

π 計算に対する動作視覚化システム

3L-1

塚崎 悟[†] 金指 文明[‡] 富樫 敦[¶]

静岡大学工学部知能情報工学科

1 はじめに

本論文では、モバイルプロセスの数学的モデルである π 計算の動作を視覚的にシミュレートするシステムを提案する。 π 計算は、Milner ら [1] によって提案されたプロセス計算であり、名前通信という概念を用いてプロセスの動作を記述する。この概念により、データだけでなく通信ポートやプロセスも受送信することが可能となり、プロセスの動的な振る舞いを記述することができる。 π 計算では、プロセスの動作を数式によって記述する。そのため、複雑なプロセスを記述できる反面、プロセスの動作を直感的に理解することが難しい。本システムは、プロセス式を視覚的に表示することによって、 π 計算の直感的な理解を支援する。

一方、ネットワークの普及により、分散・並列処理システムの開発に関する基礎理論や開発手法が重要となっている。 π 計算は分散・並列処理プロセスの数学的モデルであるため、 π 計算を用いることによりシステムの仕様を効率的に記述することができる。 π 計算で記述された仕様を基に実際に動作するシステムが構築できれば、有用なソフトウェア開発手段となる。本論文では、 π 計算のプロセス式を補助的に用いることによりプログラミングを行うフレームワーク (Field Walker)[3] を提案し、そのアプリケーション制作例として視覚化システムの設計・開発を行う。

2 視覚化システムの概要

π 計算には、1つの通信ポートから1つのデータの出入力を行う Monadic π 計算 [1] と、1つのポートで同時に複数のデータの出入力を行う Polyadic π 計算 [2] があ

$P ::= 0$	無動作プロセス (<i>inaction</i>)
$\tau.P$	静アクション (<i>silentprefix</i>)
$x(y).P$	入力アクション (<i>boundinputprefix</i>)
$\bar{x}y.P$	出力アクション (<i>outputprefix</i>)
$P + P$	選択 (<i>summation</i>)
$P P$	合成 (<i>composition</i>)
$(\nu x)P$	制限 (<i>restriction</i>)
$!P$	繰り返し (<i>replication</i>)

図 1: π 計算の構文

る。本論文では、前者の Monadic π 計算を視覚化の対象とする。図 1 に π 計算の構文を示す。

P はプロセス、 x, y は名前を表す。 0 はデッドロックで、 τ は外部からは観測できない内部アクションを示す。 α をアクションとすると、 $\alpha.P$ は α の実行後 P のように振る舞うプロセスを表す。また、 $x(y).P$ や $(\nu y)P$ にある y は、 P によって束縛された名前を表している。繰り返し演算子 $!P$ は、プロセス P の無限個の複製を並行合成することを意味する。

本システムは、大きく分けてプロセス入力・解析部、状態遷移図表示部、プロセス構造表示部の3つの処理部で構成される。プロセス入力部では、 π 計算で記述されたプロセスを入力し解析を行う。入力されたプロセスが誤りのないプロセスであるなら、解析の結果を用いてプロセスの状態遷移図を表示する。状態遷移図表示部では、1回の簡約処理を1つの状態と捉え、各状態についてのプロセスの遷移を視覚化する (図 2)。次状態への遷移は、決定的なプロセスである場合には自動的に遷移先が表示され、非決定的である場合には次に遷移する状態をユーザによって指定する。状態遷移図表示部で遷移先の状態が選択されると、プロセス構造表示部にその状態におけるプロセスの構造が表示される。プロセス間で通信が起こる場合には、アニメーションによって直感的な理解を促す (図 3)。状態遷移図とプロセスの構造を別々に表示

A Visual system for the π -calculus.

[†] Satoru Tsukasaki, Dept. of Computer Science, Shizuoka University.

[‡] Fumiaki Kanezashi, Graduate school of Science and Engineering, Shizuoka University.

[¶] Atsushi Togashi, Department of Computer Science, Shizuoka University.

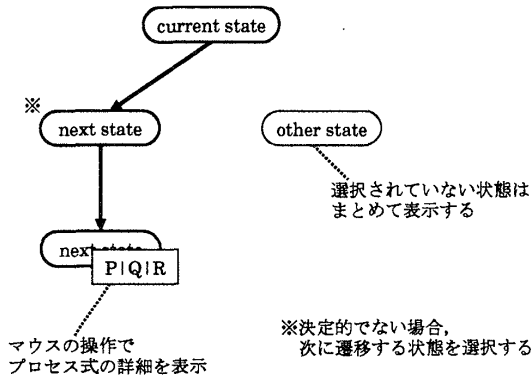


図 2: 状態遷移図表示部の描画イメージ

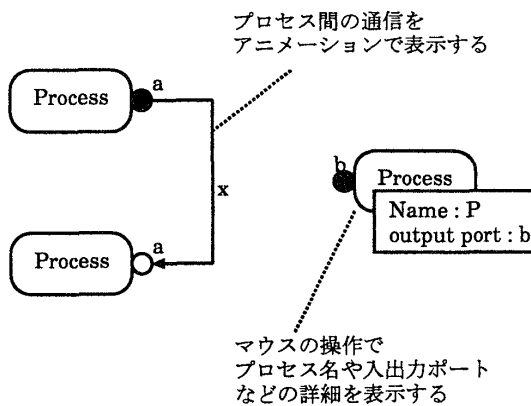


図 3: プロセス構造表示部の描画イメージ

することにより、プロセス全体の流れと、その中での細かな動きを混同することなく理解することができる。

3 実装方法

本論文での視覚化システムは、プロセス入力部や状態遷移図表示部などの個々のプログラムの部品を既存のプログラミング言語 (Java) で実装しておき、これらの部品を π 計算のプロセスとして統合することで動作する。その際、 π 計算で記述されたプロセスに基づいて動作するフレームワークを構築し、このフレームワーク上で視覚化システム的设计を行う。図 4に、このフレームワークの構成を示す。作成したプロセスを実行する際、すべてのプロセスは生成された時点で1つの管理プロセスによって管理される。管理プロセスは出力ポート名と入力ポート名を照合することによって、通信の仲介を行う。また、管理プロセス自身も1つのプロセスであるため、管理プロセス上でさらに別の管理プロセスを動作させることも

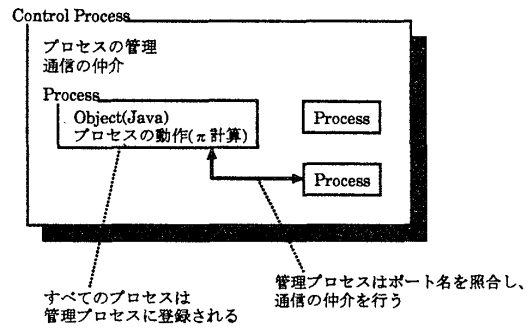


図 4: フレームワークの概要

可能である。このフレームワークを用いることで、 π 計算で設計されたプログラムをそのまま実際に動作するプログラムとして利用することができる。

4 おわりに

本論文では、視覚化の対象を Monadic π 計算に限定したが、より複雑な Polyadic π 計算についての視覚化もやりたい。また、本論文で用いたフレームワークは π 計算によるプログラミング手法を提供しているため、本システムはフレームワーク上で作成したプログラムの評価補助ツールとしても利用可能であると思われる。

謝辞 本論文は、一部文部省科学研究費 (基盤研究 (C)08680343, 重点領域研究 09245214), 電気通信普及財団, 東海産業技術新興財団, 栢森情報科学新興財団の援助による。

参考文献

- [1] Robin Milner, Joachim Parrow, David Walker, A calculus of mobile processes, Reports ECS-LFCS-89-85 and -86, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1989.
- [2] Robin Milner, The Polyadic π -Calculus: a Tutorial, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1991.
- [3] 金指 文明, 塚崎 悟, 富樫 敦, Field Walker: π 計算によるプログラミングシステム, 情報処理学会第 3 回研究会, 1998