Modeling for openness in MAS

Daniela Maria Uez
Jomi F. Hübner (advisor)

May 2017
1 Introduction

2 Our Propose

3 Example

4 Considerations
A MAS may be composed of four dimensions:
- Agents
- Environment
- Interaction
- Organization

For each MAS, dimensions may be arranged in a different way
- One or more dimension may not be relevant and are not developed
Open MAS (I)

- Commonly, an open system is the one in which agents can enter and leave the system freely at runtime.
- These new agents entering the system at runtime are unknown during the system development.
Multi-agent program contest is an open system

- The system is developed by a team in Germany
- The agents are developed by different teams around the world
- The team in Germany does not know how the agents are developed – for them, the agents are unknown during the system development
- These agents will enter into the system at runtime to take part in the contest
Open MAS (III)

- Other dimensions also may be open
  - Environmental artifacts, interaction protocols, and organizational norms also may be included at runtime
  - Elements of environment, interaction or organization dimension that is unknown during the development
Open MAS (IV)

- An open MAS is the one in which any dimension may be changed at runtime
- That means, elements of any dimension (agents, artifacts, protocols, norms, ...) unknown during development may be included at runtime
- These new elements have to be integrated into the system at runtime, working with the existent ones
Open dimensions

- MAPC Scenario (from Germany guys perspective)
  - Open at agent dimension (they are developed by others)
  - Environment is fully developed by the Germany team
  - Part of the organization is developed (some roles are defined by the contest).
    Some elements are included at runtime by the participant team organization (partially open)
  - Interaction is a non relevant dimension (nonexistent)
Designing openness

- Openness cannot be specified in a single component. It arises at runtime when new elements are included in the system.
What does it mean to design openness?
What does it mean to design openness?

- During the system design and development elements of some dimension are not known
  - They cannot be designed/developed
- But these elements will take part in the system at runtime
  - The system should be *designed* to properly “receive” these new elements
Methods (I)

Analyses

Design

Development
These methods are dependent on one single dimension. Other dimensions specification have to be translated at agent or environmental primitives. Commonly are developed in target platforms that only deal with agent or environment concepts. Methods do not offer guidelines to help the integration of new elements.
Methods (III)

- It does not prevent the design of an open MAS
  It constraints the specification of openness specific issues
- It is not easy to determine what is necessary at design time for a method properly deal with openness
1 Introduction

2 Our Propose

3 Example

4 Considerations
We propose a method for analysis, design, and development of open MAS

Each dimension is specified independently, with its own concepts and abstractions

Each of them has their own metamodel

Dimensions are connected by a set of *linking concepts*
Method Overview

Analyses

Design

Development

using JaCaMo Platform
Method Overview

Analyses

Design

Development using JaCaMo Platform
Full Metamodel

- State
- Transition
- Action
- Message
- Event
- Participant
- Protocol
- Wokspace
- Artifact
- Operation
- Event
- Message
- initiator
- generates
- count as
- Brute Fact
- Status Function
- Role
- Group
- Social Scheme
- Goal
- Norm
- Protocol
- Skill
- Perception
- Agent
- Plan
- Action
- Goal
- Belief
- Manual

Legend:
- Association
- Inheritance
- Composition
- Related Concept
- Border Association
- Runtime Association
Linking concepts (I)

- Related concepts: concepts that have similar meaning in different dimensions
Related concepts
Linking concepts (II)

- Border concepts: provide at design time specific information about how dimensions are linked at runtime
Border concepts
1 Introduction

2 Our Propose

3 Example

4 Considerations
Traveler Assistant(I)

- Traveler assistant provides information about tourist-related spots and services, indicating places to visit, eat or relax.
- Tourist and service providers (hotel, restaurants, tour operators) may negotiate prices and dates to provide services.
- Tourists may access relevant information about visiting places.
In the analysis phase, a detailed system description is defined.

This description is used as a starting point for the design phase.

In this example, the system is open at agent dimension, since each tourist or service provider may develop its own agent to take part in the system.
Example

Organization

- Assist the tourist
  - Visit tourist places
    - Choose the place to visit
      - Visit the next place on the route
    - Check suggestion list
    - Go to the place
      - Schedule a service
        - Choose a provider
          - Negotiate
            - Find a provider
              - Accept a service offered
            - Use the service
      - Enjoy local services
    - Provide the scheduled service
- Visit tourist places
- Check suggestion list
- Go to the place
- Schedule a service
- Choose a provider
- Negotiate
- Find a provider
- Accept a service offered
- Enjoy local services
- Provide the scheduled service

Tourist
- Skills
  - —

Service Provider
- Skills
  - Provide a service
Artifacts

**Map**
- visited place
- add place to visit
- delete place to visit
- itinerary next place
- next place

**Guide**
- name
- history
- available services
- related places
- tourist is in the place

**Schedule Book**
- add schedule
- delete schedule
- today
- current schedule
- next schedule
- next available date
- is date available
- schedule notification

**Manual**

**Map**
- **visited place** (operation): mark the place as visited
- **add place to visit** (operation): includes the place in the list of places to visit
- **delete place to visit** (operation): deletes the place from the list of places to visit
- **next place** (event): informs the next place in the visiting list
- **itinerary to next place** (event): informs the route to the next place in the list

**Guide**
- **name** (event): informs the place's name
- **history** (event): informs the place's history
- **available services** (event): informs the available services next to the place
- **related places** (event): informs the related places
- **tourist is in the place** (event): informs when the tourist is in the place

**Schedule Book**
- **add schedule** (operation): includes a new schedule
- **delete schedule** (operation): deletes a schedule
- **today** (event): informs the actual date
- **current schedule** (event): informs the current schedule
- **next schedule** (event): informs the next schedule
- **is date available** (event): informs if a date is available
- **schedule notification** (event): sends a notification for a near schedule
Brute Facts and Status Functions (I)

**Brute Fact**

- **Guide**
  - name
  - history
  - available services
  - related places
- **Map**
  - visited place
  - add place to visit
  - delete place to visit
  - itinerary next place
  - next place

**Schedule Book**

- today
- current schedule
- next schedule
- is date available
- schedule notification
- next available date

**Tourist Service Provider**

- assist the tourist
- visit the next place on the route
- check suggestion list
- visit tourist places
- go to the place
- choose the place to visit
- find a provider
- accept a service offered
- negotiate
- choose a provider
- enjoy local services
- use the service
- schedule a service
- provide the scheduled service

**Status Function**

- **agent** playing the role tourist
- goal **negotiate** achieved

- **count as**
  - agent is next to a visiting place
  - agent sends the message **schedule service in date**
Brute Facts and Status Functions (II)

Example

Modeling for openness in MAS

Tourist Service Provider Schedule Book

is date available?

X
date not available
date available

X

schedule service in date
X X
schedule service in date
X

price(price)
X

not accepted
accept

X

next available date

Schedule Book
add schedule
delete schedule
<<artifact>>
today
current schedule
next schedule

Schedule notification

next available date

Guide

<<artifact>>

history

available services
name
related places

Tourist is in the place

agent is next to a visiting place

count as

agent playing the role tourist

Service Provider

Tourist

Service Provider

Tourist

Guide

Schedule Book

Map
visited place
add place to visit
delete place

Schedule
add schedule
delete schedule

next available date
is date available
schedule notification

agent is playing the role tourist

Status Function
Brute Facts and Status Functions (III)

agent sends the message schedule service in date

schedule service in date

date not available

is date available?

date available

price(price)

not accepted

accepted

count as goal negotiate achieved
1 Introduction

2 Our Propose

3 Example

4 Considerations
We are developing a method to develop open MAS
Each dimension is defined by its own set of concepts
We also define a set of linking concepts that are used to link these dimensions
  - Related concepts
  - Border concepts
Conclusion (II)

- This is a work still in progress
- Some questions remain:
  - Minimal set of elements (roles + goals)
  - Roles and goals included at runtime
  - ...
That’s all, folks!

Thanks a lot!