WOA 2008

2008, November 18

Adding Organizations and Roles as Primitives to the JADE Framework

Matteo Baldoni[1], Guido Boella[1], Mauro Dorni[1], Roberto Grenna[1], Andrea Mugnaini[1], and Leendert van der Torre[2]

[1]Dipartimento di Informatica, Università degli Studi di Torino

[2]Computer Science and Communications, University of Luxembourg

UNIVERSITÀ
DEGLI STUDI
DI TORINO

ALMA UNIVERSITAS
TAURINENSIS

Outline

- Organizations and roles in MAS…
- … and in JADE
  - Organizations
  - Roles
  - Players
  - Interaction
    - Players and organization
    - Players and their roles
- Conclusions and future works

Organizations and roles in MAS… (1)

Organizations are used for:

- coordinating open multiagent systems,
- providing control of access rights,
- enabling the interaction of heterogeneous agents,
- providing suitable abstractions to model real world institutions.

Roles facilitate the coordination of agents inside an organization, and give new abilities in the context of organizations, called powers, to the agents which satisfy the requirements necessary to play them (in our vision).

Organizations and roles in MAS… (2)

A merger of models (1)

Our work is based on a merger of models:

- an ontological one, by Boella and van der Torre.
- one about the problem of formally defining the dynamics of roles, by Dastani, van Riemsdijk, Hulstijn, Dignum and Meyer

Organizations and roles in MAS… (3)

A merger of models (2)

From the first one we know that:

- To play roles it is necessary to have the capabilities for fulfilling the requirements imposed by each role.
- Roles are offered by organizations to allow a player to interact with the organization in different ways, depending on the player capabilities.
- Roles are instances which represent a part of the state of the organization offering them
- Players of roles can act in the organization via roles, by invoking the powers offered by them.
- Power are methods which can change the state of the played role, of the sibling roles, and of the organization.

Organizations and roles in MAS… (4)

A merger of models (3)

The second one, give us four operations to deal with role dynamics of playing roles.

- *Enact*: is the request that a player does to be made able to play a role
- *Deact*: finishes the association of a player with a fixed role
- *Activate*: since only one role can be played at a time, this operation allows a player to activate a certain role, and to use powers that the role gives
- *Deactivate*: it’s called when the player suspends temporarily to play a fixed role; the role cold be activated again using *Activate*

Organizations and roles in MAS… (5)

A merger of models (4)

So, in our model, organizations are first class entities of the MAS, rather than a mental constructions which agents use to coordinate themselves, or abstractions used only in the design phase of the system.

… and in JADE (1)

What is JADE?

JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language by T-Lab.

It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications. The agent platform can be distributed across machines (which not even need to share the same OS).

… and in JADE (2)

Why JADE?

JADE has many interesting characteristics:

- it’s a general purpose framework,
- it’s open source,
- agents are implemented extending the *Agent* class offering primitives for agents communication issues and scheduling of goals,
- since it’s written in Java, it has a great possibility of integration with other Java-based frameworks, tools, languages (e.g.: Jess, Jason),
- since it’s written in Java, allows us to re-use the strategy we used to design powerJava, a version of Java language extended with primitives for organizations and roles.

… and in JADE (3)

Three new classes

To extend JADE we introduce three basic classes (extending *Agent*):

- *Organization*
- *Role*
- *Player*

They contain the primitives for the coordination of players, organizations, and roles

Note that an instance of the class *Role* is not autonomous.

… and in JADE (4)

Organizations

The class *Organization* is a subclass of the JADE *Agent* class. An organization offers:

- all the roles classes whose instances agents can play inside the organization,
- all the protocols needed to communicate with agents (*players*) which want to play a role inside it,
- the behaviours to manage information about roles and their players.

… and in JADE (5)

Roles (1)

Roles are offered by organizations, and offers the protocols to communicate with the player agent, and the methods for the role programmer to use these protocols.

Roles are Java *inner classes* of the class *Organization*, like in language powerJava.

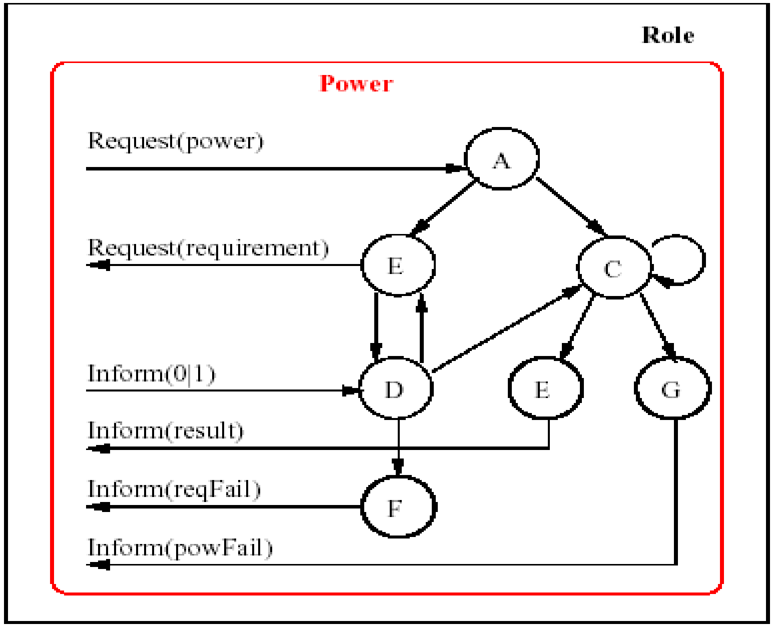
The role programmer has to define the methods (powers) which can be invoked by the player, and to specify them in a data structure used to select the correct power to be executed.

… and in JADE (6)

Roles (2)

Roles can interact with the player sending and receiving messages.

The role behaviour is a *Final State Machine Behaviour* (a typical JADE Behaviour):



… and in JADE (7)

Players (1)

Players of roles in organizations are offered by the class *Player*, which extends the JADE *Agent* class.

They:

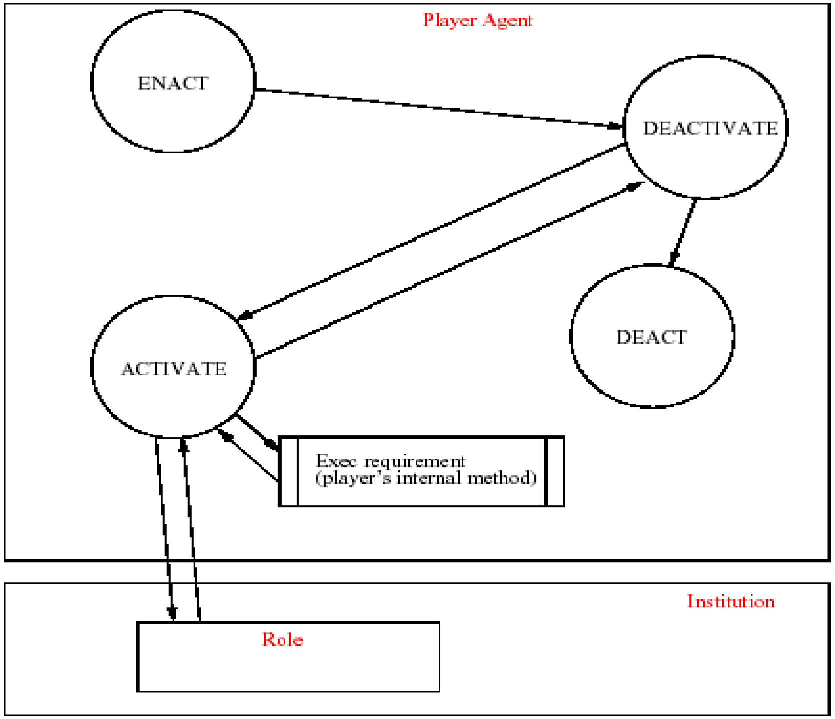
- can reside on different platforms with respect to the organization and its roles;
- need a special behaviour for playing a role.

This class allow to model the states of the role playing (enacted, active, deactivated, deacted), the transitions from one state to the others, and offers the protocols for communicating with the organization and with the role.

… and in JADE (8)

Players (2)

To interact with a role, the *Player* class implements a suitable Final State Machine Behaviour.



… and in JADE (9)

Interaction (1)

All interaction protocols use standard FIPA messages, to enable also non JADE agents to interact with organizations without further changes. The protocols are always split in two part: the side of the initiator and of the one of the responder.

We shortly describe the interactions between:

- Players and an organization
- Players and their roles

… and in JADE (10)

Interaction (2)

Players and an organization (1)

There is an enactment protocol inherited, respectively, as concerns the initiator and the receiver, from the classes *Player* and *Organization* (which is only a responder).

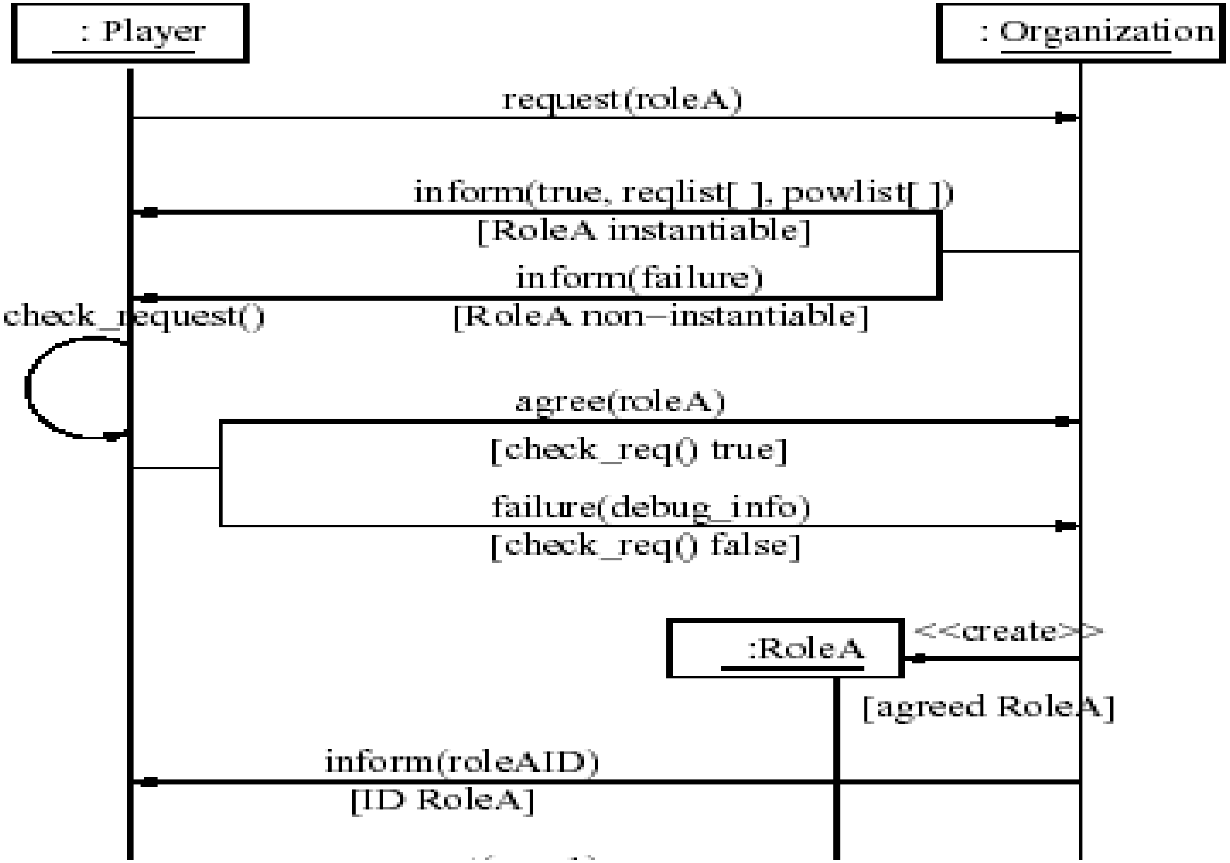
This protocol:

- forwards from the player to the organization the request of enacting a specified role,
- manages the exchange of information:
  - sending the specification of requirements and powers of the roles,
  - checking whether the player complies with the requirements.

… and in JADE (11)

Interaction (3)

Players and an organization (2)



… and in JADE (12)

Interaction (4)

Players and their roles (1)

The interaction between a player and its role is regulated by three protocols:

- the request by the role of executing a requirement,
- the invocation of a power by the player,
- the request of the role to invoke a power.

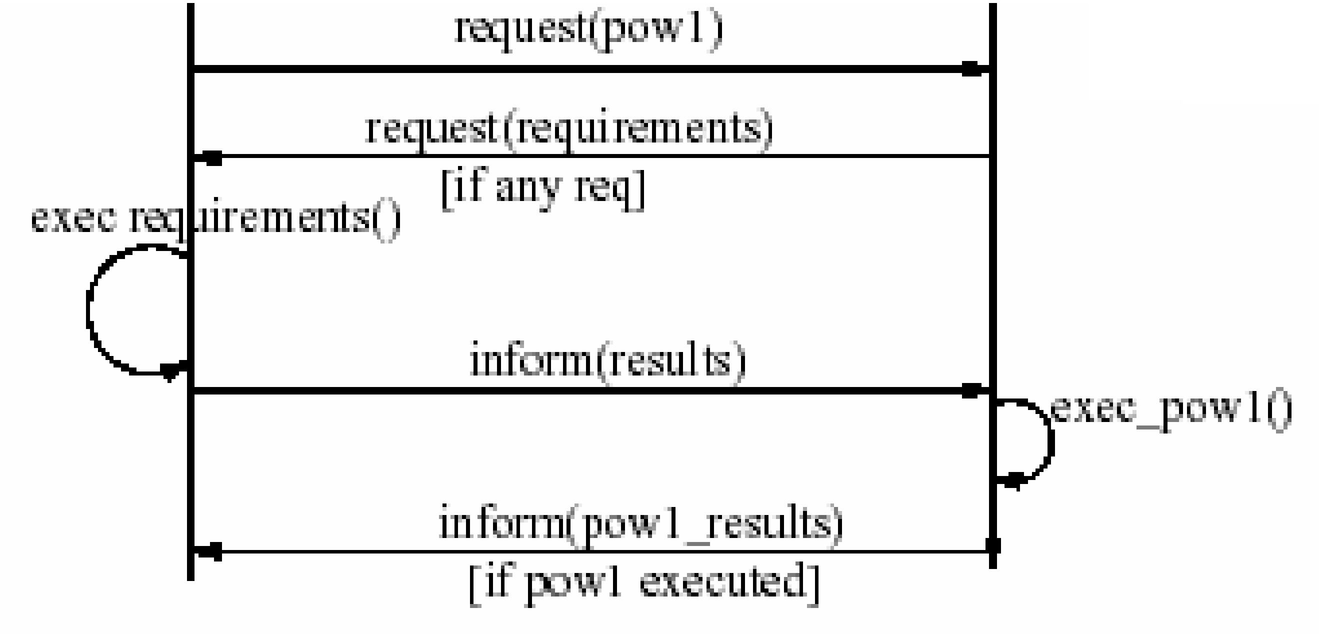
In all cases, the interaction protocol works only between a player and the role instances it plays.

… and in JADE (13)

Interaction (5)

Players and their roles (2)

Here, a player call for a power, and the role instance call for a requirement execution (e.g., an “EvaluateTask” in FIPA CNP)



Conclusions and future works (1)

- While in powerJava *Players* are objects (and they do not have the possibility of not to fulfill a requirement, for example, or not to execute a method → they are not autonomous!), in our framework *Players* are agents *totally autonomous*
- In powerJava all the execution has to be done on the same platform, while with this new solution we can have distributed solutions
- This framework works as a middle tier on which base complex and distributed applications respecting agents autonomy

Conclusions and future works (2)

Our attention is to:

- Improve the normative aspect of the framework using, for example, rule engines as Jess or languages as Jason to process goals and behaviours
- Implement a new web oriented solution for introducing the concept of role on web-services based applications
- Providing a formal model for roles to give a semantics for powerJava and powerJade

Thank you!

grenna@di.unito.it