Organizational Styles and Agent Patterns

Organizational System Architecture

- Architectural Design
- 10 Architectural Styles
- Software Quality Attributes
- Correlation Catalogue
- Examples
- Agent Patterns
Architectural Design for MAS

- Multi-agent systems: organizations of coordinated autonomous agents that interact to achieve particular, possible common goals.
- The developer thinks in terms of single agents and makes them interact one another to produce the desired behavior.
- Architectural choices made on his experience.
- Not referring to a particular organization.
- Very critical when designing huge MAS dealing with many different kinds of agents.

Architectural Styles

- Styles to guide high-level system design and discussed how they drive the composition of a system
- Pipe-filter, Main-Program & Subroutines, layered architecture,…
- Does not focus on organizational architectures.
**Organizations and MAS**

- Organizations to coordinate the actions of many individuals for some purpose.
- To develop and manage structures as business units, profitable enterprises, multi-national alliances, ...
- Not constructed with a population of identical individuals doing the same thing;
- Diversify, delegate, negotiate, manage, cooperate, compete, ...
- Agent Systems could benefit from the same organizational models and architectural designs understood in terms of organizational concepts.

**Flat Structure**

- No fixed structure and no control of one actor over another is assumed.
- Main advantage: autonomy, distribution and continuous evolution of an actor architecture.
- Key drawback: an increased amount of reasoning and communication by each participating actor.
Structure in 5

- Typical strategic and logistic components found in organizations.
- Operational core: the basic tasks and operations — the input, processing, output and direct support procedures associated with running the system.

- At the top: the apex composed of strategic executive actors.
- The support component assists the operation core for non-operational services that is outside the basic flow of operational tasks and procedures.
- The coordination component carries out the tasks of standardizing the behavior of other components, in addition to applying analytical procedures to help the system adapt to its environment.
- Actors who join the strategic apex to the operational core make up the middle agency (intermediate, relay).
**Pyramid**

- Actors at the lower levels depend on actors of the higher levels.
- Supervision from the apex.
- Managers and supervisors are only intermediate actors routing strategic decisions and authority
  - They can coordinate behaviors or take decisions only at a local level.

---

**Pyramid**

- Applied when deploying simple systems.
- Encourages dynamicity: coordination and decision are direct, not complex and immediately identifiable.
- To manage and resolve crisis situations.
  - A complex agent system faced with an intrusion from non-trustable agents could dynamically migrate itself into a pyramid organization to resolve the security problem in a more efficient way.
- Evolvability and modifiability can be implemented at low costs.
- The computation can appropriately be defined via a hierarchy of procedure definitions: related to the classical *main program and subroutines* architectural style.
**Joint Venture**

- Agreement between two or more principal partners to obtain the benefits of larger scale, partial investment and lower maintenance costs.
- Delegation of authority to a specific joint management actor that coordinates tasks and operations and manages sharing of knowledge and resources.
- Pursue joint objectives and common purpose.

---

**Joint Venture**

- Each principal partner can manage and control itself on a local dimension and interact directly with other principal partners
  - Exchange, provide and receive services, data and knowledge.
- The strategic operation and coordination of such a system and system partner actors on a global dimension are only ensured by the joint management actor.
- Outside the joint venture, secondary partners supply services or supports tasks for the organization core.
**Arm's-Length**

Agreements between independent and competitive but partner actors. Partners keep their autonomy and independence but act and put their resources and knowledge together to accomplish common goals.

No authority delegated/lost from a collaborator to another. For applications that involve a collection of independent computations whose execution proceed competitively. Can be considered a derivation of the classical communicating processes architectural style.

**Bidding**

- Competitiveness mechanisms and actors needed to run an auction.
- The auctioneer actor runs the show.
- The auction issuer issues the bidding.
- Implies fast response time and adjustability for the system.
**Takeover**

- Total delegation of authority and management from two or more partners to a single collective *takeover* actor.

- Similar to the joint venture style.
- **Difference:** In a joint venture identities and autonomies of the separate units are preserved, the takeover absorbs these critical units, no direct relationships tolerated except those involving the takeover.

---

**Hierarchical Contracting**

- Coordinating mechanisms that combine arm's-length features with aspects associated with pyramidal authority.
- Coordination mechanisms developed to manage arm's-length (independent) characteristics
- Involve a variety of negotiators, mediators and observers at different levels
  - handling conditional clauses to monitor and manage possible contingencies,
  - negotiate and resolve conflicts and
  - finally deliberate and take decisions.
- Hierarchical relationships restrict autonomy and underlie a cooperative venture between the contracting parties.
Hierarchical Contracting

- Dual and admittedly complex contracting arrangements
- To manage conditions of complexity and uncertainty deployed in high-cost-high-gain (high-risk) applications.
- Suitable for applications that involve distinct classes of layered services that can be arranged hierarchically
- Can be considered a specialization of the classical layered architectural style.

Vertical Integration

- Merges system actors engaged in related tasks at different stages of a production process.
- A merger synchronizes and controls interactions between each of the participants that can be considered as workshops.
- Suitable for applications that require a defined series of independent computations to be performed on ordered data
- Can be viewed as a specialization of the classical pipe and filter architectural style.
**Co-optation**

- Involves the incorporation of representatives of external systems into the decision-making structure and behavior of an organization.
- Organizations trading confidentiality and authority for resource, knowledge assets and support.
- Each co-optated actor has to adjust his views with the policy of the system.

**Software Quality Attributes for MAS**

- **Predictability**
  - System Actors have a high degree of autonomy in the way that they undertake action and communication in their domains.
  - Difficult to predict individual characteristics as part of determining the behavior of a complex system.

- **Security**
  - Protocols and strategies for verifying authenticity for these data sources by individual software entities are an important concern in the evaluation of overall system quality since, in addition to possibly misleading information acquired by system actors, there is the danger of hostile external actors spoofing the system to acquire information accorded to trusted domain actors.
Software Quality Attributes for MAS

● Adaptable
  • System actors may be required to adapt to modifications in their environment.
  • Dynamic introduction of a new agents unknown or the manipulations of existing agents.

● Coordinability
  • To coordinate with other actors. Two different antagonist ways:
  • Cooperativity. Actors must be able to coordinate with other actors to achieve a common purpose.
  • Competitively. Actors must be able to coordinate with other actors except that the success of one agent implies the failure of others.

Software Quality Attributes for MAS

● Availability.
  • Guard against the interruption of offered services.
  • Must actually be considered a sub-attribute of security

● Failability-Tolerance.
  • Failure of one actor does not imply a failure of the whole system.
  • To prevent failure: similar or replicated capabilities and refer to more than one agent for a specific behavior.
  • Induces redundancy in the system.
Software Quality Attributes for MAS

- **Modularity**
  - Increases efficiency of task execution, reduces communication overhead and usually enables high flexibility.
  - Implies constraints on inter-module communication.

- **Aggregability**
  - Components of other actors. They surrender to the control of the composite actor.
  - Efficient tasks execution and low communication overhead,
  - Prevents the system to benefit from flexibility.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat Structure</td>
<td>BREAK</td>
<td>BREAK</td>
<td>MAKE</td>
<td>HELP</td>
<td>HELP</td>
<td>MAKE</td>
<td>HURT</td>
<td>HELP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structure-in-5</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HURT</td>
<td>HELP</td>
<td>MAKE</td>
<td>MAKE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pyramid</td>
<td>MAKE</td>
<td>MAKE</td>
<td>HELP</td>
<td>MAKE</td>
<td>BREAK</td>
<td>HELP</td>
<td>HURT</td>
<td>HELP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joint Venture</td>
<td>HELP</td>
<td>HELP</td>
<td>MAKE</td>
<td>HELP</td>
<td>HURT</td>
<td>MAKE</td>
<td>MAKE</td>
<td>HELP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidding</td>
<td>BREAK</td>
<td>BREAK</td>
<td>MAKE</td>
<td>HURT</td>
<td>MAKE</td>
<td>BREAK</td>
<td>MAKE</td>
<td>MAKE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Takeover</td>
<td>MAKE</td>
<td>MAKE</td>
<td>HURT</td>
<td>MAKE</td>
<td>BREAK</td>
<td>HELP</td>
<td>HURT</td>
<td>HELP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arm’s-Length</td>
<td>HURT</td>
<td>BREAK</td>
<td>HELP</td>
<td>HURT</td>
<td>MAKE</td>
<td>BREAK</td>
<td>MAKE</td>
<td>HELP</td>
<td>HURT</td>
<td></td>
</tr>
<tr>
<td>Hierarch. Cont.</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td>HELP</td>
<td></td>
</tr>
<tr>
<td>Vert. Integ.</td>
<td>HELP</td>
<td>HELP</td>
<td>HURT</td>
<td>HELP</td>
<td>HURT</td>
<td>BREAK</td>
<td>BREAK</td>
<td>BREAK</td>
<td>BREAK</td>
<td></td>
</tr>
<tr>
<td>Co-optiation</td>
<td>HURT</td>
<td>HURT</td>
<td>MAKE</td>
<td>MAKE</td>
<td>HELP</td>
<td>HURT</td>
<td>HELP</td>
<td>HELP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

HELP, MAKE, HURT, BREAK, respectively model partial/positive, sufficient/positive, partial/negative and sufficient/negative contributions.
**Architectural Design**

- Global architecture in terms of interconnected subsystems.

- 3 Steps
  - **1 Macrolevel**: Organizational Styles (Organization Theory)
    - Vertical Integration, Pyramid, Joint Venture, Structure in 5, Bidding, Hierarchical Contracting, Co-optation, Takeover
  - **2 Micro level**: Patterns (Agent Community)
    - Broker, Matchmaker, Contract-Net, Mediator, Monitor, Embassy, Wrapper, Master-Slave, ...
  - **3 Assigning Actors to Agents**, Positions, Roles
**Patterns for Agents: Monitor**

- One monitor, a number of subscribers and at least one subject of interest.
- Accepts subscriptions, request notifications from subjects of interest, receive such notifications of events and alerts subscribers to relevant events.
- The subject provides notifications of state changes as requested.
- The subscriber registers for notification of state changes to distributed subjects, receive notifications with current state information, and update its local state information.
- Used in the horizontal contracting, vertical integration, arm’s-length and bidding styles implying observation requirements.

**Monitor and Broker**

[Diagram of Monitor and Broker interactions]
**Broker**

- One broker, a number of service consumers and providers.
- The broker is an arbiter and intermediary accessing services of one actor to satisfy the request of another.
- Consumers may also be in turn service providers, and inversely providers can also be consumers.
- Roles of each actors are established in the context of a particular dialogue.
- Used in the horizontal integration and joint venture styles.

**Matchmaker**

- Locates a provider corresponding to a consumer request for service, and then hands the consumer a handle to the chosen provider directly.
- The broker directly handles all interactions between the consumer and the provider.
- Here the negotiation for service and actual service provision are separated into two distinct phases.
- Used in the horizontal integration and joint venture styles.
**Matchmaker and Mediator**

- A mediator, and any number of colleague actors who play either the role of initiator or performer.
- A colleague (initiator) addresses the mediator in place of asking directly another colleague (performer).
- He has acquaintance models of colleagues and coordinates the cooperation between them.
- Each colleague has an acquaintance model of the mediator.
- A broker simply matches providers with consumers,
- A mediator encapsulates interactions and maintains models of initiators and performers behaviors over time.
- Used in pyramids, takeovers, vertical integrations and horizontal contracting: underlies direct cooperation and encapsulation features reinforcing authority mechanisms.

---

**Mediator**

- A mediator, and any number of colleague actors who play either the role of initiator or performer.
- A colleague (initiator) addresses the mediator in place of asking directly another colleague (performer).
- He has acquaintance models of colleagues and coordinates the cooperation between them.
- Each colleague has an acquaintance model of the mediator.
- A broker simply matches providers with consumers,
- A mediator encapsulates interactions and maintains models of initiators and performers behaviors over time.
- Used in pyramids, takeovers, vertical integrations and horizontal contracting: underlies direct cooperation and encapsulation features reinforcing authority mechanisms.
**Embassy**

- Involves a foreign actor, a single embassy actor and any number of local actors.
- The foreign actor requests access to a local actor from its embassy actor.
- Depending on the level of certificate provided, access may be granted or denied.
- When the access is granted, the foreign actor can submit messages to the embassy for translation.
- The content is translated in accordance with a standard ontology.
- Translated messages are forwarded to target local actors.
- The results of the query are passed back out to the foreign actors, translated in reverse.

**Embassy**

- For structure-in-5, arm’s-length, bidding and co-optation
- Take in charge security aspects between components related to the competitiveness mechanisms inherent to these styles.
**Wrapper**

- A wrapper, a number of clients and only one legacy system since the wrapper is domain-specific.
- Allows a legacy application to be coupled with a MAS.
- Interfaces the clients to the legacy by acting as a translator.
- Ensures that communication protocols are respected and the legacy system remains decoupled from the clients.
- Used in the co-optation style when one of the co-optated actor is a representative for a legacy system.

---

**Wrapper and Contract-Net**
**Contract-Net**

- Involves a manager and any number of participants.
- The manager issues a request for proposal for a particular service to all participants and then accepts "proposals" to meet the service request at a particular "cost".
- The manager selects among these proposals and indicates acceptance to exactly one participant.
- The selected participant performs the contracted work and informs the manager upon completion.
- The contract-net pattern is used in the bidding and arm’s-length style since they are based on competitive features.
- This pattern is especially used in the arm’s-length and bidding and co-optation styles due to their inherent competitive features.