A Formal Semantics for DAI Language
NUML

Xu Manwu (许满武), Lu Jianfeng (陆剑锋)
Zeng Fancong (曾凡聪) and Dai Jinwen (戴劲雯)

Department of Computer Science & Technology, Nanjing University, Nanjing 210093

Received August 11, 1994; revised December 5, 1994.

Abstract

Traditional AI systems are brittle in the sense that they fail miserably when presented with problems even slightly outside of their limited range of expertise. A powerful, extensible strategy of Distributed Artificial Intelligence (DAI) for overcoming such bounds is to put the system in a society of systems. So the ability to coordinate group activities of individuals and to communicate between each other is necessary for a language describing DAI systems. Agent-oriented language NUML is such a language. It is a specific kind of object-oriented language. To give formal semantics to NUML, there is the problem to formalise object-oriented programming paradigm which is still open. The theory of higher-order π-calculus is a concurrent computation model with sufficient capability, which provides us a mathematical tool to do the formalization. This paper tries to use higher-order π-calculus to formalise NUML.

Keywords: Formal semantics, π-calculus, logic programming, distributed AI.

1 Introduction

As E.H.Durfee states in [1]:

Artificial Intelligence (AI) has emphasized building “stand-alone systems” that can solve problems with minimal help from other systems (computer or man). These systems have traditionally been brittle, in the sense that they fail miserably when presented with problems even slightly outside of their limited range of expertise . . .

A more powerful, extensible strategy for overcoming the inherent bounds of intelligence present in any finite AI (or natural) system is to put system in a society of systems, so that it can draw on a diverse collection of expertise and capabilities in the same way that people overcome the limitations of individuals by banding together into societies that are designed to accomplish what individuals cannot. The ability to flexibly team up and coordinate group activities toward individual and collective goals is a hallmark of intelligence.

This project is supported by the National High-tech. Research & Development Program(863), National Natural Science Foundation and National Key Laboratory of Software Engineering of China.
Research in distributed artificial intelligence (DAI) concentrates on understanding the knowledge and reasoning techniques needed for intelligent coordination, and on embodying and evaluating this understanding in computer systems...

Therefore, to describe such intelligent agent systems, a candidate language, in our opinion, should provide the following features:

1. The combination of multiple programming paradigms, such as functional, logic and agent-oriented programming, to meet the requirement of applications;
2. The metareasoning mechanism to support the capabilities of reflection and introspection in running DAI systems; and
3. The construct of programs suitable for software development. Intelligent agent, for instance, is such a construct.

We have implemented an experimental language named NUML on a graph reduction architecture with an enriched λ-calculus as its base language. D-constructor expression and the concept of implicit goal are added into the base language to amalgamate logic programming paradigm with functional programming paradigm so that a unified foundation with good mathematical properties is presented. Correspondingly, we include a derivation rule for this graph reduction mechanism as well as the β-reduction rule.

Having adopted the concept of agent-oriented programming, NUML becomes a specialization of the object-oriented language. As we know, it is hard to deal with concurrency while giving formal semantics to an object-oriented language. In the case of NUML, if we use the reduction machine model mentioned above to do the formalization, we should face the same difficulty. On the basis of CCS, Milner, Parrow and Walker create the theory of π-calculus which provides a concurrent computation model with sufficient capability. And Walker gives semantics to a parallel object-oriented programming language named POOL by translating it into π-calculus. Thesom shows how Plain CHOCS (a second generation calculus for higher-order processes) can be used to give semantics to a prototype object-oriented language called D. In [6,10], Milner and Sangiorgi proved that π-calculus has the computation capability of λ-calculus by translating terms in λ-calculus into processes of π-calculus. So the necessary but difficult thing we should do to formalise NUML is to give the semantics of Horn-clause logic in π-calculus, i.e., to deal with:

- the bi-directional information passing of unification, and
- the disharmony between the execution mechanism of Horn-clause logic and the reduction mechanism of π-calculus.

In this paper, we are mainly concerned with the Horn-clause logic part of NUML and semantics of that part is given in higher-order π-calculus.

The language NUML and higher-order π-calculus are described in Sections 2 and 3 respectively. Its semantics in HOπ is given in Section 4. A reduction implementation model based on HOπ is presented in Section 5. Finally in Section 6, we end this paper with a discussion of intended further work.