Agent-Oriented Design Models II: Scenarios, Behaviour Models, and Service Models

Prof Kuldar Taveter, Tallinn University of Technology, Estonia
Please join the mailing list!

- aine.aom@lists.ttu.ee
- Subscribing: http://lists.ttu.ee/mailman/listinfo/aine.aom
Last time

- Agent models
- Acquaintance models
- Interaction models
- Knowledge models
<table>
<thead>
<tr>
<th>Agent type name</th>
<th>Tamagotchi</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>The software agent type for a digital pet</td>
</tr>
<tr>
<td><strong>Role(s)</strong></td>
<td>MyTamagotchi, FriendTamagotchi</td>
</tr>
<tr>
<td><strong>Environmental considerations</strong></td>
<td>TamagotchiShell, TamagotchiTown, Present</td>
</tr>
</tbody>
</table>
The acquaintance model for the case study of Tamagotchi
UML-style interaction diagram

Person

Tamagotchi

I am not happy

Please visit your friend

Could I come to your place?

Please come!

Good day!

Hello!

Please accept my present

Thanks for your present!

May I play with you?

OK

Playing with the friend
Interaction-sequence diagram

1. I am not happy
2. Please visit your friend
3. Could I come to your place?
4. Please come!
5. Good day!
6. Hello!
7. Please accept my present
8. Thanks for your present!
9. May I play with you?
10. OK
11. Playing with the friend
Interaction-frame diagram

Person

- I am not happy
- Please visit your friend

Tamagotchi

- Could I come to your place?
- I am sorry, you are too young
- Please come!
- Good day!
- Hello!
- Please accept my present
- Thanks for your present!
- Behave yourself!
- May I play with you?
- OK
- Sorry, I am not in the mood for playing right now
- Playing with the friend

Tamagotchi
# Knowledge attributes of a Tamagotchi

<table>
<thead>
<tr>
<th>Knowledge attribute</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tamagotchi name</td>
<td>String</td>
</tr>
<tr>
<td>Tamagotchi age</td>
<td>Integer</td>
</tr>
<tr>
<td>Tamagotchi gender</td>
<td>Enumeration (‘M’; ‘F’)</td>
</tr>
<tr>
<td>Tamagotchi generation</td>
<td>Integer</td>
</tr>
<tr>
<td>Tamagotchi weight</td>
<td>Real</td>
</tr>
<tr>
<td>Tamagotchi hunger level</td>
<td>Integer</td>
</tr>
<tr>
<td>Tamagotchi happiness level</td>
<td>Integer</td>
</tr>
<tr>
<td>Tamagotchi training level</td>
<td>Integer</td>
</tr>
<tr>
<td>Tamagotchi “life” points</td>
<td>Integer</td>
</tr>
<tr>
<td>Tamagotchi “gotchi” points</td>
<td>Integer</td>
</tr>
</tbody>
</table>
A partial knowledge model of a Tamagotchi
Overview of today

- Models at the system design layer
- The viewpoint framework
- Scenarios
- Behaviour models
- Intermezzo: Greeting scenario
- Service models
The conceptual space
Types of models at different layers

- **Motivation layer**: goal models and motivational scenarios, role and organisation models, domain models
- **System design layer**: agent models, acquaintance models, interaction models, knowledge models, scenarios, behaviour models, service models
- **Deployment layer**: agent interface and interaction specifications, data models and service models, agent behaviour specifications
## The viewpoint framework

<table>
<thead>
<tr>
<th>Viewpoint models</th>
<th>Viewpoint aspect</th>
<th>Interaction</th>
<th>Information</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction layer</td>
<td></td>
<td>Role models and organization models</td>
<td>Domain models</td>
<td>Goal models and motivational scenarios</td>
</tr>
<tr>
<td>Conceptual domain modelling</td>
<td></td>
<td>Agent models and acquaintance models, interaction models</td>
<td>Knowledge models</td>
<td>Scenarios and behavior models</td>
</tr>
<tr>
<td>Platform-independent computational design and implementation</td>
<td>Agent interface and interaction specifications</td>
<td>Data models and service models</td>
<td></td>
<td>Agent behavior specifications</td>
</tr>
</tbody>
</table>
Scenario

- Elaborates a motivational scenario
- A collective activity that models how a particular goal is achieved by agents enacting particular roles
- May contain sub-scenarios
# The scenario of socializing a Tamagotchi

<table>
<thead>
<tr>
<th>SCENARIO 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Goal</strong></td>
</tr>
<tr>
<td><strong>Initiator</strong></td>
</tr>
<tr>
<td><strong>Trigger</strong></td>
</tr>
<tr>
<td><strong>Failure</strong></td>
</tr>
</tbody>
</table>

## DESCRIPTION

<table>
<thead>
<tr>
<th>Condition</th>
<th>Step</th>
<th>Activity</th>
<th>Agent types and roles</th>
<th>Resources</th>
<th>Quality goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Visit the friend (Scenario 6)</td>
<td>Person/Owner, Tamagotchi/ MyTamagotchi, Tamagotchi/ FriendTamagotchi</td>
<td>Present</td>
<td>My Tamagotchi happy</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Find a partner for the Tamagotchi (Scenario 7)</td>
<td>Person/Owner, Tamagotchi/ Matchmaker</td>
<td>Friend_list</td>
<td>My Tamagotchi happy</td>
<td></td>
</tr>
</tbody>
</table>
## SCENARIO 6

<table>
<thead>
<tr>
<th>Goal</th>
<th>Visit the friend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiator</td>
<td>MyTamagotchi</td>
</tr>
<tr>
<td>Trigger</td>
<td>The owner has chosen to visit</td>
</tr>
</tbody>
</table>

### DESCRIPTION

<table>
<thead>
<tr>
<th>Condition</th>
<th>Step</th>
<th>Activity</th>
<th>Agent types and roles</th>
<th>Resources</th>
<th>Quality goals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Connect with another Tamagotchi</td>
<td>Person/Owner, Tamagotchi/MyTamagotchi, Tamagotchi/FriendTamagotchi</td>
<td></td>
<td>My Tamagotchi happy</td>
</tr>
<tr>
<td>Loop</td>
<td>2</td>
<td>Go to the friend’s place</td>
<td>Person/Owner, Tamagotchi/MyTamagotchi, Tamagotchi/FriendTamagotchi</td>
<td></td>
<td>My Tamagotchi happy</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Give the friend a present</td>
<td>Tamagotchi/MyTamagotchi, Tamagotchi/FriendTamagotchi</td>
<td>Present</td>
<td>My Tamagotchi happy, appropriate presenting</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Play with the friend</td>
<td>Tamagotchi/MyTamagotchi, Tamagotchi/FriendTamagotchi</td>
<td></td>
<td>My Tamagotchi happy</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Return home</td>
<td>Tamagotchi/MyTamagotchi, Tamagotchi/FriendTamagotchi</td>
<td></td>
<td>My Tamagotchi happy</td>
</tr>
</tbody>
</table>
Behaviour models

- A scenario focuses on what happens between agents
- A behaviour model addresses what an individual agent does
Two kinds of behaviour models

- Behavioural interface models
- Agent behaviour models
A behavioural interface model for a Tamagotchi

<table>
<thead>
<tr>
<th>Activity</th>
<th>Trigger</th>
<th>Pre-conditions</th>
<th>Post-conditions</th>
</tr>
</thead>
</table>
| Have meal| Meal selection by the owner | - The Tamagotchi is hungry  
- The selected meal is in the Tamagotchi Shell | - The Tamagotchi’s level of hunger has decreased  
- The Tamagotchi has consumed the meal from its Tamagotchi Shell |
| Go to the friend’s place | Request to visit by the owner | - The connection to the other Tamagotchi is operational  
- The Tamagotchi is in its Tamagotchi Shell | - The Tamagotchi is back in its Tamagotchi Shell  
- The Tamagotchi’s happiness level is increased  
- The Tamagotchi holds any items received as presents |
The abstract agent architecture revisited
The execution loop of an abstract agent

**while** the agent is unfulfilled **do**

sense the environment;
update the knowledge base;
reason;
choose actions;
act;
**end while**
An agent behaviour model of a Tamagotchi

Person

I am not happy

Please visit your friend

Tamagotchi

Tamagotchi is unhappy

Express unhappiness

R1

Connect with another Tamagotchi

There exists a connected friend

R2

R3

Visit the other Tamagotchi

R4

Request visit

R5

Go to the friend’s place

Give a present and play

Tamagotchi has enough presents

R6

Give the friend a present

Play with the friend

Tamagotchi is still unhappy

R7

Return home
Elaboration: Epistemic action

Tamagotchi

Give the friend a present

happinessLevel = happinessLevel + 40

R8
Behaviour model as a statechart

[Tamagotchi is unhappy]

Express unhappiness
- entry / send
- (I am not happy)

Please visit your friend

Connect with another Tamagotchi

[There exists a connected friend]

Visit the other Tamagotchi
Goal model for greeting

- Greeter
- Greetee
- Evaluator

- Notice
- Identify
- Formulate
- Articulate
- Evaluate

- Appropriate greeting
- Variety
- Right tone and phrase
- Accurate identification
- Timely noticing
Combined behaviour and interaction model for greeting
Greetings animation and video
Service model

- *Services* are modelled as reactive entities that provide functionality to the agents.
Agents, services, and objects

Legend

- **Agent**: 
- **Service**: 
- **Functional object**: 
- **Interaction**: 
- **Invocation**: 

Diagram:

Agents:
- A → B → C → D → E

Services:
- X → Y

Functional objects:
- 1 → 2 → 3 → 4 → 5
Services for Tamagotchis

<<Service>>
Toilet
flush()

<<Service>>
Shop
select (Food food)
Food buy()
select (Item item)
Item buy()

<<Service>>
StoreHouse
store(Food food)
Food get()
store(Item item)
Item get()

<<Service>>
Physician
cure()

<<Service>>
Mailbox
Letter getLetter()

tellOwner(String msg)
connect()
visit()
send(String msg)
givePresent(String msg, Present pres)
returnHome()

goToTamagotchiTown()
returnFromTamagotchiTown(Password pw)

<<Environment>>
TamagotchiShell

<<AgentType>>
Tamagotchi

Password
toTamagotchiTown()
returnFromTamagotchiTown(Password pw)

<<Service>>
Workplace
enter (Password pw)
Password exit()

<<Service>>
ShoppingMall
enter (Password pw)
Password exit()
select (Item item)
Password buy (Password pw)
Password exit()

<<Service>>
School
enter (Password pw)
Password exit()

<<Service>>
Pre-school
enter (Password pw)
Password exit()

<<Service>>
Theatre
enter (Password pw)
Password exit()

<<Service>>
GameCentre
enter (Password pw)
Password exit()

<<Service>>
King's Castle
enter (Password pw)
Password exit()

<<Service>>
TravelAgency
enter (Password pw)
Password exit()
<table>
<thead>
<tr>
<th>Activity</th>
<th>Service invocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Express unhappiness</td>
<td><code>Communicator.tellOwner(&quot;I am not happy&quot;)</code></td>
</tr>
<tr>
<td>Connect with another Tamagotchi</td>
<td><code>Communicator.connect()</code></td>
</tr>
<tr>
<td>Request visit</td>
<td><code>Communicator.send(&quot;Could I come to your place?&quot;)</code>, <code>Communicator.receive()</code></td>
</tr>
<tr>
<td>Go to the friend’s place</td>
<td><code>Communicator.visit()</code>, <code>Communicator.send(&quot;Good day!&quot;)</code>, <code>Communicator.receive()</code></td>
</tr>
<tr>
<td>Give the friend a present</td>
<td><code>Communicator.givePresent(&quot;Please accept my present&quot;, Present), Communicator.receive()</code></td>
</tr>
<tr>
<td>Play with the friend</td>
<td><code>Communicator.send(&quot;May I play with you?&quot;)</code>, <code>Communicator.receive()</code></td>
</tr>
<tr>
<td>Return home</td>
<td><code>Communicator.returnHome()</code></td>
</tr>
</tbody>
</table>
Today

- Workshop on creating design models (part 2)
Next time

On 31 March 2010:

- Lecture "From Agent-Oriented Modelling to Agent Programming"
- Workshop on using Jason and JADE agent platforms