Flexible hierarchical organisation of role based agents

Emmanuel Adam  Emmanuelle Grislin-Le Strugeon  Rene Mandiau

LAMIH (UMR CNRS 8530)
Université de Valenciennes, FRANCE

SARC’08
1. Background: bring MAS into human organization

2. Proposition of a holomas using roles
   - Elements of formalisation
   - Architecture

3. Dynamics in holonic multiagent organisation
   - Dynamic of roles
   - Robustness
   - Growth

4. Implementation

5. Application

6. Perspectives and Conclusion
   - Perspectives
   - Conclusion
Background: bring MAS into human organization

Application/Social Context

Integration of multi-agent organization into human organization

Administrative systems as Holonic organizations

Cognition

objective communications

a unity by service

Reaction

processes played by roles

communication with supervisor, assistants and neighbours
Background: bring MAS into human organization

Application/Social Context

Integration of multi-agent organization into human organization

Cognition

- objective communications
- a unity by service
- processes played by roles

Reaction

communication with supervisor, assistants and neighbours
Background: bring MAS into human organization

Application/Social Context

- Integration of multi-agent organization into human organization
- Administrative systems as Holonic organizations

Cognition

Communication

Objective communications

A unity by service

Processes played by roles

Communication with supervisor, assistants and neighbours
Proposition of a holomas using roles

Roles

According to the propositions in [Koestler 69] (canon, rules, skills, strategies), a role is composed of a set of rules:

- Mechanism 1: "Functional holons are governed by fixed sets of rules and display more or less flexible strategies."
- Mechanism 2: "The rules, referred to as the system's canon, determine its invariant properties, its structural configuration and/or functional pattern."
- Mechanism 3: "While the canon defines the permissible steps in the holon's activity, the strategic selection of the actual step among permissible choices is guided by the contingencies of the environment."
- Mechanism 4: "Holons on successively higher levels of the hierarchy show increasingly complex, more flexible and less predictable patterns of activity, while on successive lower levels we find increasingly mechanised, stereotyped and predictable pattern."
According to the propositions in [Koestler 69] (canon, rules, skills, strategies), a role is composed of a set of rules:

- Mechanism 1: "Functional holons are governed by fixed sets of rules and display more or less flexible strategies."
- Mechanism 2: "The rules, referred to as the system's canon, determine its invariant properties, its structural configuration and/or functional pattern."
- Mechanism 3: "While the canon defines the permissible steps in the holon's activity, the strategic selection of the actual step among permissible choices is guided by the contingencies of the environment."
- Mechanism 4: "Holons on successively higher levels of the hierarchy show increasingly complex, more flexible and less predictable patterns of activity, while on successive lower levels we find increasingly mechanised, stereotyped and predictable patterns."
Proposition of a holomas using roles

Roles

According to the propositions in [Koestler 69] (canon, rules, skills, strategies), a role is composed of a set of rules:

- **mechanism.1**: “Functional holons are governed by fixed sets of rules and display more or less flexible strategies.”
Roles

According to the propositions in [Koestler 69] (canon, rules, skills, strategies), a role is composed of a set of rules:

- **mechanism.1**: “Functional holons are governed by fixed sets of rules and display more or less flexible strategies.”
- **mechanism.2**: “The rules, referred to as the system’s canon, determine its invariant properties, its structural configuration and/or functional pattern.”
Roles

- According to the propositions in [Koestler 69] (canon, rules, skills, strategies), a role is composed of a set of rules:
  - **mechanism.1**: “Functional holons are governed by fixed sets of rules and display more or less flexible strategies.”
  - **mechanism.2**: “The rules, referred to as the system’s canon, determine its invariant properties, its structural configuration and/or functional pattern.”
  - **mechanism.3**: “While the canon defines the permissible steps in the holon’s activity, the strategic selection of the actual step among permissible choices is guided by the contingencies of the environment.”
## Proposition of a holomas using roles

### Roles

According to the propositions in [Koestler 69] (canon, rules, skills, strategies), a role is composed of a set of rules:

1. **mechanism.1**: “Functional holons are governed by fixed sets of rules and display more or less flexible strategies.”
2. **mechanism.2**: “The rules, referred to as the system’s canon, determine its invariant properties, its structural configuration and/or functional pattern.”
3. **mechanism.3**: “While the canon defines the permissible steps in the holon’s activity, the strategic selection of the actual step among permissible choices is guided by the contingencies of the environment.”
4. **mechanism.4**: “Holons on successively higher levels of the hierarchy show increasingly complex, more flexible and less predictable patterns of activity, while on successive lower levels we find increasingly mechanised, stereotyped and predictable pattern.”
Rules formalisation

we define a rule as a set of behaviours:

\[ R = (name_R, priority_R, tasks_R) \]

\[ tasks_R = \{ t_0^R, \ldots, t_{nt}^R \} \]

\[ nt = number \ of \ tasks \]
We define a Role as a set of essential rules and a set of secondary rules.

A searcher has to publish (a lot of) articles. To help the laboratory, he/she can manage the library, the projects, the phd students ....

\[
\text{role} = (\text{name, priority, KP, KE, KS, hardRules, flexibleRules})
\]

- **KP**: Pre-requirement, consequences, weight,
- **KE**: Environmental Knowledge (data)
- **KS**: Social Knowledge (roles names and constraints)

For example: a speaker has to respect the time-limit fixed by the chair-man.
Agents formalisation

Holonic Agents: Our holonic agents are defined as follows:

\[
\text{agent}_a = \left( \text{KP}, \text{KE}, \text{KS}, \text{HRA}, \text{messages}_a, \text{perception}_a, \text{rules}_a, \text{roles}_a \right)
\]

**KP**: *(Personal knowledge) = \{name; current state; individual goals(GI)\}*

**KE**: *(Environmental knowledge) = partial representations of objects of the environment.*

**KS**: *(Social knowledge) = partial representations of the acquaintances & collective goal (GC).*
Agents formalisation

Holonic Agents: Our holonic agents are defined as follow:

$$ \text{agent}_a = (KP, KE, KS, HRA, \text{messages}_a, \text{perception}_a, \text{rules}_a, \text{roles}_a) $$

**KP**: *(Personal knowledge)* = \{name; current state; individual goals (GI)\}

**KE**: *(Environmental knowledge)* = partial representations of objects of the environment.

**KS**: *(Social knowledge)* = partial representations of the acquaintances & collective goal (GC).

**HRA**: *(Holonic Roles Agent)* = Agent that manages roles of the system.
MAS formalisation

MAS

Mas and agent definition: MAS is simply defined as a set of agents.

Environment definition: \[ E = \{ object_0, object_1, ..., object_n \} \]

World definition: \[ world = (environment, mas) \]
Proposition of a holomas using roles

Architecture

General architecture of our Holonic IMAS

Architecture
General architecture of our Holonic IMAS

Architecture

- Each of our Holomas is assisted by a HRA that manages the roles
General architecture of our Holonic IMAS

Architecture

- Each of our Holomas is assisted by a HRA that manages the roles
- A HRA could be a set of HRA distributed around the Holomas
Dynamic of roles

- Priority of a role's rule increases each time the agent chooses it.
- Each agent has its own definition of a role.
- Each HRA agent checks if there are differences between prescribed roles and roles really played by agents.
- New definition of a role is kept if a sufficient number of agents have modified a role in the same way.
- A secondary rule can become a hard rule if all agents always choose it.
Dynamic of roles

- Priority of a role's rule increases each time the agent chooses it.
Dynamic of roles

- Priority of a role's rule increases each time the agent chooses it.
- Each agent has its definition of a role.
Dynamic of roles

- Priority of a role's rule increases each time the agent chooses it.
- Each agent has its definition of a role.
- Each HRA agent checks if there are differences between prescribed roles and roles really played by agents.
Dynamic of roles

- Priority of a role's rule increases each time the agent chooses it.
- Each agent has its definition of a role.
- Each HRA agent checks if there are differences between prescribed roles and roles really played by agents.
- New definition of a role is kept if a sufficient number of agents have modified a role in the same way.
Dynamic of roles

- Priority of a role's rule increases each time the agent chooses it.
- Each agent has its definition of a role.
- Each HRA agent checks if there are differences between prescribed roles and roles really played by agents.
- New definition of a role is kept if a sufficient number of agents have modified a role in the same way.
- A secondary rule can become a hard rule if all agents always choose it.
Robustness by replication in holonic multi-agent organisation

Robustness

Two critical agents: responsible for the Holonic Roles Agents & the Holomas responsible agent.

Flexible hierarchical organisation of role based agents

Adam, Grislin & Mandiau (LAMIH, CNRS) Flexible hierarchical organisation of role based...
Robustness

- Two critical agents: responsible of the Holonic Roles Agents & the Holomas responsible agent.
Robustness by replication in holonic multi-agent organisation

Robustness

- Two critical agents: responsible of the Holonic Roles Agents & the Holomas responsible agent.
  - physical replication, ping messages
Robustness by replication in holonic multi-agent organisation

- Two critical agents: responsible of the Holonic Roles Agents & the Holomas responsible agent.
  - Physical replication, ping messages
- For other agents, a watch holon-assistants, allows to detect damaged agents
Robustness by replication in holonic multi-agent organisation

<table>
<thead>
<tr>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Two critical agents: responsible of the Holonic Roles Agents &amp; the Holomas responsible agent.</td>
</tr>
<tr>
<td>- physical replication, ping messages</td>
</tr>
<tr>
<td>- For other agents, a watch holon-assistants, allows to detect damaged agents</td>
</tr>
<tr>
<td>- creation of a new agent by the responsible of the faulty agent</td>
</tr>
</tbody>
</table>
Robustness by replication in holonic multi-agent organisation

Robustness

- Two critical agents: responsible of the Holonic Roles Agents & the Holomas responsible agent.
  - physical replication, ping messages
- For other agents, a watch holon-assistants, allows to detect damaged agents
  - creation of a new agent by the responsible of the faulty agent
  - re-load of roles and links from the HRA
Robustness by replication in holonic multi-agent organisation

Robustness

- Two critical agents: responsible of the Holonic Roles Agents & the Holomas responsible agent.
  - physical replication, ping messages
- For other agents, a watch holon-assistants, allows to detect damaged agents
  - creation of a new agent by the responsible of the faulty agent
  - re-load of roles and links from the HRA
  - problem: loss of the data used during the breakdown
Growth when overload

- **Downward Growth**: The agent creates assistants and delegates some of its tasks to them.

![Diagram of agent growth](image.png)
Growth when overload

- Downward Growth: The agent creates assistants and delegates some of its tasks to them.
- Horizontal growth: An agent needs a skill and its responsible creates a neighbour.
Growth in holonic multi-agent organisation

Growth when overload

- **Downward Growth**: The agent creates assistants and delegates some of its tasks to them.
- **Horizontal growth**: An agent needs a skill and its responsible creates a neighbour.
- **Upward Growth**: To coordinate agents/systems acting towards a same goal.
Growth in holonic multi-agent organisation

**Growth when overload**

- **Downward Growth**: The agent creates assistants and delegates some of its tasks to them.
- **Horizontal growth**: an agent needs a skill and its responsible creates a neighbour.
- **Upward Growth**: To coordinate agents / systems acting towards a same goal.
- **Internal Growth**: A HoloMAS agent creates internal coordinators.

![Diagram showing growth in a holonic multi-agent organisation](image)
Well-balanced growth

Needs:
- Downward and horizontal growth must respect holonic concepts: an assistant deal with less complex roles than its responsible.
- Growth must produce a well-balanced holarchy.

Use of:
- Agent workload ($w_l$)
- Agents maximum workload thresholds ($m_{wt}$)
- Workload of a group (sub-holomas) ($g_{wl}$)
- Agent workload still available ($w_{la}$)

Note: Use of the ContractNet protocol.

Note: if $a_1$ is the assistant of $a_0$, $m_{wt_a_1} = \alpha \times m_{wt_a_0}$ with ($\alpha \in ]0, 1[$).
Well-balanced growth

Needs:

- Use of:
  - $\text{agent workload } (w_l)$
  - $\text{agents maximum workload thresholds } (m_{wt})$
  - $\text{workload of a group (sub-holomas) } (g_{wl})$
  - $\text{agent workload still available } (w_{la})$

Note: Use of the ContractNet protocol

Note: if $a_1$ is the assistant of $a_0$, $m_{wt}a_1 = \alpha \times m_{wt}a_0$ with ($\alpha \in ]0, 1[)$
Well-balanced growth

Needs:

- Downward and horizontal growth must respect holonic concepts: an assistant deal with less complex roles than its responsible
Well-balanced growth

- Needs:
  - Downward and horizontal growth must respect holonic concepts: an assistant deals with less complex roles than its responsible.
  - Growth must produce a well-balanced holarchy.
Well-balanced growth

- **Needs:**
  - Downward and horizontal growth must respect holonic concepts: an assistant deal with less complex roles than its responsible.
  - Growth must produce a well-balanced holarchy.

- **Use of:**

Note: Use of the ContractNet protocol.

Note: If \( a_1 \) is the assistant of \( a_0 \), then \( m_{wt}(a_1) = \alpha \times m_{wt}(a_0) \) with \( \alpha \in (0, 1) \).
Well-balanced growth

Needs:
- Downward and horizontal growth must respect holonic concepts: an assistant deal with less complex roles than its responsible
- Growth must produce a well-balanced holarchy

Use of:
- agent workload \((wl)\)
Well-balanced growth

- Needs:
  - Downward and horizontal growth must respect holonic concepts: an assistant deals with less complex roles than its responsible.
  - Growth must produce a well-balanced holarchy.

- Use of:
  - Agent workload \((wl)\)
  - Agents maximum workload thresholds \((mwt)\)
Well-balanced growth

- Needs:
  - Downward and horizontal growth must respect holonic concepts: an assistant deal with less complex roles than its responsible
  - Growth must produce a well-balanced holarchy

- Use of:
  - agent workload ($wl$)
  - agents maximum workload thresholds ($mwt$)
  - workload of a group (sub-holomas) ($gwl$)
Well-balanced growth

- **Needs**:
  - Downward and horizontal growth must respect holonic concepts: an assistant deals with less complex roles than its responsible.
  - Growth must produce a well-balanced holarchy.

- **Use of**:
  - Agent workload ($wl$)
  - Agents maximum workload thresholds ($mwt$)
  - Workload of a group (sub-holomas) ($gwl$)
  - Agent workload still available ($wla$)
Well-balanced growth

Needs:
- Downward and horizontal growth must respect holonic concepts: an assistant deals with less complex roles than its responsible.
- Growth must produce a well-balanced holarchy.

Use of:
- Agent workload ($wl$)
- Agents maximum workload thresholds ($mwt$)
- Workload of a group (sub-holomas) ($gwl$)
- Agent workload still available ($wla$)

Note: Use of the ContractNet protocol.
Well-balanced growth

Needs:
- Downward and horizontal growth must respect holonic concepts: an assistant deals with less complex roles than its responsible
- Growth must produce a well-balanced holarchy

Use of:
- Agent workload ($wl$)
- Agents maximum workload thresholds ($mwt$)
- Workload of a group (sub-holomas) ($gwl$)
- Agent workload still available ($wla$)

Note: Use of the ContractNet protocol

Note: if $a_1$ is the assistant of $a_0$,

$$mwt_{a_1} = \alpha \times mwt_{a_0} \quad \text{with} \quad (\alpha \in [0, 1])$$
Well-balanced growth

**procedure** `HANDLE_CFP`(*ACLMessage cfp*)

```
loadAsked ← cfp.getContent()
maxAssistantLoad ← α × mwt
wla ← mwt − holonCurrentLoad
if (wla − loadAsked) ≥ 0 ∨ ((maxAssistantLoad − loadAsked) ≥ 0 ∧ ¬holon.isLeaf())) then
    if (wla − loadAsked) > 0 then
        RETURN(wla, gwl)
    else
        RETURN(maxAssistantLoad, gwl)
    end if
else
    RETURN(Refuse)
end if
```

end procedure
Implementation of our architecture

Use of JADE platform, based on empty agents that receive behaviours. Role currently implemented as a set of behaviours (not a set of hard and flexible rules).

Core behaviours used by agents:
- HolonicMessagesManagement: owned by all agents; it allows an agent to handle messages relative to: the acquaintances updating; the role transmission; the growth.
- RegenerationManagement
- RoleManagement: behaviour is linked to the roles manager agents that could be compared to Directory Facilitator agents.
- ContractNetDelegation: linked temporarily to the initiator of the ContractNet protocol relative to the delegation of roles.
- ContractNetService: linked temporarily to the potential responders of the ContractNet protocol.
Implementation of our architecture

- Use of JADE platform, based on empty agents that receive behaviours
**Implementation of our architecture**

**Implementation**

- Use of JADE platform, based on empty agents that receive behaviours
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules)
Implementation

- Use of JADE platform, based on empty agents that receive behaviours
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules)
- Core behaviours used by agents:
Implementation of our architecture

Implementation

- Use of JADE platform, based on empty agents that receive behaviours
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules)
- Core behaviours used by agents:
  - HolonicMessagesManagement: owned by all agents; it allows an agent to handle message relative to: the acquaintances updating; the role transmission; the growth;
Implementation

- Use of JADE platform, based on empty agents that receive behaviours
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules)
- Core behaviours used by agents:
  - HolonicMessagesManagement: owned by all agents; it allows an agent to handle message relative to: the acquaintances updating; the role transmission; the growth;
  - RegenerationManagement: allows an agent to be surveyed by its acquaintances
Implementation of our architecture

**Implementation**

- Use of JADE platform, based on empty agents that receive behaviours.
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules).
- Core behaviours used by agents:
  - **HolonicMessagesManagement**: owned by all agents; it allows an agent to handle message relative to: the acquaintances updating; the role transmission; the growth;
  - **RegenerationManagement**: *allows an agent to be surveyed by its acquaintances*
  - **RoleManagement**: behaviour is linked the roles manager agents that could be compared to Directory Facilitator agents.
Implementation

- Use of JADE platform, based on empty agents that receive behaviours
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules)
- Core behaviours used by agents:
  - HolonicMessagesManagement: owned by all agents; it allows an agent to handle message relative to: the acquaintances updating; the role transmission; the growth;
  - RegenerationManagement: allows an agent to be surveyed by its acquaintances
  - RoleManagement: behaviour is linked to the roles' manager agents that could be compared to Directory Facilitator agents
  - ContractNetDelegation: linked temporarily to the initiator of the ContractNet protocol relative to the delegation of roles
Implementation of our architecture

**Implementation**

- Use of JADE platform, based on empty agents that receive behaviours.
- Role currently implemented as a set of behaviours (not a set of hard and flexible rules).
- Core behaviours used by agents:
  - **HolonicMessagesManagement**: owned by all agents; it allows an agent to handle message relative to: the acquaintances updating; the role transmission; the growth;
  - **RegenerationManagement**: allows an agent to be survey by its acquaintances
  - **RoleManagement**: behaviour is linked the roles manager agents that could be compared to Directory Facilitator agents
  - **ContractNetDelegation**: linked temporarily to the initiator of the ContractNet protocol relative to the delegation of roles
  - **ContractNetService**: linked temporarily to the potential responders of the ContractNet protocol.
Example 1 of well-balanced growth

Example 1 of well-balanced growth
Example 1 of well-balanced growth

- three initial agents: $a_1$ having a $mwt$ of 10, responsible for $a_{1.1}$ and $a_{1.2}$
Example 1 of well-balanced growth

- three initial agents: \( a_1 \) having a \( mwt \) of 10, responsible for \( a_{1.1} \) and \( a_{1.2} \)
- three roles: \( \text{HeavyRole}(HR) \), \( load = 8 \); \( \text{MediumRole}(MR) \), \( load = 5 \) and \( \text{LightRole}(LR) \), \( load = 2 \)
Example 1 of well-balanced growth

- **three initial agents**: $a_1$ having a $mwt$ of 10, responsible for $a_{1.1}$ and $a_{1.2}$
- **three roles**: $\text{HeavyRole}(HR)$, $load = 8$; $\text{MediumRole}(MR)$, $load = 5$ and $\text{LightRole}(LR)$, $load = 2$
- $\alpha = 0.75 \Rightarrow mwt_{a_{1.1}} = mwt_{a_{1.2}} = 7$
Well balanced growth : Example 1

Example 1 of well-balanced growth

- three initial agents: $a_1$ having a $mwt$ of 10, responsible for $a_{1.1}$ and $a_{1.2}$
- three roles: $\text{HeavyRole}(HR)$, load $= 8$; $\text{MediumRole}(MR)$, load $= 5$ and $\text{LightRole}(LR)$, load $= 2$
- $\alpha = 0.75 \Rightarrow mwt_{a_{1.1}} = mwt_{a_{1.2}} = 7$
- scenario = add the roles $\{HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR\}$ to $a_1$
Well balanced growth : Example 1

\[ a_1: \{HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR\}. \]
\[ \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}. \]
Well balanced growth : Example 1

\[ a_1: \{HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR\}. \]
\[ \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}. \]

\[ mwt=10 \]
1.HR(8) \[ \quad \quad \quad \quad \quad \text{Group workload} = 36 \]
4.LR(2)

\[ mwt=7 \]
2.MR(5) \[ \quad \quad \quad \quad \text{Group workload} = 14 \]
5.LR(2)

\[ mwt=5 \]
7.LR(2) \[ \quad \quad \text{a1.1.1} \]
9.MR(5) \[ \quad \quad \quad \quad \quad \text{a1.1.2} \]

\[ mwt=7 \]
3.MR(5) \[ \quad \quad \quad \quad \text{a1.2} \]
6.LR(2)

\[ mwt=5 \]
8.MR(5) \[ \quad \quad \quad \quad \quad \text{a1.2.1} \]
Well balanced growth : Example 1

\[
a_1: \{HR, MR, MR, LR, LR, LR, MR, MR, LR, LR, LR\}.
\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}.
\]

\[
\begin{align*}
1. & \text{ HR}(8) \\
4. & \text{ LR}(2) \\
2. & \text{ MR}(5) \\
5. & \text{ LR}(2) \\
7. & \text{ LR}(2) \\
11. & \text{ LR}(2) \\
9. & \text{ MR}(5) \\
10. & \text{ LR}(2) \\
12. & \text{ LR}(2) \\
\end{align*}
\]

\[
\begin{align*}
mwt &= 10 \\
mwt &= 7 \\
mwt &= 7 \\
mwt &= 5 \\
mwt &= 5 \\
mwt &= 5 \\
mwt &= 5 \\
mwt &= 3 \\
mwt &= 3 \\
\end{align*}
\]

Group workload = 42

Group workload = 16

Group workload = 16
Example 2 of well-balanced growth

Example 2 of well-balanced growth
Example 2 of well-balanced growth

- same three initial agents $a_1$, $a_{1.1}$ and $a_{1.2}$; but with $mwt_{a_1} = 8$
Example 2 of well-balanced growth

- same three initial agents $a_1, a_{1.1}$ and $a_{1.2}$; but with $mwt_{a_1} = 8$
- same three roles: $\text{HeavyRole}(HR)$, $load = 8$; $\text{MediumRole}(MR)$, $load = 5$ and $\text{LightRole}(LR)$, $load = 2$
Well balanced growth - Example 2

Example 2 of well-balanced growth

- same three initial agents $a_1$, $a_{1.1}$ and $a_{1.2}$; but with $mwt_{a_1} = 8$
- same three roles: $HeavyRole(HR)$, $load = 8$; $MediumRole(MR)$, $load = 5$ and $LightRole(LR)$, $load = 2$
- $\alpha = 0.75 \Rightarrow mwt_{a_{1.1}} = mwt_{a_{1.2}} = 5$
Example 2 of well-balanced growth

- same three initial agents \( a_1, a_{1.1}, \text{ and } a_{1.2} \); but with \( mwt_{a_1} = 8 \)
- same three roles: \( \text{HeavyRole}(HR), load = 8 \); \( \text{MediumRole}(MR), load = 5 \) and \( \text{LightRole}(LR), load = 2 \)
- \( \alpha = 0.75 \Rightarrow mwt_{a_{1.1}} = mwt_{a_{1.2}} = 5 \)
- scenario = add the roles \{ HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR \} to \( a_1 \)
Well balanced growth - Example 2

\[ a_1: \{\text{HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR}\}. \]
\[ \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}. \]

\[ \text{Group workload} = 18 \]

\[ \text{Group workload} = 5 \]

\[ \text{Group workload} = 5 \]

\[ mwt=8 \]

\[ mwt=6 \]
Well balanced growth - Example 2

\[ a_1: \{ \text{HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR} \}. \]
\[ \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 \}. \]

\[ \text{Group workload} = 26 \]

\[ \text{Group workload} = 9 \]

\[ \text{Group workload} = 9 \]
Well balanced growth - Example 2

\[ a_1: (HR, MR, MR, LR, LR, LR, LR, MR, MR, LR, LR, LR).\]
\[ (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12).\]

**Group workload = 42**
Implementation of an application

Implementation of a new application

Just have to program the behaviours

Relation between roles and behaviours, agents and roles and between agents are described in an XML file.

An application deploys the agents on the net and transmits them roles and initial data.

Currently: Turn to an OWL-S representation of roles and the HoloMAS.
Implementation of an application

Implementation of a new application

- Just have to program the behaviours
Implementation of an application

Implementation of a new application

- Just have to program the behaviours
- Relation between roles and behaviours, agents and roles and between agents are described in a xml file
Implementation of an application

Implementation of a new application

- Just have to program the behaviours
- Relation between roles and behaviours, agents and roles and between agents are described in a xml file
- An application deploys the agents on the net and transmits them roles and initial data
Implementation of an application

Implementation of a new application

- Just have to program the behaviours
- Relation between roles and behaviours, agents and roles and between agents are described in a XML file
- An application deploys the agents on the net and transmits them roles and initial data
- Currently: Turn to a OWL-S representation of roles and the HoloMAS
Applications

- Assist cooperative work into a patent department [Holomas’00] and a technological watch department [AOIS’04]

Method

1. Analysis and modelling of the human organization
Applications

- Assist cooperative work into a patent department [Holomas’00] and a technological watch department [AOIS’04]

Method

1. Analysis and modelling of the human organization
2. Improvement of the human organization based on the models
Applications

- Assist cooperative work into a patent department [Holomas’00] and a technological watch department [AOIS’04]

Method

1. Analysis and modelling of the human organization
2. Improvement of the human organization based on the models
3. Design of a MAS based on holonic principles and modeled on the human organization
Example on a case study: Information MultiAgent System
Modeling the human activities

According to the results, it should be interesting to modify the request, add/remove points to search engine, ...
Example on a case study: Information MultiAgent System
Architecture of the proposed Holonic IMAS

Adam, Grislin & Mandiau (LAMIH, CNRS)
Example on a case study: Information MultiAgent System

Exchanged messages during a delegation

Create a new assistant, link it to the RH, inform it of its neighbours and propose the role to it.

Inform other assistants of their new neighbour.

Cfp: to find an assistant available for the load of the role

agent1.1 ok

Cfp: to find an assistant available for the load of the role

none
Perspectives and Conclusion

Perspectives

Perspectives: self-* capacities

Evaluation of rules utility based on the reinforcement

Problem: always use the same rules → allow to test other rules

Criticity of the agents: only the 2 heads of the systems are said criticals → use an automatic detection of the agents criticity

Roles:
- Role complexity: specified and implemented by the programmers → automatic detection of role complexity

Perspective: automatic detection of the inter-blocking situations / non-cooperative behaviour

Implementation of the totality of our proposition

Adam, Grislin & Mandiau (LAMIH, CNRS)
Flexible hierarchical organisation of role based...
Perspectives : self-* capacities

- Evaluation of rules utility based on the reinforcement
Perspectives

Perspectives : self-* capacities

- Evaluation of rules utility based on the reinforcement
  - Problem : always use the same rules → to allow to test other rules
Perspectives : self-* capacities

Perspectives

- Evaluation of rules utility based on the reinforcement
  - Problem : always use the same rules → to allow to test other rules
- Criticity of the agents : only the 2 heads of the systems are said criticals
Perspectives : self-* capacities

Evaluation of rules utility based on the reinforcement
- Problem: always use the same rules → to allow to test other rules
- Criticity of the agents: only the 2 heads of the systems are said criticals
  → use an automatic detection of the agents criticity
Perspectives: self-* capacities

- Evaluation of rules utility based on the reinforcement
  - Problem: always use the same rules → to allow to test other rules
- Criticity of the agents: only the 2 heads of the systems are said criticals
  - → use an automatic detection of the agents criticality
- Roles:
Perspectives: self-* capacities

Perspectives

- Evaluation of rules utility based on the reinforcement
  - Problem: always use the same rules → to allow to test other rules
- Criticity of the agents: only the 2 heads of the systems are said criticals
  - → use an automatic detection of the agents criticality
- Roles:
  - Role complexity: specified and implemented by the programmers → automatic detection of role complexity
Perspectives and Conclusion

Perspectives

Perspectives : self-* capacities

- Evaluation of rules utility based on the reinforcement
  - Problem : always use the same rules → to allow to test other rules
- Criticity of the agents : only the 2 heads of the systems are said criticals
  - → use an automatic detection of the agents criticity
- Roles :
  - Role complexity : specified and implemented by the programmers → automatic detection of role complexity
  - Perspective : automatic detection of the inter-blocking situations / non cooperative behaviour
Perspectives

- **Perspectives**
  - Evaluation of rules utility based on the reinforcement
    - Problem: always use the same rules → to allow to test other rules
  - Criticity of the agents: only the 2 heads of the systems are said criticals
    - → use an automatic detection of the agents criticity
  - Roles:
    - Role complexity: specified and implemented by the programmers → automatic detection of role complexity
    - Perspective: automatic detection of the inter-blocking situations / non cooperative behaviour
    - Implementation of the totality of our proposition
Conclusion

We propose a multi-agent organisation using both notion of roles and notion of hierarchy. This proposition helps us to easily develop MAS to help actors of complex administrative systems. Improve the proposition by self-* capacities.

- Running projects
  - Develop an applicative framework for a "tangible table"
  - Assist workflow for a logistics management problem (start soon)

Adam, Grislin & Mandiau (LAMIH, CNRS) Flexible hierarchical organisation of role base
Conclusion

- We propose a multi-agent organisation using both notion of roles and notion of hierarchy.
Conclusion

- We propose a multi-agent organisation using both notion of roles and notion of hierarchy.
- This proposition help us to easily develop MAS to help actors of complex administrative systems.
We propose a multi-agent organisation using both notion of roles and notion of hierarchy.

This proposition help us to easily develop MAS to help actors of complex administrative systems.

Improve the proposition by self-* capacities.
We propose a multi-agent organisation using both notion of roles and notion of hierarchy.

This proposition help us to easily develop MAS to help actors of complex administrative systems.

Improve the proposition by self-* capacities.

running projects
Conclusion

- We propose a multi-agent organisation using both notion of roles and notion of hierarchy.
- This proposition help us to easily develop MAS to help actors of complex administrative systems.
- Improve the proposition by self-* capacities.
- running projects
  - Develop an applicative framework for a “tangible table”
Conclusion

- We propose a multi-agent organisation using both notion of roles and notion of hierarchy.
- This proposition help us to easily develop MAS to help actors of complex administrative systems.
- Improve the proposition by self-* capacities.
- running projects
  - Develop an applicative framework for a “tangible table”
  - Assist workflow for a logistic management problem (start soon)