Agents in Organisations: 
Autonomy, Regulation and Interaction

Frank Dignum
Huib Aldewereld
Virginia Dignum
M. Birna van Riemsdijk

July 11-12, 2011
EASSS 2011

Agenda

DAY 1
• Welcome
  – Tutors
  – Expectations
• Introduction
  – Background
  – Regulation vs autonomy
  – Existing models
• Agent-Centred organizations
  – Components of organization
  – OperA framework

DAY 2
• Organization-aware agents
  – Architectures
  – Support tools
• Applications
• Challenges
  – Reorganization
  – Implementation complexity

July 11-12, 2011
EASSS 2011
Welcome

• Huib Aldewereld
• Frank Dignum
• Virginia Dignum
• Birna van Riemsdijk

Expectations

• Your background
• Your experiences with organization models
• What you hope to get from this tutorial
INTRODUCTION

Motivation: commons

- Common goods: “The Tragedy of the Commons” (Hardin, 1968)
  - an important class of goods that conventional markets do not handle well
  - A resource is shared
  - None has an incentive to restrict their consumption
  - Yet over-consumption will exhaust the resource
  - Examples: water, pasture, fish, bandwidth
- A generic problem without a generic solution.
  - For a detailed set of case studies see “Governing the Commons” by Elinor Ostrom (Ostrom, 1990)
Motivation: governance

- (Good) governance: evaluating risks and monitoring compliance
- How can component actions be regulated without compromising their integrity or revealing information?
  - Contracts: service level agreements
  - Monitoring/Auditing framework
  - Roles, powers, permissions, authentication
- Software system evolution:
  - closed $\rightarrow$ semi-open $\rightarrow$ open
  - Organization structures are a non-invasive way to constrain software components in open architectures

Agents versus Organizations

- Agents $\rightarrow$ Autonomy
  - Agents are motivated by their own objectives, beliefs...
  - may take up organizational role if it serves their purposes
- Organization $\rightarrow$ Regulation
  - Organizations (too) have their own purpose
  - Exist independently of the agents populating it
Need for organization

• Do agents need organizations?
  – Do agents need to know/reason about the organization?

• Do MAS need organizations?
  – Interaction in MAS cannot be based in communication alone
  – MAS engineering require high-level agent independent abstractions
    ➔ Explicit social concepts defining the society where agents live

Relevance - 1

From the agents’ perspective, organizations:
• Insure a better integration of the agents in the system
• in order to better adapt themselves to change
• Delegation of tasks/beliefs between the agents
  – coalitions, teams, alliances...
  – That is (organizational) structures that need to be represented or exploited

• Despite or Thanks to
  – Multiple limitations : Cognitive, Physical, Temporal, Institutional,
  – Autonomy of the agents
  – The different organizations the agents take part in
Relevance - 2

From MAS perspective, organizations
- insure global behavior at the MAS level
  - In terms of cooperation, collaboration, ...
  - To be sure that the global goals of the system or collective instance are achieved
- represent observed patterns of interaction
- Despite or Thanks to
  - Multiple limitations: Cognitive, Physical, Temporal, Institutional,...
  - Autonomy of the agents
  - Descriptive or prescriptive view

Regulation versus Autonomy

- Regulated, or directed, behavior
  - Pre-determined behavior, external to agent:
    - Lack of agility
  - Do not consider differences in individual capabilities
  - Strict obedience to rules often does not get work done
- Autonomous behavior
  - Ability to make decisions about own activity
  - Individual rationality is insufficient to deal with social behavior (helpfulness, greater good, ...)
  - (Informal) structures are necessary for coordinating processes and stability
Regulation and Autonomy

• Can we have the best of both?

• Combination of individual rationality with laws of social interaction

Regulation with autonomy

• **Internal autonomy requirement:** Specify organization independently from the internal design of the agent
  – Enables open systems
  – Heterogeneous participation

• **Collaboration autonomy requirement:** Specify organizations without fixing a priori all structures, interactions and protocols
  – Enables evolving societies
  – Balances organizational needs and agent autonomy
Agent organization: Main features

- Make a clear distinction between description of organization and description of agents
- Agents are
  - dynamic, autonomous entities that evolve within organizations
- Organizations
  - Are regulative environments that constrain the behaviors of the agents
  - or: may appear as the result of agents’ activities

Requirements for agent organization models

- Reflect and support organizational needs/choices
  - Structure: roles, norms, interaction
  - Strategy: Global goals and requirements
  - Dynamics, open environments
- Satisfy internal autonomy requirement
  - Interaction structures are not completely fixed in advance
- Satisfy collaboration autonomy requirement
  - Explicit agreements concerning individual performance and interaction
Important

- individual agents will not work together just because they happen to be together
- organizational systems have global requirements and goals

- model must balance organizational aims and agent desires
- need to predict/verify overall behavior
  - describe expected behavior independently from agents: roles, scripts
  - describe enactment agreements: contracts

Models for organizations: top-down regulation

- specification
- implementation
Models for organizations: bottom-up autonomy

organization

Emerging / observed

individuals

Implementation

Models for organizations: balance autonomy and regulation

organization

landmarks

instantiation

Dynamic Specification

individuals
Components of organizations

- **Entities** (roles, positions, people, groups, components, ...)
- **Relationships** (networks, interactions, coordination ...)
- **Goals** (purpose, intention, shared, ...)
- **Norms** (culture, rules, ...)
- **Environment** (physical, social, open, dynamic, restrictive...)

Approaches - 1: e-institutions

- **Common Ontology and language**
- **Agents play roles in fixed protocols:**
  - governors
- **Norms:**
  - Constraints, cannot be violated
AMELI architecture

- **MEDIATION**
  - To facilitate agent communication within scenes.

- **COORDINATION AND ENFORCEMENT**
  - To guarantee the correct evolution of each scene.
  - To guarantee legal movements between scenes.
  - To control the obligations participating agents acquire and fulfil.

- **INFORMATION MANAGEMENT**
  - To facilitate the information agents need in the institution.
Approaches - 2: MOISE+

- **Main Components:**
  - Organisation Modelling Language
  - Organisation Management Infrastructure
  - Organisation-awareness agent mechanisms

- **Middleware for MAS:**
  - artifacts for detecting norm compliance and help coordination
Approaches - 2: MOISE+

- Declarative specification
  - Structural dimension
  - Functional dimension
  - Deontic dimension
- collecting and expressing: constraints, norms, cooperation patterns

MOISE

<table>
<thead>
<tr>
<th>role</th>
<th>deontic</th>
<th>mission</th>
<th>T</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>back</td>
<td>obliged</td>
<td>m1</td>
<td>1 day</td>
<td>get the ball, go to ... kick ...</td>
</tr>
<tr>
<td>left</td>
<td>obliged</td>
<td>m2</td>
<td>1 day</td>
<td>be placed at ... kick ...</td>
</tr>
<tr>
<td>right</td>
<td>obliged</td>
<td>m3</td>
<td>1 hour</td>
<td>kick to the goal, ...</td>
</tr>
<tr>
<td>attacker</td>
<td>obliged</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

norms (hard constraints)

structure

functioning

Key

July 11-12, 2011
EASSS 2011
Multi-Agent Organizations

- Organization exists independent of agents
- Roles used as positions
- System open for new agents
- Characteristics:
  - Result determined by combining global and individual goals
  - Social aspects determine balance between goals
  - Ontology needed
  - Explicit organizational knowledge

Organization is the leading abstraction
Organizational Modelling

OPERA

OPERA Model

• Organizational Model
  – represents organizational aims and requirements
  – roles, interaction structures, scene scripts, norms

• Social Model
  – represents agreements concerning participation of individual agents (‘job’ contracts for agents)
  – rea = role enacting agent

• Interaction Model
  – represents agreements concerning interaction between the agents themselves (‘trade’ contracts between reas)
Organizational Model

- **Social Structure**
  - roles, groups, dependencies

- **Interaction Structure**
  - scene scripts, connections, transitions

- **Normative Structure**
  - role, scene and transition norms

- **Communication Structure**
  - communicative acts, domain ontology

---

Social Model

- **Role negotiation scenes**
- Individual agents have own goals and requirements, that not necessarily meet role specification

- **Social Contract**
  - describes a specific agreement for a role enacting agent (rea)
  - Meets organizational expectations
  - Incorporates individual behavior
Interaction Model

- Script negotiation scenes
- Interaction contract
  - describes a specific performance of a scene
- Scene Instantiation
  - Reas are not exactly as the role descriptions
  - Describe the specific interaction protocol realizing landmarks and incorporating rea capabilities

**OPERA**

- Social commitment
- Interaction commitment
- Interaction pattern
- Interaction Model
Designing OPERA models

- Organizational goals imply different requirements concerning coordination
- Analysis and facilitation of social context
  - Organizational Model
  - Flexible structures: landmarks
- Analysis and facilitation of individual context
  - Social model: instantiation to individual requirements
  - Interaction model: protocols

OPERETTA and OPERA
OPERETTA: Components

- SS editor:
  - Social model specification (Roles, objectives, dependencies)
- IS editor:
  - Interaction model specification (Scenes, transitions, evolution)
- Graphical view and tree view
- Reorganization tracker
  - Building reorganization scripts
- Model validator
  - Checking model integrity, ensuring correct specifications
Example: Conference organization

- Objective:
  - Realize a scientific conference

- Organizational requirements
  - Separation of duties between program and local organization
  - Ensure scientific quality
  - Ensure large attendance
  - ...

- Agent requirements
  - Present own research
  - Receive information about new research development
  - Network
  - ...

Methodology (1)
Designing the Coordination

- Identify (functional) requirements
  - Identify global functionalities and objectives of society

- Identify stakeholders

- Create role tables
  - Important parties in achieving the organizational objective
  - Relations between parties

Social Structure
Scenario: Conference organization

- Modeling the organization of a conference
- Stakeholders
- Objectives
- Dependencies

<table>
<thead>
<tr>
<th>Role</th>
<th>Objectives</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Paper accepted</td>
<td>Program chair</td>
</tr>
<tr>
<td>Local organizer</td>
<td>Successful conference</td>
<td>Program chair, Participants...</td>
</tr>
<tr>
<td>Program chair</td>
<td>High quality conference</td>
<td>Authors, reviewers</td>
</tr>
<tr>
<td>Reviewers</td>
<td>Quality papers, independent eval</td>
<td>Program chair</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Roles

- Roles indicate important parties
- Sub-objectives are identified by means-end analysis
- Role dependencies identify interaction between roles, leads to scenes

<table>
<thead>
<tr>
<th>ROLE DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role id</td>
</tr>
<tr>
<td>Objectives</td>
</tr>
<tr>
<td>Sub-objectives</td>
</tr>
<tr>
<td>Rights</td>
</tr>
<tr>
<td>Norms</td>
</tr>
<tr>
<td>Type</td>
</tr>
</tbody>
</table>
Creating the Social Structure

Role example
Methodology (2)
Designing the environment

• Identify organizational norms
  – Responsibility analysis
  – Resource analysis
  – Trigger analysis
  – Norm specification
  – Sanction specification

• Communication:
  – Model Ontology
  – Concrete domain ontology
  – Generic communication acts

Normative & Communicative Structures

Norms

• Statutes: Abstract norms
  – Main objective of the organization,
  – Values that direct the fulfilling of this objective
  – Context
  – E.g. Fairness of exchange

• Concrete norms
  – Protocols and Rules: enable agents to comply with organizational norms
  – Constraints: cannot be violated
  – Regulations: agent can decide

• Counts-As connects Abstract-Concrete
Communication Dimension

• Abstract Level
  – Generic Terms
    • Incontextual concepts
  – Model Ontology
    • concepts of the framework itself
    • E.g. norm, rule, role, group, violation, landmark...

• Concrete Level
  – Concrete domain ontology
  – Generic communication acts

Scenario: ontology and norms

• Ontology for the conference organization
  – Author, paper, session, review, PC-member, website, deadline, notification, proceedings, ...

• Norms in the conference organization
  – Role: PC-member cannot review own papers
  – Interaction: Session chair can stop presentation if too long
  – Global: English language must be used for all communication
Creating the Environment

– Communicative structure build *automatically*;
– Or via importing existing domain ontology

Methodology (3)

Defining interaction

• Scene scripts
• Structure interaction
  – Partial ordering of scenes
  – Relationships between scenes
    • Causal dependency: Sequence of scenes
    • Synchronization: AND relation between scenes
    • Parallelism: OR relation between scenes
    • Instantiation: new scene instances

Interaction Structure
Scene script

<table>
<thead>
<tr>
<th>SCENE DEFINITION</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene identifier</td>
<td>From role dependency</td>
</tr>
<tr>
<td>Roles</td>
<td>Participants in scene</td>
</tr>
<tr>
<td>Description</td>
<td>Textual description</td>
</tr>
<tr>
<td>Results</td>
<td>Objectives of scene -&gt; relate to role dependency</td>
</tr>
<tr>
<td>Patterns</td>
<td>Partial ordering of landmarks to achieve result, for each scene result</td>
</tr>
<tr>
<td>Norms</td>
<td>From norm analysis</td>
</tr>
<tr>
<td>Rationale</td>
<td>Further information</td>
</tr>
</tbody>
</table>

Creating the Interaction Structure
Methodology (4)
Validation

• Validation checks:
  1. Meta-model constraints
     (is the model an instance of the meta-model?)
  2. OperA specific constraints
     (is the model a correct OperA model?)

• Validation intended as design-assistance
Model Tracker

- Design support for reorganization
- Tracks changes to model
- Formal change operators
- Can set milestones to mark important design states
- Exports to change scripts

Advantages

- Full separation between organizational and agent design
- Full MDE
  - Support for integration with other models
- Intuitive concepts and interfaces
  - Enables rapid, collaborative prototyping, with stakeholders/domain experts
- Flexibility
  - Landmarks as alternative to protocol specification
- Integrity
  - Formal language based on deontic, temporal logic
Conclusions

• Organizations are first-class entities in the development of agent systems
• Balance organizational and individual requirements
• Agent autonomy: role performance

• OPERA: organization design framework following these principles
• OPERETTA: modelling environment for OPERA

• More information / download:
  – http://www.operetta-tool.nl/