# The $\mathcal{RICA}$ metamodel: an organizational stance on agent communication languages

Juan Manuel Serrano, Sascha Ossowski, and Sergio Saugar

Artificial Intelligence Group
Department of Computing, ESCET
University Rey Juan Carlos
{JuanManuel.Serrano,Sascha.Ossowski,Sergio.Saugar}@urjc.es
http://www.ia.escet.urjc.es

## 1 Introduction

Several Multi-Agent System (MAS) metamodels have been proposed recently after an effort to precisely identify the major abstractions and relationships underpinning this new kind of programming paradigm. Most of them strongly rely on an organisational metaphor, whereby concepts such as roles, interactions, groups, norms, institutions, and others, play an essential role in defining the architecture of a MAS. Besides organizational abstractions, agent communication language pragmatics, namely speech acts, is an equally stable feature of MAS since their inception. However, few metamodels makes explicit reference to communicative actions, so that the interplay between organisational and communicative features is frequently implicit and left unexplained.

This position paper will summarise the Role/Interaction/Communicative Action metamodel ($\mathcal{RICA}$ for short)[1][2][3], a conceptual language which attempts to bridge the gap between Agent Communication Languages (ACL) and Organisational Models. The organizational stance on ACLs put forward by the $\mathcal{RICA}$ metamodel has also been developed with the aim of devising a principled approach to the reuse of catalogues of communicative actions and interactions protocols, such as those standardised by FIPA. In the following sections, we will first describe the structure of the $\mathcal{RICA}$ metamodel. Next, an example in the well-known conference management domain will illustrate its major features. Last, the major advantages of the approach will be summarised.

## 2 The $\mathcal{RICA}$ metamodel

According to the goals of this research, the novelty of the $\mathcal{RICA}$ metamodel does not show itself on the treatment of high-level organizational abstractions, such as groups, organizations or institutions, but on the proposed perspective on social interactions. Environmental features are not considered either. Figure 1 shows the $\mathcal{RICA}$ metamodel represented in terms of a UML class diagram. This version of the metamodel extends previous ones with the notions of *scenes*, *institutions* and *norms*, borrowing these terms from e-institution frameworks [4]. The whole metamodel includes additional constraints to the UML class diagram, together with the recursive definition model and its execution semantics. The proposed relationships of the UML diagram specify the features involved in the definition of the different types of entities. Definition completeness is not claimed either: e.g. essential features in the definition of role types such as their responsibilities, permissions or rights [5], are not considered.

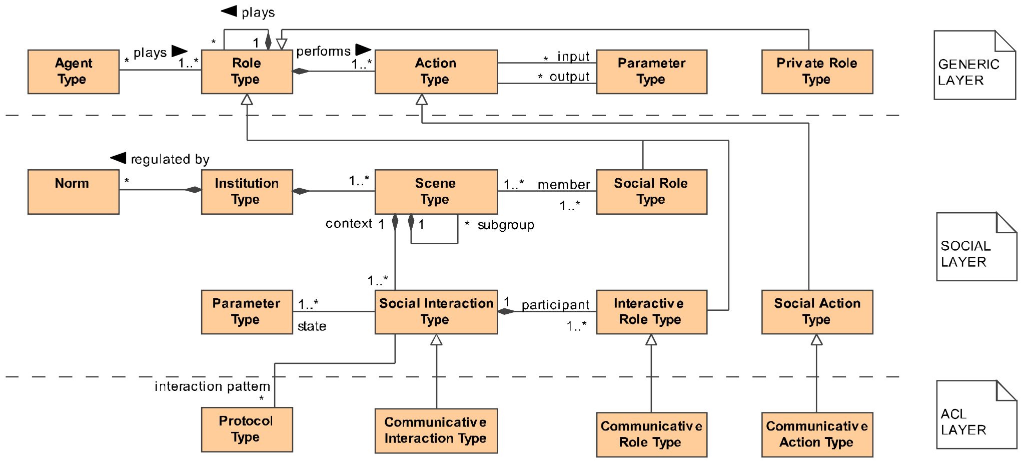

**Fig. 1.** The $\mathcal{RICA}$ metamodel

The $\mathcal{RICA}$ metamodel may be partitioned into three major layers:

- The first one includes agent types, *generic* role and action types, and non-social or private roles: roles played by agents which only require the interaction with the environment. The *subsume* relationship allows to decompose the functionality of some role into other subroles. Actions are regarded as the minimal units of behaviour in which the functionality defined by some role can be decomposed (similar to PASSI *tasks* [12]).
- The second one deals with genuine social concepts such as institutions, norms, scenes, social roles and interactions. Scenes are regarded as meeting points which allow to classify the different interactions which take place in the organization. The definition of interaction types require the identification of the kinds of roles which must be played by their participants. These roles, named *interactive roles*, are distinguished from common *social roles*, which are attached to the different scenes of the institution[1]. At this point, it may help to think of social interactions as particular kinds of *connectors* [6], which mediate the interactions among particular kinds of software *com-*

[1] Social roles must subsume some interactive role within the scenes to which they belong. Interactive roles, on the contrary, can not subsume any further role.

*ponent*: agents. In this way, interactive roles are similar to the roles defined by generic connectors: e.g. the *source* and *sink* roles of a pipe.

- The third one precisely establishes the interplay among communicative abstractions, namely communicative actions and interaction protocols, with the organizational ones introduced above. This is achieved by considering these communicative abstractions from an organizational perspective, in such a way, that communicative actions and protocols are structured around so-called *communicative roles* and *interactions*. These abstractions are particular kinds of interactive roles and social interactions, identified from agent communication dialects, i.e. from specific sets of communicative actions and protocols. In this way, the pragmatic competence required by generic interactive roles (i.e. the kind of communicative actions that an agent will perform in the context of some interaction) is inherited from the communicative roles that they must *specialise*. Similarly, the possible protocols regulating social interactions are inherited from the protocols attached to communicative interactions. Typically, communicative actions are generic, application-independent, abstractions. Hence, communicative interactions may be regarded as generic reusable connectors for agent-based systems. The full specification of CAs would require adherence to some semantic paradigm: social or intentional. Neither is incompatible with the $\mathcal{RICA}$ metamodel. Similarly, different protocol formalisms may be accommodated in the whole picture, which provides their organizational context (particularly, the kinds of roles and actions, state parameters, etc., that the protocol specification will rely upon).

## 3 An example: a Conference Management System

We will use a conference management application for illustration purposes. Figure 2 shows a simplified version of its $\mathcal{RICA}$ institutional model. Institutional agents may play four major social roles: author, reviewer, program committee member and program chair. Interactions among these types of agents are structured around two major scene types: those jointly performed by authors and PC chairs at the submission meeting room, and those happening among reviewers and PC members concerning the reviewing process. Submission of papers occur at the first one. Other possible interaction type at this scene may correspond to the requesting of the camera-ready copy. Concerning the second scene, it includes the interactions concerning the review of a given paper. These interactions would be preceded by interactions of another type aiming to elicit the competence of the reviewers[2]. Other interaction types may be considered as well: e.g. that among the PC members and the PC chair to select the PC members which are responsible for the review process of each paper.

[2] The order in which interactions occur would be regulated by the activation conditions of the different roles. Alternatively, it may be specified by a macro-protocol such as the performative structure [4].

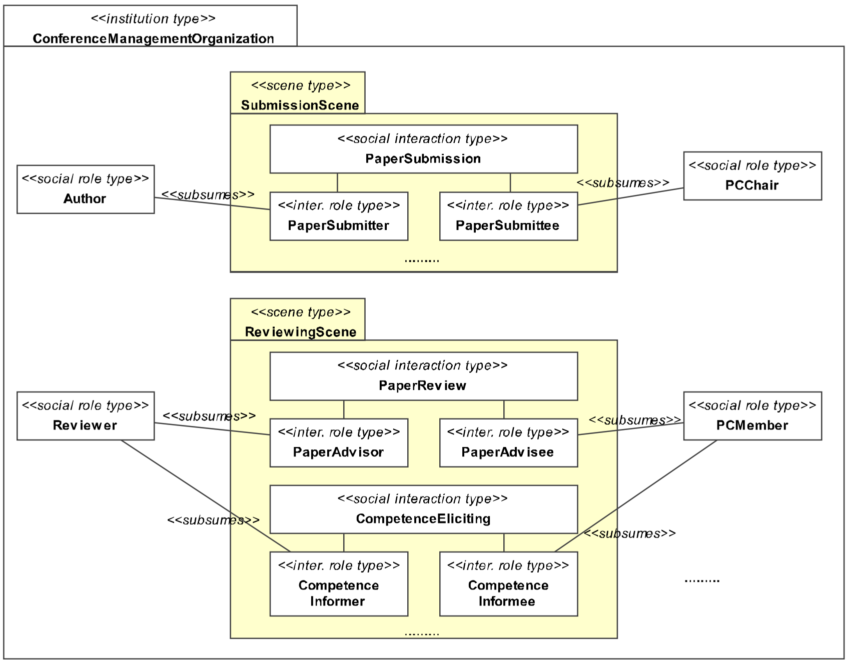

**Fig. 2.** Simple institutional model of a conference management system

This general picture of the conference organization shares many features with other analysis found in the literature for this application domain [7][5]. The major differences are found in the analysis of the different interactions. For instance, the behaviour of authors within the submission of a paper must conform to the expectations and characteristics specified by the *paper submitter* role. As figure 3 shows, an agent playing this type of role may *submit* a paper to the PC chairs. Moreover, they are also allowed to *cancel* its previous submission. Concerning PC chairs, they will evaluate if the paper *satisfies the submission constraints*. If so happens, they may express to its authors their *agreement* to *notify* them about the evaluation result. The whole interaction may be regulated by a simple *request* protocol, since submissions may be modelled as requests to be notified about the evaluation of some proposal.

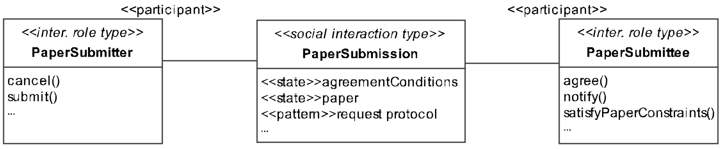

**Fig. 3.** Paper submission interaction model

Figure 3 actually showed the resulting model obtained from the recursive definition shown in figure 4. As can be observed, most of the paper submission model characteristics are actually inherited from application-independent communicative interactions: the non-standard *submission* interaction, defined from the *submit* CA and others, and the *action performing* interaction underlying FIPA CAL. Moreover, the only non-inherited component, i.e. the social action *satisfyPaperConstraints*, overrides a generic social action defined by the *requestee* communicative role.

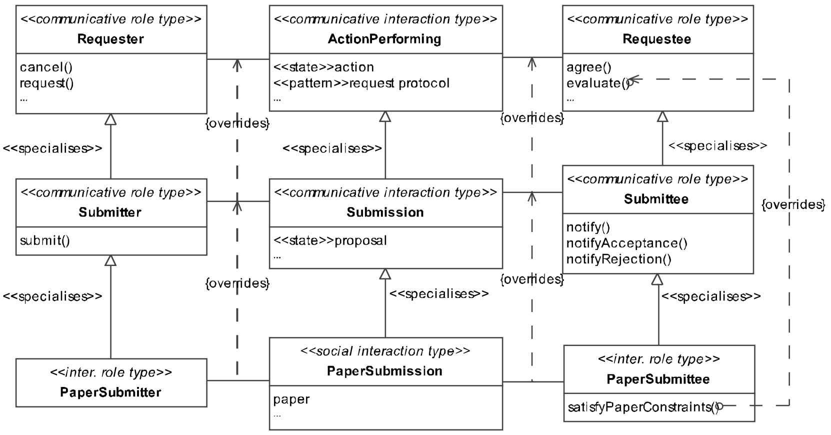

**Fig. 4.** Reuse of generic communicative interactions

## 4 Conclusion

The $\mathcal{RICA}$ metamodel offers an organizational perspective on agent communication languages, by structuring communicative actions and protocols around characteristic social roles and interactions. These mixed communicative/organizative abstractions provide the glue to smoothly integrate the ACL within the organization model of a given MAS. Furthermore, it allows for a principled reuse approach to the design of the MAS interaction space, since communicative interactions serve as micro-organizational modelling patterns [8]. Besides the communicative interactions underlying FIPA ACL (information exchange, action performing, etc. [3]), many others may be identified on the basis of non-standard performatives suggested in other domains (e.g. negotiation [9], dialogue types put forward by argumentation theorists [10], etc.).

Our analysis of agent interactions may be integrated as a refinement of other organizational frameworks, where similar abstractions are found. For instance, GAIA[5]/AGR[7]/INGENIAS[11] interactions and PASSI communications [12] resemble *social interactions* in modelling spirit. The integration of this abstraction would require introducing the *subsume* relationship, or reusing a similar one (e.g. the *dependencies* in the AGR metamodel), in order to link the new interactive roles with the social ones of the framework.

## References

1. Serrano, J.M., Ossowski, S.: A semantic framework for the recursive specification of interaction protocols. In: Coordination Models, Languages and Applications. Special Track of the 19th ACM Symposium on Applied Computing (SAC 2004). (2005)
2. Serrano, J.M., Ossowski, S.: On the impact of agent communication languages on the implementation of agent systems. Cooperative Information Agents VIII (Klusch, Ossowski & Kashyap, ed.) (2004)
3. Serrano, J.M., Ossowski, S., Fernández, A.: The pragmatics of software agents - analysis and design of agent communication languages. Intelligent Information Agents - An AgentLink Perspective (Klusch, Bergamaschi, Edwards & Petta, ed.), Lecture Notes in Computer Science **2586** (2003) 234–274
4. Esteva, M., Rodriguez, J.A., Sierra, C., Garcia, P., Arcos, J.L.: On the formal specifications of electronic institutions. In Dignum, F., Sierra, C., eds.: Agent-mediated Electronic Commerce (The European AgentLink Perspective). Volume 1191 of LNAI., Berlin, Springer (2001) 126–147
5. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multiagent systems: The Gaia methodology. ACM Transactions on Software Engineering and Methodology **12** (2003) 317–370
6. Allen, R., Garlan, D.: A Formal Basis for Architectural Connection. ACM Transactions on Software Engineering and Methodology **6** (1997) 213–249
7. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: AOSE. (2003)
8. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Organizational abstractions for the analysis and design of multi-agent systems. In Ciancarini, P., Wooldridge, M.J., eds.: AOSE. Volume 1957 of LNCS. Springer (2000) 235–252
9. Wooldridge, M., Parsons, S.: Languages for negotiation. In Horn, W., ed.: Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI-2000), Berlin, IOS Press (2000) 393–397
10. Parsons, S., McBurney, P.: Argumentation-based communication between agents. In: Communication in Multiagent Systems. (2003) 164–178
11. Pavón, J., Gómez-Sanz, J.: Agent oriented software engineering with ingenias. In Marik, V., Mller, J., Pechoucek, M., eds.: Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems, Springer Verlag (2003)
12. Cossentino, M., Sabatucci, L.: Agent system implementation. In: Agent-Based Manufacturing and Control Systems: New Agile Manufacturing Solutions for Achieving Peak Performance. CRC Press (2004)