Introduction to the Course on Agent-Oriented Modelling and Multi-Agent Systems

Professor Kuldar Taveter,
Alex Norta, PhD.,
Department of Informatics
Tallinn University of Technology
Overview of today

- Logistics
- Who am I?
- Who are you?
- Why this course?
- What is agent?
Logistics

- Lectures on Wednesdays at 10.00-11.30 in the main building of TUT, lecture hall U06A-229
- Workshops/lab classes for stationary students on Wednesdays at 12.00-13.30 in the ICT building of TUT at Akadeemia Road 15A, computer class ICT-501
- Consultation times by Prof Kuldar Taveter: on demand in the ICT building of TUT at Akadeemia Road 15A, Room ICT-634
- Consultation times by Dr Alex Norta: on demand in the ICT building of TUT at Akadeemia Road 15A, Room ICT-639
  - or Skype: alexbafana
A workshop at 12.00-13.30 in ICT-501 on the concept of agent and overview of supporting tools and miniprojects
Next time

- On 11 February 2014:
  - Lecture “The Models for Agent-Oriented Analysis: Goal, Role, and Organization Models”
  - Workshop on the analysis models
Who am I?

- Name: Kuldar Taveter
- Position: Professor, Chair of Software Engineering
- Education:
  - Dip.Eng., TUT, 1988
  - M.Sc., TUT, 1995
  - Ph.D., TUT, 2004
- Work experience:
  - 1985-1989: Institute of Cybernetics
  - 1989-1993: Private companies
  - 1993-1998: Department of Informatics of TUT
  - 1997-2005: Technical Research Centre of Finland
  - 2005-2008: The University of Melbourne, Australia
  - 2008-: Department of Informatics of TUT
  - Jan-Aug 2011: University of South Carolina, USA
- Research areas: Agent-oriented software engineering, engineering of sociotechnical systems, multiagent systems, intelligent systems, ambient intelligence, agent-based simulation
Who is Alex?

Alex Norta
Postdoctoral Researcher at Tallinn University of Technology
Finland | Research
Current
Tallinn University of Technology, Freelance
Previous
University of Oulu, SRll, Helsinki University
Education
Eindhoven University of Technology
Communication

Course webpage:
http://maurus.ttu.ee/sts/?page_id=1891

Course mailing list: aine.aom@lists.ttu.ee
kuldar.taveter@ttu.ee, alexander.norta@ttu.ee, or Skype: alexbafana
Learn how to **design** by **AOM** a **software-intensive composite artifact** that delivers the overall solution for the end users through interactions between its components and where each component follows the execution loop of an abstract agent.
What is design?

- A specification of an artifact, manifested by an agent, intended to accomplish goals, in a particular environment, using a set of components, satisfying a set of requirements, subject to constraints.
What is the artifact?

* The entity (or class of entities) being designed. Note: this entity is not necessarily a physical object.

* Classes of artifacts:
  * **physical artifacts**, both simple, such as boomerangs (single-component), and composite, such as houses (made of many types of components)
  * **processes**, such as business workflows
  * **symbolic systems**, such as programming languages
  * **symbolic scripts**, such as essays, graphic models, animations, and software
  * **laws, rules and policies**, such as a criminal code
  * **human activity systems**, such as software design projects, committees and operas
What kind of artifact?

- A sociotechnical system: composite ICT artifact supporting social processes
- Human activity system
- Components may be embedded in physical devices or products – physical artifact
- Other examples:
  - A system of e-governance services – process
  - Automation of a business or governmental process – process
What can be the artifact in the context of our course?

* Software-intensive system with components embedded in physical devices or products (e.g. IVSM, IAPM)
* Human-agent community (e.g. IVSM, IAPM)
* A physical product with substantial ICT components (e.g. MADM)
* Automation of a business or governmental process (e.g. IABM, IVGM)
* Service (system) of e-government (e.g. IVGM)
* A work product of digital design, such as a serious game or animation (e.g. MADM)
How is this course positioned?

- Design of a sociotechnical system
- The process of creating a new product to be sold by a business to its customers
- Supporting social behavior through computational systems
- A system that perceives its environment and takes actions that maximize its chances of success
- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software

Diagram:

- Product Design
- Social Computing
- Software Engineering
- Intelligent Systems
Conceptual space for designing sociotechnical systems

Motivation layer

System design layer

Deployment layer
Conceptual space for designing STS populated with concepts
Three perspectives required

- Behaviour
- Interaction
- Knowledge
Miniproject

- Design by agent-oriented modelling and prototyping or simulation of a sociotechnical system
- Desired features:
  - Open
  - Social
  - Intelligent
  - Adaptive
  - Solves a real-world problem – no dating services!
- Interdisciplinary examples from 2013: Two Mektory projects:
  - Healthminer
  - Phoenix
Assessment

- Presentation of goal model and role models – 5%
- Presentation of analysis models – 5%
- Presentation of analysis and design models – 5%
- Final presentation of miniprojects – 15%
- Submitted report approximately 2000 words of text along with appendices, figures, tables, graphs and examples of source code – 20%
- Final written exam – 50%
- Teams of 3-4 students and topics for miniprojects should be agreed and **short proposals** should be sent to Prof Taveter by the 18th of February
Workshops / lab classes

- Using JADE (http://jade.tilab.com/) or some other relevant platform of your choice like Jason (http://jason.sourceforge.net/wp/) for implementing your miniproject
- Presentations on miniprojects
- Help and advice on miniprojects


Agent-oriented modelling

The Art of Agent-Oriented Modeling
Leon S. Sterling and Kuldar Taveter
The book’s mission

To address how computing can support individuals and social organizations in the environment where the computing is:

- Pervasive;
- Deployed over a range of devices;
- With multiplicity of users

Approach for engineering sociotechnical systems that are:

- Open;
- Intelligent;
- Adaptive
What kinds of sociotechnical systems are we interested in?

* Open
  * Self-interested human (and man-made?) agents
  * Different availabilities

* Social:
  * Social relationships between human and man-made agents
  * Man-made agents represent and/or assist human agents
  * Incentives for human (and man-made?) agents:
    * Extrinsic: money or tax benefits
    * Social: public accreditation or leader-board position
    * Intrinsic: personal interest in a social cause, altruism, or hobby

* Adaptive:
  * Uncertainty, ambiguity, bias
  * Reasoning about trust and reputation
  * Information collection, fusion, and decision-support

* Intelligent
Why this course?

Grand social challenges (after Prof Mike N. Huhns):

- Economy: prices, taxes, and credit
- Energy: distribution and allocation
- Environment: pollution control, carbon allowances, and water allocation
- Transportation: people and goods
- Telecommunication systems
- What is common for these challenges?
“Service-Oriented” “Cloud” “Ecosystems” + “Big Data” + “Mobility”
Where does SOCC-automation make sense?

For tedious, repetitive human action in value chain

Trust & reputation are part of subjective, individual urge

Influenced by data, information, personal experience

Information/application-logistics by discrete SOCC-automation
Why this course?

- The world is naturally distributed…
- … but the applications are centralized!
- Let us take shopping apps
Price and Feedback Database

Price information

Price information

Price information

Price information

Price information

Price information

Price information
What kinds of problems am I talking about?

- Allocation of scarce resources (e.g., electric power, parking spaces, emergency care, transportation)
- Distributed situation assessment (e.g., traffic jams)
- Decentralized decision-making (e.g., finding a grocery store, legal decision-making, taxation, intelligent traffic control)
- Cooperative information exchange (currently no context, no help by other users)
What kinds of systems would be needed?

- Distributed systems that obtain control information from the edge of the system to provide desired behaviors at the edge of the system.

Examples:
- Electricity distribution
- Grocery shopping
- Healthcare
- Transportation
- Traffic control
- Telecom bandwidth
More examples
(“Smart Society” project)

* An application that helps you find local citizens who provide their private vehicle to take you to a place that you wish to visit and automatically coordinates the needs of different groups of tourists, in order to optimize time and travel expenses...
* An innovative application for your smart-phone, that recommends you the best itinerary from your current position to a point of interest, that cleverly combines public transport, car sharing, bike sharing and other services, and makes you reach also peripheral places, taking into account your user profile and preferences.
* An innovative application for your smart-phone that allows you to quickly request and obtain information in real time with the help of local citizens who, through their experience, make their time and their knowledge available to the visitors during their tours...
* An innovative application for your smart-phone that allows you to take part in a treasure hunt within the city, by suggesting the best places to visit according to your profile and your personal interests, and to win real prizes such as discount coupons for museums, exhibitions, restaurants, hotels, etc.
* An innovative application for your smart-phone that allows you to quickly find a real tour guide that suits you, in other words, an expert who offers you a customized tour in the city where you are, and together with other like-minded people, according to your profile and your interests...
Two kinds of systems

- Predominantly client-server systems
- Increasingly more peer-to-peer systems
- Peer-to-peer subsumes file sharing
Client-server systems

- **Client:**
  - queries;
  - waits;
  - receives;

- **Server:**
  - waits;
  - serves;
  - does not query.
Peer-to-peer systems

- All nodes are equal
- Each node can work as a client or as a server
- Each node can offer (computing, memory, bandwidth, information) resources
EXAMPLE: Biggest P2P system

https://bitcoin.org/bitcoin.pdf

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they’ll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.
Client-server vs. peer-to-peer

- Peer-to-peer resources are distributed
- Peer-to-peer is more robust
- Peer-to-peer is more secure
- Peer-to-peer is trickier to test
- Peer-to-peer is trickier to manage
- Hybrid network: peer-to-peer network + client-server directory
Application areas of P2P systems

- Communication
  - Skype
- File sharing
  - BitTorrent
- Gaming
- Security

http://prism-break.org/en/
There is no pure peer-to-peer
What is **Agent**?

- An active entity as opposed to a passive entity
- An entity that can act in the environment, perceive events, and reason
- An entity that acts on behalf of someone or somebody

http://en.wikipedia.org/wiki/Agent#In_computer_science_and_artificial_intelligence
Agent

Agent is an entity that perceives and affects its environment and performs reasoning.

Agent is:
- reactive;
- proactive;
- social.

Agent interacts in an asynchronous way.
The abstract agent architecture
The execution loop of an abstract agent

\[\textbf{while} \text{ the agent is unfulfilled} \ \textbf{do} \]
\[
\begin{align*}
\text{sense the environment;} \\
\text{update the knowledge base;} \\
\text{reason;} \\
\text{choose actions;} \\
\text{act;}
\end{align*}
\]
\[\textbf{end while}\]
Anthropomorphic qualities

- Beliefs
- Responsibilities
- Expectations
- Capabilities
- Goals
- Desires
- Intentions
Example agents
Mobile devices

- Personal
- Social

- Are aware of their location at least up to the granularity of a GSM cell
- Can perceive their immediate environment through embedded light, sound, and motion sensors
- Tablets and cell phones available for experimentation in miniprojects!
Next time

- On 11 February 2014:
  - Lecture “The Models for Agent-Oriented Analysis: Goal, Role, and Organization Models”
  - Workshop on creating AOM analysis models with Microsoft Visio