Roles in the Context of Multiagent Task Relationships

George Thomas
Department of Computer Science
University of Iowa
Iowa City, IA 52242
george-thomas@uiowa.edu

Andrew B. Williams
Department of Computer and Information Sciences
Spelman College
Atlanta, GA 30314
williams@spelman.edu

Abstract

In computer science, there currently exist several definitions of roles. We introduce roles from the perspective of a task relationship defined as a connection between entities via a set of tasks. In our view, a role is a component of a task relationship within the context of a cooperative multiagent system or organization. We describe a model of roles that approaches the meaning of what a role is from a computational perspective in task allocation and treats roles as participants in a task relationship. We utilize notation from linguistics to introduce how roles can influence each other. We argue that the key features of current models of roles can be mapped into our task relationship model, and we illustrate how this approach satisfies most of the desired properties of roles.

Introduction

The notion of a "role" has been used widely for a long time across a broad spectrum of domains that range from Sociology, Organizational Theory and Computer Science to Linguistics and the Thespian Arts. Within Computer Science itself, roles are used in Databases, Programming languages, Ontologies and Multiagent systems. Yet there are a plethora of definitions for a role within the same subfield, let alone across different fields. Roles have been used to describe parts played by an actor, a named pattern of behavior or simply a series of tasks to be executed. We examine the concept of a role from a computational perspective in multiagent systems; primarily, we seek to investigate how the introduction of roles within a multiagent problem can improve the tractability of that problem by exploiting its natural structure.

A simple definition of roles describes them as *sets of related tasks*. The key notion here is the word "related" which indicates that there is some knowledge embedded in the partitioning of a task domain across the constituent roles of an organization. And this knowledge sheds insight on both the domain characteristics, as well as how cooperative agents working within the organization should exploit these characteristics to allocate the tasks efficiently.

We focus on domains that can be represented by a set of tasks to be completed. Our interest is in implementing efficient task allocation mechanisms in such systems. We investigate how introducing the roles into such a system can improve its computational tractability. In brief, the introduction of roles into a task domain provides the following benefits:

1. Roles help to prune the search space for solutions by reducing the allocatable tasks to an agent. If an agent is assigned a role, it knows that certain tasks will not be assigned to it, and these can be deleted from consideration.
2. Roles introduce domain prior knowledge into the problem.
3. Roles can act as a kind of longer term task schedule. Being assigned a role gives an agent a better idea of the tasks that will likely be assigned in the future.

With this motivation, we seek to come up with a model for this purpose. We restrict our notion of roles by choosing to view them as components in a relational descriptor. A role fully specifies a relationship between entities. We argue that viewing a role as a relationship does not limit its power or applicability. Instead, we claim that such a generalized view serves as a superclass or "superview" into which other, more specific, views can be mapped.

This paper is organized as follows. We give an overview of the related work on roles from a Computer Science perspective. We then describe our model and illustrate some of its properties. We intuitively show how some of the main views on roles can be mapped into ours. We close with a discussion on our model.

Related Work

Research in roles spans a wide range of ideas with significant variation between them. (Kinny *et al,* 1992) quotes (Miller *et al*, 1960) as saying "A role is one of those wonderful concepts apparently able to tolerate any number of alternative definitions, so one more should cause no more trouble." This is indeed true as the wide variation of role definitions in the existing literature on roles illustrates. (Tambe, 1997) defines a role as an abstract specification of a set of activities an individual or subteam undertakes in service of the team's overall activity. (Vlassis, 2003) calls a role a masking operator on the action set of an agent given a particular state, i.e. it simply serves to reduce the available actions of an agent at a particular time.

(Zambonelli *et al,* 2003) see a role as an "abstract description of an entity's expected function". A role has the attributes of *responsibility* (its functionality), *permissions* (its rights and access to resources), *activities* (its internal actions) and *protocols* (its formalized interaction with other roles).

(Steimann, 2000) discusses three main view of roles prevailing in the literature: as named places in relationships (the name of a relationship gives a role), as specializations or generalizations (a role is not a type), and as an independent type whose instances are endowed with state and behavior but have no identity. Our notion is very similar to a combination of the first and third descriptions: a role exists within a relationship and has a specific behavior. (Pacheco and Carmo, 2003) view a role from the deontic perspective with a role possessing obligations, permissions and prohibitions. They do not view a role as an agent; instead they see roles and agent properties as directly correlated, with an agent theoretically being able to play a role for every property it possesses.

(Zhu, 2003) focuses on human roles and Computer Supported Cooperative Work and presents a number of definitions for roles. Of particular relevance to our work is the description of a role as the abstraction of an entity's behavior that consists of a subset of the entity's interactions along with a set of constraints on those interactions.

(Odell *et al,* 2003) states that roles are formed when a recurring set of behaviors is found useful in a group context; basically, a reusable set of actions becomes standardized as a role.

(Masolo *et al,* 2004) defines a role as a concept that can be "played" in a contingent, temporary manner when an entity enters into a relationship with other entities. The main features of this viewpoint are that roles are properties, are anti-rigid possessing dynamic qualities, are linked to contexts and are relational in nature. This is very similar to our notion except for the description as a property.

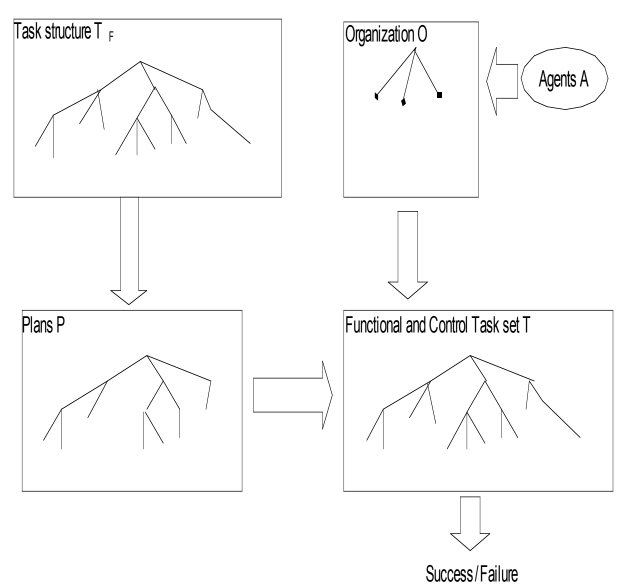
(Boella and Van der Torre, 2004) view roles as abstract agents possessing capabilities, beliefs and goals. These beliefs and goals may be different, and are usually subsets, of the beliefs and goals of the organization the role belongs to. Similarly, these beliefs and goals of a role may be different from the beliefs and goals of an agent who plays the role; but the agent subjugates its own beliefs and goals to those of the role it is playing for as long as it plays the role. An agent takes on the "persona" of its role in hope of being rewarded by its employing organization for fulfillment of the goals of its role, and to avoid being sanctioned for non fulfillment thereof. This reification of a role allows a role to have its own beliefs and goals and to be treated as an abstract entity.

Approach

We first discuss the intuition behind our approach. We argue that many problem domains can be decomposed into a set of tasks and their inter-relationships. We consider such task-oriented domains where the decomposition of the overall objectives into a set of tasks to be achieved is feasible. These tasks may be classified as goals in other terminology. The key to using multiple agents to efficiently solve the overall problem lies in allocating the tasks to these agents in a manner that exploits the existing, and dynamically evolving, structure of the problem domain. Roles are an intuitive way to exploit the problem domain structure. We hypothesize that using roles allows the implicit introduction of domain knowledge into any solution system; in addition, roles constrain the task space of an agent by restricting that agent to a subset of the overall task space; this "intelligent" restriction serves to then improve the computational tractability of the problem. The idea of using roles thus has been used in (Spaan *et al*, 2002) but the roles there constrained a set of actions; we apply the roles to a set of tasks. A further advantage we gain is that role assignment helps to serve as a predictor of future tasks to be executed, allowing reasoning for long term task allocation.

Our approach takes an organizational view of roles with a role as the primary building block in an organization. An organization is structurally a hierarchy of roles, but semantically, is a mapping from a group of heterogeneous agents onto a problem domain composed of tasks and task interrelationships. Our key thesis here is that the design of an organization or team to solve a problem domain *cannot* be separated from the problem domain it is designed to solve *or* the agents that will be available to solve that problem.

Figure 1



Consider a problem domain characterized by an environment E, with a set of tasks T_F. The tasks have a hierarchy and interrelationships. Let P be a subset of the tasks with a particular ordering among the tasks and let the execution of P result in the solution of the problem; i.e. P is a particular plan to solve the problem. We can further relax this and say that P is a class of plans (with related characteristics) to solve the problem. Let A be the set of agents available to work on the problem. Then an organization O is a mapping from A onto an augmented task set T. T is the union of the functional task set T_F along with a set of additional control tasks T_C required to ensure coherence of the organization. The organization O is composed of a set of roles. This process is represented in Figure 1.

The organization introduces an additional set of control tasks but this very introduction allows the organization to efficiently execute the functional tasks, and is thus an acceptable overhead. The idea of control and functional specialization in roles was covered by (Tidhar *et al,* 1998) and we extend this idea to tasks.

The idea we present here is that an effective organization cannot be separated from the structure of the problem it is trying to solve or the agents available to solve that problem. A well designed organization will have a structure that considers both of these components. A "good" organization will thus restrict the actions of agents to stay within the actions space that achieves the goals of the problem. The closer the agents are restricted to the ideal of *exactly* those actions that satisfy the overall goals, the better the organization.

Formal Model

We present a preliminary formalization of these concepts and expand on the meanings of some of our terms here.

An *entity* is a participant in a system that can exert influence or be influenced in that system. A system refers to the problem domain under consideration.

An *environment*, e, describes the conditions that are prevailing in the problem domain. The set of agents in the system is described by the set $\mathcal{A}$. The set of entities, $\mathcal{S}$, in the system is described by the set $\{\mathcal{A} \cup \{e\}\}$.

A *task* is a set of actions, executed in sequence, that achieve something positive in a problem environment. Every task includes a set of properties that fully specify it. These properties can include precedences, dependencies, duration, etc.

A *problem description* $\mathcal{D}$ is a tuple $<e, \mathcal{T}_F>$ where $\mathcal{T}_F$ is a set of tasks organized in a hierarchy, with interrelationships between these tasks. These interrelationships can be constraints such as sequential or parallel constraints, etc. Execution of these tasks in the environment, in accordance with their constraints leads to a solution to the problem.

Informally, a role is now an *association* between a set of tasks and a set of entities. This association is basically the formalization of a relationship. We can capture this more formally by defining a *role-task-relationship*.

A role-task-relationship (RTR) can be defined as a tuple $<a_r, \mathcal{T}_r, \mathcal{P}_r>$ where a_r is a placeholder called the *active role*, $\mathcal{T}_r$ the set of tasks role a_r is restricted to performing, and $\mathcal{P}_r$ is the set of *passive roles* – roles that are affected or influenced in some way by the execution of any of the tasks in $\mathcal{T}_r$. This representation is borrowed from linguistics in a description of the form:

```
<subject><verb><object>
```

where the active role is the subject or executor of the verb(task set) and the passive roles are the direct, or indirect, objects of the actions. Hence, this becomes:

```
<active-role><task-set>[<passive-role>]
```

A role, therefore, is a participant in a RTR and can be either active or passive. We view roles always within the context of a set of tasks; the association between the role "players" is via a set of tasks, and this is the multiagent task relationship.

An agent playing an active role is restricted to the task sets described in the RTR. We restrict the selection of an entity for an active role to be from $\mathcal{A}$ but the passive roles are selected from $\mathcal{S}$. We choose to model the environment as an entity eligible to play a passive role to depict the fact that the execution of a task, while it may not directly, or indirectly affect other agents, can alter the environment. Both active and passive roles serve as place holders for actual entities to be bound to and freed from them during actual execution.

As stated earlier, an organization is now a hierarchy of roles, with links or dependencies between these roles. An organization structure introduces a further set of tasks that deal with decision making and flow of information within the organization. We call these control tasks, represented by the set $\mathcal{T}_C$. The set of tasks over which a role is defined is taken from the union of the functional tasks (from the problem description) and the control tasks $\mathcal{T}_C$. Therefore

$$\mathcal{T} = \{\mathcal{T}_F \cup \mathcal{T}_C\}.$$

Let us illustrate our model by an example. Consider an example of a supervisor. This can be described as follows:

```
<supervisor><{supervises}>[<supervisee>]
```

Here the agent playing a supervisor role is restricted to the task of supervises. A supervisee is influenced by the execution of this task.

Another example is the robotic soccer environment where we assume there are the following tasks:

t_1= kick ball at opponent's goal
t_2= pass to a midfielder
t_3= stay in position to receive a pass

Then we can define an *attacker* role as:

$$<attacker><\{t_1,t_2,t_3\}>[<env>,<pass\text{-}receiver>]$$

The env plays a passive role for task t_1 and midfielder plays the passive role of pass-receiver for task t_2. There is no passive role for task t_3.

Properties

Roles have properties such as responsibilities, which can be considered their task sets; permissions or access rights to the various resources or resource constraints under which they must operate; and obligations whereby their commitments can be modeled. This can allow a role to be initialized with starting beliefs, and agents playing that role must operate under these beliefs. The duration and life cycle of a role can also be modeled here.

(Hubner *et al,* 2002; Odell *et al,* 2004; Pacheco and Carmo, 2003; Ferber, 1999) introduce different types of interrelationships between roles. Basically, all the interrelationships that are possible at the task level are possible at the role level too. Our model allows incorporation of these features.

Our model must support dependencies between roles, such as implication whereby if an agent holds role *x*, it also holds role *y*; and incompatibility whereby holding role *x* implies that role *y* cannot be held (Pacheco and Carmo, 2003). It also supports a hierarchy of roles whereby roles can be subdivided into sub roles that inherit many of the parent properties, or are specializations of the parent role. If roles are identical, we can group them under a common role and give a cardinality to that role.

Roles can have interrelationships between them. These can be AND constraints where role *x* and role *y* have to both be executed for successful goal achievement. We can subdivide AND constraints into those cases where role *x* has to be executed first and then role *y* next (sequence constraints); or role *x* and role *y* have to be executed simultaneously (parallel execution constraints); or the case where order is unimportant as long as both are executed. OR constraints, where either role *x* or role *y* or both can be executed, and XOR constraints, where exactly one of role *x* or *y* should be executed but not both, are also possible.

Roles can have knowledge of other roles (Hubner *et al,* 2002; Ferber, 1999). A role can model another role in its reasoning, can communicate with, or exercise authority over other roles. In addition, an agent executing a role can be active or inactive in that role. Inactivity in a role can be seen as a suspension (Odell *et al*, 2004). The idea of suspension and resumption, classification and reclassification (role allocation and reallocation) can be seen as modeling the temporal dynamics of the role and granting flexibility to the organization.

This discussion raises the question of automated role discovery that is outside the scope of this paper but we state some of the issues involved in answering this question here.

Roles involve relationships. How do we discover these relationships in a problem domain? Is it based on the hierarchical task decomposition of a domain, or the degree of independence between sub-task sets within a hierarchy, or the structure imposed by precedence ordering between tasks? How do we introduce the capabilities of available agents into this consideration?

Discussion and Justifications

We argue that our approach captures the essential features of roles. In our opinion, these are:

1. A relationship of some kind is involved; we argue that this is primarily some kind of a task relationship.
2. This relationship must be viewed in the context of a group or organization, even if only two agents are involved. Roles outside an organization or group structure are meaningless.
3. Roles are dynamic, temporary bindings.

Comparison with Other Approaches

We argue that we can model most of the other approaches with this view. In this section, we contrast our model with the three main views of roles.

The Database Entity-Relation model views roles as named places in relationships, with dynamic classification possible. This fits into our model with a set of roles and a task relationship. Our model views roles as more than just named places, but sees a role as possessing intrinsic properties.

The view of roles as an abstract behavior specification with responsibilities, permissions, activities and protocols can be modeled again with a role's tasks and properties. While we do not view a role as an agent, we view them as more than just properties as is captured in the deontic notion of roles in (Pacheco and Carmo, 2003).

The view of roles as independent types with state and behavior but no identity is the most similar to our model. We differ from (Boella and Van der Torre, 2004) who view a role as an agent with its own beliefs, but we do view them as instances to be adjoined with the entities playing the roles. We argue that the view of a role as an agent can be reduced to this. We see this as the Java idea of an *interface*. The convention is to have all interfaces end with the suffix *–able*, the idea being that it captures a behavior, but is not a type by itself; yet every object being referred to by that interface exhibits the seeming beliefs and goals of that interface for as long as it is referred to by that interface name. This also allows an agent to play multiple roles simultaneously and whenever it is referred to with a role, it is assumed to have only the properties and behavior expected from that role.

Analysis of Characteristics

We briefly examine our model through the grid of role characteristics presented in (Steimann, 2000).

1. Our roles have their own properties and behavior.
2. They depend explicitly on relationships.
3. An agent may play different roles simultaneously.
4. An agent may play the same role several times, but not simultaneously as that does not make sense in our view.
5. An agent may acquire and abandon roles dynamically within the context of the overall organization. The sequence in which the roles are abandoned or acquired is also subject to the restrictions imposed by the organization.
6. As long as an agent has the minimal capabilities required for a role, it can play it. This means that heterogeneous agents can play the same role.

7. We do not allow roles to play roles as this does not make sense in our view. This position seems to arise when a role is viewed as an agent itself. We allow the specialization of roles into sub-roles but this is not the same.
8. Roles can be transferred from one object to another via a process of relinquishing and acquiring but not direct transfer.
9. The state and features of an agent are role specific when it is addressed from the perspective of that role. A role will hide or restrict access to features that are not relevant to it.
10. Different roles do not share identity if they are not related via specialization.

Conclusion

We present a model that approaches the meaning of roles from a task relationship perspective within the context of an organization. We argue that every relationship is centered on some shared set of actions or tasks; the exact relationship can be a variety of different interactions among those tasks but the central notion of a set of tasks remains. We can then define roles as the subjects or objects in such task relationships. These relationships are situated within an organization that, we argue, must be cognizant not just of the task structure and environment properties of the problem domain it was designed to function in, but also of the capabilities, number and properties of the agents that are available to work as its members. Both the number of agents and the environment and task conditions may be dynamic, and the organization must adapt accordingly. Roles provide stability for tractable robust computation and also accord flexibility to handle dynamic conditions.

We intend to formalize our model and to empirically demonstrate how it can be used to improve tractability in task oriented domains, primarily for task allocation among cooperating agents. Automated discovery of roles and efficient methods for allocations of these roles are also future areas of research. Developing models to capture and analyze the various interactions and dependencies in the tasks of a problem domain will also be required.

References

Boella, G., Van der Torre, L.W.N. 2003. Attributing mental attitudes to normative systems. AAMAS: 942-943

Ferber, J. 1999. *Multi-Agent Systems*. Addison Wesley.

Hübner, J.F., Sichman, J.S., and Boissier, O. 2002. A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. SBIA: 118-128

Kinny, D., Ljungberg, M., Rao, A., Sonenberg, E., Tidhar, G.and Werner, E. 1992. Planned Team Activity, LNAI 830.

Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., and Guarino, N. 2004. Social Roles and their Descriptions. KR: 267-277

Miller, G.A., Galanter, E., and Pribman. K.H. 1960. *Plans and the Structure of Behaviour*. Holt, Rinehart and Winston.

Odell, J., Van Dyke Parunak, H., and Fleischer, M. 2003. The Role of Roles in Designing Effective Agent Organizations. Software Engineering for Large-Scale Multi-Agent Systems, Garcia, A.; Lucena, C.; Zambonelli, F.; Omicini, A.; Castro, J. (eds.), LNCS 2603, Springer, Berlin, pp 27-38.

Odell, J., Van Dyke Parunak, H., Breuckner, S., and Fleischer, M. 2004. Temporal Aspects of Dynamic Role Assignment. Agent-Oriented Software Engineering (AOSE) IV, P. Giorgini, Jörg Müller, James Odell, eds., LNCS 2935, Springer, Berlin.

Pacheco, O and Carmo, J. 2003. A Role Based Model for the Normative Specification of Organized Collective Agency and Agents Interaction. J. Autonomous Agents and Multi-Agent Systems 6(2): 145-184.

Spaan, M. T. J, Vlassis, N., and Groen, F. C. A. 2002. High level coordination of agents based on multiagent Markov decision processes with roles. In A. Saffiotti, editor, IROS'02 Workshop on Cooperative Robotics, Lausanne, Switzerland.

Steimann, F. 2000. On the representation of roles in object-oriented and conceptual modelling. Data Knowl. Eng. 35(1): 83-106

Tambe, M. 1997 Towards Flexible Teamwork Journal of Artificial Intelligence Research, Volume 7, Pages 83-124

Tidhar, G., Heinze, C. and Selvestrel, M. 1998. Flying together: modeling air mission teams, Applied Intelligence, Volume 8, Number 3.

Vlassis, N. 2003. A concise introduction to multiagent systems and distributed AI. Informatics Institute, University of Amsterdam.

Zambonelli, F., Jennings, N.R., and Wooldridge, M. 2003. Developing multiagent systems: The Gaia methodology. ACM Trans. Softw. Eng. Methodol. 12(3): 317-370

Zhu, H. 2003. A Role Agent Model for Collaborative Systems. IKE: 438-444