Multi-aspect modeling of services for disability students in a smart city context

TODORKA GLUSHKOVA
DIMITAR STOYANOV

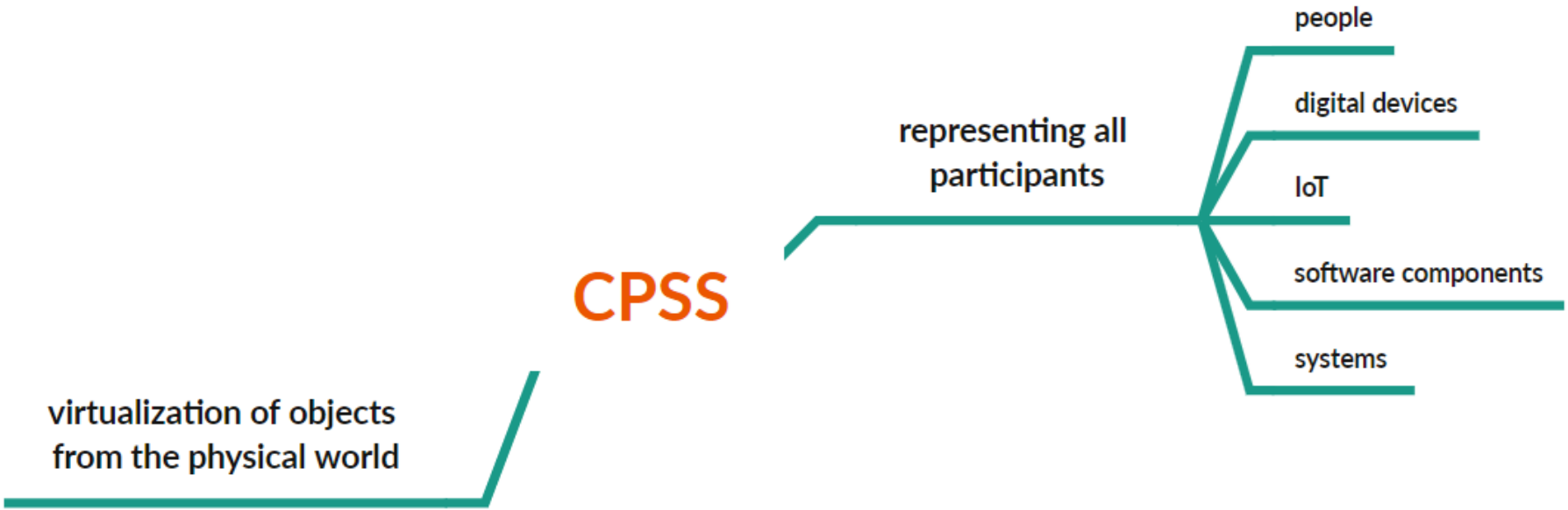
INTRODUCTION

- New trends in the development of the digital age are related to the IoT and CPSS paradigms.
- One of the most popular applications of these technologies is related to the need to create intelligent systems managing the processes in a "smart city"
- In such an integrated environment, it is necessary to model and implement different services that include different aspects of the physical and virtual worlds.

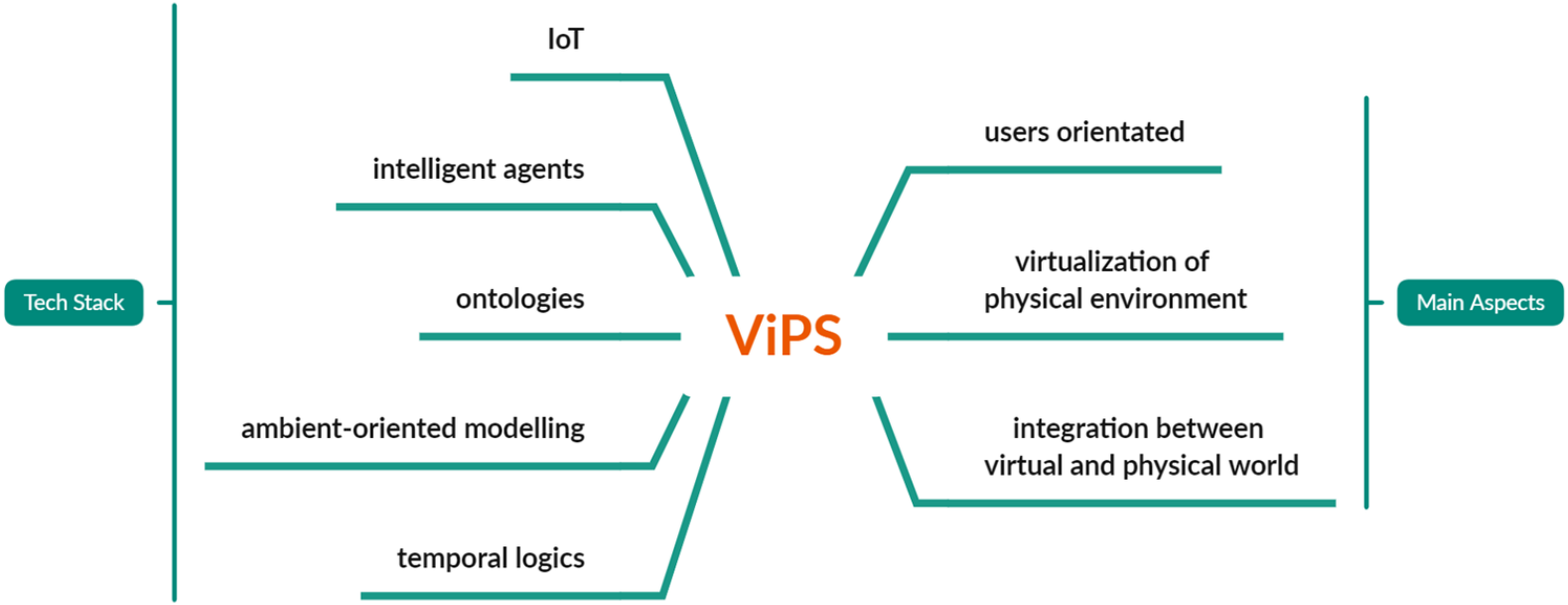
GENERAL CONCEPTS

- Cyber-Physical Social System (CPSS)
- Virtual Physical Space (ViPS)
- Adaptation of ViPS

CYBER-PHYSICAL SOCIAL SYSTEM (CPSS)



TECH STACK AND MAIN ASPECTS



ADAPTATION OF VIPS

SMART CITY

multi-agent system

TNet

personal assistant

LoRa

integration with IoT devices

AmbiNet

Multi-aspect formal modeling

- Interaction of different domains - transport, healthy environment, education.
- We model two physical spaces - the airspace, divided into sectors for air cleanliness monitoring, and the transport network in the city, through which the student's journey will take place.
  - **TS** is a Transport Subspace. It is a graph with vertices - transport units

    $\{te^i \mid i=1,\ldots,n\}$ and ***edges*** – pairs of two vertices;
  - **AS** is an Air Subspace. It is a graph structure also with vertices–air zones $\{z^i \mid i = 1,\ldots,m\}$.
- **SPACE = TS ∪ AS.**

Multi-aspect formal modeling

- To determine the degree of air pollution in each of the zones, we define
  pf: AS ⟶ R, which assigns a real number to each zone of AS.
  - if **pf (z)> PT**, then it is polluted ;
  - if **pf(z) ≤ PT**, then it is pure (or z is “clean zone”).
- Using the transport network in **TS** we define the routes as the set of all sequences of transport nodes $\mathbf{R_i = (te^i0, te^i1,..., te^ig)}$ in the graph TS connecting the starting and destination points in the city.
- There are two approaches to solving the problem:
  - To generate routes in the TS, for each node at which the status of the corresponding air zone in the AS is checked.
  - To search for connectivity in the AS involving only clean air zones and then to generate routes in the TS that include only transport nodes located in these “clean zones”.

Multi-aspect formal modeling

- According to the accepted formulation of the problem, a network of air zones has been built over the city of Plovdiv, in each of which a sensor measuring station functions.
- Since these air zones in AS are much smaller number than the transport nodes in the transport space TS, it is more appropriate to choose the second approach.
- We initially look for connectivity between the zones above the student's starting and target locations, generating a graph structure including only the clean air zones.
- From TS we generate a connected subgraph whose nodes lie only in the clear air zones from AS

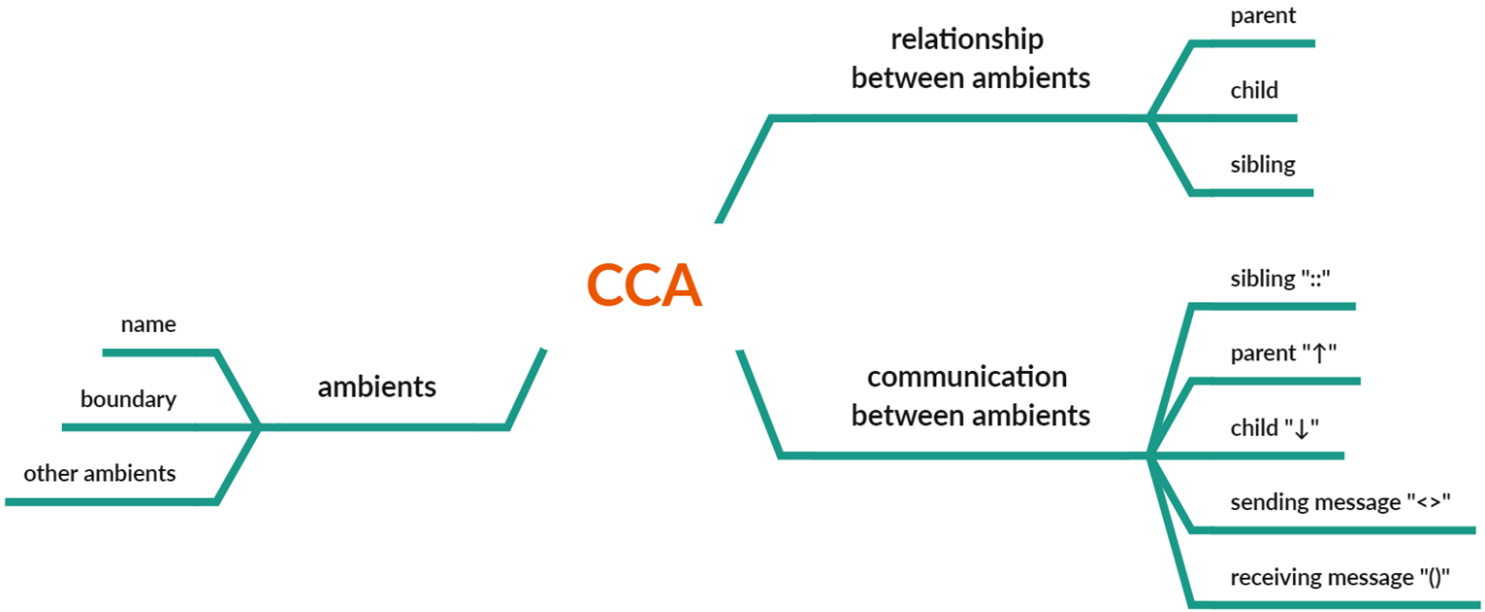
Multi-aspect formal modeling

- To generate clean air connectivity and possible "healthy" routes in TS, we use a modified **A* algorithm**.
- If it is not possible to ensure clear connectivity **in AS** or to generate a healthy route **in TS**, the user's personal assistant will receive the necessary information.
- For modeling in **AmbiNet**, we use Ambient Oriented Modeling and more specifically the mathematical notation ***CCA (Calculus of Context-Aware Ambients)***
- For modelling of temporal aspects of things in **TNet** we use ***Interval and Temporal Logic and Tempura.***

MODELING AND VIRTUALIZATION

- Calculus of Context-aware Ambients (CCA)
- Smart City Ambients
- CCA Syntax
- CCA Editor

CCA


CCA
relationship between ambients
parent
child
sibling
communication between ambients
sibling "::"
parent "↑"
child "↓"
sending message "<>"
receiving message "()"
ambients
name
boundary
other ambients

SMART CITY AMBIENTS

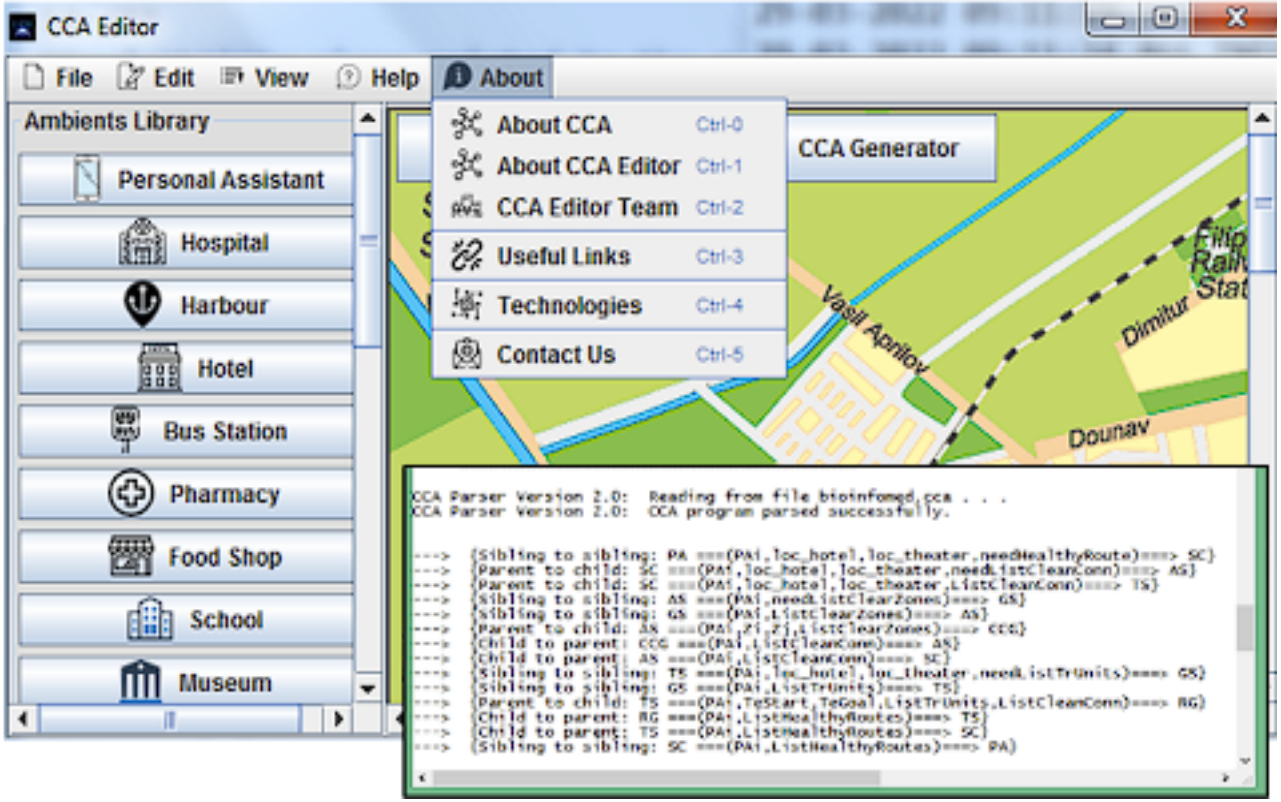
Name	Description	Process
SC	Smart City Space	P_{SC}
TS	Transport Subspace.	P_{TS}
AS	Air Subspace	P_{AS}
GS	Guard System in the Smart City Space	P_{GS}
CCG	Clear Connectivity Generator	P_{CCG}
RG	Route Generator in the TS	P_{RG}
PA	Personal Assistants for users	P_{PA}

CCA SYNTAX

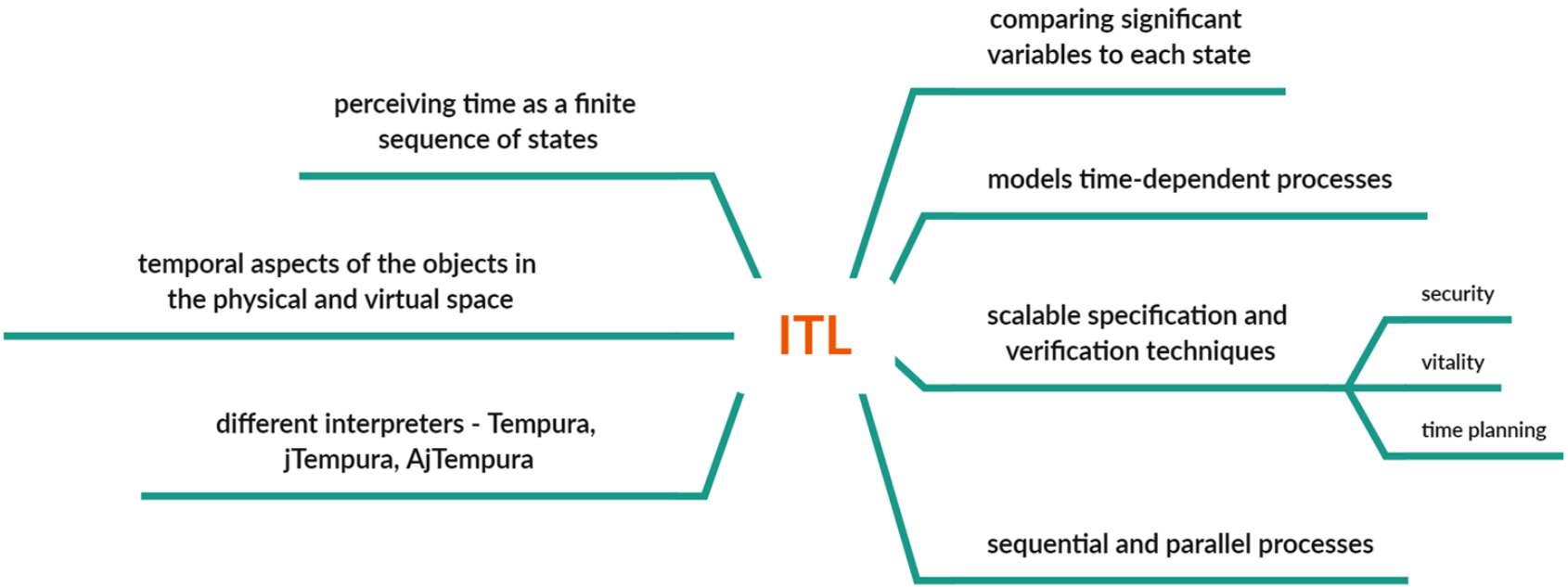
$$P_{AS} \cong \left(\begin{array}{l} SC \uparrow (PA_i, z_i, z_j, needCleanConn).GS ::< PA_i, needCleanZones > .0 \mid \\ GS :: (PA_i, ListClZones(z_k)).0 \mid \\ ((pf(z_i) > PT) or (pf(z_j) > PT))?(resp = no).SC \uparrow < PA_i, resp > .0 \mid \\ \left(\begin{array}{l} ((pf(z_i) < PT) and (pf(z_j) < PT))? \\ CC\_G \downarrow < PA_i, z_i, z_j, ListClZones(z_k), needCleanConn > .0 \mid \\ CC\_G \downarrow (PA_i, r\_CCG). \mid \\ \left(\begin{array}{l} (r\_CCG <> no)?(r\_AS = ListConnZones).SC \uparrow < PA_i, r\_AS > .0 \mid \\ (r\_CCG == no)?(r\_AS = no).SC \uparrow < PA_i, r\_AS > .0 \end{array} \right) \end{array} \right) \end{array} \right)$$

$$P_{AS} \cong \left(\begin{array}{l} SC \uparrow (PA_i, z_i, z_j, needCleanConn).GS ::< PA_i, needCleanZones > .0 \mid \\ GS :: (PA_i, ListClZones(z_k)).0 \mid \\ ((pf(z_i) > PT) or (pf(z_j) > PT))?(resp = no).SC \uparrow < PA_i, resp > .0 \mid \\ \left(\begin{array}{l} ((pf(z_i) < PT) and (pf(z_j) < PT))? \\ CC\_G \downarrow < PA_i, z_i, z_j, ListClZones(z_k), needCleanConn > .0 \mid \\ CC\_G \downarrow (PA_i, r\_CCG). \mid \\ \left(\begin{array}{l} (r\_CCG <> no)?(r\_AS = ListConnZones).SC \uparrow < PA_i, r\_AS > .0 \mid \\ (r\_CCG == no)?(r\_AS = no).SC \uparrow < PA_i, r\_AS > .0 \end{array} \right) \end{array} \right) \end{array} \right)$$

CCA EDITOR


CCA Editor
File
Edit
View
Help
About
Ambients Library
Personal Assistant
Hospital
Harbour
Hotel
Bus Station
Pharmacy
Food Shop
School
Museum
About CCA Ctrl-0
About CCA Editor Ctrl-1
CCA Editor Team Ctrl-2
Useful Links Ctrl-3
Technologies Ctrl-4
Contact Us Ctrl-5
CCA Generator
Vasil Aprilov
Dimitur
Dounav
CCA Parser Version 2.0: Reading from file bioinformed.cca . . .
CCA Parser Version 2.0: CCA program parsed successfully.
---> {Sibling to sibling: PA ===(PA1,loc_hotel,loc_theater,needHealthyRoute)===> SC}
---> {Parent to child: SC ===(PA1,loc_hotel,loc_theater,needListCleanConn)===> AS}
---> {Parent to child: SC ===(PA1,loc_hotel,loc_theater,ListCleanConn)===> TS}
---> {Sibling to sibling: AS ===(PA1,needListClearZones)===> GS}
---> {Sibling to sibling: GS ===(PA1,ListClearZones)===> AS}
---> {Parent to child: AS ===(PA1,Zi,Zj,ListClearZones)===> CCG}
---> {Child to parent: CCG ===(PA1,ListCleanConn)===> AS}
---> {Child to parent: AS ===(PA1,ListCleanConn)===> SC}
---> {Sibling to sibling: TS ===(PA1,loc_hotel,loc_theater,needListTrUnits)===> GS}
---> {Sibling to sibling: GS ===(PA1,ListTrUnits)===> TS}
---> {Parent to child: TS ===(PA1,TeStart,TeGoal,ListTrUnits,ListCleanConn)===> RG}
---> {Child to parent: RG ===(PA1,ListHealthyRoutes)===> TS}
---> {Child to parent: TS ===(PA1,ListHealthyRoutes)===> SC}
---> {Sibling to sibling: SC ===(PA1,ListHealthyRoutes)===> PA}

ITL

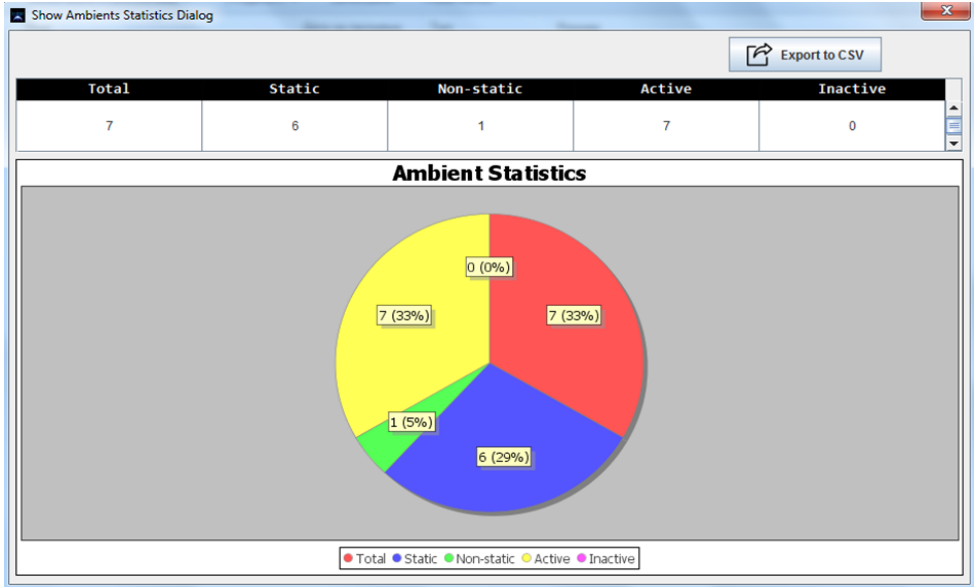

ITL
perceiving time as a finite sequence of states
temporal aspects of the objects in the physical and virtual space
different interpreters - Tempura, jTempura, AjTempura
comparing significant variables to each state
models time-dependent processes
scalable specification and verification techniques
security
vitality
time planning
sequential and parallel processes

ITL STATES AND PROCESSES DEFINITION IN TEMPURA

```
define university() = {exists University, Student,SC,AmbiNet,AS,TS,QTime:
{University = u_closed and list(Student, nstudents) and stable(struct(Student))
and (forall i<nstudents: Student[i] = outside)and list(Qtime, nstudents)
and stable(struct(Qtime)) and forall i<nstudents: {list(Qtime[i], nAmbiNet)
and stable(struct(Qtime[i]))} and (forall i<nstudents :
(forall j<nAmbiNet : Qtime[i][j] = notthere)) and list(AmbiNet, nAmbiNet)
and stable(struct(AmbiNet)) and (forall j<ntp:AmbiNet[j] = AmbiNet_not_active) and Timer = 0 and
school_open() and (forall i<nstudents: student_idle(i)) and ... }
set print_states=true.
```

Results

- The results of testing and verification give us the opportunity to analyze the interactions between the participating interactive components (ambients), the detection of inconsistencies and problems already at the stage of preliminary modeling and verification.



CONCLUSION

- Ambient-oriented modeling has as its main purpose the preliminary development, testing and verification of the main scenarios.
- Ambients virtualize objects in CPS and present them in a single unified way in their context-dependence and dynamism.
- They can "move" in CPSS and become part of or include other Ambients, which allows them to describe processes in their dynamism and change.
- This dynamism and temporal dependence determine the use of ITL, which describes the state of objects dynamically in the process of their change.
- It is necessary to implement a single complex system for context-oriented modeling of various aspects of the considered processes in terms of both the temporal and spatial aspects of things, as well as in terms of context-dependent events, rules and policies.

AMBIENT-ORIENTED MODELING

Q&A

THANK YOU!