Agent Roles: From Methodologies to Infrastructures

Mariachiara Puviani, Giacomo Cabri, Letizia Leonardi
Dipartimento di Ingegneria dell’Informazione
Università di Modena e Reggio Emilia
Modena, Italy, 41100
{mariachiara.puviani, giacomo.cabri, letizia.leonardi}@unimore.it

ABSTRACT

Role is an important and useful concept to define the collaboration in agent systems, since it allows to abstract from physical agents and to focus on their behaviour. So, the role concept is exploited in agent methodologies and agents infrastructures, but often in different and discontinue ways. The aim of this paper is to map the role concepts proposed in different agent methodologies, and also mapping other concepts related to role. Moreover, we try to extend this mapping also to agent infrastructures, so to help developers choosing the right methodologies and infrastructures. Providing a comparison (and outlining the contrasts) will be important and useful as a starting point to understand and to try to unify foundational agent concepts.

KEYWORDS: Role, Methodology, Infrastructure, Agent

1. INTRODUCTION

Nowadays roles are very important to build agent systems, like MASs (Multi Agent Systems) or Agent Societies. Agents can be properly developed in a system that provides roles, because of their ability to guide the behaviour of agents in the system itself, in terms of communication rules, constraints and norms that are strictly connected to roles. So, having a role in a society or in a system is essential for an agent to not being isolated or only “external” agents in the system.

When starting his/her work, the developer can be supported by methodologies to easily build the system assigning the right roles to the agents. But the main problem when using an existing methodology is that not all the infrastructures have a mapping of the chosen roles; so all the work done with that methodology turns out to be useless in the next step of the development.

Starting from these considerations, we studied some existing agent methodologies, and some infrastructures that can be useful to build agent systems. We focused on the definition of roles as found in methodologies, and on the way in which infrastructures use roles: the aim is of trying to map the role concept between methodologies and infrastructures.

In this paper, first we present the concept of role in general (Section 2). Then we place this concept in methodologies (Section 3), and in infrastructures (Section 4). At the end we try to map roles from methodologies to infrastructures, giving to developer some guidelines on which methodology is better to use and which infrastructure is better to associate to it; or, in case of a chosen infrastructure, which methodology can help in the role mapping (Section 5 and 6). Section 7 presents some related work.

2. ROLES

Roles are very important in agent systems, where they represent a useful abstraction to describe the behaviour common to (possibly different) entities, in this case agents. They reveal even more useful when defining collaboration models for agents.

According to [18] we can find out two definitions of role in MASs, taken from two different perspectives:

1. Concept: a role is a constraint under which an agent takes part in some interactions and evolves in a certain way. Especially in MAS, an agent behaves under its bound roles.
2. Implementation: a role is an encapsulation of certain attributes and behaviours of the agent it is bound to.

Roles define rights and duties that an agent has in the system, and can be seen as a sort of abstraction that captures different views of agents [17] and can evolve easily and independently from each other.
These definitions allow us to underline some characteristics of agent systems, MASs in particular:

- An agent can play more than one role at the same time;
- Agents can dynamically change their role;
- Agents, not roles, take actions;
- Roles are not isolated but connected to other roles;
- Roles are the way through which other agents can understand how to communicate with the agent that plays the role;
- Roles help reuse.

From these characteristics, we understand the importance of roles, whose features are well outlined in [17]. Often, especially in methodologies, roles are put together to make organizations: this permits to represent how agents can interact and communicate with each other, under their roles. Essentially, an organization of roles can be viewed and described as a super role, which can be broken down into sub-roles.

For the relevance that roles have for methodologies, it is very important to implement them also in infrastructures.

Unfortunately, the definition that a role has in a methodology is not always the same it has in the infrastructure used for the system implementation. Therefore, it can be very important to map role concepts from methodologies to infrastructures, to help developers choosing the tools for system developing.

3. ROLES IN METHODOLOGIES

Several agent methodologies have adopted the concept of role in their phases. We have analysed some interesting and common methodologies, starting from [6] and [7], and in the follow we describe how these methodologies represent roles and, with the help of metamodels, what they are related to.

3.1. GAIA

In GAIA [19], a collection of roles is an organization. All roles are considered atomic constructs, and cannot be defined in terms of other roles, but are then defined in terms of responsibilities, permissions, activities and protocols. Responsibilities define role’s functionalities; permissions are the “rights” which allow the role to
perform its responsibilities; activities are computations that can be executed by roles; and protocols define interactions between roles.

A role gives an agent a well-defined position in the organisation, with an associated set of expected behaviours. In addition, the portion of environment with whom the agent needs to interact in order to accomplish its roles is just determinated by the agent specific role, and by its current status.

A role is an abstract construct used to conceptualise the system, but does not necessarily have any direct realisation within the system.

In GAIA there is also a role model, first created in the Analysis phase and redefined in the Design phase, which defines the topology of the interaction patterns and the control regime of the organizational activities.

Figure 1 represents the GAIA metamodel and underlines the entities connected to the concept of role.

3.2. PASSI

In PASSI (Process for Agent Societies Specification and Implementation) [9], role is a social concept: a collection of tasks performed by agents in following a sub-goal or offering some services to the other members of the society. An agent can play several functional roles during its interactions with other agents to achieve its goals.

PASSI during the first of its five models, System Requirements Model, identifies roles creating a series of sequence diagrams, exploring the responsibilities of each agent through role-specification scenarios. Then, it describes roles in the Agent Society Model where class diagrams are used to show the roles played by agents, the tasks involved, communication capabilities and inter-agent dependencies.

Figure 2 represents the PASSI metamodel.

3.3. SODA

Also SODA (Societies in Open and Distributed Agent spaces) [15] has the concept of roles. In its Architectural Design phase, the main goal is to assign responsibilities for achieving tasks to roles, and responsibilities for providing functions to resources. Then, in order to attain tasks, a role should be able to perform actions.

Figure 3 reports the SODA metamodel, where we can see the concept of role, together with some related entities.
3.4. Tropos

In Tropos [3], the role notion is strictly connected to actors: agent, role, and position are in fact specializations of the concept of actor. A position can cover from 1 to N roles, whereas an agent can play from 0 to N roles and occupy from 0 to N positions. These dependences are introduced when an actor diagram is defined (Early Requirements phase).

In other words, a role is defined as an abstract characterisation of the behaviour of a social actor within some specialised context or domain of endeavour.

Figure 4 represents the Tropos metamodel.

4. ROLES IN INFRASTRUCTURES

Infrastructures become very important when the necessity to build an agent system emerges from concrete application needs. Therefore, a bottom-up approach starts from very concrete aspects such as protocols, languages, runtime supports, etc.

It is very important to have the possibility to use the concept of role also in infrastructures. Not only because this concept turns out to be useful in different phases of the development, but also in order to achieve continuity between the different phases of the development, avoiding fragment solutions difficult to maintain. In the following, we present how some agent-oriented infrastructures represent this concept, supported by infrastructures metamodels. In fact each infrastructure would have a different definition and responsibilities for a role to support collaboration in agent-systems.

4.1. CArtAgO

CArtAgO (Common Artifact for Agent Open environment) [16] first introduces the concept of workplace, which is an organisational layer on top of workspaces. A workspace is an open set of artifacts and agents, as a workplace is the set of roles and organisational rules being in force in a workspace. Roles can be played by agents inside the workplace, and they
may or may not give permissions to agents to use some artifacts or to execute some specific operations on selected artifacts.

In Figure 5 the readers can see the CArtAgO metamodel.

4.2. JADE

When studying JADE, we can say that the concept of behaviour is similar to the one of role, but still not the same. Behaviour is used to model a generic task, and it is specialised in CompositeBehaviour and SimpleBehaviour. Also these concepts are then specialised as readers can see in Figure 6 reporting the JADE metamodel.

4.3. MARS

JADE (Java Agent DEvelopment Framework) [1] does not have directly the concept of role, being more focused on agent, which is the key concept of the architecture.
MARS (Mobile Agent Reactive Spaces) [4], like JADE, does not have the direct concept of role, but we can say that the concept of identity is very similar. The main difference between the two is that the concept of identity represents only agent rights in the system, but not agent duties.

Identity is one of the three components that characterises an access event, along with tuple and operation type. Identity is strictly connected to agent: an agent can play one or more identities.

In Figure 7 readers can see the MARS metamodel.

4.4. RoleX

RoleX (Role eXtension) [12] is based on the concept of role. A role is defined as a set of actions that an agent playing that role can perform to achieve its task, and a set of events that an agent is expected to manage in order to act as requested by the role itself.

RoleX defines descriptors to grant a high level of abstraction in assuming, playing and discarding roles. A role descriptor is an object that describes what the role does, but not how, or with which operations it is done; it also includes the descriptors of the associated actions and events.

Descriptors are useful to hide to the agent programmer, the physical class that implements a role, and to the agent, the physical location of the role implementation.

4.5. TuCSoN

In TuCSoN (Tuple Centers Spread Over Networks) [14], the concept of role derives from the Agent Coordination Context (ACC). ACC works as a model for the agent environment, by describing the environment where an agent can interact. It also enables and rules the interactions between agents and the environment, by defining the space of admissible agent interactions. ACC has first to be negotiated by each agent with the MAS infrastructure, and then the agent specifies which roles to activate: if the agent request is compatible with the current organisation rules, a new ACC is created, configured according to the characteristics of the specified roles, and is released to the agent for active playing inside the organisation.

In a role-based organizational context, the ACC makes it possible to represent the roles that an agent is playing inside an organisation, and then its responsibilities, permissions and interactive behaviours.

Figure 9 represents the TuCSoN metamodel.

5. MAPPING THE ROLE CONCEPT

We started our work from the concepts found out from methodologies and infrastructures. Initially, we have tried to map entities between methodologies, moving down from the concept of role. Then the mapping is extended to
the related concepts because it is impossible to build a role, without looking at other features. However, it is not easy to extend this theory because not all the methodologies have the same entities. We began from the ones with the same name, and then we tried to do the mapping between entities with different names but with the same meaning. This is not always possible; sometimes the concept is similar but not exactly the same.

### Table 1. Mapping Methodologies

<table>
<thead>
<tr>
<th>GAIA</th>
<th>PASSI</th>
<th>SODA</th>
<th>Tropos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>Role</td>
<td>Role</td>
<td>Role</td>
</tr>
<tr>
<td>AgentType</td>
<td>Agent</td>
<td>Agent</td>
<td>Agent</td>
</tr>
<tr>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Goal</td>
</tr>
<tr>
<td>Service</td>
<td>Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Environment</td>
<td>Scenario</td>
<td>Workspace</td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td>Communication</td>
<td>Interaction</td>
<td></td>
</tr>
<tr>
<td>Resource</td>
<td>Resource</td>
<td>Resource</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>AgentAction</td>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>Responsibility</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We can see looking at Table 1, it is not always possible to map all entities, like Position that is present only in the Tropos methodology, or Responsibility that is only in GAIA. We can also notice that the concept of Goal, which is present in all the studied methodologies, does not appear in their metamodel (only Tropos deals with it directly), the other methodologies use the concept of Task as actions to do to achieve goals, so we can map Goal and Task together.

The PASSI concept of Scenario is not so clear if can be mapped on GAIA Environment and SODA Workspace, also if the concepts are not so different, so we have pointed it out for a possible but not sure mapping.

[8] and [13] have tried to map all the entities of a specific methodology (PASSI and SODA) in one or more infrastructures. Starting from these works, we have extended them, trying to map the concept of role and some related ones, in four methodologies, and then from them to five infrastructures. The mapping is reported in Table 2.

We can notice that some concepts present in methodologies, cannot have the correspondence in infrastructures, like Goal/Task that is strictly connected only to methodologies and does not appear in infrastructures.

As we can see from Table 2, a lot of cells are empty because it is not possible to find the mapping concept. As a consequence, we can see that it is very difficult to map infrastructures’ entities in methodologies’ ones. Nevertheless, it is also very important when a mapping is found because it allows connecting methodologies and infrastructures.

### 6. RELATED WORK

In literature, we have not found approach like ours, but there are different ones, that are important for studying the role concept.

Software agents, like objects, processes, organizations, and people, can play roles and can therefore be incorporated into a role model. An important work to this aim, is [11], that presents an overview on agent roles and role models, as a new abstraction for analysing and designing agent systems.

In that paper it is justified why the role modelling is appropriate for intelligent agent systems:

- role models emphasize social or interactive behaviour;
- roles in a role model work together to accomplish their goal;
- role models are patterns that should be documented and shared;
- roles and role models provide an abstraction that can unify diverse aspects of a systems;
- role model synergy integrates roles and may be valuable for agent design;
- role model dynamics can be employed to model mobility, adaptive behaviour, context switching, and other important aspects of agent systems.

That work is very important for trying to incorporate a role model in a methodology or in an infrastructure.

Another important work for agent-based systems is [18], which proposes a method based on roles that will be useful to use role in methodologies and infrastructures, because of the standardisation of the concept of role, which it proposes.

It clarifies the role concept, with its meanings and properties, and proposes a method RoMAS to realize its potential. That paper proposes a role-based modelling language tailored to explicitly separate role from agent conceptually and linguistically; roles exist throughout the whole process of a MAS development.

Other approaches that are less correlated to our work could be [17] and [5].
7. CONCLUSION AND FUTURE WORK

This work has started from methodologies, but it will be really interesting to start from the infrastructures point of view, looking at the concept of role, and at the concepts related to.

From this preliminary work, we can only suggest to agent system developers which are the most useful infrastructures to use, once they have chosen a specific methodology, and the concepts to use to translate the entities used to describe their system. This kind of mapping is interesting to understand the use of the role concept, and to use it in an ideal world. While for real-world applications, it will be interesting to map methodologies and infrastructures, more closely, suggesting which are the best methodologies, and related infrastructures to use for building agent systems, under some specific constraints.

Now we are trying to evaluate our mapping, using a simple case study, a Conference Management Systems, and then we will try to do a specific mapping, under chosen constraints.

Another important aspect is that sometimes, the lack of correspondences in concepts comes from the gap that is present between methodologies and infrastructures. In order to fill this gap, one of the possibilities is exactly to map all the concepts existing in methodologies to the ones in infrastructures, and vice versa. Starting from this work, we will try to fill the entire concept map, which will allow adopting existing methodologies and existing infrastructures for building agent systems, without create just another one methodology or infrastructure.

ACKNOWLEDGEMENTS

Work supported by the Italian MiUR in the frame of the PRIN project “MEnSA - Agent oriented methodologies: engineering of interactions and relationship with the infrastructures”.

REFERENCES


Table 2. Mapping Infrastructures

<table>
<thead>
<tr>
<th>GAIA</th>
<th>PASSI</th>
<th>SODA</th>
<th>Tropos</th>
<th>CAiAGo</th>
<th>JADE</th>
<th>MARS</th>
<th>RoleX</th>
<th>TarCSnN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>Role</td>
<td>Role</td>
<td>Role</td>
<td>Role</td>
<td>Behaviours</td>
<td>Identity</td>
<td>Role</td>
<td>Role Descriptor</td>
</tr>
<tr>
<td>AgentType</td>
<td>Agent</td>
<td>Agent</td>
<td>Agent</td>
<td>Agent</td>
<td>Agent - Agent</td>
<td>Agent</td>
<td>Agent</td>
<td>Agent</td>
</tr>
<tr>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
<td>Task</td>
</tr>
<tr>
<td>Service</td>
<td>Services</td>
<td>Environment</td>
<td>Scenario</td>
<td>Workspace</td>
<td>Workspace - Workspace</td>
<td>Workspace - Workspace</td>
<td>Workspace - Workspace</td>
<td>Workspace - Workspace</td>
</tr>
<tr>
<td>Communication</td>
<td>Communication</td>
<td>Interaction</td>
<td>Interaction</td>
<td>Interaction</td>
<td>Reaction</td>
<td>Reaction</td>
<td>Reaction</td>
<td>Reaction</td>
</tr>
</tbody>
</table>