



# Ambient-Oriented Modeling Helps Disabled Students

**Todorka Glushkova,  
Plovdiv University “Paisii Hilendarski”**

**[glushkova@uni-plovdiv.bg](mailto:glushkova@uni-plovdiv.bg)**

- Fourth Industrial Revolution determines the need to create new Cyber-Physical Spaces (CPS) that provide dynamic interaction between the real and virtual world.
- These systems take account of changes in the physical world and adapt this information to the goals, desires and needs of the individual users.
- Software IoT architecture includes various types of active and passive components - mainly agents, but also services, modules, ontologies
- The development of such environments requires the creation of new approaches, models, methods and technologies to integrate the individual components of the system.

- Virtualization of “Things”:
  - Objects with attributes (inherent characteristics of things)
  - Additional aspects- Space, Time, Events
- The monitoring of environment changes enables a more adaptive learning process for all students, especially for disabled people.
- The formal representation of the space characteristics of "things" in the new IoT architecture - **AmbiNet**

- Due to hardware and software complexity and heterogeneity, the construction of IoT applications is associated with serious risks.
- It is inappropriate to directly develop such systems, without preliminary modeling and prototyping process.
- We will describe an appropriate approach to modeling IoT systems, known as Ambient-Oriented Modeling (AOM).
- In this presentation we will focus on creating an Ambient-Oriented Model for disabled students who attend university classes with their wheelchairs.

An Ambient is an identity with:

- **Limitation** - limited location, where action is happening;
- **Inclusion** – one ambient can be included in other one.
- **Mobility** - an ambient can change its location in the ambient hierarchy.

The ambient will be presented as a structure with:

- **Identifier** - mandatory element for identification and control.
- **Processes** that work directly in the ambient and in some aspects control it.

# Ambient-Oriented Modeling (AOM)

- There are different formal systems that are used for AOM. Basic formalism is  $\pi$ -calculus, which presents a type process-calculations.
- Calculus of Context-Aware Ambients(CCA) enables mobile ambients to respond of changes in the environment.
- CCA Ambient is an identity that is used to describe an object or component - process, device, location, etc.
- The Ambient has a name, a boundary, and may contain other ambients, and be included in another ambient.
- There are three possible relationships between two ambients - parent, child and sibling.

# Ambient-Oriented Modeling (AOM)

- Ambients can exchange messages with each other.
- The notation "::" is a symbol for sibling ambients;
- "↑" and "↓" are symbols for parent and child;
- "<>" means sending, and "()" - receiving a message.
- The CCA has four syntax categories:
  - locations  $\alpha$ ,
  - opportunities  $M$ ,
  - processes  $P$  and
  - contextual expressions  $k$ .

- The role of the personal IoT assistant has been incredible increasing for disabled students who attend lectures with their wheelchairs.
- The wheelchair has different physical sensors to collect information about changing of parameters as current location, temperature etc., and interacts with educational systems via student's PA.
- We will look at the wheelchair as a separate ambient including two sub-ambients: the student (through his PA) and the mobile device for communication with educational space.



- We develop both common to all students services such as: e-Lecture; e-Test; e-Schedule etc., as well as specific services for different groups of students.
- We will look at two specific services for disabled students:
  - Route generation for moving to the study hall
  - Provide specific up-to-date information on changing the location of study halls and problems with the operation of automatic doors, elevators and ramps.

# The example

- For example let's the PA receives information about the lecture on Artificial Intelligence, which will take place in a 422 study hall at 9 o'clock.
- We will use the following Ambients:
  - PA -personal assistant;
  - OA - Operative assistant;
  - AS - Analytical Subspace;
  - ANet – AmbiNet, child ambient of AS;
  - GA- Guard assistant;
  - IoTN - IoT Nodes in real world,
  - Cart- intelligent student's wheelchair.

- After recognizing of wheelchair, the system activates PA, which takes care of the delivery of all educational services and learning resources to the mobile device, like all other students.
- The specific services for these students are mainly related to his mobility in the physical space of the university campus.
- Once the student receives the list of all services from his PA, he 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 zones has collection of sensors that dynamically provide actual information to Guard Assistants (GA).

- When PA receives information about the upcoming event, it sends a message to ANet for generating of an appropriate route.
- ANet starts a bidirectional communication process with the GA for providing of up-to-date information from the physical world.
- After receiving the list of currently active zones from the GA, ANet generates a list of appropriate routes and sends it to the PA.
- From the received information, the student chooses a route and sends it to the Cart Ambient.
- When the wheelchair moves in the physical campus, GA tracks its location and, when it is close to some of the IZ, activates the actuators associated with opening the doors, providing a lift, etc.
- If in real time any of the IZ changes its status and becomes inactive, GA promptly informs PA for choosing a new route.

- The route is a sequence of important zones (IZ1, IZ2 ...).
- The search process can be realized through different algorithms as 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 is to set a list with limited number of IZ (islands) through which the search is done with fewer steps.
  - Identify a set of important zones (Islands)  $IZ_1, IZ_2, \dots, IZ_k$
  - Find paths from a current location to  $IZ_1$ ; from  $IZ_{j-1}$  to  $IZ_j$  for each integer  $j \in [2, k]$  and from  $IZ_k$  to the destination.
- This algorithm does not guarantee the detection of an optimal route but it ensures the discovery of one or several possible routes.

We can modelled CCA processes as follows:

$$\begin{aligned}
 P_{PA} &= \left( \begin{array}{l} !AS :: (lecture\_AI, room\_422, time\_9).OA :: \langle location, room\_422, PAi \rangle .0 \mid \\ !OA :: (listRoutes).Cart :: \langle listRoutes \rangle .0 \mid \\ Cart :: (Route, OK).0 \end{array} \right) \\
 P_{OA} &= \left( \begin{array}{l} !PA :: (location, room\_422, PAi).AS :: \langle location, room\_422, PAi \rangle .0 \mid \\ AS :: (listRoutes, PAi).PA :: \langle listRoutes \rangle .0 \end{array} \right) \\
 P_{AS} &= \left( \begin{array}{l} !PA :: \langle lecture\_AI, room\_422, time\_9 \rangle .0 \mid \\ !OA :: (location, room\_422, PAi).GA :: \langle location, room\_422, PAi \rangle .0 \mid \\ GA :: (location, room\_422, listIZ, PAi).ANet \downarrow \langle location, room\_422, listIZ, PAi \rangle .0 \mid \\ !ANet \downarrow (listRoutes, PAi).OA :: \langle listRoutes, PAi \rangle .0 \end{array} \right) \\
 P_{ANet} &= \left( !AS \uparrow (location, room\_422, listIZ, PAi).AS \uparrow \langle listRoutes, PAi \rangle .0 \right) \\
 P_{GA} &= \left( \begin{array}{l} !AS :: (location, room\_422, PAi).IoTN :: \langle location, room\_422, PAi \rangle .0 \mid \\ IoTN :: (location, room\_422, listIZ, PAi).AS :: \langle location, room\_422, listIZ, PAi \rangle .0 \end{array} \right) \\
 P_{IoTN} &= \left( !GA :: (location, room\_422, PAi).GA :: \langle location, room\_422, listIZ, PAi \rangle .0 \right) \\
 P_{Cart} &= \left( PA :: (listRoutes).PA :: \langle Route, OK \rangle .0 \right)
 \end{aligned}$$

For describing of CCA processes was created the programming language ccaPL. The interpreter of ccaPL is developed as Java application. CcaPL program realization of scenario:

```
PA [AS::recv(lecture_AI,room_422,time_9).OA::send(location,room_422,PAi).0|
    OA::recv(listRoutes).Cart::send(listRoutes).0| Cart::recv(Route,OK).0]]
OA [!PA::recv(location,room_422,PAi).AS::send(location,room_422,PA).0|
AS::recv(listRoutes,PAi).PA::send(listRoutes).0]]
AS [PA::send(lecture_AI,room_422,time_9).0|
OA::recv(location,room_422,PAi).GA::send(location,room_422,PAi).0|
GA::recv(location,room_422,ListIZ,PAi).ANet#send(location,room_422,ListIZ,PAi).0|
ANet#recv(ListRoutes,PAi).OA::send(ListRoutes,PAi).0|
ANet[AS@recv(location,room_422,ListIZ,PAi).AS@send(ListRoutes,PAi).0]]]
GA [AS::recv(location,room_422,PAi).IoTN::send(location,room_422,PAi).0|
IoTN::recv(location,room_422,ListIZ,PAi).AS::send(location,room_422,ListIZ,PAi).0]]
IoTN [GA::recv(location,room_422,PAi).GA::send(location,room_422,ListIZ,PAi).0]]
Cart [PA::recv(ListRoutes).PA::send(Route,OK).0]
```



- Based on the main version, we developed a simulator for verification the scenario described above.
- Notation " $A \text{ === } (X) \text{ ===} \rightarrow B$ " 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 hierarchy of Ambients.
- The animator, implemented into the simulator allows visualization of ambients, their location and processes.
- The processes of ambients can be traced to each step of the scenario and make it possible to identify the inconsistencies and errors.

# The simulator

University Campus

STEP BY STEP RUN

PA

OA

AS

GA

IoTN

Cart

Команден прозорец - java Cca\_parser ruse2.cca

```
---> {Sibling to sibling: AS ===(lecture_AI,room_422,time_9)===> PA}
---> {Sibling to sibling: PA ===(location,room_422,PAi)===> OA}
---> {Sibling to sibling: OA ===(location,room_422,PA)===> AS}
---> {Sibling to sibling: AS ===(location,room_422,PA)===> GA}
---> {Sibling to sibling: GA ===(location,room_422,PA)===> IoTN}
---> {Sibling to sibling: IoTN ===(location,room_422,ListIZ,PA)===> GA}
---> {Sibling to sibling: GA ===(location,room_422,ListIZ,PA)===> AS}
---> {Parent to child: AS ===(location,room_422,ListIZ,PA)===> ANet}
---> {Child to parent: ANet ===(ListRoutes,PA)===> AS}
---> {Sibling to sibling: AS ===(ListRoutes,PA)===> OA}
---> {Sibling to sibling: OA ===(ListRoutes)===> PA}
---> {Sibling to sibling: PA ===(ListRoutes)===> Cart}
---> {Sibling to sibling: Cart ===(Route,OK)===> PA}
```

Message

PA::rcv(ListRoutes).PA::send(Route,OK).0

OK

Message

!PA::rcv(location,room\_422,PAi).AS::send(location,room\_422,PA).0| PA::rcv(location,room\_422,PAi).AS::send(location,room\_422,PA).0

OK

- This approach makes it possible to verify the interaction between different components of the learning environment and to correct inconsistencies and ineffective actions at the modeling and prototyping stage.
- The future plans of the team are related to the development of 3D AOM - editor and simulator for modelling and testing of different scenarios
- However, the main future direction is applying the experience to new domains- for modelling of smart city, smart farming and medicine, tourist guide, etc.

Thank you!  
Questions?