computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

AMBINET – AN ENVIRONMENT FOR AMBIENT-ORIENTED MODELING

Todorka Glushkova [1], Stanimir Stoyanov [1, 2], Asya Stoyanova-Doycheva [1], Vanya Ivanova [1], Lyubka Doukovska [2]

[1] Faculty of Mathematics and Informatics, University of Plovdiv, Bulgaria Blvd., 236, 4027 Plovdiv, glushkova@uni-plovdiv.bg, stani@uni-plovdiv.net, astoyanova@uni-plovdiv.net, vantod@abv.bg, l.doukovska@mail.bg

Paper history:
Received 9 May 2019
Received in revised form 28 August 2019
Accepted 10 September 2019
Available online 30 September 2019

Keywords:
Calculus of Context-aware Ambients;
Cyber-Physical-Social Systems;
Intelligent agents;
Smart city.

Abstract: The concept of the Internet of Things (IoT) is closely related to the concepts of Cyber-Physical System (CPS) and Cyber-Physical-Social System (CPSS). A key feature of these technologies is the integration of the virtual and physical world. In this paper, an environment for ambient-oriented modeling called AmbiNet is presented. The environment AmbiNet is implemented as a component of the reference architecture known as Virtual Physical Space (ViPS) that can be adapted for CPSS applications in various domains, for example a smart city, a personal touristic guide, or education. The need for virtualization of things from the physical world in a formal way is also considered. In the paper, the usability of the environment is demonstrated by modeling of services delivered to tourists in an intelligent city. The architecture of ViPS is also briefly described. Furthermore, the virtualization and modeling of spatial aspects through the AmbiNet formalism is demonstrated by an example.

1. INTRODUCTION

Cyber-Physical Systems (CPS) and Internet of Things (IoT) are closely related concepts. The IoT paradigm will make it possible for virtually any thing around us to exchange information and work in synergy with each other [1] bringing the communication on a higher semantically enriched level – it is no longer just a transportation method for meaningless messages; it becomes an act of knowledge exchange and accumulation. This paradigm easily applies to each dynamic environment that is capable of self-adjustment based on sensed changes and analyzed knowledge gained and shared among the components that inhabit it [2]. CPSs stress the tight coordination between the computational and physical resources and the integration of the virtual and physical worlds [3]. By placing the person (the user) in the center of such spaces, they become Cyber-Physical-Social Spaces (CPSSs). From a software architecture point of view, a CPSS includes a variety of 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, context-aware, 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 DeLC (Distributed eLearning Center) has been used for years. 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. 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. A component for presenting the location of objects of interest in the physical world was implemented using a formal ambient-oriented approach. The use of this component is demonstrated to assist disadvantaged

learners. Summing up the experience of constructing VES, we began developing a reference architecture known as Virtual Physical Space (ViPS) [4] that could be adapted to various CPSS-like applications such as a smart seaside city, a personal tourist guide [5, 22], and education. 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. To present the spatial aspects of “things”, we have decided to use an ambient-oriented modeling approach. In this paper, we present the AmbiNet environment that supports the chosen approach.

Due to the hardware and software complexity, as well as heterogeneity, the development of a CPSS application implies serious risks. It is inappropriate, inefficient, and financially disadvantageous to directly build such an application. Previous modeling would be of great help, where the system model was researched and analyzed to identify weaknesses and errors. In the case of VES, we are going to explore various approaches for building an integrated modeling environment. Usually, events, time, and locations are considered as fundamental concepts modeling the application. In this paper we present the modeling of spatial aspects of “things” using the AmbiNet environment.

The rest of the paper is organized as follows: a short review of ambients and ambient intelligence is considered in Section 2, which is followed by an overall description of the ViPS architecture in Section 3. Section 4 addresses the architecture of AmbiNet. Section 5 demonstrates the application of AmbiNet to model a tourist service in a smart city. Finally, Section 6 concludes the paper.

2. AMBIENT – ORIENTED MODELING

Due to characteristics such as restriction (a limited location where an interesting calculation happens), input (an ambient can be nested into other ambients), and mobility (an ambient can be moved from one location to another), ambients are a suitable means for modelling of spatial aspects of “things” in an IoT system. Ambient intelligence (AmI) makes the day-to-day information environment of users more "sensitive" to their problems and peculiarities by adding environmental sensors and actuators. AmI is growing rapidly as a multidisciplinary topic of scientific and practical interest [6]. The basic idea behind AmI is that applications operating in an environment are able to use real-time aggregated information and background data accumulated over time to make decisions in favor of users. In an AmI system, to provide flexibility, adaptability, predictability, and acceptable interface, Artificial Intelligence plays a key role. In [7], intelligence is highlighted as a basic aspect of an AmI system being defined as a digital environment that proactively but reasonably supports people in their everyday life. A typical example are personal assistants that, depending on the situation, are able to provide proactive help or exercise restraint. Sensible requirements may be abilities to recognize the user, to learn or know preferences and likes, to show empathy to the moods of users; they are termed as the "5Ws" (Who, Where, What, When and Why) in [8].

The convergence of AmI and IoT [9] aims to build environments, in which intelligent Web-enabled objects collaborate by exchanging their own data and services, thus providing comprehensive services to users. According to the vision of AmI and IoT, the devices are widespread in the environment in the form of sensors, actuators, intelligent appliances, active labeled objects, or mobile robots. These devices are interconnected and can always be accessed from anywhere through a variety of small portable mobile devices. Compared to classic web services, AmI and IoT provide not only computing and software resources but also various physical and event control capabilities [10]. Convergence can be especially effective for disabled people, the elderly, children, or patients where the ubiquitous and the environment can provide multiple reactive or proactive services to improve the quality of life. Using IoT and AmI technologies, the number of user support services is steadily increasing. These services are dynamic due to frequent changes occurring in the environment.

A suitable approach to modeling IoT systems is Ambient-Oriented Modeling (AOM) and the fundamental formalism used in various AOM systems is the π–calculus as a kind of process calculus [11]. A basic concept in the calculus is that of a channel where a channel is able to move through other channels. Processes are modeled as channel movements. Despite some shortcomings, the concepts of π–calculus are a solid basis for creating a number of formalisms. In [12], a version of π-calculus is presented extended with primitives, which allow migration between naming locations. The Join-Calculus [13] reformulates the π-calculus with a clearer idea of the interaction locations. An important extension known as Ambient Calculus (AC) adopts a mobility paradigm [14]. In AC, ambients are hierarchically structured, agents are limited to ambients, and ambients move under the control of agents. A novelty is the movement of self-contained nested environments that include background data and real-time computing as opposed to more commonly used techniques that move individual agents or objects. The Calculus of

Boxed Ambients (CBA) [15] occurs as a successor of AC.

IoT and pervasive computing are new paradigms to make computers integrated into everyday life activities. Pervasive computing adds the concept of mobility and context-awareness to the distributed systems. Within a system, users and devices can be mobile. In order to understand the context, it is required that the system be able to sense the context in its environment and use the information to adapt its behavior to the current situation. The Calculus of Context-aware Ambients (CCA) is a process calculus based on the notion of ambients. An ambient is an entity describing any object or component (e.g. person, process, device, location, etc.) in a system defined as a bounded place where computations happen [16]. It has a name, a boundary, and can have built-in ambients. An ambient can be mobile and it has the ability to communicate with other ambients. CCA is used to model ambients in terms of process, location and capability. CCA extends CBA with new constructions to allow mobile ambients to respond to changes in the environment, in which they are running. Consequently, mobility and awareness of context are of particular importance. To represent the properties of CCA processes, a programming language is available. Contextual expressions can be used to ensure fulfillment of a specific opportunity only under certain conditions of the environment, i.e. in a particular context.

3. VIPS IN A NUTSHELL

Creating a CPSS application addresses, besides everything else, the notions of “things” and “intelligent objects”. Furthermore, another essential aspect is the integration of the physical and virtual worlds. In order to implement such an integration, each physical "thing" has to be “virtualized” in the virtual world. Usually, the virtualization and digitization of any real physical object depends on the specific domain of interest. 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. The architecture also takes into account events related to these objects as well as their temporal and spatial aspects. The virtualization of physical objects is mainly supported by the ViPS middleware. A “virtualized thing” is modeled as a complex structure including not only the thing’s features but also characteristics such as events, time, and location. In the Digital Libraries Subspace information is saved about the domains of interest, for example, objects related to tourism, medical assistance, transport, eco-agriculture and aquaculture, etc. In the architecture, the analytical subspace is the central part consisting of 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 by using of Interval Temporal Logics (ITL)

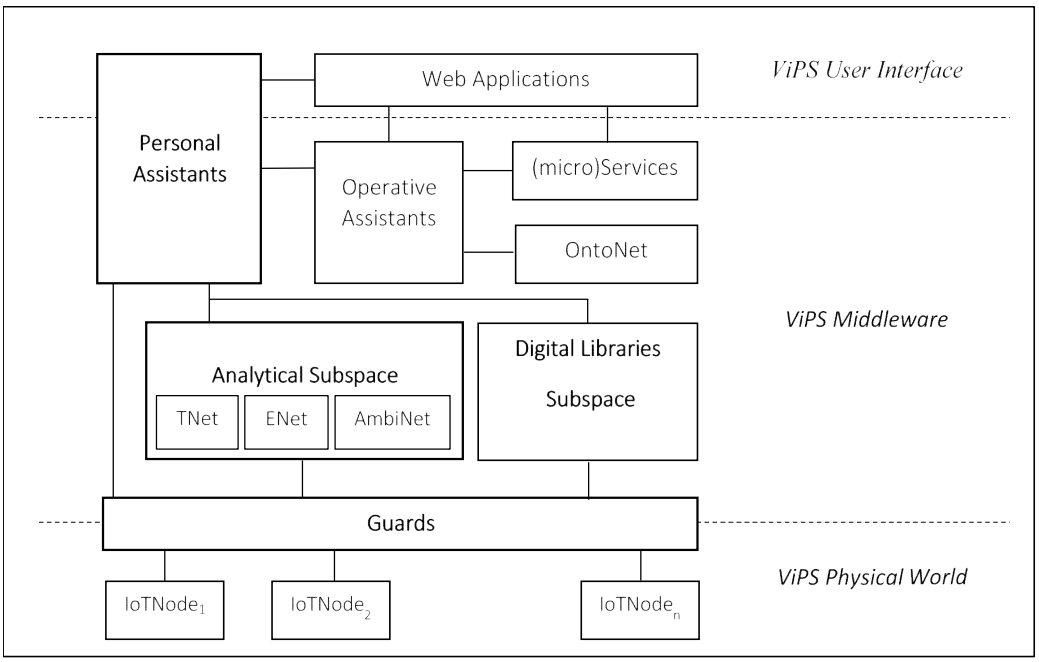


Figure 1 – Architecture of ViPS

[17].

- AmbiNet models the spatial characteristics of "things" and events as a network of ambients.

The operative assistants provide access to the resources of both subspaces and interact both with the user's 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 appropriate 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 OntoNet layer is a hierarchy of ontologies, which represents the essential features of things. Furthermore, the relationships between the "things" are specified in the OntoNet.

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.

4. AMBINET ENVIRONMENT

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 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, a tourist guide, or education. Electronic services can 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 (Fig. 2.) consists of software modules to assist users in:

- Visually building CCA models;
- Conducting experiments with these models.

The core of the system is the AmbiNet ccaPL Interpreter [18]. This is an interpreter of the formal modeling language, based on the Calculus of Context-aware Ambients (CCA). In the first version of the interpreter [19] users had to prepare the models in the ccaPL language, which is a difficult and tedious task. To facilitate the development of AmbiNet, it delivers a specialized visual CCA modeling editor called the AmbiNet Editor.

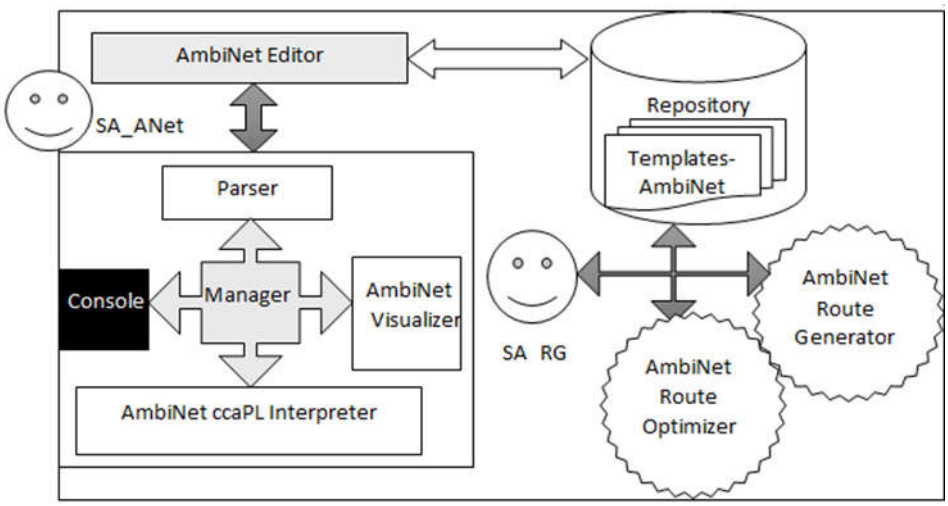


Figure 2 – AmbiNet Architecture

Depending on their location, ambients can be static and dynamic. 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 sub-ambient structure of faculties, corps, lecture rooms, and so on. Dynamic 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 dynamic ambients. Typically, if they change their location, then all the “ambient-children" move as well.

An essential requirement is that this editor can be adapted for use in different application areas. For this purpose, it works with pre-prepared templates. Each template is a static specification of a certain class of things from the domain of interest. For example, for a smart city, it is advisable to develop "stuff" templates such as buildings, stops, crossings and traffic lights, public transport vehicles, etc. The ready templates are used to visually create CCA models. Depending on the type of presentation, "things" can be parameterized to generate a particular instance that will become an element of the model. Each template of a "thing" has a visual representation (icon) that the designer works with. The icon can be positioned in the model, removed, or moved.

Multiple templates for a problem area are stored in a separate repository. When adapting the editor for a selected problem area, the corresponding repository is updated, giving designers access to the AmbiNet Repository templates.

Building CCA models with the visual editor takes the following steps:

- The icons of static objects are placed in the work area; it is also possible to build hierarchical structures of ambients with different levels of inclusions;
- Icons (templates) are parameterized to generate a specific instance (ambient);
- The relationships between ambients are specified. These can be, for example, communication channels of different types, messaging capabilities, etc.;
- The created model is saved in the appropriate repository.

The editor can run at the repository level; what is more, it can add, remove, and update existing templates.

In the first version of AmbiNet, the visualization of experimental results (modeling) was part of the interpreter itself. The results were presented mainly in textual form (primitive animation was also possible), which made it difficult to analyze them. In the current version, a standalone component called the current version of AmbiNet Visualizer is being developed, displaying the experimental results in a way that is creative and easy to understand and interpret (e.g. virtual reality).

Another component of AmbiNet is the AmbiNet Route Generator. In many cases, in addition to presenting the location, it is necessary to look for an appropriate route that needs to be accomplished in the physical space. In a CPSS, the real world is modeled through an ambient-oriented context-aware network. Especially in AmbiNet, for each individual context – a city, an 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.

Different methods of searching for routes are known in the specialized literature. Here, the idea of "island search" is appropriate, because in CPSS applications the status and usability of the devices involved in the route are of particular importance (especially for the disadvantaged). In a nutshell, our approach to generating routes in virtual physical spaces includes the following steps:

- Static network – the nodes of this network are all existing installations (devices, vehicles) designed to support the movement of people in the area under consideration. The nodes themselves are modeled as ambients; ribs are transitions between these nodes.
- Dynamic subnet – Depending on the actual and target location of the user, this subnetwork is dynamically selected. It includes only those devices that can be directly used in the specific case.
- Possible routes – A variety of possible routes are generated in the dynamic network. A possible route includes only those nodes that can help reach the target location. For example, people in wheelchairs need devices that provide wheelchair accessibility.

The AmbiNet Route Optimizer works in close connection with the AmbiNet Route Generator. From the possible routes a user may select a specific one that can be executed. In order to determine the route's performance, the AmbiNet Route Optimizer interacts with its ambients to check their actual instant operating capability.

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

Figure 3 – Template of a Smart City with Ambients on it

the next section in the demonstration of AmbiNet for a particular service (Fig. 3.).

5. AMBINET IN ACTION

Let AMB be the set of ambients in the space. An ambient A = <name, location, type, parent, P (a), attr> ∈ AMB where:

- **name** is the ambient identifier;
- **location** is the location of the ambient in the physical world; for virtual ambients, location = null;
- **type** is the type of ambient (static, dynamic, abstract);
- **parent** is the parental ambient; if there is none, parent = null;
- **P (a)** is the plurality of the ambient's processes;
- **attr** is a set of additional attributes of the ambient. Personal assistants, operational assistants, guards, and other virtual components can be represented as ambients with type=dynamic and location=null.

An ambient A1 is a child of ambient A, if A = parent (A1), i.e. the representation of the ambient hierarchy may be a recursive structure. Let the ambients A1 and A ∈ AMB be given. We define the following two operations:

- A1 ↓ A (in) – ambient A1 enters the boundary of ambient A and becomes its child, i.e. A = parent (A1);
- A1 ↑ A (out) – ambient A1 goes out of A boundaries as A and A1 become a sibling, i.e. parent (A) = parent (A1)

The ambients A1 and A2 ∈ AMB can communicate with each other by sending and receiving messages as follows:

- $A1 \xrightarrow{<message>} A2$, if A2=parent(A1) or A1=parent(A2) or parent(A1)=parent(A1) (*)
- For example, let us consider ambients A1, A2, A3, and A4 as:
  - A1=<Hotel, loc_A1, static, null, P(A1), attr1> – a Hotel
  - A2=<Wheelchair, loc_A2, dynamic, Hotel, P(A2), attr2> – the user's wheelchair in the Hotel
  - A3=<422room, loc_A3, static, Hotel, P(A3), attr3> – a room in the Hotel
  - A4=<Lift, loc_A4, dynamic, Hotel, P(A4), attr4> – a Lift in the Hotel

Ambients A1 and A3 are static but ambients A2 and A4 are dynamic; A1 has no parent in the current context (parent (A1) = null), while A2, A3, and A4 are children of A1, i.e. A1 = parent (A2) = parent (A3) = parent (A4). The location of dynamic ambients may change; it is captured by appropriate sensors and transmitted dynamically. Let us assume that the user's wheelchair enters the elevator. Then:

A2=<Wheelchair, loc_A2, dynamic, Lift, P(A2)> as Lift=parent(A2) and loc_A2=loc_A4. Recursively: Hotel = parent (parent (A2)). If ambient A2 wishes to send a message to ambient A3, this cannot be done directly, as A2 and A3 meet none of the conditions (*) and are not in any of the tolerable relationships: parent-child or child-child. The transmission of the message can only be done by retransmission from the parents in the hierarchy. In this case:

$A2 \xrightarrow{<message>} A4$, because A4=parent(A2); after that $A4 \xrightarrow{<message>} A3$, as parent(A4)= parent(A3).

Modeling the infrastructure of a "smart" city is a complex task [20, 21]. To develop such a model, it is necessary to virtualize different types of "things" such as those related to people's ability to travel, specific public places, special facilities for people with disabilities, water cleanliness monitoring facilities and other environmental problems, and so on. Through such a model we can explore the behavior of the services that an intelligent city can offer. Each user has his/her personal assistant that offers him/her personalized help and support when using the services in the intelligent city. This can be done applying the following procedure:

- Using the mobile device, the user selects the desired service provided by the Smart City;
- The Smart City interacts with the personal assistant to prepare a personalized service plan for the user, taking into account his/her location, desired destination, ways of travel, the weather, and other circumstances.

Let us take a look at the following simple scenario: A user with movement difficulties is housed in a hotel and wishes to visit the ancient fortress over the city. He/she has an intelligent wheelchair and a personal assistant with which he/she communicates through his/her mobile device.

After communicating with the CPSS space of the Smart City, the personal assistant creates a plan to achieve the goal and, following the generated route, assists to move the wheelchair from the hotel to the fortress.

Upon arrival at the fortress, the wheelchair recognizes some life-threatening signs and by communicating with the Smart City, the personal assistant provides transportation of the tourist to the City Hospital. We will look at the two parts of this scenario:

Part 1: Finding a suitable route to visit the fortress. The trip can be done by public transport, by taxi or by using of the pedestrian promenade;

Part 2: Providing the fastest route for transporting the tourist to the local hospital. The journey will be made with a "smart" ambulance that will interact dynamically with other objects such as "smart" traffic lights, emergency teams, the emergency ward in the "smart" Hospital, and so on, to provide adequate emergency medical care.

The processes of the main CCA – ambients that model the two parts of the described scenario, are shown in Fig. 4.

The interactions between the participating objects are presented as a ccaPL program and the AmbiNet ccaPL Interpreter is launched to simulate and visualize the model under consideration. A simulator of the described scenario in console mode is presented in Fig. 5. An animator interacting with the hierarchical ambient structure is visualized in Fig. 6.

$$P_{PA} \cong \begin{pmatrix} WCh ::< getLocation > .WCh :: (location).0 \mid \\ AmbiNet ::< PAi, location, WantToVisitFort > .0 \mid \\ AmbiNet :: (hasTaxi, openFortMuseum). \\ AmbiNet ::< PAi, location, getRouteToFort > .0 \mid \\ AmbiNet :: (OptimalRoute).WCh ::< startMoving > .0 \mid \\ WCh :: (location, LifeThreateningIndicators). \\ GA ::< PAi, location, TransportToHospital > .0 \mid \\ GA :: (OptimalRoute).WCh ::< OptimalRoute > .0 \end{pmatrix}$$

$$P_{WCh} \cong \begin{pmatrix} PA :: (getLocation).PA ::< location > .0 \mid \\ PA :: (startMoving).! ::< PAi, MovementByRoute > .0 \mid \\ PA ::< location, LifeThreateningIndicators > .0 \mid \\ PA :: (OptimalRoute).0 \end{pmatrix}$$

$$P_{GA} \cong \begin{pmatrix} AmbiNet :: (PAi, getActiveZones).AmbiNet ::< PAi, ListActiveZones > .0 \mid \\ PA :: (PAi, location, TransportToHospital).Hosp ::< PAi, location, needAmbul > .0 \mid \\ Hosp :: (PAi, AmbulStatus\_OK).TrL ::< PAi, TrafficLightStatus > .0 \mid \\ TrL :: (PAi, TrafficLightStatus\_OK).AmbiNet ::< PAi, location, TrafficLightStatus, AmbulStatus > .0 \mid \\ AmbiNet :: (PAi, OptimalRoute).PA ::< OptimalRoute > . \\ Hosp ::< PAi, OptimalRoute > .TrL ::< PAi, OptimalRoute > .0 \end{pmatrix}$$

$$P_{AmbiNet} \cong \begin{pmatrix} PA :: (PAi, location, WantToVisitFort).Fort ::< PAi, visitFort > .0 \mid \\ Fort :: (PAi, FortMuseumWorkingTime).Taxi ::< PAi, TravelByTaxi > .0 \mid \\ Taxi :: (PAi, hasTaxi).PA ::< hasTaxi, openFortMuseum > .0 \mid \\ PA :: (PAi, location, getRouteToFort).GA ::< PAi, getActiveZones > .0 \mid \\ GA :: (PAi, ListActiveZones).RG \downarrow < PAi, ListActiveZones, getRoutes > .0 \mid \\ RG \downarrow (PAi, ListRoutes).RO \downarrow < PAi, ListRoutes, getOptimalRoute > .0 \mid \\ RO \downarrow (PAi, OptimalRoute).PA ::< OptimalRoute > .0 \mid \\ GA :: (PAi, location, TraffLightStatus, AmbulStatus). \\ GA ::< PAi, OptimalRoute > .0 \end{pmatrix}$$

$$P_{Fort} \cong \left(AmbiNet :: (PAi, visitFort).AmbiNet ::< PAi, FortMuseumWorkingTime > .0 \right)$$

$$P_{Taxi} \cong \left(AmbiNet :: (PAi, TravelByTaxi).AmbiNet ::< PAi, hasTaxi > .0 \right)$$

$$P_{Ambul} \cong \left(Hosp :: (PAi, OptimalRoute).0 \right)$$

$$P_{Hosp} \cong \begin{pmatrix} GA :: (PAi, location, needAmbul).GA ::< PAi, AmbulStatus > .0 \mid \\ GA :: (PAi, OptimalRoute).Ambul \downarrow < PAi, OptimalRoute > .0 \end{pmatrix}$$

Figure 4 – Main CCA Process

The modeling of both parts of the scenario is done by interacting with the AmbiNet subspace and its operative assistants. The ViPS Guard system is used when there is a need for up-to-date information on the state of ambients from the physical world. AmbiNetEditor interacts with the AmbiNet Repository, where templates of the physical city are stored with all static ambients. According to the current scenario, a particular template (such as a map of the city with static ambients on it) is used, on

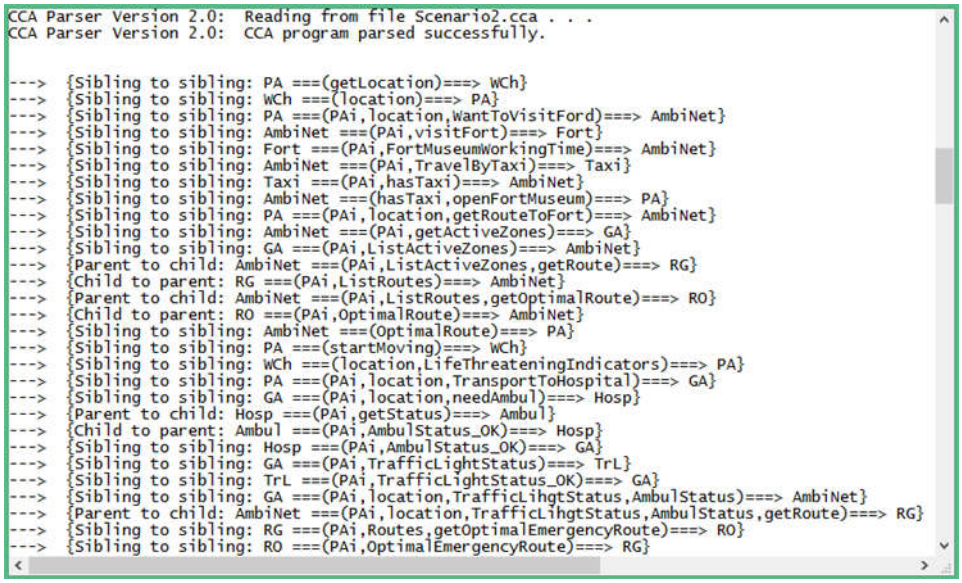


Figure 5 – A simulator of the Scenario in Console Mode

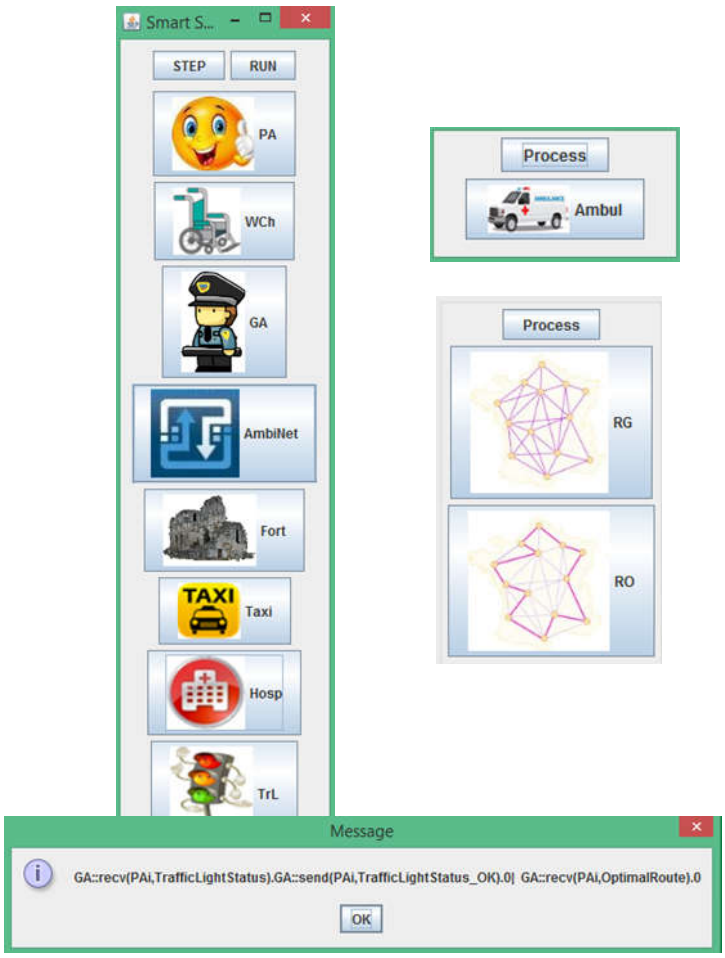


Figure 6 – AmbiNet Animator Interacting with the Ambient Hierarchy

which the dynamic ambients are modeled. The AmbiNet Route Generator component is used to generate routes, and the AmbiNet Route Optimizer – to find the optimal route. After modeling the scenarios, the interactions between the participating objects are presented as a ccaPL program and the AmbiNet ccaPL Interpreter is started to simulate and visualize the model under consideration.

Let us look at the following ambients:

A1=<Hotel, loc_hotel, static, none, P(Hotel), attr_Hotel> – a Hotel in the Smart City

A2=<wheelchair, loc_hotel, dynamic,Hotel, P(W), attr_W> – a tourist wheelchair

A3=<ramp, loc_ramp, static, none, P(ramp), attr_ramp> – a Ramp for wheelchair

A4=<Fort, loc_fort, static, none, P(Fort), attr_Fort> – an Ancient fortress

A5=<taxi, loc_taxi, dynamic, none, P(taxi), attr_taxi> – a Taxi

A6=<Hospital, loc_hosp, static, none, P(hosp), attr_hosp> – a City Hospital

A7=<Ambulance, loc_amb, dynamic, none, P(amb), attr_amb> – an Ambulance

A8=<Traff_Light, loc_tl, static, none, P(tl), attr_tl> – a Traffic light

All physical objects are equipped with sensors that transmit information about the availability and connectivity of the object (Fig. 3). In addition to the physical ambients in the script, there are also several ambassadors that have no representatives in the physical world such as:

A9=<PA, null, abstract, none, P(PA), attr_PA> – a Personal Assistant

A10=<ANet, null, abstract, AS, P(ANet), attr_ANet> – AmbiNet

A11=<RG, null, abstract, ANet, P(RG), attr_RG> – a Route Generator

A12=<RO, null, abstract, ANet, P(ANet), attr_RO> – Route Optimizer

A13=<GA, null, abstract, none, P(GA), attr_GA> – Guards

6. CONCLUSION

The article demonstrates an ambient-oriented approach that can be followed for modelling of CPSS applications in various domains. More specifically, the modeling of electronic services delivered to tourists in a smart city is considered. The experience that we have gained from the application of AmbiNet to various examples shows that ambient-oriented modeling is an adequate and effective approach to supporting the development of CPSS-like applications in a diversity of domains.

7. ACKNOWLEDGEMENT

The authors wish to acknowledge the partial support of the MES by the Grant No. D01-221/03.12.2018 for NCDSC – part of the Bulgarian National Roadmap on RIs, and the National Program "Young Scientists and Postdoctoral Students" of the Ministry of Education and Science in Bulgaria.

8. REFERENCES

[1] N. Chilamkurti, S. Zeadally, H. Chaouchi (eds.), *Next-Generation Wireless Technologies: 4G and Beyond*, Springer, London, 2013, 281 p.

[2] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, J. Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*, Cisco Systems, Inc., 2017.

[3] F.Y. Wang, "The emergence of intelligent enterprises: From CPS to CPSS," *IEEE Intelligent Systems*, vol. 25, no. 4, pp. 85–88, Jul./Aug. 2010.

[4] S. Stoyanov, A. Stoyanova-Doycheva, T. Glushkova, E. Doychev, "Virtual physical space – an architecture supporting internet of things applications," *Proceedings of the XX-th International Symposium on Electrical Apparatus and Technologies SIELA'2018*, Bourgas, Bulgaria, 3-6 June 2018, pp. 1-4, DOI: 10.1109/SIELA.2018.8447156

[5] T. Glushkova, M. Miteva, A. Stoyanova-Doycheva, V. Ivanova, S. Stoyanov, "Implementation of a personal Internet of Thing tourist guide," *American Journal of Computation, Communication and Control*, vol. 5, no. 2, pp.39-51, June 2018.

[6] C. Augusto, D. Cook, "Ambient intelligence: Applications in society and opportunities for AI," *Lecture Notes Tutorial given during 20th International Joint Conference on AI – IJCAI'07*, Hyderabad, India, 2007, pp. 1-28.

[7] H. Raffler, Other perspectives on ambient intelligence, 2006. [Online]. Available at: www.research.philips.com/password/archive/23/pw23_ambintel_other.html.

[8] K. Brooks, "The context quintet: narrative elements applied to context awareness", *Proceedings of the International Conference on Human Computer Interaction (HCI'2003)*, Erlbaum Associates, Inc., 2003, pp. 1-6.

[9] ISTAG, Ambient intelligence: From vision to reality, European Commission Report, 2003 [Online]. Available at: http://cordis.europa.eu/fp7/ict/istag/reports_en.html).

[10] A. Yachir, Y. Amirat, A. Chibani, N. Badache, "Event-aware framework for dynamic services

discovery and selection in the context of ambient intelligence and Internet of Things," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 85-102, Jan 2016.

[11] R. Milner, J. Parrow, D. Walker, "A calculus of mobile processes," *Inform. and Comput*., vol. 100, issue 1, pp. 1-77, 1992.

[12] J. Riely, M. Hennessy, "A typed language for distributed mobile processes," *Proceedings of the 25th Annual ACM Symposium on Principles of Programming Languages*, 1998, pp. 378-390.

[13] C. Fournet, G. Gonthier, "The reexive CHAM and the join-calculus," *Proceedings of the 23rd Annual ACM Symposium on Principles of Programming Languages*, 1996, pp. 372-385.

[14] L. Cardelli, A. Gordon, "Mobile ambients," *Theoretical Computer Science*, vol. 240, pp. 177–213, 2000.

[15] M. Bugliesi, G. Castagna, S. Crafa, "Access control for mobile agents: The calculus of boxed ambients," *ACM Transactions on Programming Languages and Systems*, vol. 26, issue 1, pp. 57–124, 2004.

[16] F. Siewe, H. Zedan, A. Cau, "The calculus of context-aware ambients," *Journal of Computer and System Sciences*, vol. 77, issue 4, pp. 597-620, 2011.

[17] B. Moszkowski, "Compositional reasoning using interval temporal logic and tempura," Lecture Notes in Computer Science, Springer, vol. 1536, 1998, pp. 439-464.

[18] F. Siewe, *ccaPL: a Programming Language for the Calculus of Context-aware Ambients*, Technical Report, Software Technology Research Laboratory, De Montfort University, 2011, pp. 1–11.

[19] M. H. Al-Sammarraie, *Policy-based Approach for Context-aware Systems*, Ph.D. Thesis, Leicester, UK: STRL, De Montfort University, 2011.

[20] S. De, Y. Zhou, I. L. Abad, K. Moessner, "Cyber–physical–social frameworks for urban big data systems: A survey," *Applied Sciences*, pp. 1-26, 2017.

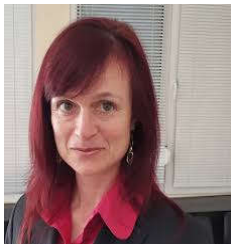
[21] P. Chamoso, A. González-Briones, S. Rodríguez, & J. M. Corchado, "Tendencies of technologies and platforms in smart cities: A state-of-the-art review," *Wireless Communications and Mobile Computing*, pp. 1-17, 2018.

[22] G. Stamatescu, I. Făgărăşan, A. Sachenko, "Sensing and data-driven control for smart building and smart city systems," Journal of Sensors, vol. 2019, article ID 4528034, pp. 1-3, 2019. https://doi.org/10.1155/2019/4528034

***Todorka Glushkova**, PhD in Artificial Intelligence, Associate Professor in the Computer Systems Department, University of Plovdiv "Paisii Hilendarski". Current research interests include Modeling of Context-aware CPSS Spaces.*

***Stanimir Stoyanov**, PhD in Artificial Intelligence and Software Engineering, Full Professor and Head of the Computer Systems Department, University of Plovdiv "Paisii Hilendarski". Research interests include Intelligent systems, modeling of Context-aware CPSS Spaces.*

***Asya Stoyanova-Doycheva**, PhD in Software Engineering, Associate Professor in the Computer Systems Department, University of Plovdiv "Paisii Hilendarski". Research interests in Software Engineering.*



***Vanya Ivanova**, PhD in Educational Sciences, Assistant Professor in the Department of Education in Mathematics, Informatics and Information Technology, University of Plovdiv "Paisii Hilendarski". Research interests in computer modeling.*

***Lyubka Doukovska**, PhD and DSc in Radar Signal Processing, Full Professor and Head of the Intelligent Systems Department in the Institute of Information and Communication Technologies, Bulgarian Academy of Sciences. Research interests: Intelligent systems and modeling.*