LIMITATIONS IN AUML'S ROLES SPECIFICATION

Jean-Luc Koning & Ivan Romero Hernández
Institut National Polytechnique de Grenoble, LCIS Research Laboratory
50 rue Laffemas, BP 54, 26902 Valence cedex 9, France
{Jean-Luc.Koning,Ivan.Romero}@esisar.inpg.fr

Abstract Roles have gained a fair amount of attention from researchers in the multiagent system domain, given its recurrent appearance on most application examples using an agent-oriented approach. This attention is understandable, because the role an agent takes within any given system defines every one of its actions, i.e., what it thinks and what it says.

The Agent-UML specification language presents a notion of Role that could be related to previous works such as actors and objects. However, AUML gives roles a totally different, more agent-oriented approach, by considering that roles are a *dynamic property* of the entities conforming the system (agents).

This paper focuses on the limitations of the current AUML specifications and its related implications on dynamic roles.

1. A Pragmatic Definition of a Role

Roles have gained a fair amount of attention from researchers in the multiagent system (MAS) domain, given its recurrent appearance on most application examples using an agent-oriented approach. This attention is understandable, because the role an agent takes within any given system defines every one of its actions, i.e., what it thinks and what it says.

The notion of role affects many spheres of the agent activity, and not only one specific aspect. In the internal sphere it affects the agent's goals, desires and intentions —using a BDI terminology [5]. While it is evident that a complete notion of role needs to include those more abstract notions to be considered as such, it is impossible right now (given the state of the art), to give a thorough definition that encompasses all the possible dependency relations between roles and all the subtle aspects of agency. For this reason, we are going to choose a pragmatic approach: to give a definition of our own, that follows closely what it is interesting within our research work.

For our interests, **a role is a finite stereotypical behavior an agent takes through its existence within a MAS system.**

By *stereotypical* we mean a *repetitive, more or less predictable* sequence of messages and processing that happens when some condition is met, and by *finite* we mean it doesn't matter how complex the agent gets, there is always a condition known as *final state*. We do not focus on the actual mechanisms that specify how an agent reaches the beginning of any of these stereotypical behaviors, because these are directly linked to the cognitive sphere of the agent, which is well beyond a mere observationally descriptive approach as this one.

We are however, interested in (1) proving that once the system makes the decision of acquiring a stereotypical response, this chosen behavior eventually reaches its final state, and if not, (2) in detecting how this condition come to be in order to prevent it.

2. Roles are Everywhere

As mentioned above, the notion of role has, indeed, gained a central *role* in some recent publications related to the agent oriented software engineering domain [4]. In the particular case of the AUML proposal, roles are everywhere [6].

AUML presents a notion of Role that could be related to previous works. It is already present on other domains of research, in particular that of *concurrent object actors* [1] [2] and within the more usual object oriented (OO) approach. In the actor approach, the system is defined by modular subdivision and detailed specification of any possible message sequence that any given entity could produce through time. This follows a revisited analogy to the real-world theater piece, where every possible dialog is perfectly specified to every possible actor. In the case of monolithic OO systems, a notion of role comes along with that of *class*, which allow a slightly role-like specification of the system in the form of public methods and encapsulation, but not necessarily enforcing a well defined sequence of message (method) exchange.

However, AUML gives roles a totally different, more agent-oriented approach, by considering that roles are a *dynamic property* of the entities conforming the system (agents).

By *dynamic property* we mean a transient, time dependent attribute related to an agent. In that view, an agent could decide to take or leave a role depending on its mental state, even allowing agents to take or leave a role in the middle of an ongoing interaction process.

3. Limitations of the Current AUML Specification

As far as we are concerned, this possibility gives birth to some very particular problems unlike those already present on the distributed systems or concurrent objects domain, and likely unseen anywhere else. We have essentially identified four problems.

Role dynamics awareness. If we have systems where an agent could choose to take another role, every other agent already involved inside an interaction process with the changing agent, should be aware of that change, before pursuing the normal flow of the interaction and any notation pretending to represent this should be explicit enough to represent it. There are some conceivable automatic delegation schemes applicable to particular application examples [3], that could eventually allow an agent to leave its role without affecting the overall function of the rest of the system, but in most cases we believe an explicit notification process will be indispensable.

Interaction integrity. Once all the agents know a role change has happened, there must be a mechanism ensuring all the necessary roles are *properly* taken within the system, and if roles are or are not *vital* or *indispensable* it would be good to have a explicit way to know it. It is easy to see that any improperly made role-change could severely affect the overall functioning of the system. On the real world, people within an organization usually do not leave their assigned functions without delegating them first, if those functions are relevant enough to the overall functioning of the system. It is also important to note that not all roles are as important and others. For example, within an auction market scenario, it could happen that auctioneers and vendors could desist whenever they wish, even exchange places, but auction house managers can not do so.

Mental state consistency. When roles are *relevant* to the system functioning, it could be necessary to ensure that the mental state of any possible delegate agent taking over this role is compatible and consistent with the original. This is a necessary condition if we want to assure that the delegate is able to take over the function left by one of their peers.

Location dependency. In some cases agents are located within an environment, performing a very specific task related to its abstract or real location. These agents could take another role, but not leave a particular vital one which justifies their very existence.

4. Implications of AUML Dynamic Roles

We believe that the dynamic nature of roles, as defined by the AUML proposal, has some interesting implications to the kind of multiagent systems AUML is capable of modeling. If we accept the fact that roles could change over time, according to the agent's mental state, we are also implying that there are mechanisms embedded within the final system to handle this role change.

Strangely enough there is very few, if any, publications on these issues. At this point, we essentially suggest two possible directions.

An explicit role-change notification mechanism made implicit. There should be a way for the MAS as a whole to know when an agent has changed its role within the system, in order to assure a minimal functioning.

Usually in most MAS it is assumed that an agent knows which role it has and that it is able to detect which role their peers have too, either guessing by the sequence of messages they provide or through a explicit request. But most importantly, once an interaction process begins, it is executed until its end, without sudden roles changes. So we could insure that agent's mental state is consistent at the end of the interaction.

If we accept that an agent's role (or roles) could change over time according to the agent's mental state, even within an ongoing interaction process, we are implying that there is an implicit mechanism to notify the others that a change took place. It is either that or an advanced delegation mechanism.

An advanced delegation mechanism. If agents could change role transparently without explicitly notifying to their peers every time they change their role, there must be a delegation mechanism taking the responsibility for them. This is in order to insure the continuity of the system function.

Those mechanisms *know* which roles belong to which agents and are capable of locating an agent capable of taking over the function given away by a third one. Such mechanisms *do exist* in some experimental multiagent systems [3], but are relatively far from the usual approach taken by most MAS platforms; where agents have well defined communication channels. They must find out by themselves if there is another agent capable of satisfying their needs; for example, through a ContractNet interaction protocol.

These advanced delegation systems have another severe limitation: mental state consistency is hard to satisfy. It restricts the kind of roles that could be delegated to a small subset of all the imaginable scenarios.

Besides, such delegation schema only works for a rather limited number of settings,i.e., in case there is no strong dependance between the internal variables (agent's mental state), like for instance in middle-ware agents or proxys.

5. Mental State Consistency and Communication Integrity

Left apart the fact that AUML seems to already suggest some kind of infrastructure within the modeled system —which is against the intended purpose of platform independence behind such a notation—, there still is the problem of mental state consistency and communication integrity on dynamic role-change scenarios.

If we use a simplified internal versus external approach to the agent domain, it is not surprising to see in every MAS example available, that an agent's mental state is closely related to its history of interactions, and that this mental state is also decisive on the course those interactions will take from the present. The very notion of agent is based on this idea that the capacity of agents learning from past experiences (through interaction with the environment or with other agents) and the ability to choose what to do next according to this experience.

For specialists interested in multiagent system modeling, there are many methodological approaches that are more or less satisfying depending on their particular interests. For people interested in interaction, most agent oriented systems could be modeled in a manner analogous to most communicating systems, i.e., using a model checking approach (EFSM, Boochy automatas, Petri nets, CCS, Pi-calculus) and validated through model checkers, or a fully logics approach validated through theorem proving systems.

References

[1] G. Agha and P. V. Thati. Actors: A model for reasoning about open distributed systems. In *Formal methods for distributed processing: An object oriented approach*, chapter 8. Cambridge University Press, 1986.

[2] Jean-Pierre Briot. Actalk: A testbed for classifying and designing actor languages in the smalltalk-80 environment. In S. Cook, editor, *ECOOP-89*, pages 109–129, Nottingham, 1989. Cambridge University Press.

[3] Denis Jouvin and Salima Hassas. Dynamic multiagent architecture using conversational role delegation. In *International Workshop on Agent Oriented Software Engineering (AOSE-2003)*, LNCS, Melbourne, Australia, July 2003.

[4] Elizabeth A. Kendall. Agent roles and aspects. In *ECOOP Workshop*, page 440, 1998.

[5] D. Kinny, A. Rao, and M. Georgeff. A methodology and modeling technique for systems of bdi agents. In W. Van de Velde and J. Perram, editors, *Agents Breaking Away: 7th Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-96)*, volume LNAI 1038, pages 56–71, Eindhoven, The Netherlands, 1996. Springer-Verlag.

[6] James Odell, Harry Van Dyke Parunak, and Mitch Fleischer. The role of roles in designing effective agent organizations. In Alessandro Garcia, C. Lucena, F. Zambonelli, A. Omicini, and J. Castro, editors, *Software engineering for large-scale multiagent systems*, volume 2603 of *Lecture Notes on Computer Science*, pages 22–28. Springer, Berlin, 2003.