Roadmap – Where we go next!

- Background
- Intra-Agent Modeling
- Inter-Agent Modeling
- Supra-Agent Modeling
  - Group and Role Concepts
    - Introduction
    - Hierarchical Groups
    - Aalaadin Approach
    - ZEUS Role Model
    - GAIA Role Schema
    - Roles as Sets of Policies
Groups (Teams):

- Set of agents who purposefully interact for some period of time
- Sociological perspective:
  - A (small) set of humans being involved in social relationships and pursuing common goals for some period of time
Introduction

Supra-Agent Modeling • Group and Role Concepts

Role:

- abstract representation of an agent’s (or organization’s) function
- multiple agents may own the same role, an agent may own multiple roles
- examples: personal assistant, mediator, information broker, ...
- sociological perspective, I (interactionist view)
  - behavioral pattern applied in specific situations
  - analogy: role playing, as actors in movies
  - central: typification of situations + identification of situations
Role (Cont’d):

- sociological perspective, II (systemtheoretical view)
  - set of expectations held by certain persons or groups of persons with regard to the behavior and the qualities of the occupant of some (social) position
  - position = designated location in a social system (described in terms of relationships to other persons)
  - expectations may have normative power, expressed in terms of rights and duties (and sanctions) associated with positions and roles
Definition:

- “group = group leader AND group members”
  “group member = agent OR group”

Example:
Core model:

- **Agent**
  - contains Group
  - handles Role

Group and Role Concepts

- **Role characteristics:**
  - **uniqueness** (role can be unique or multiple within a group)
  - **competences** (conditions an agent must fulfill to be able to play a role)
  - **capacities** (properties an agent owns when playing a particular role)
Aalaadin Approach (Ferber et al. 1997)

Supra-Agent Modeling • Group and Role Concepts

- Group structure: defined by
  - set of available roles
  - description of valid role-role interactions
  - a language supporting role-role interactions

- Organizational structure: defined by
  - set of available group structures
Complete model:

- **Organizational Structure**
  - **Organization**
    - contains
  - **Group Structure**
    - instantiates as
    - **Group**
      - is member
      - contains
      - **Agent**
        - instantiates as
        - **Role**
          - handles
          - **Interaction**
            - is defined between
            - contains
  - **Agent Class**
    - instantiates as

Core Concepts
ZEUS role model consists of

- **role diagram**: diagram overviewing roles and their relationships (interaction, containment, inheritance)
- **collaboration diagram**: description of interaction sequences
- **role description** (for each role):
  - role name
  - relationships to other roles
  - responsibilities and collaborators
  - external interfaces (services provided)
  - prerequisites (e.g., necessary protocols)
role = abstract description of an agent’s expected function

components of the GAIA role schema:

- role name, description in natural language
- protocols and activities in which the role plays a part
- **permissions**: type and amount of resources that can be exploited by a role owner (“what can and can’t be spent”)
- **responsibilities**:
  - liveness responsibilities: activities that aim at achieving desirable state of affairs (e.g., “maximize profit”, “always respond when being asked”)
  - safety responsibilities: activities that aim at avoiding undesirable state of affairs (e.g., “never spent more money than allocated”)
Roles as Sets of Policies (Lupu 1998)

- role = set of policies relating to the same subject (called manager position domain)

- types of policies:
  - **authorization policies** specify which actions subjects are permitted or prohibited to invoke on target objects
    - positive authorizations, e.g. “students are allowed to access course databases”
    - negative authorizations, e.g. “students are not allowed to access system files”
types of policies (Cont’d):

- **obligation policies** specify which actions subjects must or must not perform.
  - **positive obligations**, e.g. “director must protect interests of the company”
  - **negative obligations**, e.g. “students must not delete (= is obliged not to delete) system files”

Language for policy specification. Illustration:

```plaintext
O-- n:@/test-engineers { DiscloseTestResults() } @/analysts + @/developers
  when n.testing_sequence == in-progress
```

⇒ Negative obligation policy, saying that test engineers must not disclose test results to analysts or developers when the testing sequence is still in progress.
Roadmap – Where we go next!

- Background
- Intra-Agent Modeling
- Inter-Agent Modeling
- Supra-Agent Modeling
  - Group and Role Concepts
  - Norms
    - Introduction
    - Deontic Logic
norms = commonly shared expectation about appropriate and inappropriate behavior

can be understood as rules or guidelines that regulate interactive behavior

degree of regulation achieved through norms depends on their strengths and persistence

strength usually correlated/associated with sanction

- positive sanctions (reward)
- negative sanctions (punishment)

Examples: permissions, responsibilities, obligations, and prohibitions

KEY ISSUE: norms versus autonomy (norm compliance)
Deontic Logic

Operators: \( \mathbf{O} \) (obliged), \( \mathbf{P} \) (permitted), \( \mathbf{F} \) (forbidden)

as usual: \( p \supset q \) means “\( \neg p \) or \( q \)”,
\( p \equiv q \) means “\( (p \supset q) \) and \( (q \supset p) \)”

Axioms of the “standard system of deontic logic”:

- KD0: all (or enough) tautologies of propositional calculus
- KD1: \( \mathbf{O}(p \supset q) \supset (\mathbf{O}p \supset \mathbf{O}q) \)
- KD2: \( \mathbf{O}p \supset \mathbf{P}p \)
- KD3: \( \mathbf{P}p \equiv \neg \mathbf{O}\neg p \)
- KD4: \( \mathbf{F}p \equiv \neg \mathbf{P}p \)
- KD5 (modus ponens): \( p, p \supset q \vdash q \)
- KD6 (O-necessitation): \( p \vdash \mathbf{O}p \)
Roadmap – Where we go next!

- Background
- Intra-Agent Modeling
- Inter-Agent Modeling
- Supra-Agent Modeling
  - Group and Role Concepts
  - Norms
  - Organizational Patterns
    - Introduction to Patterns
    - Micro-level Patterns
    - Macro-level Patterns
    - Comparing Patterns
    - In-Class Exercises
Pattern = reusable (core of a) solution to a recurring problem

Pattern = relation between a certain context, a problem, and a solution (Buschmann et al. 1997)

Context:
- situations in which the problem occurs

Problem:
- problem specification ("what needs to be solved?")
- various viewpoints

Solution:
- static aspects (spatial configuration)
- dynamic aspects (run-time behavior)
Advantages of patterns (Buschmann et al. 1997):

- document experience
- identify and specify abstractions
- provide common vocabulary and understanding
- support construction process
- help to build complex and heterogeneous software
- help to manage software complexity

Here: distinction between micro vs macro-level patterns according to number of involved actors and generality. No sharp borderline in general!
Organizational primitives (Malone 1990):

- **wechselseitige Adjustierung**
  - A ← B ← A

- **direkte Beaufsichtigung**
  - A → B → C → A

- **Standardisierung**
  - B ← C ← B
Organizational primitives (Cont’d):

- **mutual adjustment**
  - no agent does have dominant control
  - collaboration
  - communication-intensive, but flexible

- **direct supervision**
  - A controls B and C
  - A “mediates” B and C
  - less communication, reduced flexibility

- **standardization**
  - A “defines” behavioral guidelines/rules
  - minimal communication and low flexibility
Notational preliminaries (see Broker, Matchmaker, etc. on the following slides)

- based on i* (Yu 1994)
- roles/positions represented as circles
- $A \rightarrow B$ means “A depends on B”
- four types of dependencies:
  - goal dependency (ovals)
  - task dependency (hexagons)
  - resource dependency (rectangles)
  - softgoals (clouds)
Broker (Kolp et al. 2001):
Matchmaker (Kolp et al. 2001):
Micro-level Patterns

Monitor (Kolp et al. 2001):

- Subscriber
- Forward Subscribed Change
- Notify Change
- Subject
- Monitor
Mediator (Kolp et al. 2001):
Embassy (Kolp et al. 2001):
Further (families of) patterns:

- Group/Team patterns
- Wrapper patterns
- Contract-net patterns
- Traveling patterns
- Meeting patterns
- Mobility patterns
- ...

see also UML (collaboration diagram, group, etc.)
Macro-level Patterns

Structure-in-5 (Kolp et al. 2001):
Joint Venture (Kolp et al. 2001):
Macro-level Patterns

Bidding (Kolp et al. 2001):

- Issuer
- Best Possible Bid
- Run Auction
- Auctioneer
- Service Auctionned
- Start Bid at the lowest price
- Bid Higher
- No Higher Bid
- Bidder_1
- Bidder_2
- Bidder_n
Hierarchical Contracting (Kolp et al. 2001):
Macro-level Patterns

Co-optation (Kolp et al. 2001):

- **Foreign Services**
  - **Contractor_1**
    - **Knowledge Sharing**
  - **Support**
    - **Cooptated_2**
    - **Cooptated_n**
  - **Provides Assets**
    - **Cooptated_1**
    - **External Resource**

- **Contractor_n**
Macro-level Patterns

Pyramid (Kolp et al. 2001):

- Apex
  - Delegate Responsibilities
    - Manager
      - Route Delegation
        - Operator_1
      - Resolve Conflicts
        - Operator_2
  - Strategic Authority
    - Supervisor
      - Coordinate
        - Operator_3
      - Monitor
        - Operator_4
Macro-level Patterns

Arm’s-Length (Kolp et al. 2001):
Macro-level Patterns

Hierarchies (Malone 1987):

PRODUKT-HIERARCHIE

FUNKTIONALE HIERARCHIE

Produktmanager  Funktionsmanager

Ausführende von Aufgaben unterschiedlichen Typs
Macro-level Patterns

Markets (Malone 1987):

- Dezentraler Markt
- Zentralistischer Markt

- Produktmanager
- Funktionsmanager
- Ausführende von Aufgaben unterschiedlichen Typs
Networks:

- “neither market nor hierarchy”
- e.g., strategic alliances, in/outsourcing, etc.
- **Strategic view**: Network = “long-term purposeful arrangement among distinct but related for-profit organizations that allow those firms in them to gain or sustain competitive advantage” (Jarillo 1988)
- **Behavioral view**: Network = “pattern of social relations over a set of persons, positions, groups, or organizations” (Sailer 78)
- **Prevalent DAI view**: interaction structure and pattern which is mainly a result of contracting
Macr-o-level Patterns

Markets vs Hierarchies vs Networks (e.g. Powell 1990):

<table>
<thead>
<tr>
<th>feature</th>
<th>markets</th>
<th>hierarchies</th>
<th>networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>coordination</td>
<td>prices</td>
<td>routines</td>
<td>relational</td>
</tr>
<tr>
<td>relation</td>
<td>competition</td>
<td>authority</td>
<td>mutual interest</td>
</tr>
<tr>
<td>conflict resolution</td>
<td>haggling</td>
<td>supervision</td>
<td>reputation</td>
</tr>
<tr>
<td>actor preferences</td>
<td>independent</td>
<td>dependent</td>
<td>interdependent</td>
</tr>
<tr>
<td>commitment level</td>
<td>low</td>
<td>medium to high</td>
<td>medium to high</td>
</tr>
<tr>
<td>flexibility</td>
<td>high</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>goal</td>
<td>individual</td>
<td>global</td>
<td>both are possible</td>
</tr>
</tbody>
</table>
Further (families of) macro-level patterns:

- matrix patterns
- patterns for functional/divisional structuring
- patterns for geographical structuring
- ...

⇒ organization theory!
Requirements relevant for comparison (Kolp et al. 2001):

- predictability
- security
- adaptability
- cooperativity
- competitiviry
- availability
- failability-tolerance
- modularity
- aggregability
Comparing Patterns

Requirements relevant for comparing patterns (Cont’d):

<table>
<thead>
<tr>
<th></th>
<th>s-i-5</th>
<th>pyr</th>
<th>jo-ve</th>
<th>bidd</th>
<th>ar-le</th>
<th>hi-co</th>
<th>co-op</th>
</tr>
</thead>
<tbody>
<tr>
<td>predictability</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>security</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>adaptability</td>
<td></td>
<td></td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>cooperativity</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>competitiveness</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>availability</td>
<td>+</td>
<td>+</td>
<td>++</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>failability-tolerance</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>++</td>
<td></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>modularity</td>
<td>++</td>
<td>-</td>
<td>+</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>aggregability</td>
<td>++</td>
<td>++</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>
Comparing Patterns

Further criteria for deciding on appropriateness of patterns:
- problem size
- environmental dynamics
- ability of individual agents

Choice influences (according to Malone) ...
- ... production costs
- ... communication costs
- ... costs for adaptation
In-class Exercises

Use illustrations considered in first part of this course and ...

► ... identify application-specific groups and roles!
  - For each group and role: what is its purpose and value?
  - What abilities should an agent have to be a “good” group member or role owner?
  - Specify the groups and roles you identified, using ZEUS and Gaia.

► ... identify organizational micro/macro patterns and compare them!