangleq \begin{pmatrix} !::(lecture\_OOP, room\_424, time\_10).AS ::< location, room\_424, PAi >.0\,| \\ !AS :: (ListRoutes).0 \end{pmatrix} \quad (1)$$

$$P_{AS} \triangleq \begin{pmatrix} !PA :: (location, room\_424, PAi).ANet \downarrow < location, room\_424, PAi >.0\,| \\ !ANet \downarrow (ListRoutes, PAi).PA ::< ListRoutes >.0\,| \\ ANet \downarrow (room\_424, PAi).GA ::< room\_424, PAi >.0\,| \\ GA :: (ListIZ, PAi).ANet \downarrow < ListIZ, PAi >.0 \end{pmatrix} \quad (2)$$

$$P_{ANet} \triangleq \begin{pmatrix} !AS \uparrow (location, room\_424, PAi).AS \uparrow < room\_424, PAi > .0 \,| \\ !AS \uparrow (ListIZ, PAi).SA\_RG \downarrow < location, ListIZ, PAi > .0 \,| \\ !SA\_RG \downarrow (ListRoutes, PAi).AS \uparrow < ListRoutes, PAi > .0 \end{pmatrix} \quad (3)$$

$$P_{SA\_RG} \triangleq \left(!ANet \uparrow (location, ListIZ, PAi).ANet \uparrow < ListRoutes, PAi > .0 \,| \right) \quad (4)$$

$$P_{GA} \triangleq \begin{pmatrix} !AS :: (room\_424, PAi).IoTN ::< PAi > .0 \,| \\ IoTN :: (ListIZ, PAi).AS ::< ListIZ, PAi > .0 \end{pmatrix} \quad (5)$$

$$P_{IoTN} \triangleq \left(!GA :: (PAi).GA ::< ListIZ, PAi > .0 \right) \quad (6)$$

For describing of CCA processes, we use the programming language ccaPL. The interpreter of ccaPL has been developed as a Java application. Based on the main version [1], we developed a special simulator for verifcation the scenario described above. The notation “A === (X) ===> B” means that Ambient “A” sends an “X” message to Ambient “B”. “Child to parent”, “Parent to child”, and “Sibling to sibling” provide information about the relationship between sender A and recipient B according to the hierarchy of ambients. The scenario that we presented has the following ccaPL program realization:

```
PA[     AS::send(location,room_424,PAi).0| AS::recv(ListRoutes).0] |
AS[
PA::recv(location,room_424,PAi).ANet#send(location,room_424,PAi).0|
ANet#recv(room_424,PAi).GA::send(room_424,PAi).0|
GA::recv(ListIZ,PAi).ANet#send(ListIZ,PAi).0|
ANet#recv(ListRoutes,PAi).PA::send(ListRoutes).0|
ANet[
AS@recv(location,room_424,PAi).AS@send(room_424,PAi). 0|
AS@recv(ListIZ,PAi).SA_RG#send(location,LiztIZ,PAi).0|
SA_RG#recv(ListRoutes,PAi).AS@send(ListRoutes,PAi).0|
SA_RG[
ANet@recv(location,ListIZ,PAi).ANet@send(ListRoutes,PAi).0|
]]]|
GA[
AS::recv(room_424,PAi).IoTN::send(PAi).0|
IoTN::recv(ListIZ,PAi).AS::send(ListIZ,PAi).0]|
IoTN[GA::recv(PAi).GA::send(ListIZ,PAi).0]
```

Implementation of the CCA-model in the ccaPL environment allows to track the processes of the participating ambients as well and the sequence of the sent and received messages between them (Fig. 3).

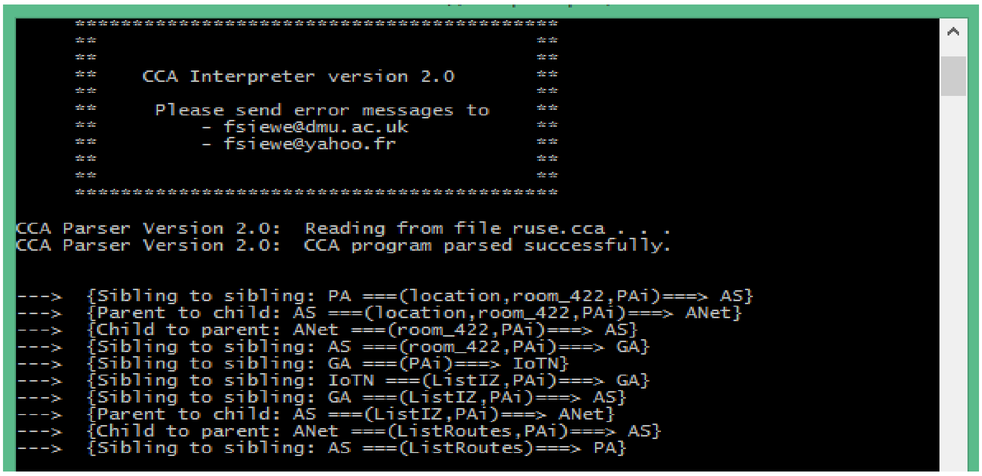

Fig. 3 Implementation of the scenario in the simulator

The animator allows visualization of the participating ambients, their location and processes. The processes of all participating ambients can be traced to each step of the scenario implementation, which makes it possible to immediately identify errors and inconsistencies in modelled processes and interactions.

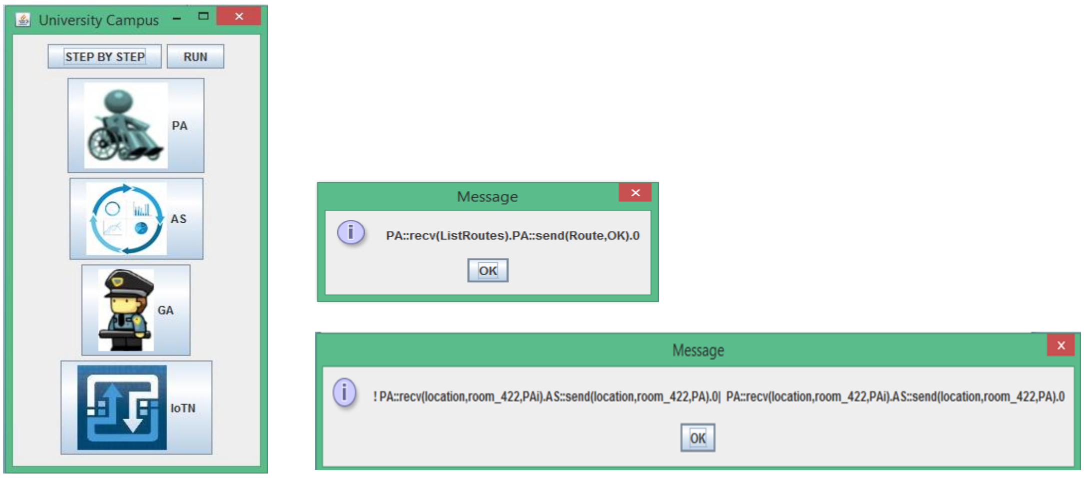

Fig. 4 Implementation of the scenario in the animator

As can be seen in Fig. 3 and Fig. 4, the visualization of the ambients' interaction in the current version of the ccaPL simulator is not clear and readable enough for all users who will use the model. This once again justifies the need to develop a special AmbiNet Visualizer, where the interactions and dependencies between the participating ambients will are presented in an animated or schematic form.

Conclusion and future directions

For years we have been using DeLC in real education. In line with our practice, ignoring the realities in the surrounding physical environment is one essential defect, particularly disadvantageous for a group of users, namely disabled learners. At the same time, the rapid enforcement of the concepts of CPSS and IoT, supporting the integration of virtual and physical spaces, reveals new opportunities to overcome this disadvantage. As a result, in this article we present a new architecture of an IoT ecosystem that can effectively assist people with disabilities in their movement. This new opportunity is demonstrated by an example scenario.

Our future intentions are mainly about improving architecture to have full support for virtualization. Essential to build IoT applications is the way to virtualize “things” from the physical world, that is, representation of physical objects in the virtual world. Therefor the virtualized things must be provided with spatial and temporal attributes, as well as with corresponding events. It is especially useful to support virtualization by formal means. In this way ViPS could be adapted to new IoT services for disabled people, not only within the university campus but within a smart city for example.

Acknowledgments

The research is partly supported by the Bulgarian Ministry of Education and Science under Grant No. D01-221/03.12.2018 for NCDSC – part of the Bulgarian National Roadmap on RIs, and by Project FP19-FMI-002 “Innovative ICT for Digital Research Area in Mathematics, Informatics and Pedagogy of Education” of the Scientific Fund of the University of Plovdiv.

References

1. Al-Sammarraie M. H. (2011). Policy-based Approach for Context-aware Systems, Ph.D. Thesis, Leicester, UK: STRL, De Montfort University.
2. Ashton K. (2009). That ‘Internet of Things’ Thing. In the Real World Things Matter More Than Ideas, RFID Journal, 22, 65-78.
3. Cardelli L., A. D. Gordon (2000). Mobile Ambients, Theoretical Computer Science, 240, 177-213.
4. Companions-project, https://cordis.europa.eu/project/rcn/96289/factsheet/en (access date 31 July 2019).
5. Cook D. J., J. C. Augusto, V. R. Jakkula (2009). Ambient Intelligence: Technologies, Applications, and Opportunities, Pervasive and Mobile Computing, 5(4), 277-298.
6. Glushkova T., M. Miteva, A. Stoyanova-Doycheva, V. Ivanova, S. Stoyanov (2018). Implementation of a Personal Internet of Thing Tourist Guide, American Journal of Computation, Communication and Control, 5(2), 39-51.
7. Glushkova T., S. Stoyanov, I. Popchev, S. Cheresharov (2018). Ambient-oriented Modeling in a Virtual Educational Space, Comptes rendus de l’Academie bulgare des Sciences, 71(3), 398-406.
8. Google Now, http://www.digitaltrends.com/mobile/how-to-use-google-now/ (access date 31 July 2019).
9. Guglev Z., S. Stoyanov (2018). Hybrid Approach for Manipulation of Events in the Virtual Referent Space, Computer Science and Communications, 7(1), https://csc.bfu.bg/index.php/CSC/article/view/210 (access date 31 July 2019).
10. Microsoft Cortana, https://www.microsoft.com/en-us/cortana (access date 31 July 2019).
11. Milner R., J. Parrow, D. Walker (1992). A Calculus of Mobile Processes, Parts 1-2, Inform and Comput, 100(1), 1-77.
12. Moszkowski B. (1998). Compositional Reasoning Using Interval Temporal Logic and Tempura, LNCS, 1536, 439-464.
13. PAL, Personal Assistant for a healthy Lifestyle, http://www.pal4u.eu/ (access date 31 July 2019).
14. PAL, The PAL Framework, https://pal.sri.com/, (access date 31 July 2019).
15. Panchanathan S., S. Chakraborty, T. McDaniel (2016), Social Interaction Assistant: A Person-centered, Approach to Enrich Social Interactions for Individuals with Visual Impairments, IEEE Journal of Selected Topics in Signal Processing, 10(5), 942-951.

16. Raffler H. (2006). Other Perspectives on Ambient Intelligence, www.research.philips.com/password/archive/23/pw23_ambintel_other.html (access date 31 July 2019)
17. Santos J., B. Silva, J. Rodrigues, J. Casal, K. Saleem (2015), Internet of Things Mobile Gateway Services for Intelligent Personal Assistants, 17th International Conference on E-health Networking, Application & Services (HealthCom), Boston, MA, USA, October 14-17, 311-316.
18. Siewe F., H. Zedan, A. Cau (2011). The Calculus of Context-aware Ambients, Journal of Computer and System Sciences, 77(4), 597-620.
19. Siri, http://www.apple.com/ios/siri/ (access date 31 July 2019).
20. Stoyanov S. (2012). Context-aware and Adaptable eLearning Systems, Internal Report, Software Technology Research Laboratory, De Montfort University, Leicester, UK.
21. Stoyanov S., A. Stoyanova-Doycheva, T. Glushkova, E. Doychev (2018). Virtual Physical Space – an Architecture Supporting Internet of Things Applications, XXth International Symposium on Electrical Apparatus and Technologies (SIELA 2018), June 3-6, Bourgas, Bulgaria, doi: 10.1109/SIELA.2018.8447156.
22. Stoyanov S., D. Orozova, I. Popchev (2016). Virtual Education Space – Present and Future, Jubilee Scientific Conference with International Participation, Burgas Free University, 410-418.
23. Stoyanov S., D. Orozova, I. Popchev (2018). Internet of Things Water Monitoring for a Smart Seaside City, XXth International Symposium on Electrical Apparatus and Technologies (SIELA 2018), June 3-6, Bourgas, Bulgaria, doi: 10.1109/SIELA.2018.8447084.
24. Stoyanov S., I. Ganchev, I. Popchev, M. O'Droma (2005). From CBT to E-learning, Journal Information Technologies and Control, No. 4, Year III, 2-10.
25. Stoyanov S., I. Popchev (2015). E-learning Infrastructures, Technosphere Journal, 4(30), 38-45.
26. Stoyanov S., I. Popchev, E. Doychev, D. Mitev, V. Valkanov, A. Stoyanova-Doycheva, V. Valkanova, I. Minov (2010). DeLC Educational Portal, Cybernetics and Information Technologies, 10(3), 49-69.
27. Tokunaga S., H. Horiuchi, K. Tamamizu, S. Saiki, M. Nakamura, K. Yasuda (2016). Deploying Service Integration Agent for Personalized Smart Elderly Care, 15th IEEE/ACIS International Conference on Computer and Information Science, Okayama, Japan, June 26-29, 897-902.

Assoc. Prof. Todorka Glushkova, Ph.D.
E-mail: glushkova@uni-plovdiv.bg

Todorka Glushkova received M.Sc. Degree in Informatics, Plovdiv University Paisii Hilendarski, Bulgaria in 2003 and Ph.D. Degree in Informatics and Computer Sciences in 2011 from Plovdiv University too. Since 2013 she has been working in Faculty of Mathematics and Informatics in Plovdiv University. From 2018 she is an Associated Professor. Her current research interests are in the field of modeling of context-aware educational spaces, adaptive e-learning services and e-learning resources.

Prof. Stanimir Stoyanov, D.Sc.
E-mail: stani@uni-plovdiv.bg

Prof. Stanimir Stoyanov is Member of the Institute of Electrical and Electronic Engineers (IEEE) and of the Association for Computing Machinery (ACM). He received his doctoral degrees from the Humboldt University of Berlin, Germany and the De Montfort University of Leicester, UK. Currently, he is a Lecturer in the Plovdiv University Paisii Hilendaski, Bulgaria, where he also acts as a Head of the Department of Computer Systems. His research interests include intelligent agents, agent- and service-oriented architectures, IoT, context-aware and adaptable software, eLearning tools and environments. He has published more than 200 and 5 books.

Acad. Ivan Popchev, D.Sc.
E-mail: ipopchev@iit.bas.bg

Academician Ivan Popchev since 2003 has been a member of the Bulgarian Academy of Sciences. He received Ph.D. Degree in 1967, with a thesis “Studies on the Performance Index of Large Scale Control Systems” and D.Sc. in 1980 with a thesis “Multicriterial Synthesis of Control Systems”. He has more than 480 papers presented at world congresses, conferences, symposia and workshops and/or published in international journals /periodicals/, proceedings and books in English, German, French, Russian, Korean and Bulgarian. His research interests are in the fields of computer sciences, mathematical modelling, decision sciences and control theory.