MaCMAS/UML Metamodels: A Methodology Fragment for the Analysis Stage of Large Complex/Complicated Multi–Agent Systems* WORKING DOCUMENT 19-June-04

Joaquín Peña and Rafael Corchuelo

The Distributed Group
University of Seville
Avda. de la Reina Mercedes, s/n. Sevilla 41.012 (Spain)
Phone: +34 954 55 38 65, Fax: +34 954 55 71 39
E–mail: joaquinp@us.es, web page: www.tdg-seville.info

1 Introduction

A MAS Metamodel describes the elements/concepts and their relationships that a certain methodology use in their models. It is a crucial element of a methodology since it describes concisely and without ambiguity how models for a certain system can be built (instantiated) and the constraints in the use of their constituents elements. Metamodels are more crucial, if possible, when defining a methodology fragment as it is our case. A Methodology fragment can not be used without integrating it with other fragments or methodologies. In this scene, establishing the relations between elements in metamodels taken into account represents a good starting point to later integrating software processes, notation, and so on.

MASs are inherently complex. Most researches agree on that complexity is a direct consequence of the high degree of interaction this kind of systems presents. Our approach is focused on dealing with complexity derived from interactions. As a consequence we have describe a methodology fragment to model the acquaintance organisation of a MAS without taking into account the structural organisation. The main contributions of our approach regarding others are:

1. Interactions are the central concept of our fragment, and thus, of all metamodels we propose. It improves our ability to deal with systems with a high interaction degree.
2. An interaction can relate an arbitrary number of roles (called multi-Role Interactions) and can be refined by several finer–grain interactions. Both facts allow us to perform a layered description of the system which improves our ability to deal with complexity through an iterative and incremental process.

* The work reported in this article was partially supported by the Spanish Ministry of Science and Technology under grant TIC2003- 02737-C02-01.

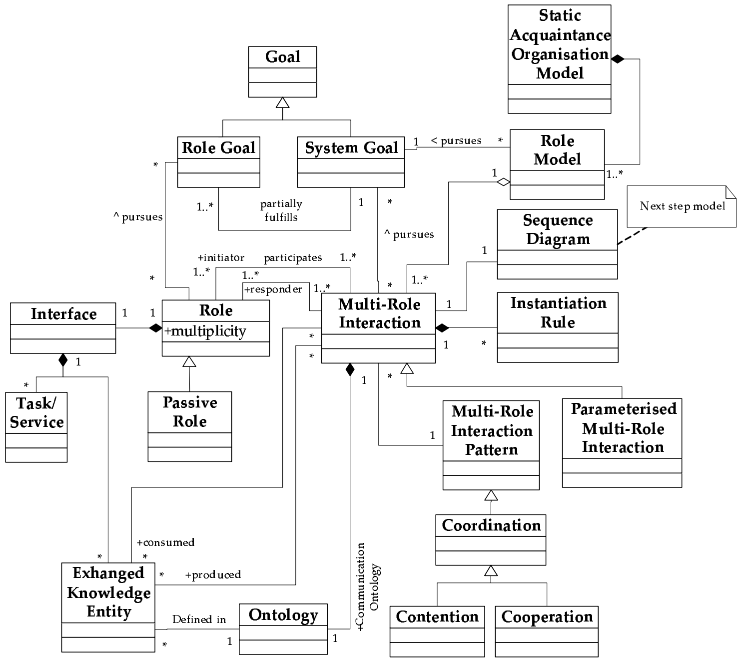

Fig. 1. Acquaintance Organisation View: Role Model Meta-Model

3. We provide a set of automatic algorithms to compose, decompose interactions. It helps the transition between layers.
4. Interactions correlates with systems goals. It allows to establish a clear traceability between requirements and final models produced by our approach.
5. Interactions can be reused at any level of abstraction.

2 The MaCMAS/UML Metamodels and Models

In order to represent interactions abstractly we propose *multi–Role Interactions* (mRI) [5,4]. mRIs are first class modelling elements in our models and are used as the minimum building block at static and dynamic aspects of modelling. Their use is crucial for performing an iterative and incremental layered approach since mRIs can be described internally by means of finer–grain mRIs, or several of them can be abstracted by coarse–grain one (see Figure 6).

An mRI is an *institutionalised pattern of interaction* that represents abstractedly the fulfillment of a system goal without detailing how this achievement is carried out. Thus, using mRI as the minimum modelling element we do not have to take into account all the details required to fulfill a complex system goal nor the messages that are exchanged at stages where these details have not been identified clearly or even are not known (we do not use message sequences in

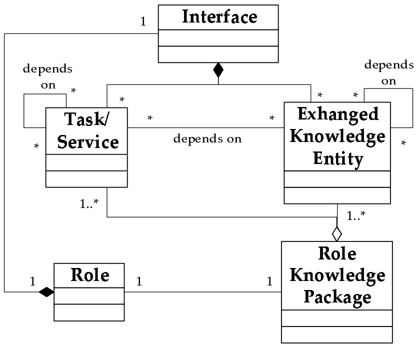

Fig. 2. Knowledge Dependencies Meta-Model

our fragment but we leave it for next fragments). This direct correlation between system goals and mRIs allows us to establish a clear traceability between goal-oriented requirement documents and analysis models and it also improves our ability to model congruent systems [3]. As it is shown in the Meta Model of Figure 1, an mRI is related with the following modelling artifacts:

1. the system goal it fulfills,
2. the knowledge consumed and produced by the interaction,
3. the interaction ontology describing the knowledge that is processed in the interaction.
4. the coordination mechanism it follows: contention or cooperation,
5. an instantiation rule which shows the constraints over the agents that may play the roles in the interaction.
6. and the roles that participates on it (notice that we do not include the agent concept since we leave it for the next fragment where role must be mapped onto agents):
   (a) the goals of roles,
   (b) the cardinality of roles (number of agents that may play it),
   (c) the initiator/s role/s, the role interfaces: knowledge and services provided by each role,
   (d) a guard for each role that represent if the agent playing a role wants to participate in the mRI or not.

mRIs are represented with UML 2.0 collaborations [2, pag. 132] as all the models we propose. We propose three views of the acquaintance organisation: Two for representing the static and dynamic aspects of the organisation and a third for representing the relation between models in different abstraction layers. We use the following models:

a) Static Acquaintance Organisation View: It shows the interaction relationships between roles in the system statically and the knowledge processed by them. It comprises the following UML models:

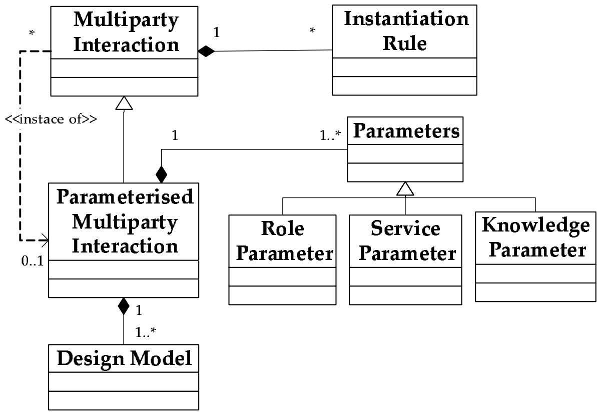

Fig. 3. Parameterised mRI Meta-Model

Role Models: It is described in metamodel of Figure 1. It shows a set of agent roles collaborating by means of several mRIs. Notice that a role model may also contain a single mRI.

Ontology: It shows the ontology shared by roles in a role model. It is used to add semantics to the knowledge owned and exchanged by roles (see [1] for an UML–based ontology notation).

Knowledge view: Its metamodel is depicted in Figure 2. It shows the knowledge processed by each role in an mRI and their dependencies. It consists of a set of UML 2.0 packages, one for each role, where we place the knowledge and services each role provides and their dependencies. It is used to understand how the interaction is carried out with out entering into details and to perform dependency decomposition (see [4] for further details).

Parameterised mRIs: Its metamodel is depicted in Figure 3. Some interactions patterns can be generalised to reuse them. Roles, knowledge and services can be parameterised which allows to reuse them with concrete roles/agents and certain knowledge and services. It is represented using UML 2.0 collaboration templates [2, pag. 509]. The instances of parameterised mRIs can be used on Role Models.

b) Behaviour of Acquaintance Organisation View: The behavioral aspect of an organisation shows how the interactions in a certain role model sequence. It is represented by two equivalent models (the metamodel of both views is depicted in Figure 4):

Behaviour of a role: It represents separately the behaviour of each role in a role model showing how the mRIs of the role sequence. It is represented using UML 2.0 ProtocolStateMachines [2, pag. 422]. It is used to focus on a certain role ignoring others.

Behaviour of a role model: It represents the order of mRIs in a role model with a centralised description. It is represented using UML 2.0

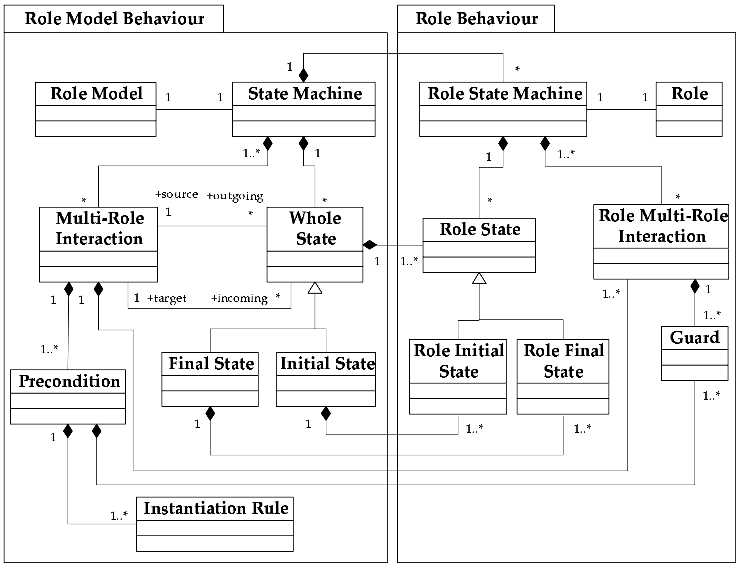

Fig. 4. Behaviour Organisation View Meta-Model: Role and Role Model Behaviour

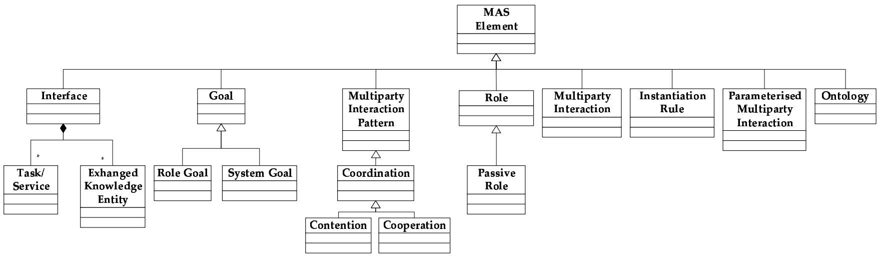

Fig. 5. MAS Elements Meta-Model

StateMachines [2, pag. 446]. It is used to understand easilily the whole behaviour of a sub–organisation.

c) **Traceability view:** This model shows how models in different abstraction layers relate. As it is shown in the meta–model of Figures 6 and 5, it shows how mRIs are abstracted, composed or decomposed by means of *classification* (when a parameterised mRI is instantiated), *aggregation*, *generalisation* or *redefinition*. Notice that we usually shows only the relations between interactions beacuase they are the focus of modelling, but all the elements that conform an mRI can be also related. Finally, since an mRI presents a direct correlation with system goals, traceability models clearly shows how a certain requirement system goal is refined and materialised.

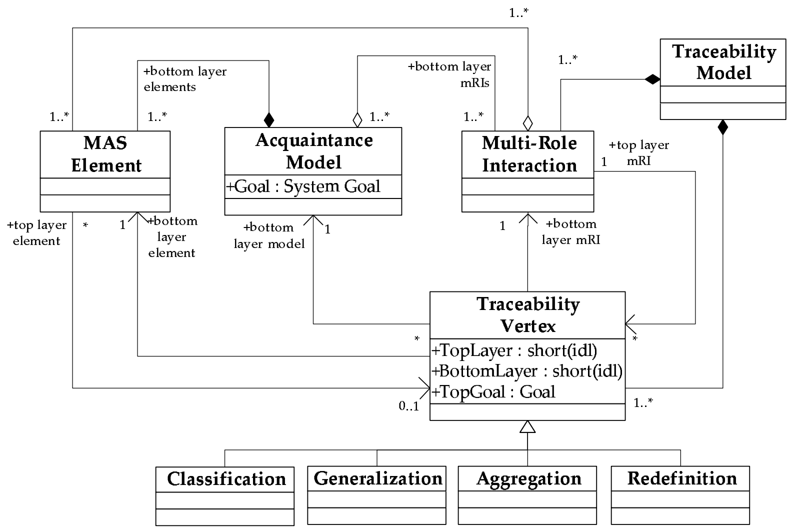

Fig. 6. Traceability Meta-Model

References

1. S. Cranefield, S. Haustein, and M. Purvis. Uml-based ontology modelling for software agents. In *Proceedings of the Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents*, 2001.
2. O. M. G. (OMG). Unified modeling language: Superstructure. version 2.0. Final adopted specification ptc/03–08–02, OMG, August 2003. www.omg.org.
3. H. Parunak, S. Brueckner, M. Fleischer, and J. Odell:. A design taxonomy of multi-agent interactions. In P. Giorgini, J. P. Müller, and J. Odell, editors, *IV International Workshop on Agent-Oriented Software Engineering (AOSE'03)*, volume 2935 of *LNCS*, pages 123–137. Springer–Verlag, 2003.
4. J. Peña, R. Corchuelo, and J. L. Arjona. A top down approach for mas protocol descriptions. In *ACM Symposium on Applied Computing SAC'03*, pages 45–49, Melbourne, Florida, USA, 2003. ACM Press.
5. J. Peña, R. Corchuelo, and J. L. Arjona. Towards Interaction Protocol Operations for Large Multi-agent Systems. In M. Hinchey, J. Rash, W. Truszkowski, C. Rouff, and D. Gordon-Spears, editors, *Proceedings of FAABS 2002*, volume 2699 of *LNAI*, pages 79–91, MD, USA, 2003. Springer–Verlag.