Software components for a dialogue multiagent system

Maxime Morge
PhD Candidate
ENS Mines Saint Etienne (France)
morge@emse.fr

Steven Collins
Research Engineer
ENS Mines Saint Etienne (France)
collins@emse.fr

Abstract

This work proposes a software component approach to design a dialogue multiagent system.

1 Introduction

Interaction is widely recognized as the most important issue to design complex software (Singh, 1997). In a software engineering viewpoint, a MAS software system is made up of multiple independent and encapsulated loci of control (i.e. agents) interacting with each other in the context of a specific application.

MAST (MultiAgent System Toolkit) is a project developed in our laboratory (Vercouter et al., 2003). The goal of this project is to provide a software framework to simplify the implementation of agent-based applications. MAST is composed of different modules :

- DeMas : agent platform for distributed execution providing low level services like registry (`Agent Identifier AID`) or message transport ;
- AdMas : an administrator's tool to supervise the deployment and the execution through observation of DeMas services ;
- GeMas : a library of software components providing models and tools coming from academic works, to build applications ;
- MeMas : an Integrated Development Environment (IDE) for building applications based on GeMas.

2 MASTComponents

This library of MASTComponents is based on the vowels approach where : (1) facet "agent" provides several models of reasoning, (2) facet "environment" is a model for perception and action, (3) facet "interaction" is a dialogue system, (4) facet "organization" : an organizational model for multiagent systems and (5) facet "user" : a Graphical User Interface (GUI) for agents.

Figure 1 shows the UML class diagram that explains the relationships between the classes that represent components, roles and events. A MASTComponentRole, an abstraction of services provided by a component, specifies behaviours implemented by a MASTComponent. A MASTComponentRole is a member of one or several Facets. The MASTEvents link MASTComponentRoles in the same Facet.

This paper focuses on the InteractionFacet.

3 InteractionFacet

The dialogue system proposed is for an argumentation system (Parsons et al., 2002), a generic model of reasoning. Figure 2 reports the UML object diagram that explains the relationships among the objects that represent MASTComponentRoles in the InteractionFacet, and InteractionEvents between them.

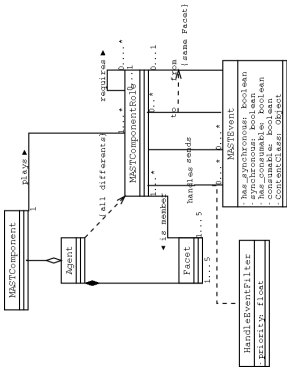

Figure 1: UML Class Diagram of a MASTComponent

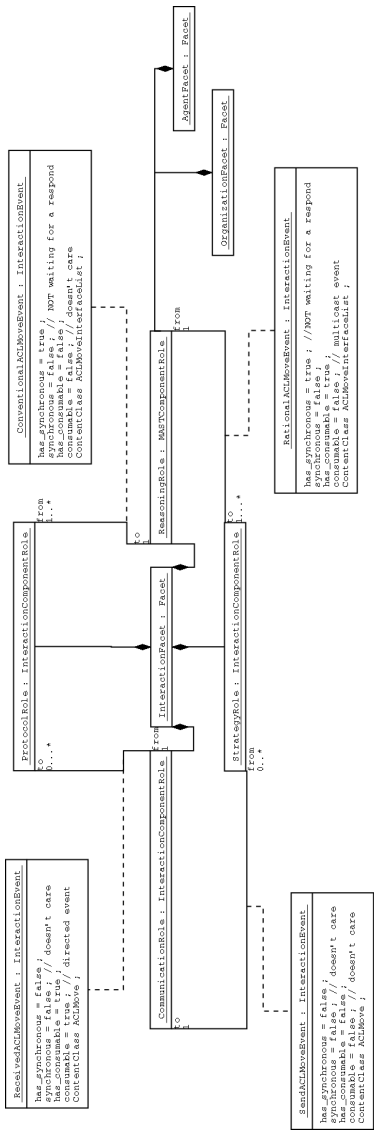

Figure 2: UML Object Diagram of InteractionFacet

an ACLMove is identified by a Move Identifier (MID) which can be referred to later, and contains : (1) a speaker (AID), (2) some hearers (AIDs), and (3) a reply field : a MID or empty if the ACLmove is the first one. an ACLMove follows (4) a conversational protocol (ProtocolRole) and contains (5) an illoctionary act made up of a performative and a content.

An InteractionComponentRoles don't necessarily have all the fields of their ACLMove specified. They exchange ACLMoveInterfaces or a ACLMoveInterfaceList (a collection of ACLMoveInterfaces). A CommunicationComponentRole receives (resp. sends) ACLMoves coming from (resp. to) other agents. A ProtocolComponentRole manages a conversation protocol. It handles the latest received ACLMove and computes a collection of conventional answers. A ReasonningComponentRole handles this collection and selects a subset of chosen ACLInterfaceMove. A StrategyComponentRole chooses an ACLMove, the best depending on the strategy it implements.

4 Conclusion

The goal of this work is to provide a generic way to specify a dialogue multiagent system. Instantiation of this software components is a work in progress.

References

[Parsons et al.2002] Simon Parsons, Michael Wooldridge, and Leila Amgoud. 2002. An analysis of formal inter-agent dialogues. In *Proc. of the first international joint conf. on autonomous agent and multiagent systems*, volume part 1, pages 394–401.

[Singh1997] Munindar Singh. 1997. Interaction-oriented programming for the web. In *Proc of the international conference on Cooperative Information Systems (CoopIS)*, May.

[Vercouter et al.2003] L. Vercouter, O. Boissier, S. Collins, M. Morge, C. Carabelea, B. Bratu, G. Muller, A. Mounier, P. Beaune, and F. Gaultier. 2003. Mast : a multi-agent system toolkit. Technical report, ENS. Mines St Etienne.