CASA - Structured Design of a Specification Language for Intelligent Agents

Stephan Flake and Christian Geiger

C-LAB VIS (Visual Interactive Systems), Paderborn, Germany
E-mail: {flake,chris}@c-lab.de, http://www.c-lab.de/vis

1 Introduction

Agent based technologies in the sense of distributed computing becomes increasingly relevant in academic and industrial research and development. Our multi agent system CASA focuses on the specification of complex plans that describe the behavior of agents. The design of CASA is based on concepts from concurrent logic programming that were used to extend the well known BDI agent approach AgentSpeak(L) [2]. Rao's AgentSpeak(L) is a specification language similar to horn clauses and can be viewed as an abstraction of an implemented BDI system. Our work is based on the assumption that AgentSpeak(L) demonstrates a successful reengineering approach of an implemented multi agent system that is now given a formal specification. By extending this specification with new features for complex plans it was possible to derive an efficient implementation that supports these new features.

2 Design of the CASA Specification Language

Specification: CASA agents perceive events/messages and select *relevant plans* with individual weights for handling these events/messages. Plans are modeled as clauses with additional guard predicates for testing the applicability of the clause (*applicable plans*). Such guard predicates may be arbitrary complex, i.e., the reduction of a guard may include the evaluation of another plan (deep guards). Such complex speculative computations may also include communicative acts with other agents. Based on the contextual conditions in guard predicates different plan types can be distinguished: *reactive plans* only have simple tests as guards, *deliberative plans* allow speculative computations within an agent and *communicative plans* allow to communicate with other agents. CASA uses a hierarchy for plan selection (reactive $>$ deliberative $>$ communicative). An agent can execute several plans at a time and elements of a single plan can be processed sequentially or in parallel. Additionally, plans can be suspended by other plans and have a special exception section if an applicable executed plan fails. The features of a CASA agent are formally defined based on extended guarded horn clauses and the cycle of operation is specified by an abstract interpreter. For ease of use we developed a simple textual format that allows an efficient modeling.
Modeling: The textual CASA definition format defines the initial agent state and is divided into four sections: First, functions for selecting events, plans, and intentions at run time execution have to be declared. Initial goals of the agent are declared in the

second section. Each of these goals will be instantiated together with an applicable plan. For parallel plan execution a multistack data structure is used to handle the instantiated plans (named as intentions) as separate entities. Initial facts and plans are listed in the third and fourth section.

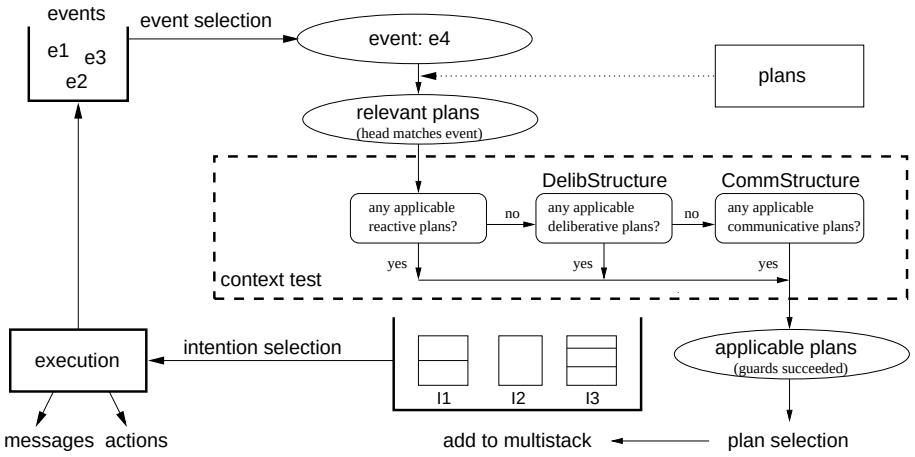

Fig. 1. CASA Interpreter Cycle

Handling speculative computations is one of the major aspects of the abstract CASA interpreter. Speculative computations appear in the context test for relevant plans whenever guards of deliberative or communicative plans have to be checked. Two independent additional components are introduced in order to manage speculative computations: the *DelibStructure* (resp. *CommStructure*) is holding elements composed of a goal and all relevant deliberative (resp. communicative) plans to check. For each of these elements a new instance of the interpreter is generated and executed in parallel to the other cycles. The operational semantics of the CASA interpreter are best described by means of the interpreter cycle shown in Figure 1.

Implementation: CASA is implemented with JDK 1.1.8, integrating modules of M. Huber's JAM library. A parser written in JavaCC reads the textual CASA specification, sets the initial state of CASA agents and starts the execution on the CASA interpreter. CASA agents are integrated into the MECCA framework, an agent management system that implements the FIPA ACL standard.

Validation: As a first case study we presented a simple application taken from holonic manufacturing [1]. Future work will concentrate on the development of visual tools for the design of CASA agents and the application in the area of flexible manufacturing systems and intelligent user interfaces.

References

1. S. Flake, Ch. Geiger, G. Lehrenfeld, W. Mueller, V. Paelke. Agent-Based Modeling for Holonic Manufacturing Systems with Fuzzy Control. *NAFIPS'99*, New York, 1999.
2. A.S. Rao. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. *7th European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands, 1996.