Models and Tools for Mulan Applications

Lawrence Cabac, Till Dörges, Michael Duvigneau,
Christine Reese, Matthias Wester-Ebbinghaus

University of Hamburg, Department of Computer Science,
Vogt-Kölln-Str. 30, D-22527 Hamburg
`http://www.informatik.uni-hamburg.de/TGI`

Abstract In this work we describe the development process of multi-agent application design and implementation with Mulan. Our approach can be characterized as model driven development by using models in all stages and levels of abstraction regarding design, implementation and documentation. Both, standard methods from software development as well as customized ones are used to satisfy the needs of multi-agent system development.

1 Introduction

The agent metaphor is highly abstract and it is used to develop software engineering techniques and methodologies that particularly fit the agent-oriented paradigm. Flexibility and autonomy of an agent's problem-solving capabilities, the richness of agent interactions and the (social) organizational structure of a multi-agent system as a whole must be captured.

Our approach borrows several ideas from well-known methodologies (AO, OO) as well as concepts from conventional modeling techniques (UML). We integrate our techniques and tools with the model-based implementation of the Mulan framework, which facilitates Petri net-based programming. In the Mulan architecture, agents and multi-agent systems are modeled through the high-level Petri net formalism of reference nets.

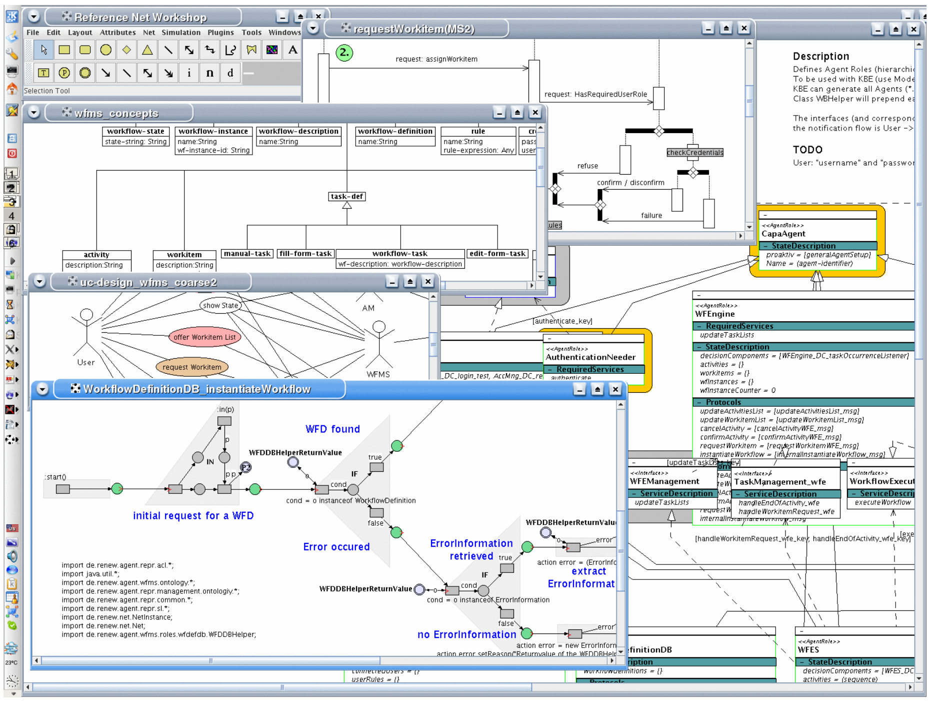
The result of those efforts is a development methodology that continuously integrates our philosophy of Petri net-based and model-driven software engineering in the context of multi-agent systems. In Section 2 we introduce the basic conceptual features of multi-agent application development with Mulan. The particular techniques, models and tools are presented in Section 3.

2 Concepts of Application Development with Mulan

The reference net-based multi-agent system architecture Mulan (Multi Agent Nets) structures a multi-agent system in four layers, namely *infrastructure*, *platform*, *agent* and *protocol* [2,3]. In a multi-agent application the organizational structure has to be defined, such that responsibilities for all aspects of the system are specified. The general perspectives in the area of multi-agent systems are the structure, the interactions, and the terminology. These perspectives are orthogonal with connecting points at some intersections. The structure of a multi-agent system is given by the agents, their roles, knowledge bases and decision components. The behavior of a multi-agent system is given by the interactions of the agents, their communicative acts and the internal actions related to the interactions. The terminology of a multi-agent system is given as a domain-specific ontology definition that enables agents and interactions to refer to the same objects, actions and facts. Without a common ontology, interactions are impossible.

3 Techniques, Models and Development Tools

In this section we describe the techniques applied during the various stages of multi-agent application development with Mulan. Most of the techniques and tools mentioned can be seen in Figure 1.



```
import de.renew.agent.repr.acl.*;
import java.util.*;
import de.renew.agent.wfms.ontology.*;
import de.renew.agent.repr.management.ontologiy.*;
import de.renew.agent.repr.common.*;
import de.renew.agent.repr.sl.*;
import de.renew.net.NetInstance;
import de.renew.net.Net;
import de.renew.agent.wfms.roles.wfdefdb.WFDDBHelper;
```

Figure 1. Models

The *coarse design* is done mainly in open discussions. The results are captured in simple lists of system components and agent interactions. This culminates in a *use case diagram*. The *structure of the multi-agent application* is refined using a *role diagram*. This kind of diagram uses features from class diagrams and component diagrams. The terminology of a multi-agent system is used in form of an *ontology definition* by the agents to communicate with each other and for their internal representation of the environment. The facts about an agent's environment are located in its *knowledge base*. The *behavior* of the system components is specified using *agent interaction protocol diagrams* (AIP, integrated into PAOSE in [1]). Additionally *Decision Components* (DC) can be used to encode behavior not directly related to communication between agents. Last, but not least *monitoring and debugging* tools are developed in our group.

4 Conclusion

In this paper we presented an integrated approach to multi-agent application development with Mulan, which is based on Petri nets and agent-oriented software engineering. The approach allows for multiple levels of abstraction. The tools that are used during the development process support all phases of development with modeling power and deployment facilities, although some of the tools still have prototypical character.

For the future, we follow several directions to refine the approach. On the practical side, we look into further developments, improvements and integration of tools and techniques. On the conceptual side, we want to expand the multi-agent oriented approach to other aspects of the development process like project organization and agent-oriented tool support. Following these directions, we want to achieve symmetrical structures in all three aspects of software development: the system, the development process and the project organization.

References

1. Lawrence Cabac, Daniel Moldt, and Heiko Rölke. A proposal for structuring Petri net-based agent interaction protocols. In Wil van der Aalst and E. Best, editors, *24th International Conference on Application and Theory of Petri Nets, Eindhoven, Netherlands, June 2003*, volume 2679 of *LNCS*, pages 102–120. Springer-Verlag, June 2003.
2. Michael Köhler, Daniel Moldt, and Heiko Rölke. Modelling the structure and behaviour of Petri net agents. In J.M. Colom and M. Koutny, editors, *Proceedings of the 22nd Conference on Application and Theory of Petri Nets 2001*, volume 2075 of *LNCS*, pages 224–241. Springer-Verlag, 2001.
3. Heiko Rölke. *Modellierung von Agenten und Multiagentensystemen – Grundlagen und Anwendungen*, volume 2 of *Agent Technology – Theory and Applications*. Logos Verlag, Berlin, 2004.