Agent-oriented modeling for social grocery shopping and other societal information systems

Prof. Kuldar Taveter, Tallinn University of Technology, Estonia
Who am I?

- Name: Kuldar Taveter
- Position: Professor, Chair in 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
  - 2011 : Department of Computer Science and Engineering of SCU
- Research areas: requirements elicitation and analysis, agent-oriented modelling, fast prototyping, agent-based simulation, ontologies
Basic Facts about Estonia

- North-East Europe
- Capital Tallinn
- Population 1,34 mio
- Area 45,000 km², comparable to the Netherlands and Denmark
- Parliamentary republic, independence Feb 24 1918
- EU, May 1 2004
- Schengen treaty, Dec 21 2007
- Euro zone, Jan 1 2011
People and society

- Nordic mindset
- Peaceful and hard-working people
- Safe and stable society
- 70/30% of population native Estonian/Russian-speakers
- Foreign languages widely spoken: English, Russian, German, Finnish....
- 3 million tourists visit Estonia every year
Nature and country

- 4 seasons
- Well-preserved nature
- 1520 islands
- 1000 lakes...
• Advanced IT society - free Internet access in many public areas, on coaches, trains, etc.


• The headquarters of Skype lie in Tallinn
Higher (tertiary) education is offered at universities and professional higher education institutions.

Ca 2/3 of the age group study in higher education institutions - there are ca 68 000 students in Estonia.

There are 8 universities in Estonia.

All institutions have introduced a bachelor-master (3+2) structure for most study programmes.

Growing number of English taught programmes are offered, especially at Master level.
Universities offering international academic programmes

Public
Estonian Academy of Arts – www.artun.ee
Estonian Academy of Music and Theatre – www.ema.edu.ee
Estonian University of Life Sciences – www.emu.ee
Tallinn University – www.tlu.ee
Tallinn University of Technology – www.ttu.ee
University of Tartu – www.ut.ee

Private
Estonian Business School – www.ebs.ee
University Nord – www.nord.ee
Tallinn University of Technology

- Founded as an engineering college in 1918
- Acquired university status in 1936
- The second largest university in Estonia with about 14,200 students, 2,000 employees and with more than 54,000 graduates
- Courses taught in Estonian, English, and Russian
- International students ~5%
- 134 Bachelor’s, Master’s, and Doctoral degree programs
- The biggest faculty of economics and business administration in Estonia
Faculties

- Civil Engineering
- Power Engineering
- Information Technology
- Chemistry and Materials Technology
- Mechanical Engineering
- Mathematics and Natural Sciences
- Social Sciences
- School of Economics and Business Administration
International programs taught in English

- **Bachelor’s:**
  - International Business Administration
  - Law
  - International Relations

- **Master’s:**
  - MBA
  - Law
  - International Relations and European Studies
  - Technology Governance
  - Software Engineering
  - Cyber Security
  - Industrial Engineering and Management
  - Environmental Management and Cleaner Production
  - Materials and Processes of Sustainable Energetics
  - Health Care Technology
  - Design and Engineering
Department of Informatics


- Faculty
  - 33 members
    - 4 professors
    - 6 associate professors
    - 6 lecturers
    - 7 assistant lecturers
    - 10 researchers
  - Qualifications:
    - PhD: 16 members
    - M.Sc.: 17 members, among them 10 PhD students
Department of Informatics: Composition

- Chair of Information Systems (ass. prof. E. Eessaar):
- Chair of Software Engineering (prof. K. Taveter)
- Chair of the Foundations of Informatics (prof. R. Kuusik)
- Chair of Knowledge-based Systems (prof. J. Tepandi)
- Chair of Information Security (prof. A. Buldas)
- Data Mining Laboratory
- Laboratory of Socio-Technical Systems
  - Evolutionary IS by agents
  - Agent-based simulation of asymmetric threats
  - Agent-based simulation of aircraft turnaround
- Laboratory of Web Services
Department of Informatics: Teaching

- B.Sc. and M.Sc. in Informatics,
- B.Sc. and M.Sc. in Business information technology
- PhD in Informatics
- General informatics (8 ECP) in a majority of study programs offered by TTU
- An introductory course in information systems (5 ECP) for students at the faculties of information technology, social sciences, and economics and business administration
- Special courses in informatics (e.g., C++ and OOP, basic course in Internet, M.Sc. and PhD seminars, etc.)
- Further training in informatics (fee-charging courses)
Department of Informatics: Research

- Information systems (Government-funded project “Model-based Creation and Management of Evolutionary Information Systems”)
- Data mining (the method of monotone systems, used in several industry-oriented research projects)
- Web-services (application-oriented research projects in e-billing, power engineering, etc.)
- Intelligent systems and software agents (tactile feedback devices, methods of problem domain analysis and design of multiagent systems and agent-based simulation systems)
# PhD studies

<table>
<thead>
<tr>
<th>Year</th>
<th>Admissions</th>
<th>Defenses</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>2005</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>2006</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>2007</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2008</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2009</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>2010</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>2011 (plan)</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>
Agent-oriented modelling

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

- To address how computing can support social organizations in the environment where the computing is:
  - Pervasive;
  - Deployed over a range of devices;
  - With multiplicity of users
- Approach for engineering software systems that are:
  - Open;
  - Intelligent;
  - Adaptive
The “agent” metaphor

- 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
The abstract agent architecture
Concepts of AOM
## Model types of AOM

<table>
<thead>
<tr>
<th>Viewpoint models</th>
<th>Viewpoint aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstraction layer</strong></td>
<td><strong>Interaction</strong></td>
</tr>
<tr>
<td><em>Conceptual domain modeling</em></td>
<td>Role models and organisation models</td>
</tr>
<tr>
<td><em>Platform-independent computational design</em></td>
<td>Agent models and acquaintance models, interaction models</td>
</tr>
<tr>
<td><em>Platform-specific design and implementation</em></td>
<td>Agent interface and interaction specifications</td>
</tr>
</tbody>
</table>
## Mapping Prometheus to viewpoint framework

<table>
<thead>
<tr>
<th>Viewpoint models</th>
<th>Viewpoint aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstraction layer</strong></td>
<td><strong>Interaction</strong></td>
</tr>
<tr>
<td>Conceptual domain modeling</td>
<td>Analysis</td>
</tr>
<tr>
<td></td>
<td>Overview</td>
</tr>
<tr>
<td></td>
<td>Diagram, System Roles Diagram</td>
</tr>
<tr>
<td><strong>Platform-independent computational design</strong></td>
<td>Agent</td>
</tr>
<tr>
<td></td>
<td>Acquaintance Diagram, Interaction</td>
</tr>
<tr>
<td></td>
<td>Diagrams, Protocol Diagrams,</td>
</tr>
<tr>
<td></td>
<td>System Overview Diagram</td>
</tr>
<tr>
<td><strong>Platform-specific design and implementation</strong></td>
<td>Event Descriptors</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Mapping Tropos to viewpoint framework

<table>
<thead>
<tr>
<th>Viewpoint models</th>
<th>Viewpoint aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstraction layer</strong></td>
<td>Interaction</td>
</tr>
<tr>
<td>Conceptual domain modeling</td>
<td>Actor Diagram</td>
</tr>
<tr>
<td>Platform-independent computational design</td>
<td></td>
</tr>
<tr>
<td>Platform-specific design and implementation</td>
<td>Agent Interaction Diagrams</td>
</tr>
</tbody>
</table>
### Mapping MaSE to viewpoint framework

<table>
<thead>
<tr>
<th>Viewpoint models</th>
<th>Viewpoint aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Abstraction layer</strong></td>
<td>Interaction</td>
</tr>
<tr>
<td>Conceptual domain modeling</td>
<td>Sequence Diagrams, Role Model</td>
</tr>
<tr>
<td>Platform-independent computational design</td>
<td>Protocol Diagrams, Agent Class Diagram</td>
</tr>
<tr>
<td>Platform-specific design and implementation</td>
<td>Plan Diagrams, Deployment Diagrams</td>
</tr>
<tr>
<td></td>
<td>Information</td>
</tr>
<tr>
<td></td>
<td>Goal Diagram, Use Cases, Role Model</td>
</tr>
<tr>
<td></td>
<td>Concurrent Tasks, Agent Class Diagram</td>
</tr>
<tr>
<td></td>
<td>Behavior</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The case study of social grocery shopping

- Customers post the prices they paid for their groceries (this could be automated by querying the RFID tags of the items) and QoS information.
- A prospective shopper enters a grocery list and obtains a pointer to the store(s) with the lowest total price (and best service).
- Each customer has an agent representing his/her interests and interacting with the agents of the other customers.
Overall goal model

- **Customer**
  - **Shopping**
    - **Store**
    - **Assistant**
      - **Creating shopping list**
      - **Finding stores**
      - **Deciding stores**
      - **Transacting**
Creating shopping list

Customer

Creating shopping list

Adding a product

Picking a product from typical shopping list

Creating typical shopping list

According to the need

Fast

Simple

Adding a product to typical shopping list

Frequently bought

Compiler
Finding stores

According to the shopping list

Close

Optimal

Calculating the route

Calculating the fuel cost

Customer

Compiler

Calculator

Calculator
Deciding stores

- Decision-maker
- Customer
- Deciding stores
  - Preferred by customer
  - High QoS
- Creating shopping basket
  - Advantageous
  - With up-to-date data
- Adding a product
  - High-quality
Transacting

Customer

Transacting

Store

Register

Minimal participation
Safe
Registered transaction
Reliable
Posting transaction information
Simple

Includes QoS
Fast
Minimal participation
# Role model for Customer

<table>
<thead>
<tr>
<th>Role</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The role of a customer buying groceries</td>
</tr>
</tbody>
</table>
| Responsibilities | Creating the shopping list  
|               |   - Adding a product to the shopping list  
|               |   - Picking a product from the typical shopping list  
|               | Determining preferences  
|               | Confirming the stores found by the Compiler  
|               | Confirming the decisions made by the Decision-maker  
|               | Driving to the stores  
|               | Making transactions  
| Constraints   | For picking products from the typical shopping list, the typical shopping list must have been created for the Customer  
|               | To find the most advantageous shopping baskets, the Customer should consider the stores found by the Compiler and the decisions made by the Decision-maker  
|               | To benefit from the transaction information posted by other customers, the Customer must authorize posting of his/her transaction information |
# Role model for Compiler

<table>
<thead>
<tr>
<th>Role</th>
<th>Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>The role of shopping list and store list compiler</td>
</tr>
<tr>
<td>Responsibilities</td>
<td>Creating and managing the typical shopping list by the buyer</td>
</tr>
<tr>
<td></td>
<td>- Add a product to the typical shopping list</td>
</tr>
<tr>
<td></td>
<td>Storing shopping lists for statistics and data mining</td>
</tr>
<tr>
<td></td>
<td>Finding potential stores with the help of the Calculator</td>
</tr>
<tr>
<td>Constraints</td>
<td>The shopping list by the Customer must be considered when finding potential stores</td>
</tr>
<tr>
<td></td>
<td>The typical shopping list must be considered when finding potential stores</td>
</tr>
<tr>
<td></td>
<td>The proximity of stores must be considered when finding potential stores</td>
</tr>
<tr>
<td></td>
<td>For creating the typical shopping list, the Customer must have created shopping list(s) before.</td>
</tr>
</tbody>
</table>
# Role model for Decision-maker

<table>
<thead>
<tr>
<th>Role</th>
<th>Decision-maker</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The role of decision-maker about the stores and their shopping baskets</td>
</tr>
<tr>
<td><strong>Responsibilities</strong></td>
<td>Comparing potential shopping baskets for the stores found by the Compiler Deciding the stores Deciding the shopping basket for each store selected</td>
</tr>
<tr>
<td><strong>Constraints</strong></td>
<td>The preferences by the Customer must be honored when deciding about the stores and their shopping baskets The transaction information posted by other customers, including QoS, must be considered when deciding about the stores and their shopping baskets The transaction information should be considered in the order of its age The fuel costs computed by the Calculator must be considered when deciding about the stores and their shopping baskets The shopping baskets created should be maximally advantageous for the Customer The shopping baskets should include high-quality products with up-to-date data</td>
</tr>
</tbody>
</table>
Organization model
Design models

- Agent and acquaintance models
- Interaction models
- Knowledge models
- Scenarios and behavior models
Agent and acquaintance model
Results from initial experiments by Prof Huhns and Hongying Du

- Savings up to 21% can be obtained by social grocery shopping!
Societal information systems

- Software agents represent members of the society
- The tasks:
  - Regulation (e.g., banking)
  - Allocation of scarce resources (e.g., energy, parking spaces, emergency care)
  - Distributed situation assessment (e.g., traffic jams)
  - Decentralized decision-making (e.g., grocery shopping, choosing healthcare providers)
- Open distributed systems
Conclusions

- AOM is appropriate for problem domain analysis for societal information systems
- Full potential of social networks has not been yet employed
- Further experiments required in different areas
- AOM -> NetLogo mapping?