On Programming Organization-Aware Agents

Andreas Schmidt Jensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

November 20, 2013

12th Scandinavian AI conference
Doctoral Symposium
1. Background

2. Motivation

3. Aim & Approach

4. Results

5. Ongoing & Future work
Background

- Intelligent agents
Intelligent agents

- Can act and sense in their environment
- They are:
  - Proactive
  - Reactive
  - Autonomous
  - Social
- Beliefs, Desires and Intentions (BDI)
Background

- Intelligent agents
  - Can act and sense in their environment
  - They are:
    - Proactive
    - Reactive
    - Autonomous
    - Social
  - Beliefs, Desires and Intentions (BDI)
- Multi-agent systems
Background

- Intelligent agents
  - Can act and sense in their environment
  - They are:
    - Proactive
    - Reactive
    - Autonomous
    - Social
  - Beliefs, Desires and Intentions (BDI)
- Multi-agent systems
  - Multiple agents
  - The whole is greater than the sum of its parts
Open societies lets any agent participate
Motivation

- Open societies lets any agent participate
- No control and no way to reach global objectives
Motivation

- Open societies lets any agent participate
- No control and no way to reach global objectives
- Organizations
Motivation

- Open societies lets any agent participate
- No control and no way to reach global objectives
- Organizations
  - Agent objectives meets system expectations
  - Structures the agents into roles and groups
  - Rights and norms
  - Improves coordination and cooperation
  - May prevent autonomy
Motivation

- Open societies lets any agent participate
- No control and no way to reach global objectives
- Organizations
  - Agent objectives meets system expectations
  - Structures the agents into roles and groups
  - Rights and norms
  - Improves coordination and cooperation
  - May prevent autonomy
- How to respect (or deliberately ignore) organization?
Motivation

- Open societies lets any agent participate
- No control and no way to reach global objectives
- Organizations
  - Agent objectives meets system expectations
  - Structures the agents into roles and groups
  - Rights and norms
  - Improves coordination and cooperation
  - May prevent autonomy
- How to respect (or deliberately ignore) organization?
  - Middleware
  - Reasoning capabilities $\Rightarrow$ Organization-aware agents
Aim & Approach

- **Main goal:** Organization-Aware Agents

- **Theoretical**
  - Organizational models: OperA – Moise$^+$ – ISLANDER
  - Specification and verification: Logic of Agent Organizations

- **Practical**
  - Agent frameworks: Jason – GOAL – Jadex
Results

Conflicts in decision making


Formalizing organizational models


Organizational reasoning


Adding Organizational Reasoning to Agents

Conflicts in decision making

Other agents

Obligations → Agent’s influences ← Desires
Conflicts in decision making

**Simple solution:** A priori ordering.
Conflicts in decision making

**Simple solution:** A priori ordering.

- Desires before obligations $\rightarrow$ *Selfish agent*
- Obligations before desires $\rightarrow$ *Social agent*
Conflicts in decision making

**Simple solution:** A priori ordering.
- Desires before obligations $\rightarrow$ *Selfish agent*
- Obligations before desires $\rightarrow$ *Social agent*

**Better:** Consequences of being in different situations
Conflicts in decision making

**Simple solution:** A priori ordering.
- Desires before obligations $\rightarrow$ *Selfish agent*
- Obligations before desires $\rightarrow$ *Social agent*

**Better:** Consequences of being in different situations
- $\neg work \rightarrow fired$
- $work \rightarrow \neg fired$
Conflicts in decision making

- Conflicts arise in the agent deliberation process
- Rules of preference and expectation are specified
- Model generation
- Conflicts resolved using expected consequences
  - In some cases the agent violates its obligation.
  - In other cases it ignores its desire.
Formalizing organizational models

- Formal model required for agent reasoning
  - Models such as OperA and MoISE\textsuperscript{+}
- We have shown correspondence with certain improvisational theatrical performances (my talk tomorrow)
- Multi-agent programming languages based on variants of Prolog (Jason, GOAL)
## Formalizing organizational models

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>role((r, O))</td>
<td>Role (r) with objectives (O).</td>
</tr>
<tr>
<td>dependency((r_1, r_2, o, t))</td>
<td>Dependency between roles (r_1) and (r_2) for objective (o) and dependency type (t).</td>
</tr>
<tr>
<td>scene((s, R, Res))</td>
<td>Scene script (s) with roles (R) and results (Res).</td>
</tr>
<tr>
<td>rea((a, r, s))</td>
<td>Agent (a) enacts role (r) in scene (s).</td>
</tr>
</tbody>
</table>

...
Formalizing organizational models

\texttt{responsible}(\texttt{Obj}, \texttt{Scene}, \texttt{Role}) :-
\hspace{1cm} \texttt{scene}(\texttt{Scene}, \texttt{Roles}, \texttt{Objectives}),
\hspace{1cm} \texttt{member}(\texttt{Role}, \texttt{Roles}), \texttt{member}(\texttt{Obj}, \texttt{Objectives}),
\hspace{1cm} \texttt{role}(\texttt{Role}, \texttt{RoleObjectives}), \texttt{member}(\texttt{Obj}, \texttt{RoleObjectives}).

\texttt{delegate}(\texttt{Me}, \texttt{Objective}, \texttt{Scene}, \texttt{OtherAg}, \texttt{Type}) :-
\hspace{1cm} \texttt{rea}(\texttt{Me}, \texttt{MyRole}, \texttt{Scene}), \texttt{rea}(\texttt{OtherAg}, \texttt{OtherRole}, \texttt{Scene}),
\hspace{1cm} \texttt{dependency}(\texttt{MyRole}, \texttt{OtherRole}, \texttt{Objectives}, \texttt{Type}),
\hspace{1cm} \texttt{member}(\texttt{Objective}, \texttt{Objectives}).
Organizational reasoning

Beliefs

Desires

Organizational reasoning

Org. options

Org. actions

Intentions
Organizational reasoning in GOAL

Option consideration and organizational deliberation:

\[
\text{forall } \text{bel(rea}(A,R,S), \text{ responsible}(0,S,R), \text{ active}(0)) \text{ do insert(option}(A,0,S)).
\]

if bel(option(_,injuredLocated,__)) then adopt(injuredLocated).

Delegation:

if a-goal(in(X)), bel(room_blocked(X), rea(Me,R,S),
\hspace{1cm} \text{delegate(Me,blockingFanRemoved,S,Other,__)})
then send(Other, !do(blockingFanRemoved)).

Same objective:

\[
\text{forall } \text{a-goal(injuredLocated), bel(rea}(A,R,S),
\hspace{1cm} \text{responsible(injuredLocated,S,R)) do } \{
\text{forall } \text{<injured found> do send}(A, \text{<location>}).
\text{forall } \text{<room checked> do send}(A, \text{<room>}). \}
\]
AORTA: Adding Organizational Reasoning to Agents

- Conflicts in decision making
- Formalizing organizational models
- Organizational reasoning

AORTA
AORTA: Adding Organizational Reasoning to Agents

- **Organizational formulas**
  - \(\text{org(objective(injuredFound, medic))} \land \neg \text{bel(injuredFound)}\)

- **Actions**
  - \(\text{consider}(\phi), \text{enact}(\alpha, \rho), \ldots\)

- **Reasoning rules**
  - \(\text{org(role}(r, Os) \land \forall o(o \in Os \rightarrow \text{bel(cap}(o))) \Rightarrow \text{consider}(\text{rea}(\alpha, r))\)

- **Transitions**
  - \(\rho \Rightarrow a_{O \in OR} \quad \langle \Sigma, \kappa, \sigma, \gamma \rangle \models \mathcal{L}_R \rho \quad \mathcal{T}_O(a_{O, \kappa, \sigma, \gamma}) = \gamma' \quad \langle \Sigma, \kappa, \sigma, \gamma \rangle \rightarrow \langle \Sigma, \kappa, \sigma, \gamma' \rangle\)
Ongoing & Future work

- AORTA
  - Prototype
  - Integration with existing tools (e.g. GOAL)
  - Verification
- Deciding between organizational and agent objectives
  - The multi-agent case
  - Allow for more expressive objectives and consequences
  - Integrate with AORTA
- Applications
  - Computer games (e.g. real-time strategy)
  - Theatrical improvisation
Thank you for your attention