Roadmap – Course Overview

- Introduction
- Cooperation Models and Mechanisms
- Communication
- Micro-Level Architectures

- Macro-Level Structures
- Development Methods
**Macro-Level Structures**

- Computational Roles and Groups
  - What is a Role?
  - What is a Group?
  - Hierarchical Groups
  - Aalaadin Approach
  - ZEUS Role Model
  - GAIA Role Model
  - Roles as Sets of Policies

- Organizational Patterns
What is a Role?

General characterization

- a role is an 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
What is a 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
A group (team) is a set of agents who purposefully interact for some period of time.

Sociological perspective:

A group is a (small) set of humans being involved in social relationships and pursuing common goals for some period of time.
Hierarchical Groups (Weiß 1994)

Definition:

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

Example:
Aalaadin Approach (Ferber et al. 1997)

Macro-Level Structures • Group and Role Concepts

Core model:

- **Agent**
  - is member of **Group**
  - handles **Role**

- **Group**
  - contains **Role**

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 (Cont’d)

Macro-Level Structures ● Computational Roles and Groups

- 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
Aalaadin Approach (Cont’d)

Complete model:

- **Organizational Structure**
  - holds
  - instantiates as
  - defines

- **Organization**
  - contains

- **Group Structure**
  - instantiates as

- **Group**
  - contains
  - is member

- **Role**
  - handles

- **Interaction**
  - is defined between

- **Agent Class**
  - instantiates as

- **Agent**

Core Concepts
ZEUS Role Model (Collis et al. 1999)

- **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)
GAIA Role Schema (Wooldridge et al. 2000)

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:

```
  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.
Macro-Level Structures

- Computational Roles and Groups
- Organizational Patterns
  - What is a Pattern?
  - Short-range Patterns
  - Medium-range Patterns
  - Long-range Patterns
  - Comparing Patterns
What is a Pattern?

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

In this course: informal (and fuzzy) distinction between short/medium/long-range patterns, according to the number of involved agents. Roughly:

*group size / company size / multiple-companies size*
Organizational primitives (Malone 1990):

- Wechselseitige Adjustierung
- Direkte Beaufsichtigung
- Standardisierung
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):
Monitor (Kolp et al. 2001):
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.)
Structure-in-5 (Kolp et al. 2001):
Joint Venture (Kolp et al. 2001):
Bidding (Kolp et al. 2001):

- **Issuer**
  - Service Auctionned
  - Best Possible Bid
    - Run Auction

- **Auctioneer**
  - Start Bid at the lowest price
    - Bidder_1
  - Bid Higher
    - Bidder_2
  - No Higher Bid
    - Bidder_n
Hierarchical Contracting (Kolp et al. 2001):
Co-optation (Kolp et al. 2001):
Pyramid (Kolp et al. 2001):

- Apex
- Manager
  - Delegate Responsibilities
    - Operator_1
  - Resolve Conflicts
    - Operator_2
- Supervisor
  - Strategic Authority
    - Operator_3
  - Coordinate
    - Operator_4

- Monitor
Arm’s-Length (Kolp et al. 2001):
Long-range Patterns

Hierarchies (Malone 1987):

- Produktmanager
- Funktionsmanager
- Ausführende von Aufgaben unterschiedlichen Typs
Markets (Malone 1987):

DEZENTRALER MARKT

ZENTRALISTISCHER MARKT

 Produktdirector
 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
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!
Comparing Patterns

Macro-Level Structures • Short-range Patterns

Criteria relevant for comparison (Kolp et al. 2001):

- predictability
- security
- adaptability
- cooperativity
- competitiveness
- availability
- failability-tolerance
- modularity
- aggregability
Comparing Patterns (Cont’d)

Criteria 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 (Cont’d)

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