# RoMAS: A role-based modeling method for multi-agent system

In this paper, we present a new concept of role, clarify its meaning and properties, analyze its significance in MAS (Multi-Agent System) and propose a method RoMAS to realize its potential. Our method supports dynamic binding between role and agent.

## 1. Role in MAS

We want to model MAS based on role. In Gaia, a role is “defined by four attributes: responsibilities, permissions, activities, and protocols.” Thus, it is an implicit definition.

Our insight is that an *entity* (i.e. object in OO or agent in MAS) cannot exist by itself, and it should bind some roles. We propose a definition of role in MAS:

Role:

1. From the conception perspective, a role is a constraint under which an agent takes part in some interactions and evolves in a certain way. In MAS, an agent behaves under its bound roles.
2. From the implementation perspective, a role is an encapsulation of certain attributes and behaviors of the agent it is bound to.

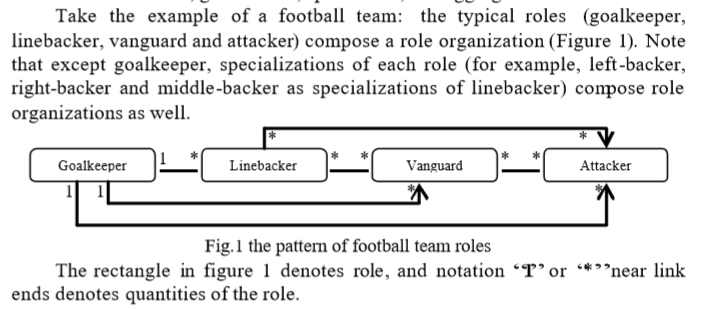
The characteristics of this definition are:

1. An agent can have more than one role at one time;
2. An agent can dynamically change its roles;
3. Role does not take actions while it is the agent that takes actions;
4. Roles are not isolated: there must be other roles related to it;
5. Role acts as a “window” of agent, through which other agents know the way to interact with the agent;
6. Roles provide a facility for efficient reuse.

## 2. Role Organization in MAS

Roles along with the communication paths among them compose a role organization. Role organization is helpful in the following aspects:

1. It represents how agents interact. Each agent acts and communicates under its role or roles.
2. It abstracts interactions in MAS.
3. As a frame in which agents bind certain roles, it enables the agents to change their roles dynamically.
4. Essentially, role organization is an ad hoc role. It can be instantiated, generalized, specialized, and aggregated.

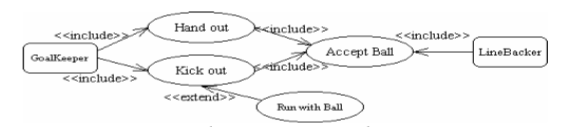


## 3. Role-Based MAS Development

We propose a role-based modeling language tailored to:

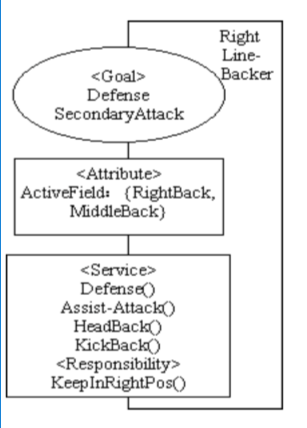
1. explicitly separate role from agent conceptually and linguistically;
2. roles exist throughout the whole process of MAS development. The main development process in natural language is as follows:
3. Captures use cases;
4. Identify roles from use cases;
5. Constructs role organization;
6. For each role, if the appropriate agent does not exist, then go to (e); else
7. Binds roles to agents
8. Describes dynamic properties of bind relation between agents and roles
9. Go to (f)
10. Generates agents according to roles; Go to (d).i.
11. Generates codes for agents with roles bound;

### 3.1 Capture Use Cases

Use cases outline the system events and their interactions.

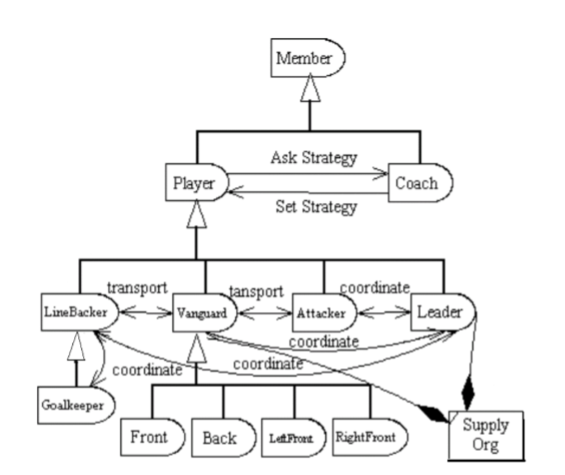
The figure represents a use case (there should be more in real system), in which rectangle denotes Actor and ellipse denotes Use Case. In this example, Goalkeeper actor includes Hand Out use case or Kick Out use case to pass the football; Linebacker actor includes Accept Ball use case to get the football. In some possible conditions, Kick Out use case may be extended by Run With Ball use case.

### 3.2 Identify Roles

Roles can be identified from use cases.

The figure shows an example notation of role, in which the right top text is the role’ s name, the ellipse with text shows the goals the role takes, the middle rectangle with text shows the attributes, the bottom rectangle with text shows the services the role provides and its responsibility.

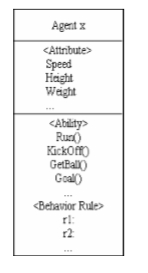
### 3.3 Construct Role Organizations

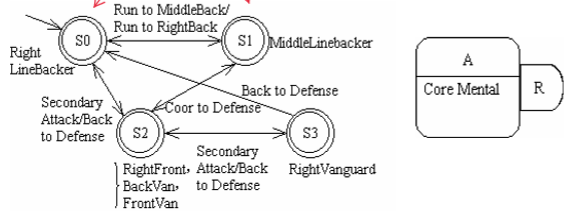
Roles are not isolate. Every role communicates and interacts with other roles. Besides, roles can be specialized or aggregate to other roles. Inheritance and aggregation associations respectively denote specialize/generalize and aggregate/decompose relations among roles.

The figure shows a role organization, in which rectangle with a semi-circle denotes role; arrow line denotes communication path, triangle denotes inheritance relation, diamond denotes aggregation relation, and rectangle with a line on left-top corner denotes organization.

### 3.4 Bind Roles to Agents

For each role, the appropriate agent may exist or not. For existing agents, they are classified to agent classes directly. An agent has a name, attributes, actions and behavior rules.





An agent can change its roles dynamically. To make this property clear, we apply finite automata to describe agent’s role transitions (Figure in the middle). Circle represents state, which denotes the roles bound to the agent. Arrow line denotes state transition, which is executed under the conditions or messages listed near the line. The notation “}” / “{” denote OR / AND relation of the roles indicated on its right-side. For “}”, only one of the roles is bound to the agent; for “}”, all the roles are bound to the agent.

The figure on the right describes initial binding of role to agent. The rectangle is a compact form of agent and the rectangle with semi-circle is a compact form of role.