

発表概要

エージェント開発環境 RXF における
slicing に基づくデバッガについて大 園 忠 親[†] 新 谷 虎 松[†]

本研究は、マルチエージェントシステム開発環境 RXF の実装研究である。RXF の開発において、知的で自律したエージェントから構成されたマルチエージェントシステムの直感的な開発を支援することを目標としている。そのようなエージェントの実現のために、制約論理型言語に基づくエージェント記述言語と、リフレクションに基づくエージェントの実行環境を実装した。複数のエージェントを並行して実行させるためと、エージェントが複数の処理を並行処理するためのスレッド機能も提供している。本エージェントのデバッグにおいて、分散環境などで並行動作する複数のスレッドのデバッグをする必要がある。並行動作するプログラムにおけるデバッグに関する問題点として、非決定性によるバグの再現の困難さと、デバッガによる副作用を原因とするバグ発見の困難さがある。本研究では、slicing に基づくことによって、これらの問題を軽減することが可能なデバッガを実装している。slicing とは、特定の変数の値に関係しているプログラムの部分を特定するための手法である。従来の slicing に基づくデバッガは、手続き型言語へ適用されていたが、本研究では、論理型言語への適用を試みる。本システムにおける slicing に基づくデバッガの特徴は、論理型言語におけるトレイルスタックの利用と、バックトラックや assert と retract への対応である。

A Debugger with Program Slicing in the Agent
Development Environment RXFTADACHIKA OZONO[†] and TORAMATSU SHINTANI[†]

We have developed a new programming environment, RXF, building a multi-agent system (MAS). RXF is used for realizing intuitive development of MASs that consist of reflective agents. The reflective agents are intelligent and autonomous agents based on a reflection. RXF provides capabilities of a constraint-logic programming language, a multi-thread and a reflection mechanism for realizing reflective agents. In constraint-logic programming, programmers can represent various data by using predicates and can process numerical data. The multi-thread is used to implement concurrent programs of agents. MAS Programmers may debug concurrent programs of MASs. The main problems associated with debugging concurrent programs are the "probe effect" and nonrepeatability. The probe effect refers to the fact that any attempt to observe the behavior of concurrent programs may change the behavior of that programs. The nonrepeatability means that different executions with the same data will result in different results. We attempt to implement a new debugger based on the slicing to reduce the problems. The slicing is an effective technique for narrowing the focus of attention to the relevant parts of a program during the debugging process. The feature of our debugger is support for a back track of logic programming and built-in predicates: assert and retract. Our debugger utilizes a trail stack of Prolog to reduce overhead in dynamic slicing. We show the implementation of a slicing on logic programming and our examination on debugging MASs using the debugger.

(平成 11 年 8 月 5 日発表)

[†] 名古屋工業大学 知能情報システム学科
Intelligence and Computer Science, Nagoya Institute of
Technology