Agent-Oriented Modelling (AOM) for Designing New Software-intensive Products

Professor Kuldar Taveter, Head of the Sociotechnical Systems’ Laboratory, Department of Informatics, Tallinn University of Technology, Estonia
Who am I?

- **Name:** Kuldar Taveter
- **Position:** Professor in Software Engineering, Head of the Laboratory of Sociotechnical Systems
- **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
  - Apr – May 2016: Shanghai University for Science and Technology, China
- **Research areas:** Agent-oriented software engineering, engineering of sociotechnical systems, multiagent systems, intelligent systems
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 is the artifact in our course?

- Software-intensive system with components embedded in physical devices or products
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
How to represent a vision?
Agent-oriented modeling
(MIT Press, 2009)

The Art of Agent-Oriented Modeling
Leon S. Sterling and Kuldar Taveter
# Notation for goal models

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Goal" /></td>
<td>Goal</td>
</tr>
<tr>
<td><img src="image2" alt="Quality goal" /></td>
<td>Quality goal</td>
</tr>
<tr>
<td><img src="image3" alt="Role" /></td>
<td>Role</td>
</tr>
<tr>
<td><img src="image4" alt="Relationship between goals" /></td>
<td>Relationship between goals</td>
</tr>
<tr>
<td><img src="image5" alt="Relationship between goals and quality goals" /></td>
<td>Relationship between goals and quality goals</td>
</tr>
</tbody>
</table>
Tourist advisor
Goal model for tourist advisor

**Value proposition**

- **Authentic**
  - Tourist

- **Have a tour**
  - GPS
    - Service Provider
  - Advisor

**Precise**

- **Determine the location**
- **Point out sightseeing spots**

**Profile honoured**

- **Loved by locals**

**Reputation enhancing**

- **Check tourist profile**
- **Check the rating**
- **Match tourist profile**
- **Guide the tourist**
- **Rate the spot**
Smart parking
Goal model for smart parking

Value proposition

Secure
Advantageous
Maximal societal welfare

Preferences honored

Find a parking spot

Check the options
- Up-to-date

Decide
- Optimal

Direct the driver
- Maximal granularity

Register arrival
- Accurate

Initiate payment
- Precise

Register departure
- Accurate

Stop payment
- Precise

Parked smartly

Commuter Parking Assistant

Parked Service Provider

GPS Service Provider

Parking Service Provider

Parking Service Provider
Fair grocery shopping

Product information

Product information

Product information

Fair grocery shopping
Fair grocery shopping
Goal model for fair grocery shopping

Value proposition

Customer

Societally

Performs shopping

Advantageously

Store

Assistant

Perform shopping

Join the system

Create shopping list

Find potential stores

Decide stores’ shopping baskets

Decide the route

Buy products

Exchange price and quality information

Easy

According to the need

Close

Minimal overall price

Optimal

Secure

Minimal participation

Simple

Quality products
First an Idea
Then a Vision
Then a Product Backlog
How to create Product Backlog?
How to manage Product Backlog?
The methodology of Agent-Oriented Modelling (AOM)
Conceptual space for design

- Motivation layer
- System design layer
- Deployment layer
Conceptual space populated with concepts
Agile Agent-Oriented Modeling (Kuldar Taveter, Tanel Tenso)

- Problem domain (product backlog) is presented as a goal model
- A goal model connects functional requirements, quality requirements, and roles
- A goal model is constantly updated within iterations of an agile design process
User stories

- As a user playing some role, I must be able to perform some activities [in order to achieve some goal]
Examples of user stories

- As a user playing some role, I must be able to do something in order to achieve some goal
- Example 1: As a Receptionist I want to Register patient to Monitor health condition
- Example 2: As a Seller I want to Ship order to Provide product
Other examples

- Issue Management
- Simulation of crisis management (EU FP7 project)
Goal model for the application of issue management
User stories for the sub-goal “Manage Release Vehicles”

- As a (human playing the role of) Release Admin, I must be able to add a new Release Vehicle to manage release vehicles;
- As a Release Admin, I must be able to change Release Vehicles to manage release vehicles;
- As a Release Manager, I must be able to see a list of Release Vehicles to manage release vehicles;
- As a Release Manager, I should not be able to edit a list of Release Vehicles to manage release vehicles;
- As a Release Manager or Release Admin, I should be able to sort a list of Release Vehicles into the ascending or descending order to manage release vehicles.
A goal model for the application of crisis management simulation

Set Up Simulation
- Define Players Training
  - Set Up Player Type And Algorithm
- Define Training Format For Player
- Specify Levels of Preparedness

Run Simulation
- Specify Crisis Location
- Define Crisis Development

Gather Simulation Feedback

Simulate various crisis scenarios

NOTE: for example these levels of preparedness will be used when running simulation to determine the response by players in crisis situation
User stories for the sub-goal “Set Up Player Type and Algorithm”

- As a Simulation Model Administrator, I want to define the “Snowstorm Training” type of training for a player of the “Emergency Service” type to set up player type and algorithm.
- As a Simulation Model Administrator, I want to define other types of training for a player of the “Emergency Service” type to set up player type and algorithm. NOTE: training types are “Earthquake Training”, “Chemical Burn Training”, etc.
A goal model for the functionality of the application

- **Cost-efficient**
- **Mitigate Snowstorm Effects**
  - **Prepare**
    - **Educate Population**
    - **Train Emergency Services**
      - **Perform Evacuation Exercises**
      - **Perform Large Scale Search Rehearsals**
      - **Perform Communication Rehearsals**
  - **Response**
    - Emergency Services
Elaborated goal model

Emergency Services

Mitigate Snowstorm Effects

Governments

Response

Restore

Timely

Coordinated

Fast

Adequate

Governments

Analyzed
# Example of Product Backlog

<table>
<thead>
<tr>
<th>As a</th>
<th>I want to</th>
<th>So that (I can)</th>
<th>Business Value</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR Manager</td>
<td>Publish new vacancies</td>
<td>Find candidates</td>
<td>80</td>
<td>20</td>
</tr>
<tr>
<td>Job Hunter</td>
<td>Apply for a job</td>
<td>Quickly apply for a job</td>
<td>80</td>
<td>40</td>
</tr>
<tr>
<td>HR Manager</td>
<td>Triage applicants</td>
<td>Politely eliminate unpromising candidates</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>Googlebot</td>
<td>effectively find and index all postings</td>
<td>Ensure that internet searchers can find job postings on this site</td>
<td>50</td>
<td>13</td>
</tr>
<tr>
<td>System Admin</td>
<td>quickly recognize and analyze system</td>
<td>ensure rapid resolution of technical problems</td>
<td>30</td>
<td>20</td>
</tr>
</tbody>
</table>
Example of Sprint Backlog

<table>
<thead>
<tr>
<th>User Story</th>
<th>Tasks</th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a member, I can read profiles of other members so that I can find someone to date.</td>
<td>Code the ...</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design the ...</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Meet with Mary about ...</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design the UI</td>
<td>12</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automate tests ...</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code the other ...</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>As a member, I can update my billing information.</td>
<td>Update security tests</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Design a solution to ...</td>
<td>12</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Write test plan</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Automate tests ...</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code the ...</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Hands-on-exercise (Alternative 1): Elaborate the below goal model into user stories!
Hands-on exercise (Alternative 2): Create a goal model for your product idea
Artifacts in agile methodologies

- Lean Startup: *validated learning* vs. working software
- Lean UX: *delivered value* vs. working software
Iterations of Scrum

- Product Backlog
- Sprint Backlog
- Sprint Planning
- 1 - 4 Week Sprints
- Daily Scrum
- Sprint Review
- Sprint Retrospective
- Potentially Shippable Product Increment
Conclusions

- Presentation and elaboration of problem domain is of critical value
- Analysis should be included in the iterative loop
- Sprint backlog has a goal
- User stories describe the implementation of (business) goals
- User stories are divided into tasks
- Business goal = Potentially Shippable Product Increment