Situation aware interactive agents

Merik Meriste
Tartu University Institute of Technology
Estonian Centre on Dependable Computing
**Problem en gros**

Situation aware networked systems embedded in an environment - situated cooperating networked artefacts.

**How to obtain, sustain common operating picture in a networked environment under conditions of uncertainty?**

**Directions:**

1. Deriving system models in the presence of inconsistent, incomplete, or conflicting information concerning their functioning and interaction

2. Developing the underpinning ‘foundations’ of engineering of situation aware autonomous components and systems
Tools?

- Software engineering since 60s
  - software entities under full control
- Practice today?
  - Software embedded in open, networked, hardly predictable complex environment
  - High inherent complexity of applications
  - The problem of effective development
- Software evolution
  - module, procedure, object, component, generic component, …
- NB! Invocation of an entity always by an external event
- Autonomous software (?)
  “software being able to handle its complexity on its own” [Weiss, 2005]
Autonomous software?

Self-governing software entities:
“autonomous” from Greek term “auto + nomos” (”self + law”)

- Encapsulating invocation control!
- Trends today?
  - Autonomic computing
  - Service oriented computing
  - Agent based computing
  - Context aware computing
  - ....

NB! It will be component’s decision how to react to an external event (cf. traditional concepts!)

NB! Autonomous software is at its start
- today more vision than practice in software industry
Agents are **advanced components** that can be set a specific task but then given the **autonomy** to set goals and carry out the operations necessary to achieve them.

Agents can respond to changes in their environment, generate and attempt to achieve goals, and have the capacity to **interact** and **cooperate** with other agents.

Inevitable features of agents are persistency, interaction with an environment and situational awareness.
Multi-agent system
**Ad-hoc networks of smart sensors**

Motes are autonomous devices with three key abilities:
- to sense the environment via sensors,
- to compute – a processor and some memory,
- to communicate – wireless communication via radio

Motes serve as sensors - “smart” motes receive context information from active peers (beacons, sensors, motes):
- Sensor data collected by peer motes in the area
- Data available on the area from external data sources (maps, weather reports, etc)

Our interests – ad hoc networks of spatially smart sensors

Test area: **TTU main building III floor**
Agent based software development: *Kratt*

- Experimental development environment *Kratt* for situation-aware autonomous agents and agent communities
- Implemented in C#, C++
- Serves for the development of situation aware interactive components for application software
- Application areas –
  - situation aware information agents,
  - interactive GIS tools (*Kaardikratt*),
  - networked situation aware autonomous agents (sensors, motes), tracking systems, etc.
Agent in Kratt

- **Communicator**: Exchange of messages with other agents
- **Manager**: Control of agent activities
- **Actor**: Initiating services
- **Monitor**: Monitoring events and activities
Specification of agents

- Based on primitive templates
  - property,
  - service,
  - agentmessage,
  - agent description

- Templates are generic - serve for (re)configuration of agents for a system
Spatial agents

Agents acquire the information from different GIS, IS, DB and, handle this information and, can be dynamically applied to satisfy requests of an application:

- *Map agents* search, process and deliver fragments of the base raster map

- *Vector agents* manage the vector slices activated interactively on the user's view of the map

- *Information agents* manage the map objects related information interacting with particular DB agents

- *Tracking agents* transmit positions of mobile objects
Map agents in Kaardikratt

- An application is configured dynamically, depending on available active agents
- Agent Management System (AMS) is also an agent
- Agent is specified and adjusted by means of generic templates

Agents on maps of Estonia
Our approach

Aims: Architecture, tools and development environment for situation aware autonomous software entities and dynamic systems of them.

Basis of the approach: Models of autonomous computation; experiments in situation aware autonomous software - prototypes of agent engine and development environment, applications (map agents, *ad hoc* networks of motes, …)
Our team & partners

Joint research group with three nuclei:
- RT systems group (Leo Motus), Tallinn University of Technology
- Agent Lab (Merik Meriste), Tartu University Institute of Technology
- Institute of Software Technology (Walter Dosch), University of Lübeck

Active cooperation on interactive computation models with Brown University (P.Wegner), Univ. of Connecticut (D.Goldin)
Computations - a feature space?

- Interaction
- Topological time
- Topological and metric time
- Three concepts of multiple metric times
- Time awareness
- Conventional models of computation
- Algorithmic
- Partially controlled
- Proactive
- Adaptive
- Nonalgorithmic
- Time constraint non-algorithmic
Interactive computation [P. Wegner]

Sequences of input-output pairs \((i_1, o_1), \ldots, (i_k, o_k), (i_{k+1}, o_{k+1}), \ldots\)

- Input \(i_{k+1}\) depends on output \(o_k\)
- Input elements are supplied dynamically by an external mechanism not under the control of the computation

**Sequential interactive**: a stream of I/O pairs

\[(i_1, o_1), (i_2, o_2), \ldots\]

**Inherently interactive**: multiple streams

\[(i_{11}, o_{11}), (i_{12}, o_{12}) \ldots\]

\[\ldots\]

\[(i_{k1}, o_{k1}), (i_{k2}, o_{k2}) \ldots\]

**Situation aware interactive** ... ?
A sample of taxonomy
Problem *en gros* - a challenge!

Situation aware networked systems embedded in an environment - situated and cooperating smart networked artefacts – modelling, simulation, design, control.

How to obtain, sustain common operating picture in a networked environment under conditions of uncertainty?

**Directions:**

1. Deriving system models in the presence of inconsistent, incomplete, or conflicting information concerning their functioning and interaction
2. Developing the underpinning ‘foundations’ of engineering of situation aware autonomous components and systems