
発表概要

エージェント指向並列言語 Orgel の静的解析による最適化

山本 繁弘[†] 大野 和彦[†] 中島 浩[†]

我々はマルチエージェント型の並列プログラミング言語 Orgel の設計・開発を行っている。Orgel は、並行/並列の実行単位であるエージェントが、抽象的な通信路であるストリームによって明示的にメッセージ送信を行う言語である。Orgel では、プログラマが問題をエージェントという並列処理単位に切り分けることに加えて、多くの並列言語と異なりストリームのエージェントへの接続をすべて宣言的に記述する。このため、プログラムの実行モデルは非常に明確である。エージェント内の処理は C 言語を拡張した文法で記述し、逐次に行われる。したがって、プログラマは Orgel の枠組により実行モデルを構築し、個々の処理内容を従来のスタイルで記述するだけで、通信のタイミングなどに苦慮することなく容易に並列プログラミングを行うことができる。Orgel では宣言的な記述により、コンパイル時に並列実行モデルが明確になっている。そのため、並列モデルが宣言的でない他の多くの並列言語に比べて詳細な静的解析が可能であり、より強力な最適化を施すことができる。本稿では、エージェント間を流れるメッセージの依存関係を解析する手法と、その結果を利用して通信やエージェントのスケジューリングを最適化する手法について述べる。

Optimizing an Agent-oriented Parallel Programming Language Orgel with Static Analysis

SHIGEHIRO YAMAMOTO,[†] KAZUHIKO OHNO[†]
and HIROSHI NAKASHIMA[†]

We are developing a parallel programming language named Orgel based on multi-agent paradigm. In the execution model of Orgel, a set of agents are connected with abstract communication channels called streams. The agents run in parallel sending asynchronous messages through the streams. In an Orgel program, each unit of parallel execution is specified as an agent by the programmer. The connections among agents and streams are declaratively specified. Thus, parallel execution model is clear. Each agent executes sequential code described using a C-extended syntax. Therefore, the programmer designs execution models on the agent framework, and describes details in usual programming style. This feature leads to easy parallel programming without sufferig from timing bugs. The declarative specification of parallel execution model in Orgel also makes the static analysis much effective, compared to other parallel languages. With the result of such analysis, strong optimization is possible. In this presentation, we describe a scheme for dependency analysis among messages. With the analysis result, we also show the optimization of communications and scheduling of the agents.

(平成 12 年 8 月 2 日発表)

[†] 豊橋技術科学大学
Toyohashi University of Technology