Ao Dai :
Agent OrienteD Ambient Intelligence

Diego Salomone, Thi Thuy Nga Nguyen, Andrei Olaru

under the supervision of prof. Amal El Fallah Seghrouchni

Lip6, University Pierre et Marie Curie, Paris

22.06.2010

- AmI
- Context-awareness
- Agents
- CLAIM
- Agentification
- Interaction
- Anticipation
- Ontologies
- Conclusion
- Demo

Ao Dai : Agent OrienteD Ambient Intelligence

overview

Ambient intelligence is a ubiquitous electronic environment that supports people in their daily tasks, in a proactive, but invisible and non-intrusive manner.

[Ramos et al., 2008, Weiser, 1993]

Ambient Intelligence

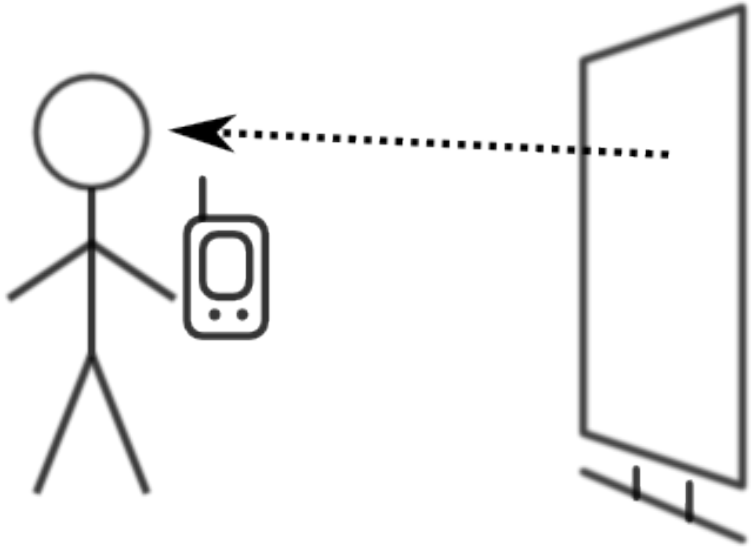
Example scenarios:

The large screen can be used to display context-aware advertisements...

Ambient Intelligence

- Context-awareness
- Agents
- CLAIM
- Agentification
- Interaction
- Anticipation
- Ontologies
- Conclusion
- Demo

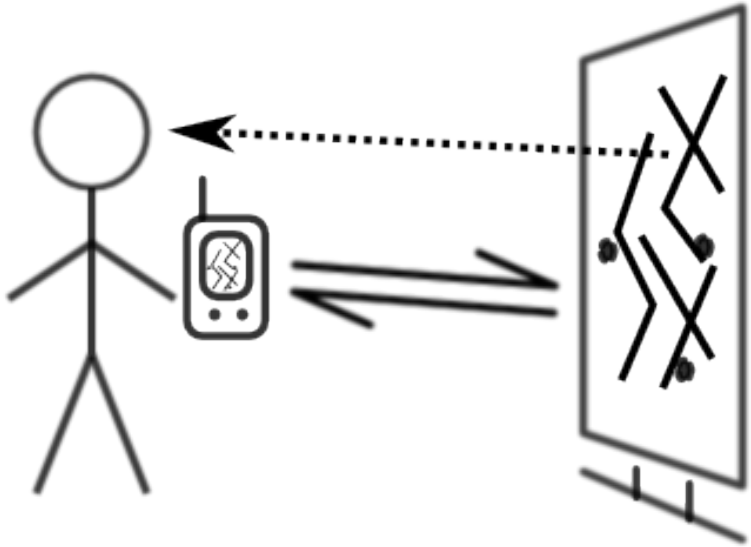
Example scenarios:



...or to draw attention of the user...

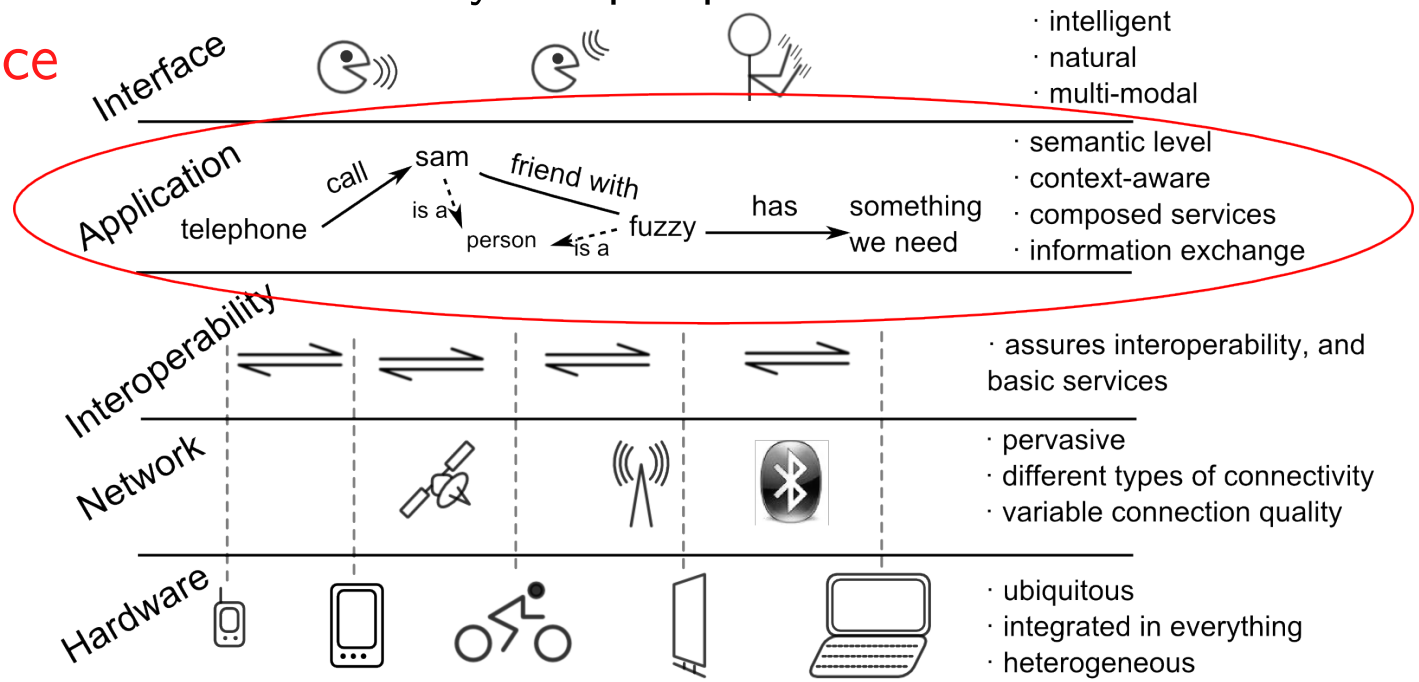
Ambient Intelligence

Example scenarios:



...to show an interactive map for which the mobile phone is too small [Canut et al., 2009]...

A layered perspective on AmI



[Seghrouchni, 2008]

The applicative (or "intelligent") layer can use AI methods and techniques like software agents and ontologies.

[Ramos et al., 2008].

Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. [Dey and Abowd, 2000]

Aspects: [Chen and Kotz, 2000]

- physical aspect (location, conditions)
- temporal aspect
- user profile and preferences
- social aspect
- computing resources
- activity
- associations (e.g. time – place – activity) [Henricksen et al., 2002]

Relevance of new information is related to its compatibility with the user's context.

· can be considered as a measure of proximity in space, time, activity, social relations, preferences and available resources.

In the Ao Dai project, we have so far considered:

- the spatial location of the user
- the user's preferences
- the available computing resources

Software agents are an appropriate implementation for AmI, considering they satisfy the needs of AmI in terms of:

- reactivity
- proactivity
- autonomy
- anticipation
- reasoning

Software agents are an appropriate implementation for AmI, considering they satisfy the needs of AmI in terms of:

- reactivity
- proactivity
- autonomy
- anticipation
- reasoning

Agents also offer beliefs, goals, intentions and easier implementation of a human-inspired behaviour.

Software agents are an appropriate implementation for AmI, considering they satisfy the needs of AmI in terms of:

- reactivity
- proactivity
- autonomy
- anticipation
- reasoning

Agents also offer beliefs, goals, intentions and easier implementation of a human-inspired behaviour.

For Ao Dai, we use CLAIM + Sympa as agent-oriented programming language and platform.

· Agent-Oriented programming language

- Created by Alexandru Suna, during his Thesis in Paris 6 [Suna and El Fallah Seghrouchni, 2007]

· Eases the programming task involving a Multi-Agent System

· Objectives

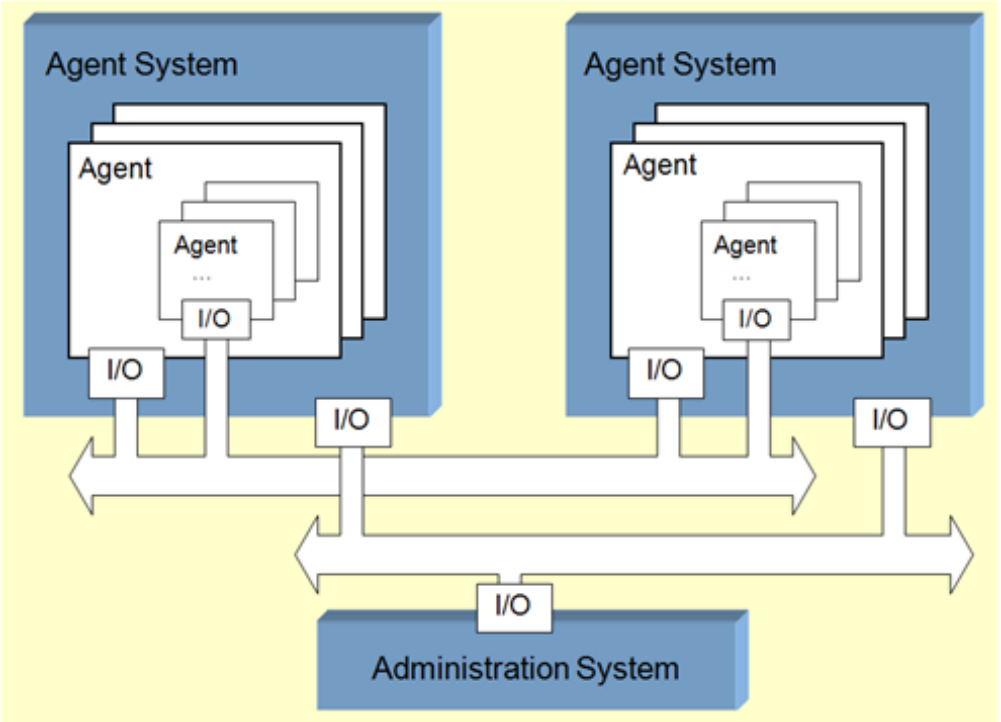
- Intelligence, Communication and Mobility
- Network Distribution and Adaptability
- Possibility of a Formal Verification

CLAIM is based on explicit declaration of agent's characteristics:

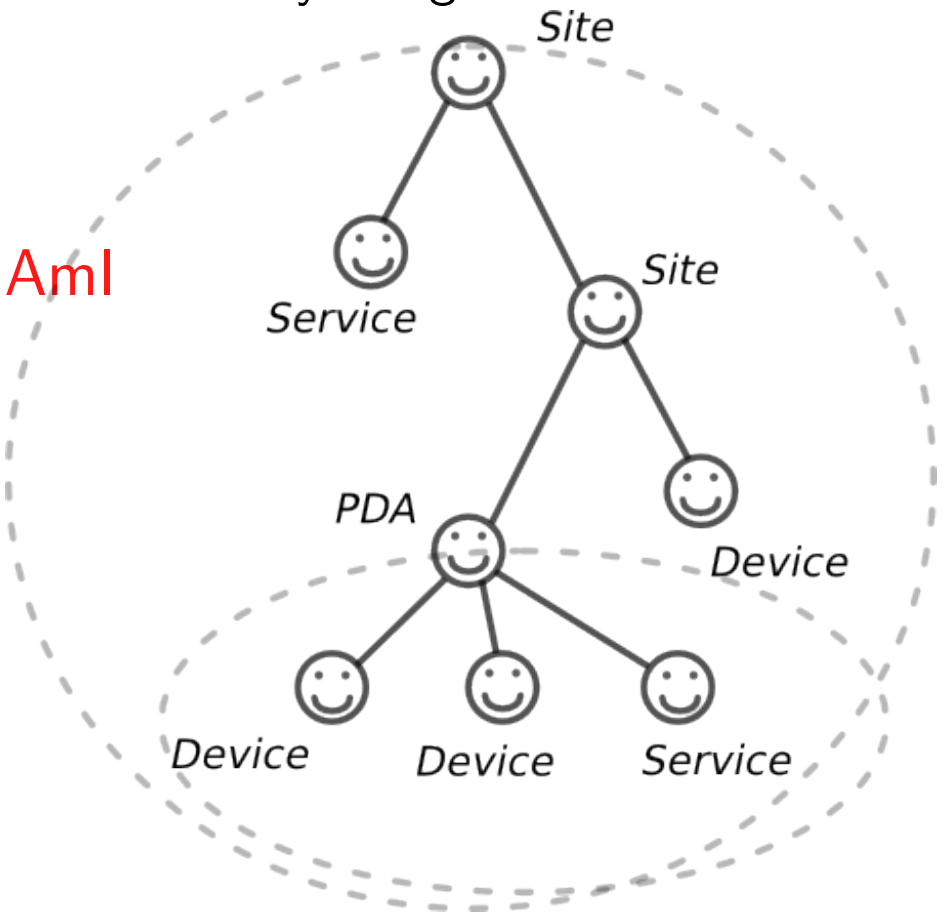
- Capabilities
- Procedures
  - Conditions
  - Triggers
  - ...

```
defineAgentClass className(?v1, ?v2, ..., ?vn) {
        authority = null | authority ;
        parent = null | agentName ;
        knowledge = null ;| {knowledge1; ...; knowledgek}
        goals = null ;| {goal1; ... ; goalg}
        messages = null ; {qMessage1; ... ; qMessagem}
        capabilities = null ; {capability1 ... capabilityc}
        agents = null ; | {agName1, agName2, ..., agNamea}
        effects = null ; | {effect1, effect2, ..., effecte}
}
```

Works on top of a Java layer, giving direct access to Java resources if needed

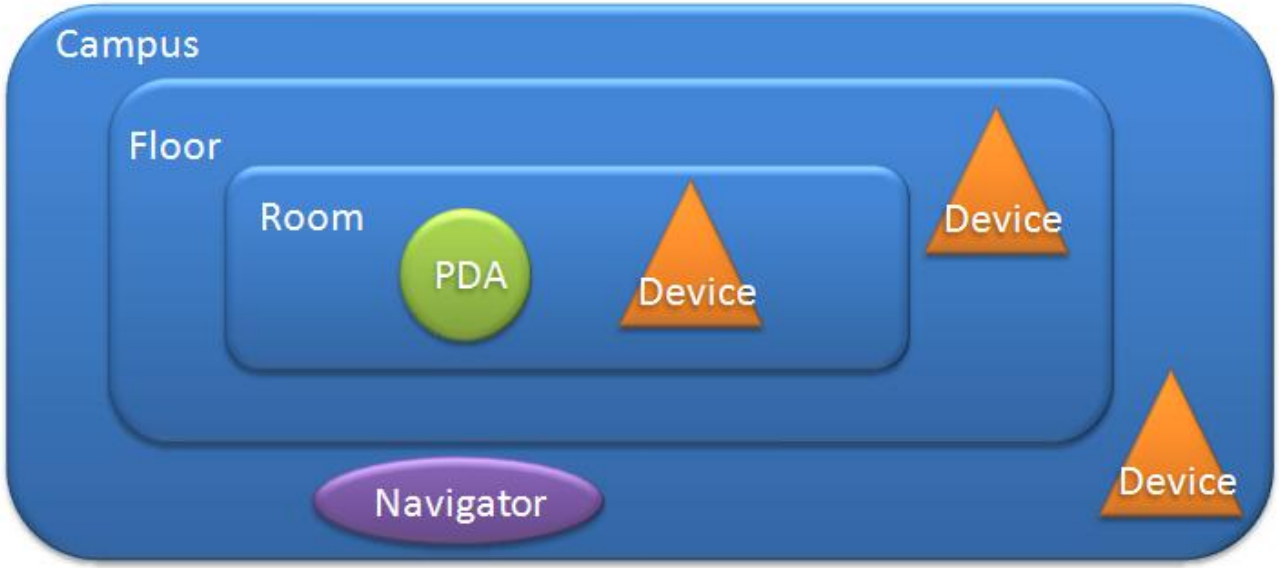


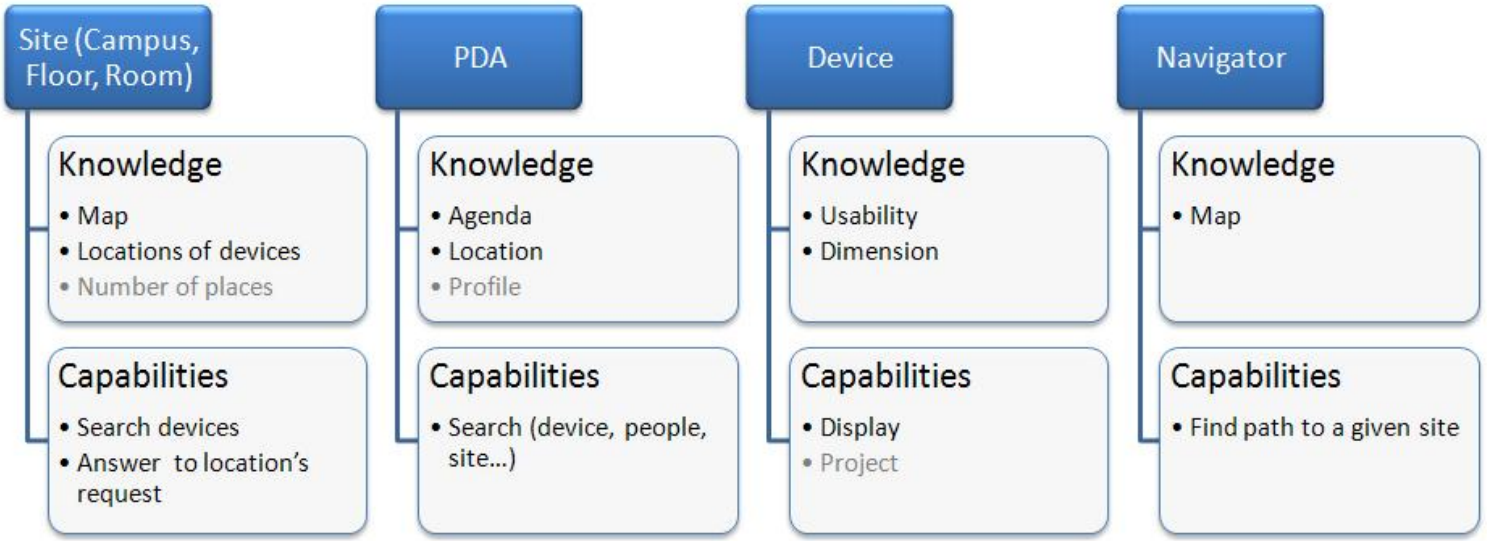
Model context-awareness in terms of location and resources as a hierarchy of agents.



An agent for each site, PDA, and device.

· 4 types of agents:

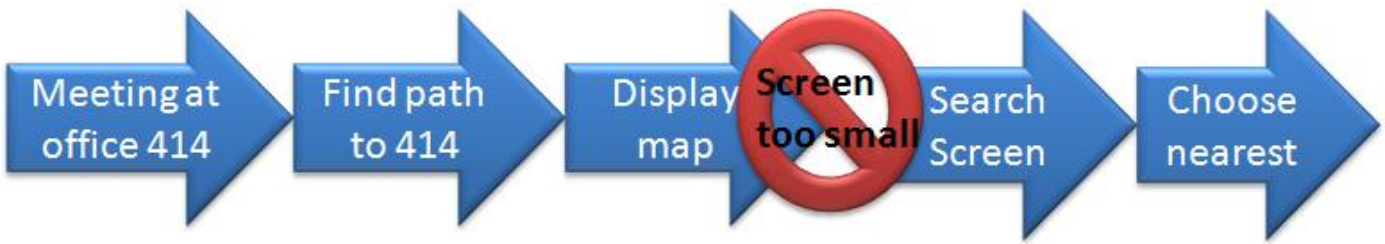




Agent PDA

- Actions are based in agenda of user and context.
  - Context: position of user, status of environment, ...
- Capability: search for device
  - Can search by capability and by criteria created by its own, according to task and context

Example:



Agent Agenda (sub-agent of PDA):

- Reads agenda of user (stored in PDA)
- Extracts tasks
- Activates tasks in PDA agent when it's time

Example:

thu 17/6/2010, 10:00-11h:00 meeting at room 414; 14:00-17:00 course at room 418

· 2 tasks: (meeting,10:00,room 414), (course, 14:00,room 418)

· At 10:00, agenda will inform PDA to activate the action correspond with task "meeting" (find path to room 418)

Agent Site

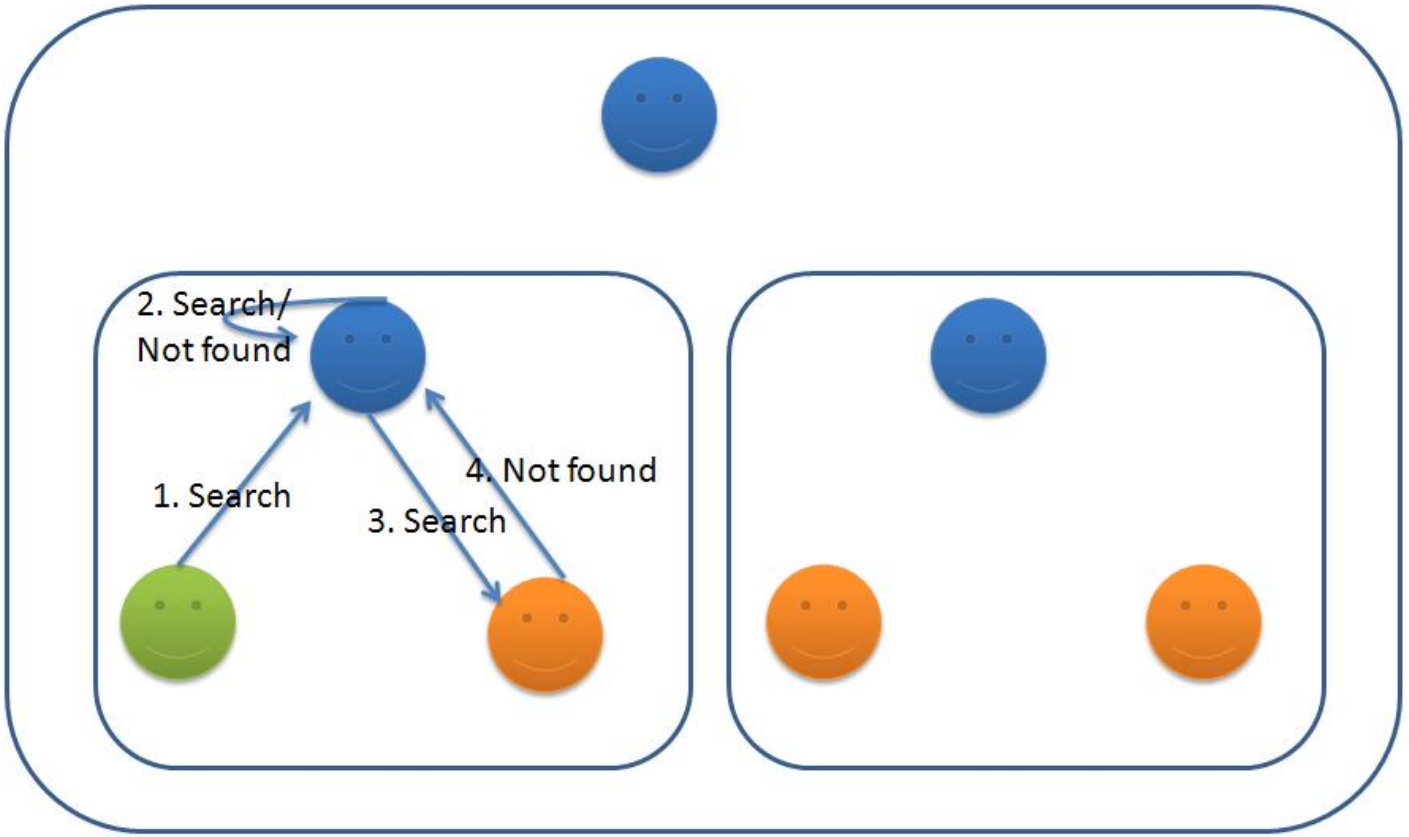
- Can be a room, or a floor, or a campus according to attribute "type" of agent.
- Creates on demand a Navigator agent to help PDA agent in navigating when PDA is in site.
- Behavior "search devices":
  - If site has capability correspond with capability in the request, and satisfied the request, it answers immediately
  - If not, it can search in its children. If its children don't have neither, it searches in its parent.
  - After the search, it sends name of all the devices found to the seeker

Agent Navigator

- Is created by a site, with the knowledge of map of site, for a specific PDA
- Behavior: find path from actual position of PDA to a new location

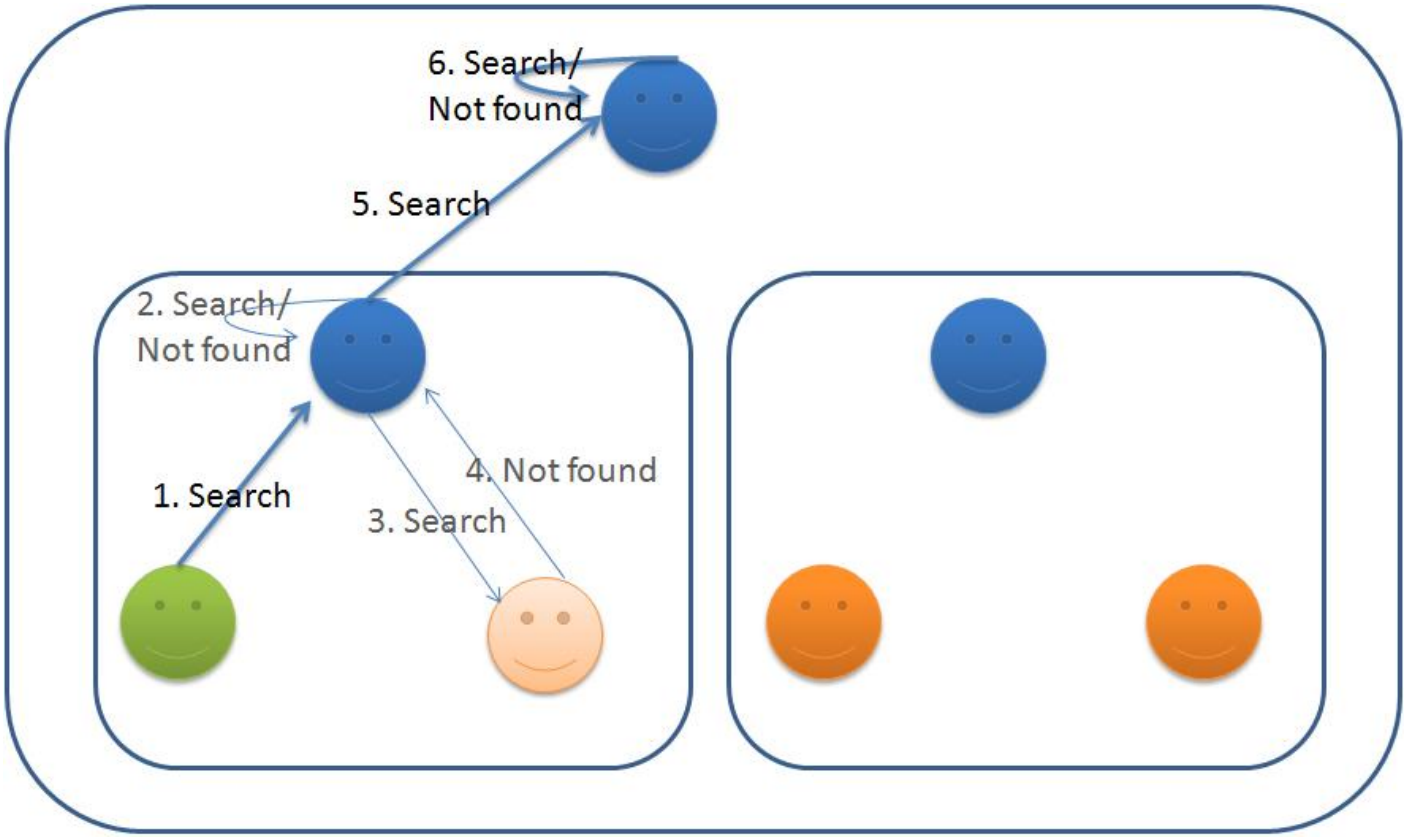
· Agent interacts only with its parent or its children

Example: Search



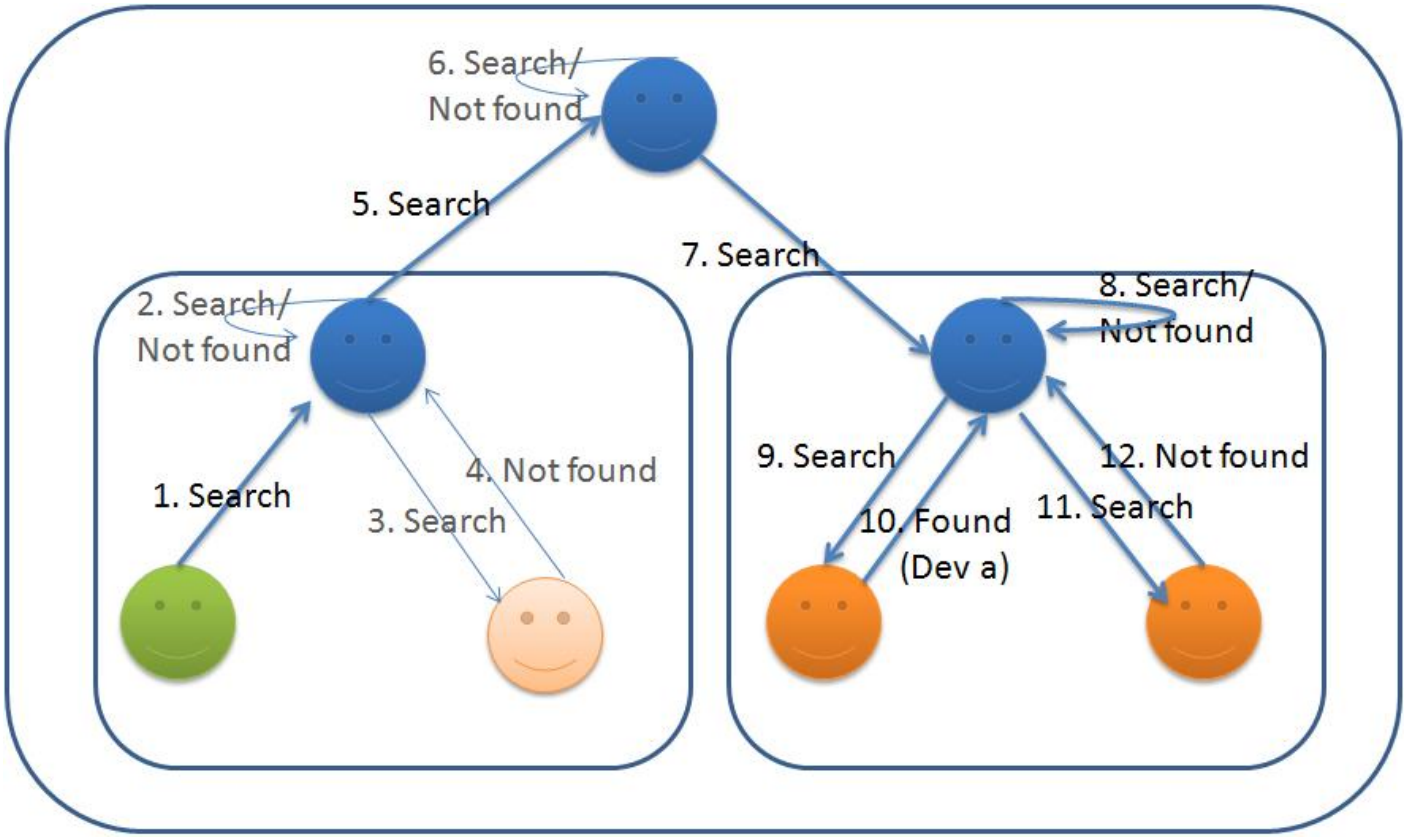
· Agent interacts only with its parent or its children

Example: Search



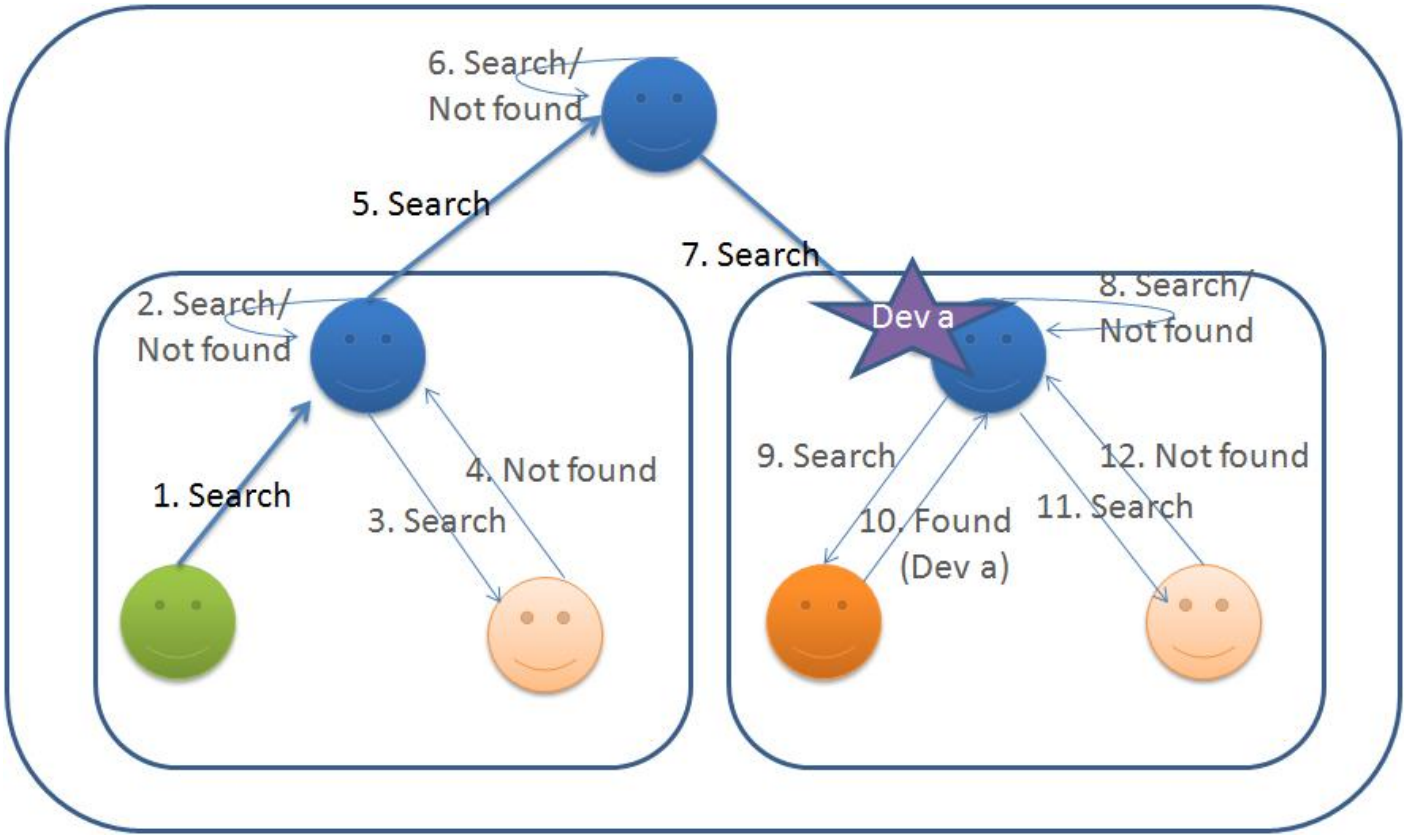
· Agent interacts only with its parent or its children

Example: Search



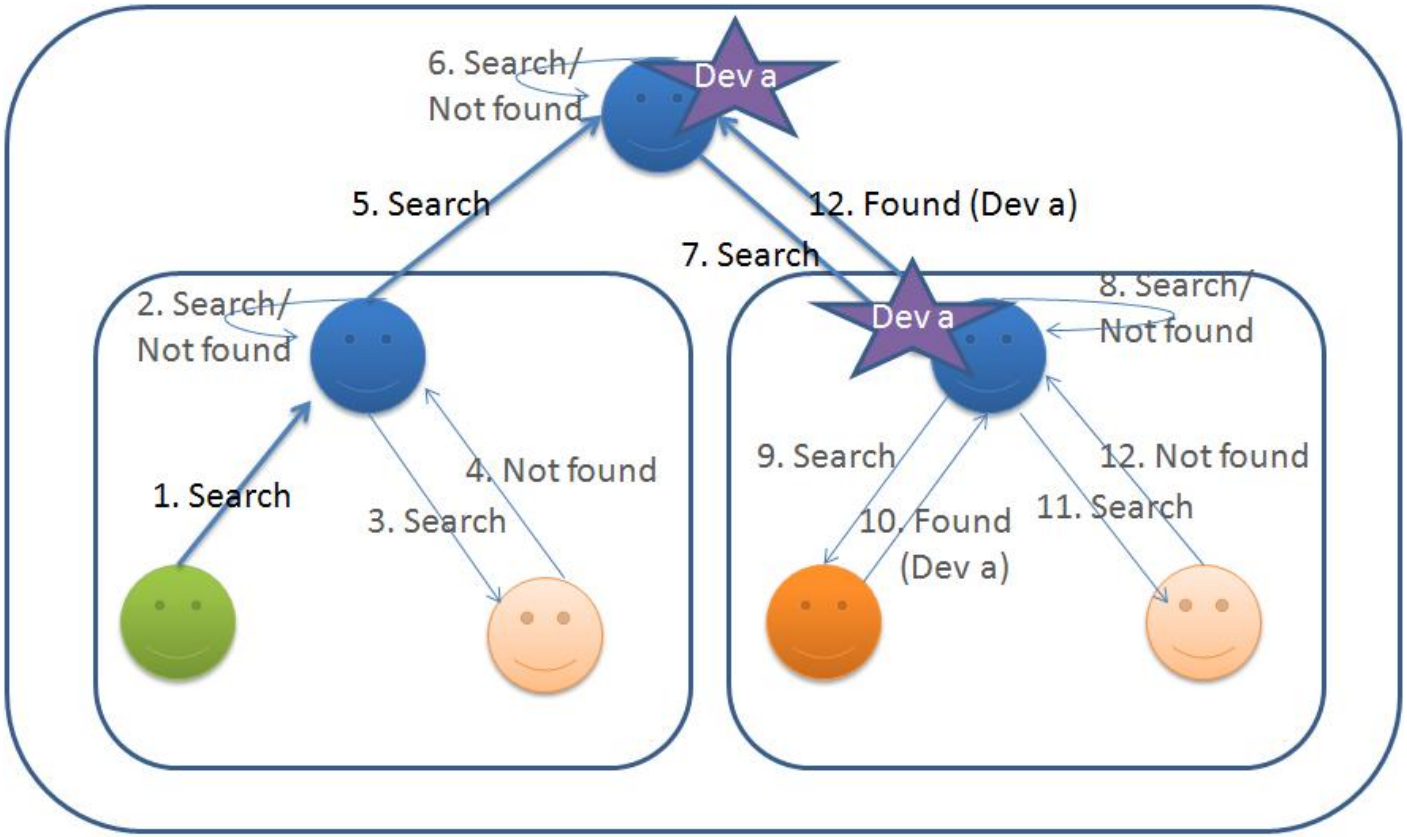
· Agent interacts only with its parent or its children

Example: Search



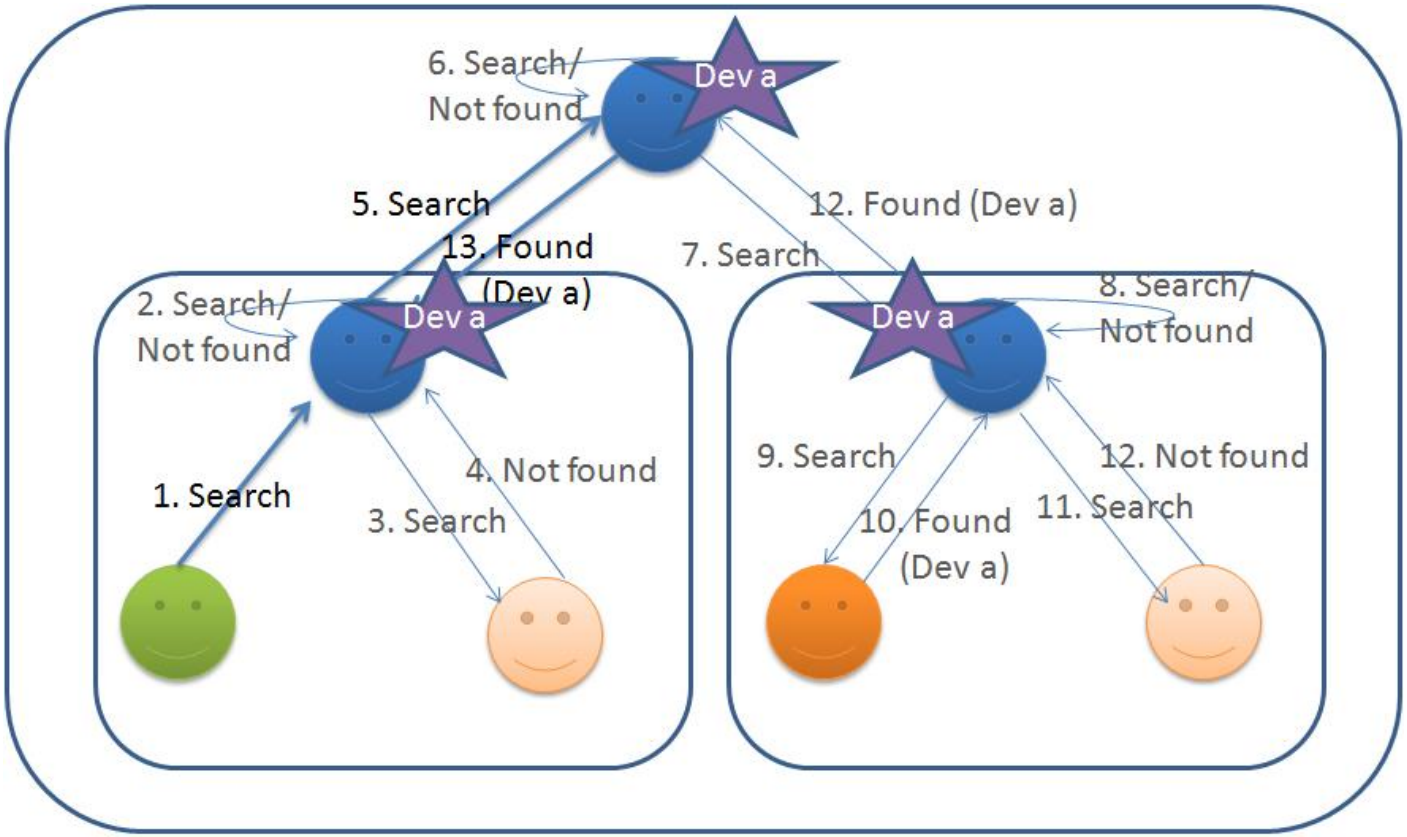
· Agent interacts only with its parent or its children

Example: Search



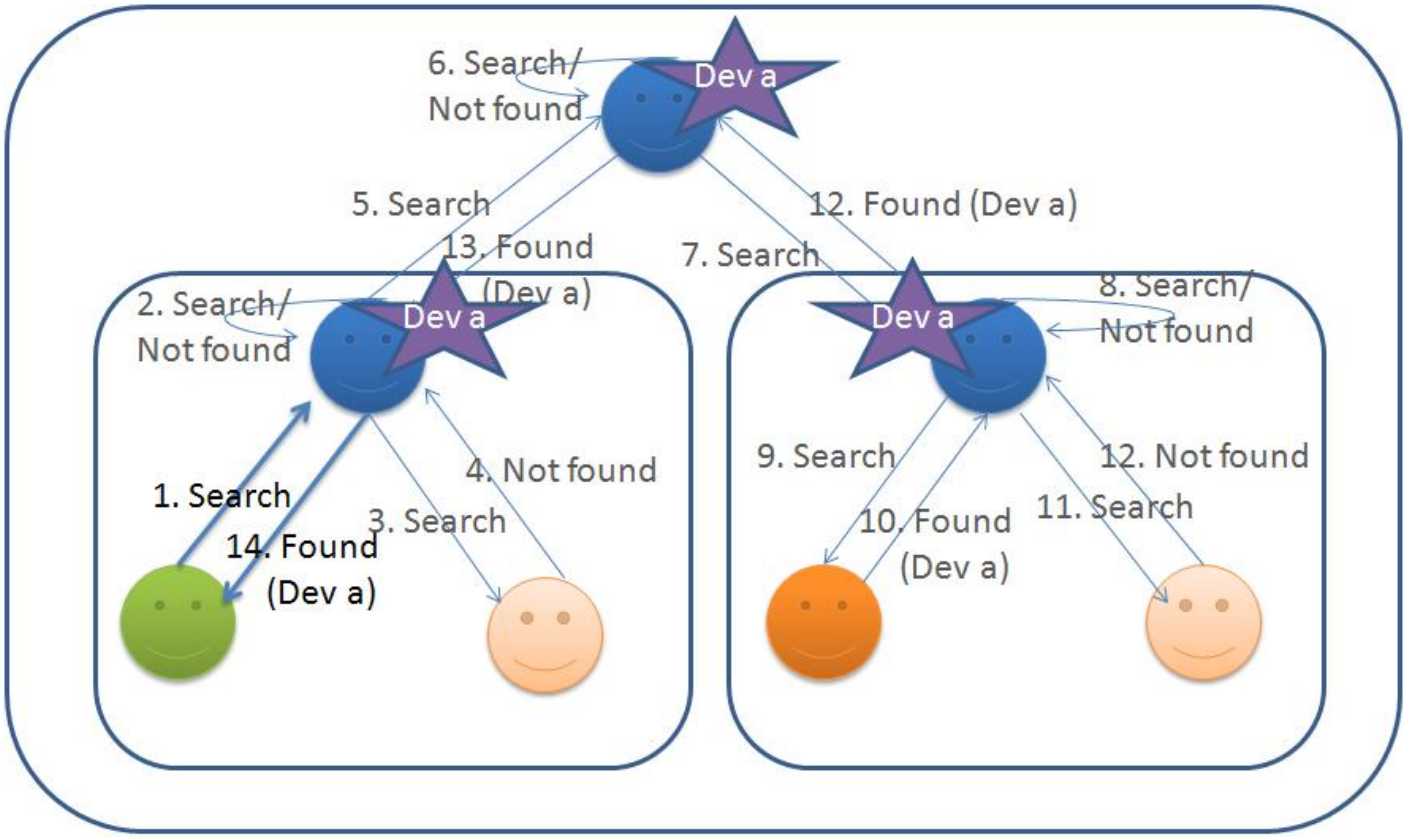
· Agent interacts only with its parent or its children

Example: Search



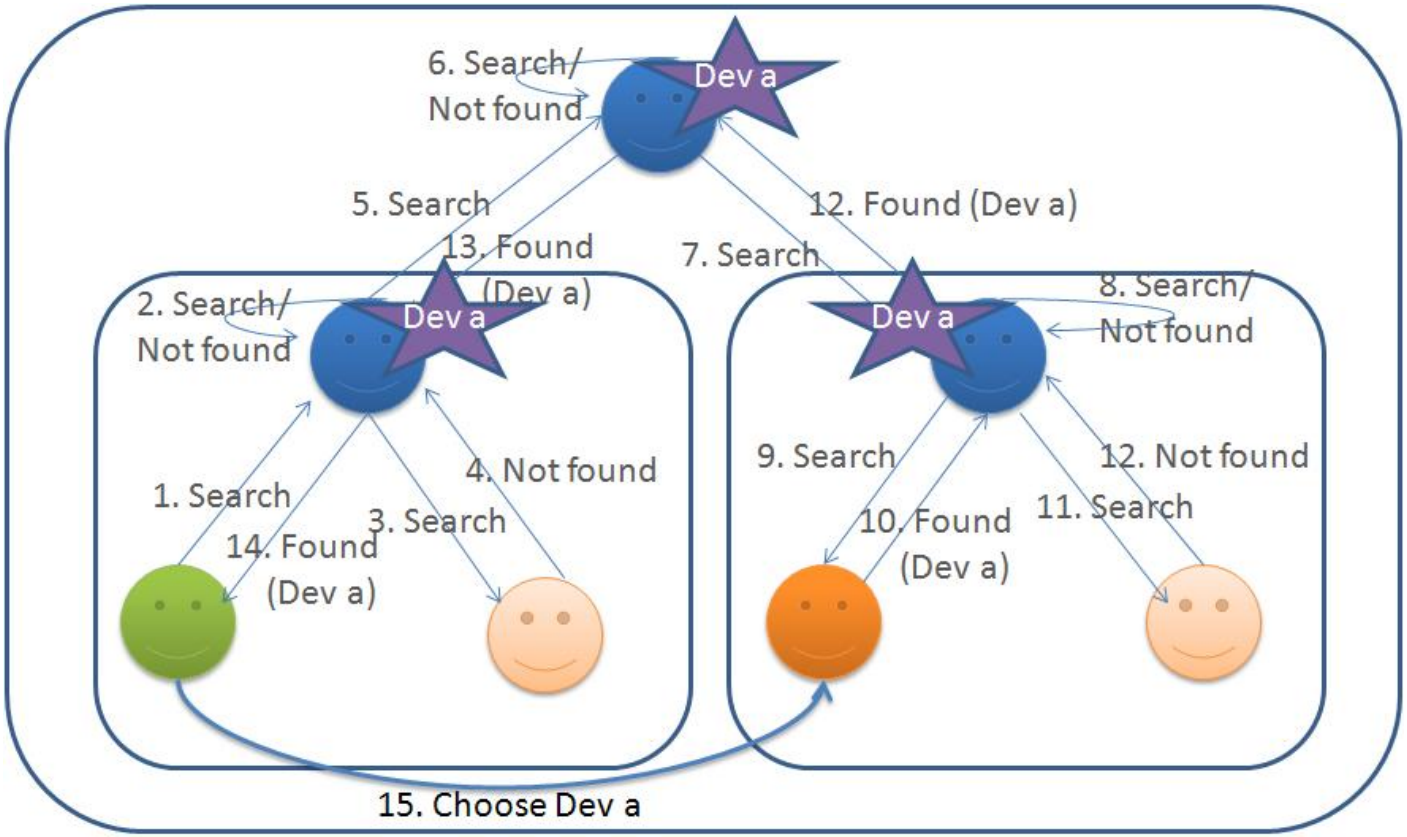
· Agent interacts only with its parent or its children

Example: Search



· Agent interacts only with its parent or its children

Example: Search



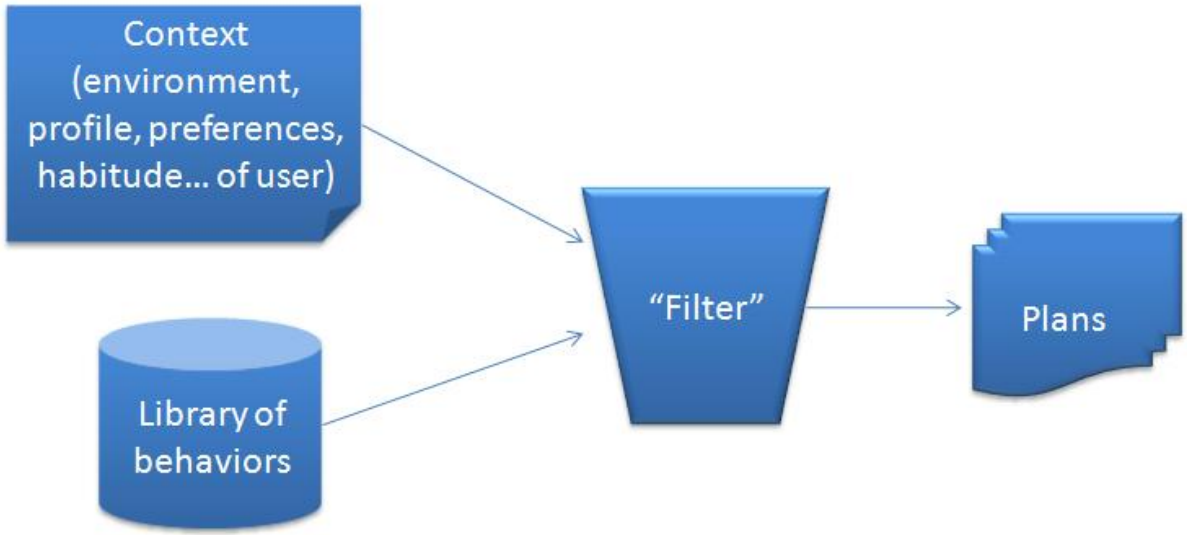
Future Work:

- Search:
  - Multi-criteria
  - Flexible criteria: based in preferences of user and in context
- Anticipation
  - Anticipatory system: *[...] a system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's predictions pertaining to a latter instant*

    [Rosen, 1985]
  - Anticipation is a future-oriented action, decision, or behavior based on a (implicit or explicit) prediction

    [Pezzulo, 2008]

- AmI
- Context-awareness
- Agents
- CLAIM
- Agentification
- Interaction
- Anticipation
- Ontologies
- Conclusion
- Demo



To work with context, we must have a representation:

- First order logic
- ontology
- graphical models
- ...

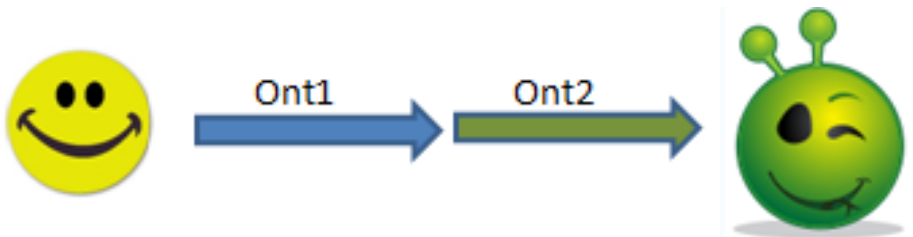
Ontology based models are flexible and robust

· Semantics representation (concepts, facts)

· Combine the assets of logic-based models and object-oriented technology [Krummenacher et al., 2007]

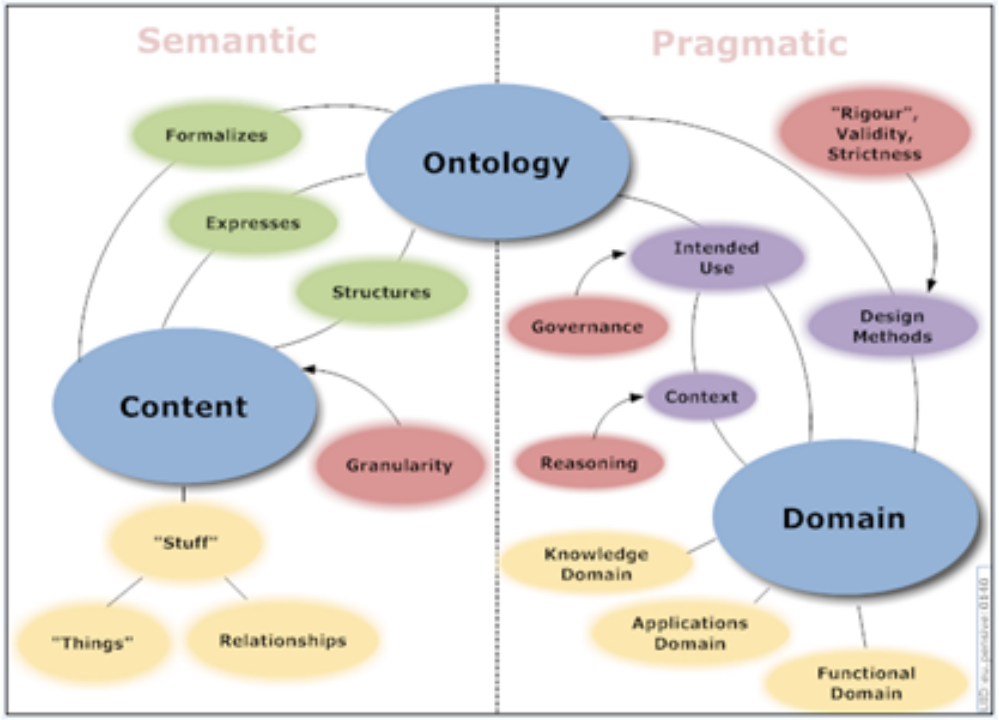
Open System Requirement:

· The agents heterogeneity imposes the possibility to work with different ontologies



Future work in Ao Dai: Add ontology processing capacity to CLAIM:

- Choose a representation (OWL, XWL, ...)
- Implement alignment, construction, comparison

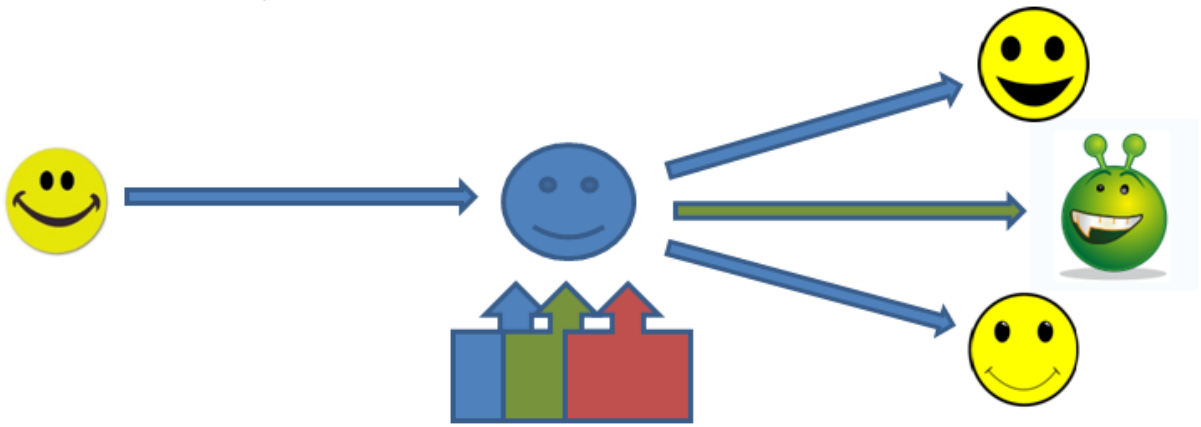


· Study the benefits of each topology

· Proceed with concrete tests to determine the best (or most appropriated) to each situation: Centralized (server), decentralized, hybrid

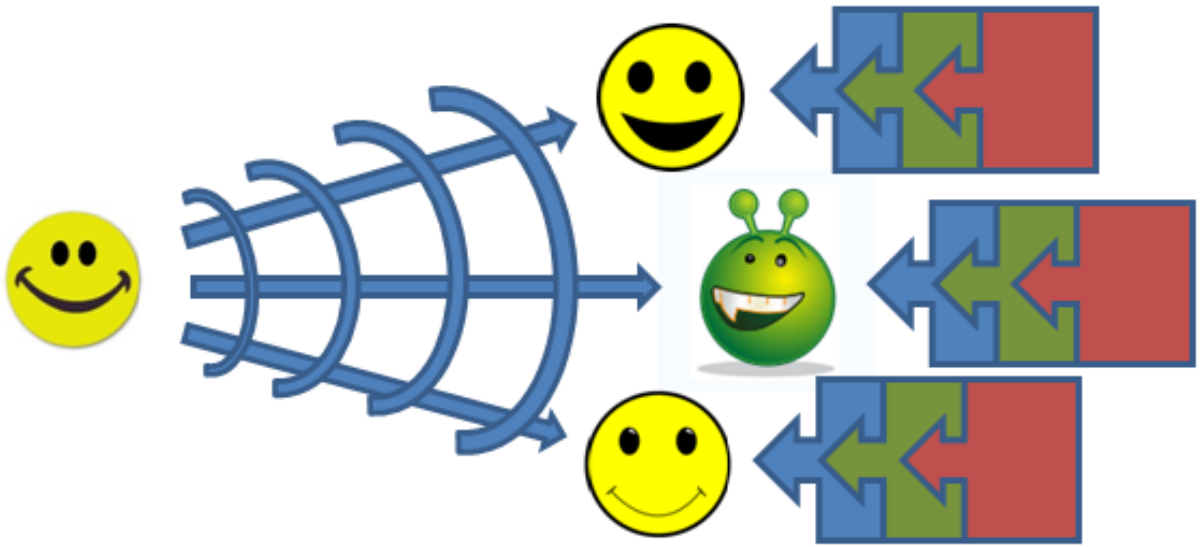
· Study the benefits of each topology

· Proceed with concrete tests to determine the best (or most appropriated) to each situation: Centralized (server), decentralized, hybrid



· Study the benefits of each topology

· Proceed with concrete tests to determine the best (or most appropriated) to each situation: Centralized (server), decentralized, hybrid



Our goal: Build an agent-based infrastructure, implemented in CLAIM, for an Ambient Intelligence system.

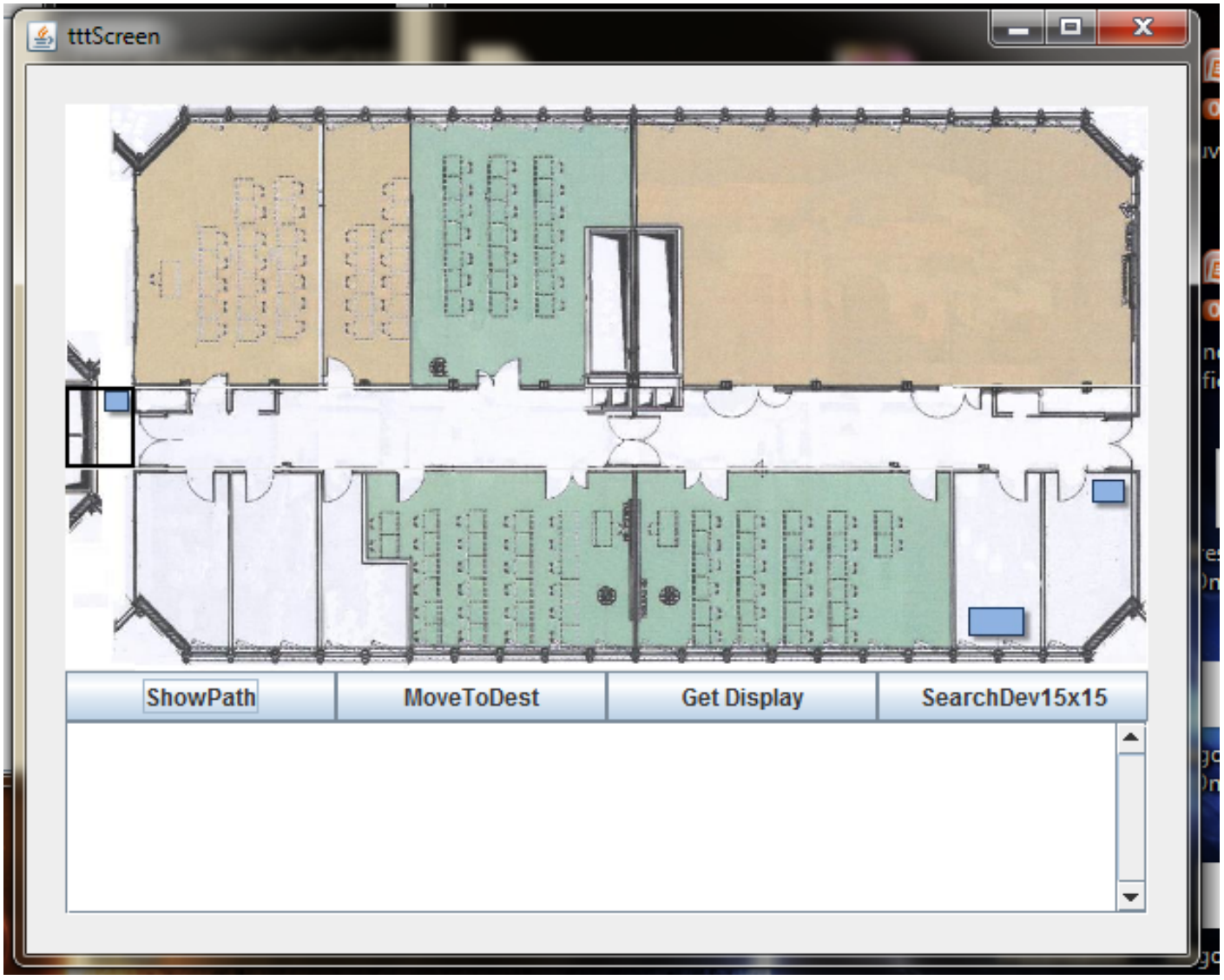
What was done: a first version, implemented in CLAIM, that offers context-awareness in terms of location and available resources.

Future work: implementation of ontologies for knowledge representation, consideration of other types of context (like social context) and anticipation of user's intentions. Also, integration of actual personal devices in the system.

- AmI
- Context-awareness
- Agents
- CLAIM
- Agentification
- Interaction
- Anticipation
- Ontologies
- Conclusion
- Demo

- DEMO -

- AmI
- Context-awareness
- Agents
- CLAIM
- Agentification
- Interaction
- Anticipation
- Ontologies
- Conclusion
- Demo



Canut, M.-F., Dubois, E., Glize, P., Sénac, C., and Truillet, P. (2009).

Systemes sociotechniques ambiants : du scenario a la maquette.
Ecole d'Ete Intelligence Ambiante.
Atelier.

Chen, G. and Kotz, D. (2000).

A survey of context-aware mobile computing research.
Technical report, Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

Dey, A. and Abowd, G. (2000).

Towards a better understanding of context and context-awareness.
CHI 2000 workshop on the what, who, where, when, and how of context-awareness, pages 304–307.

Henricksen, K., Indulska, J., and Rakotonirainy, A. (2002).

Modeling context information in pervasive computing systems.
Lecture notes in computer science, pages 167–180.

Krummenacher, R., Lausen, H., Strang, T., and Kopeckỳ, J. (2007).

Analyzing the modeling of context with ontologies.
International Workshop on Context-Awareness for Self-Managing Systems.

Pezzulo, G. (2008).

Anticipation and anticipatory systems: an introduction.

Ramos, C., Augusto, J., and Shapiro, D. (2008).

Ambient intelligence - the next step for artificial intelligence.
IEEE Intelligent Systems, pages 15–18.

Rosen, R. (1985).

Anticipatory systems.
Pergamon Press New York.

Seghrouchni, A. E. F. (2008).

Intelligence ambiante, les defis scientifiques.
presentation, Colloque Intelligence Ambiante, Forum Atena.

Suna, A. and El Fallah Seghrouchni, A. (2007).

Programming mobile intelligent agents: An operational semantics.
Web Intelligence and Agent Systems, 5(1):47–67.

Weiser, M. (1993).

Some computer science issues in ubiquitous computing.
Communications - ACM, pages 74–87.

Thank you!

Any Questions?