Dimensions of Organizational Coordination

Andreas Schmidt Jensen [a] Huib Aldewereld [b] Virginia Dignum [b]

[a] Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Delft University of Technology, Delft, The Netherlands

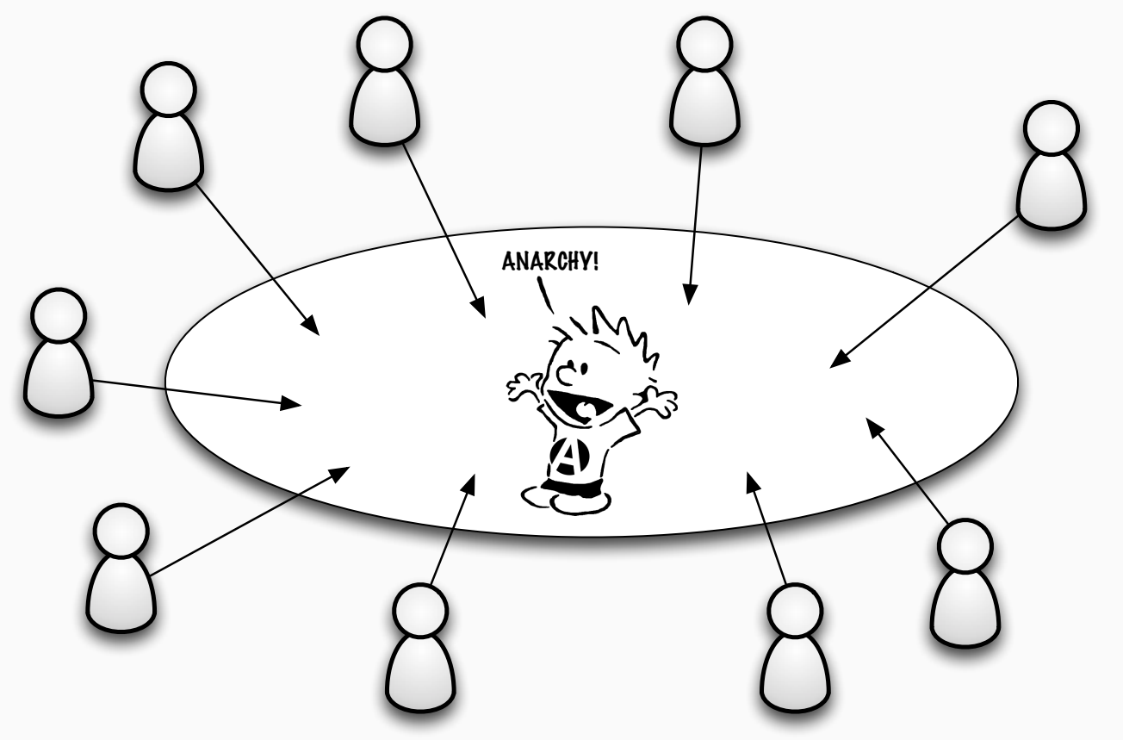
Abstract

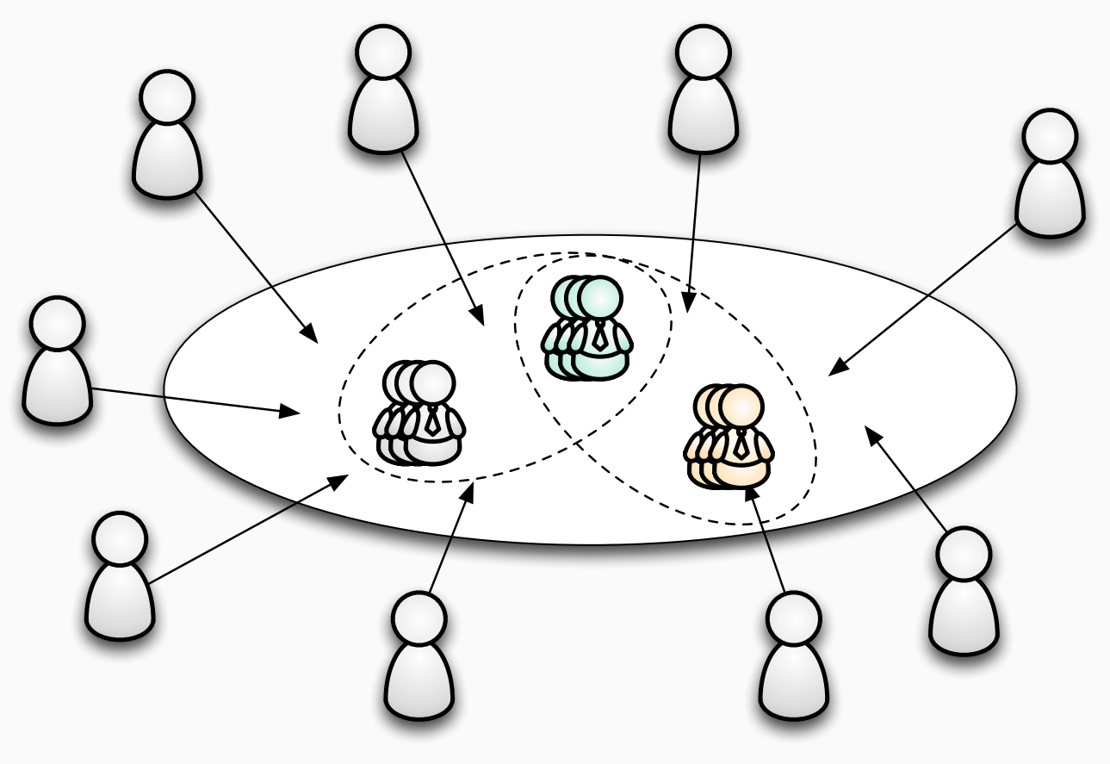
It is hard, if not impossible, to assume anything about agents' behavior in a society with heterogeneous agents from different sources. Organizations are used to restrict and guide the agents' actions such that the global objectives of the society are achieved. We discuss how agents can be supported to include organizational objectives and constraints into their reasoning processes by considering two alternatives: agent reasoning and middleware regulation. We show how agents can use an organizational specification to achieve organizational objectives by delegating and coordinating their activities with other agents in the society, using the GOAL agent programming language and the OperA organizational model.

1. Motivation

We consider *Open Societies*

- Accessible by anyone
- No control over the agents entering it.
- Not possible to ensure completion of global objectives.
- Hard to assume any kind of agent behavior.





Organizations

- Often used in multi-agent systems for controlling agents entering a society.
- A more or less abstract description of what is expected of the agents in the society.
- Ensure certain objectives are achieved.
- Contains predefined boundaries that should not be violated.

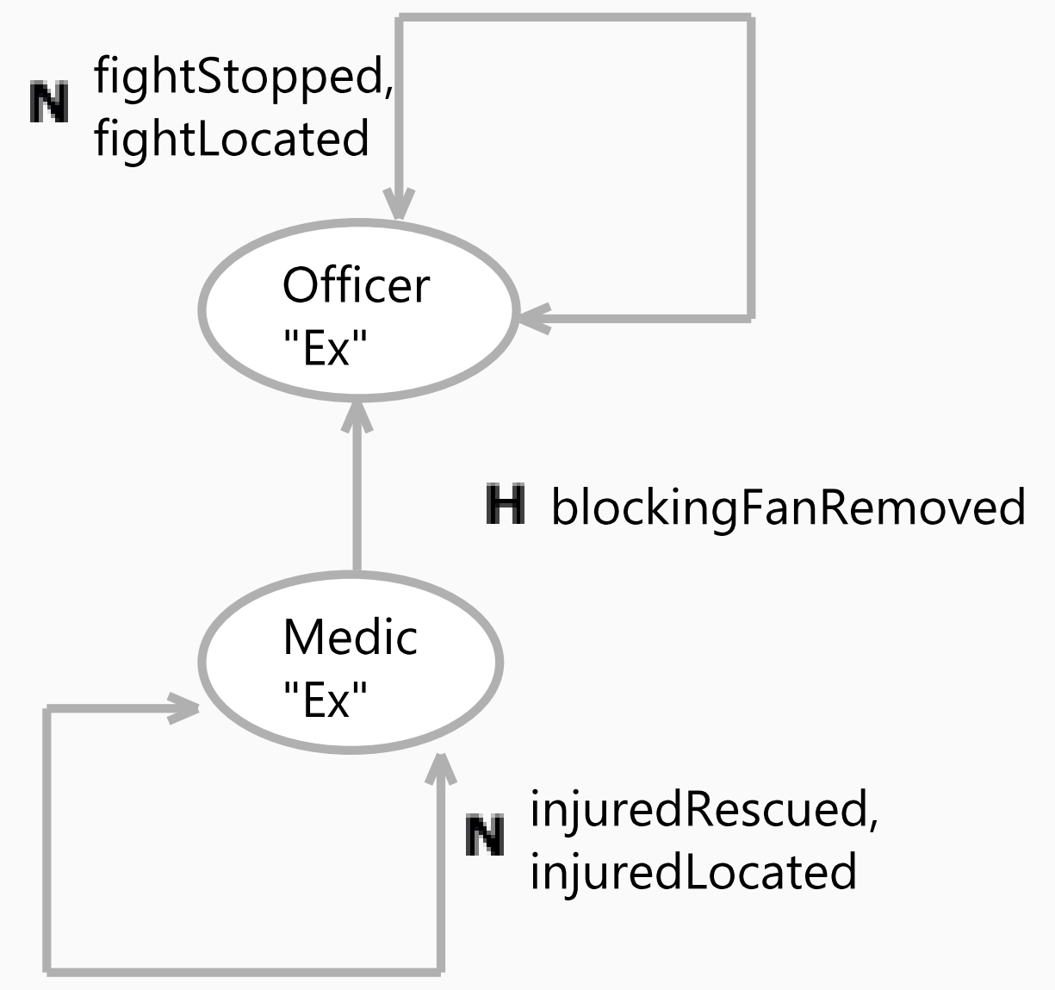
Question: How can agents in an organization coordinate their organizational objectives in order to complete them as efficiently as possible?

Scenario: Crisis Response

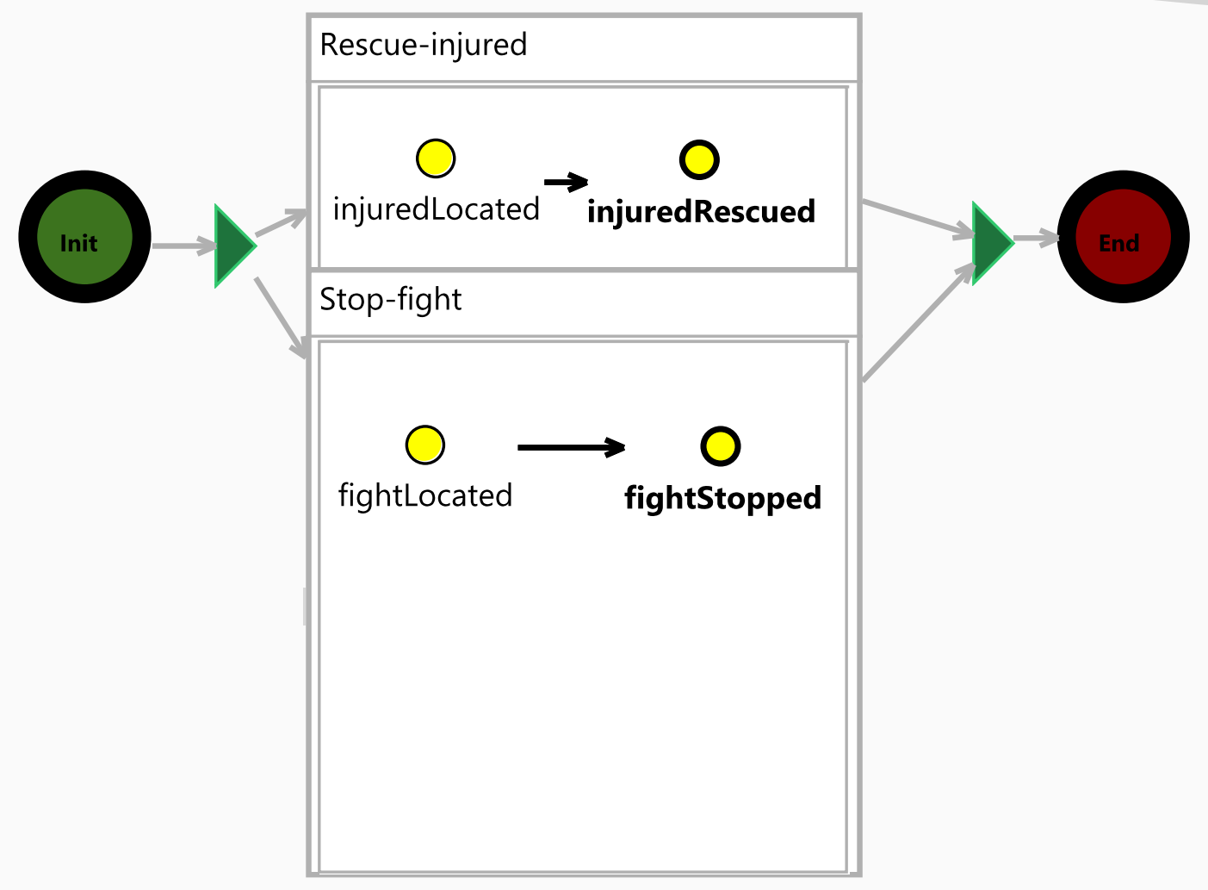
After a match between Feyenoord and Ajax, groups of fans are fighting and some of the fans are badly hurt. The authorities have been contacted, and a number of medics and police officers (the first-responders) have arrived. The medics are supposed to help the injured, while the police officers are supposed to break up the fight. However, fans of one group will not allow medics to help injured from the other group.

2. Organizational Modeling

We model the scenario using the **OperA model** [1], which proposes an expressive way for defining open organizations distinguishing explicitly between the organizational aims, and the agents who act in it.



Social structure



Interaction structure

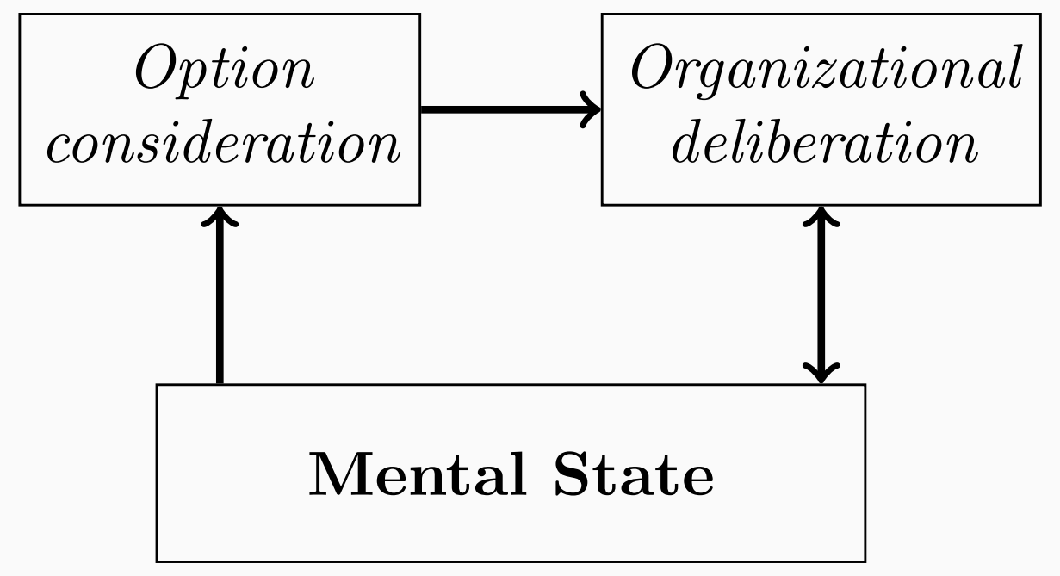
3. Dimensions

- Agents are assumed to be able to understand and reason about an organization, e.g. by:
  - → Making them organization-aware.
  - → Using organizational artifacts [5, 3].
  - → Using a middleware [4].
- The organization expects that agents playing such role adheres to its norms and attempts to achieve its objectives.
  - → Can agents deviate from expectations?
  - → If so, how are they monitored and sanctioned?

Regulated	Monitoring Sanctions	Distributed monitoring Distributed sanctioning
Restricted	Step-by-step orders	Coordination Organizational reasoning
	Middleware	**Agent**

4. Organizational Reasoning

- Option consideration phase:
  - → Which objectives is the role I enact responsible for?
  - → Which of my objectives are active (e.g. when a precondition holds)?
- Organizational deliberation phase:
  - → Should I commit to this objective?
  - → Who can I coordinate with?
  - → Can I delegate a task to another agent?
- The agent's mental state is used in both phases.



- Simple rules lets agents take the organization into account:
  - → Which objectives is my role responsible for?
  - → To whom can I delegate a certain objective?

```
responsible(Obj, Scene, Role) :-
        scene(Scene, Roles, Objectives),
        member(Role, Roles), member(Obj, Objectives),
        role(Role, RoleObjectives), member(Obj, RoleObjectives).

delegate(Me, Objective, Scene, OtherAg, Type) :-
        rea(Me, MyRole, Scene), rea(OtherAg, OtherRole, Scene),
        dependency(MyRole, OtherRole, Objectives, Type),
        member(Objective, Objectives).
```

5. Guiding Agents

The following code samples (based on the GOAL agent programming syntax [2]) allow agents to perform organizational reasoning.

Option consideration and organizational deliberation: An objective is an option when the agent is enacting a role responsible for the objective, and the objective is active. The agent simply commits to the `injuredLocated` objective, once it is an option.

```
forall bel(rea(A,R,S), responsible(O,S,R), active(O)) then insert(option(A,O,S)).
if bel(option(__,injuredLocated,__)) then adopt(injuredLocated).
```

Delegation: The agent will delegate `blockingFanRemoved` to another agent.

```
if a-goal(in(X)), bel(room_blocked(X), rea(Me,R,S), delegate(Me,blockingFanRemoved,S,Other,__))
then send(Other, !do(blockingFanRemoved)).
```

Dependency coordination: The agent has completed an objective, which was delegated from another agent, and should inform that agent.

```
if bel(reached(O), delegate(Other, O, S, Me), rea(Other,R,S)) then send(Other, :reached(O, S)).
```

Same objective: The agent is commited to `injuredLocated` and will inform other agents responsible for the same objective about its progress.

```
forall a-goal(injuredLocated), bel(rea(A,R,S), responsible(injuredLocated,S,R)) do {
    forall <injured found> do send(A, <location>).
    forall <room checked> do send(A, <room>).  }
```

Same scene: Agents participating in a scene should inform other agents participating in the same scene about their progress.

```
forall bel(option(A1,O,S), reached(O),rea(A2,__,S)) do send(A2, :reached(O, S)).
```

6. Conclusion

- Enables both the agent and the middleware approach.
  - → If nothing is disclosed, the middleware can mostly provide "step-by-step" guidance.
  - → If the agent discloses everything, a middleware will resemble an organization-aware agent.
- The ultimate way of bringing the control back to the agents is to allow them to reason about the organization themselves.
  - → Our building blocks allow this, either by letting agents disclose their beliefs to a middleware, or by integrating the building blocks in the agents.

References

[1] H. Aldewereld and V. Dignum. OperettA: Organization-oriented development environment. In *Languages, Methodologies, and Development Tools for Multi-Agent Systems*. Springer, 2011.

[2] K. V. Hindriks. Programming Rational Agents in GOAL. *Multi-Agent Programming: Languages, Tools and Applications*, 2, 2009.

[3] J. F. Hübner, O. Boissier, R. Kitio, and A. Ricci. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, 20(3), 2010.

[4] J. F. Hübner, J. S. Sichman, and O. Boissier. S-moise+: A middleware for developing organised multi-agent systems. In *COIN I, volume 3913 of LNAI*. Springer, 2006.

[5] A. Ricci, M. Viroli, and A. Omicini. Programming MAS with artifacts. In *Proceedings of the Third international conference on Programming Multi-Agent Systems*, ProMAS'05, Springer, 2006.