

2S-6

π 計算から Java への変換系を用いた分散システム開発手法

玉井 徹† 加藤 暢‡ 樋口 昌宏†

近畿大学大学院総合理工学研究科エレクトロニクス系工学専攻†

近畿大学理工学部情報学科‡

1 はじめに

π 計算とは並行に動作するシステムの振る舞いを形式的に記述することができるプロセス代数の一種である。本研究ではπ 計算から Java のプログラムに変換する処理系を構築した。この変換系を活用することにより、信頼性の高いシステム構築が可能となることが期待される。しかし、π 計算では GUI やファイル入出力を記述することができないため、実用的で実行可能なシステムが直接得られるわけでない。そのため、直接 Java で記述された補助的なプログラムを別に用意し、変換で得られるプログラムと併せて利用する必要がある。ここで必要になる補助的なプログラムとはどのようなものを調べるため、本研究ではファイルダウンローダーのπ 計算式による記述、π 計算式から Java プログラムへの変換、Java プログラムレベルでの処理記述の追加を行った。

2 π 計算の構文規則・遷移規則

π 計算の構文規則と遷移規則は文献 [1] により以下のよう

に与えられる。

構文規則

$$\text{PREFIX } \alpha ::= \bar{a}x \mid a(x) \mid \tau$$

$$\text{AGENT } P ::= 0 \mid \alpha.P \mid P + P \mid \{P \mid P\} \mid$$

if $x = y$ then P | if $x \neq y$ then P

$$\mid (\nu x)P \mid A(y_1, \dots, y_n)$$

遷移規則

$$\text{PREFIX } \frac{}{\alpha.P \xrightarrow{\alpha} P} \quad \text{SUM } \frac{P \xrightarrow{\alpha} P' \quad Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} P'}$$

$$\text{MATCH } \frac{\text{if } x = x \text{ then } P \xrightarrow{\alpha} P'}{P \xrightarrow{\alpha} P'}$$

$$\text{PAR } \frac{P \xrightarrow{\alpha} P', \text{bn}(\alpha) \cap \text{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\text{COM } \frac{P \xrightarrow{a(x)} P', Q \xrightarrow{\bar{a}u} Q'}{P \mid Q \xrightarrow{\tau} P'\{u/x\} \mid Q'}$$

$$\text{RES } \frac{P \xrightarrow{\alpha} P', x \notin \text{fn}(\alpha) \cup \text{bn}(\alpha)}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'}$$

On Developing Distributed Systems based on Pi to Java Translator speaker: Toru Tamai
 †Kinki University Interdisciplinary Graduate School of Science and Engineering
 ‡Kinki University School of Science and Engineering Department of Informatics

3 ファイルダウンローダ

本研究では実用的なシステムの一例として、ファイルダウンロードシステムを取り上げる。このシステムの概要を図 1 に示す。

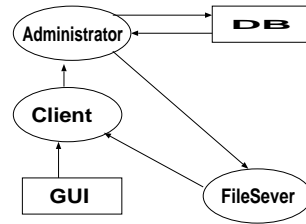


図 1: ファイルダウンロードシステム

Client は GUI からユーザの要求するファイル名と保存したいファイル名を受け取り、Administrator に要求するファイル名と自身のアドレスを渡す。Administrator はそのファイルを実際に持っている FileServer を DB に問い合わせその FileServer に Client のアドレスとファイル名を渡す。FileServer は Client のアドレスに対し実際のファイルを送信する。

3.1 π 計算による記述

このファイルダウンロードシステムをπ 計算式で記述すると以下の様になる。ただし変換の際、オブジェクト参照、ファイルへのデータ書き出し、ライブラリ (補助的なプログラム) の呼び出しの3つの動作は区別する必要があるため、次の様な記述の区別を導入した。 $\overline{\text{name1}\text{name2}\text{name3}}$: $\text{name1}\text{name2}$ というファイル名で name3 という data を保存。 $\text{name1}[\text{name2}]$: name2 を引数とした name1 というあらかじめ用意したライブラリメソッドの呼び出し。この形式に従い変換系は適切なコードを出力する。

```
Client:  $\overline{\text{gui}(\text{remotename})} . \overline{\text{gui}(\text{filename\_local})} .$ 
 $\overline{\text{cfs}} \text{ remotename} . \overline{\text{cfs}} \text{ myport} .$ 
 $\overline{\text{myport}(\text{getfile})} . \overline{\text{filename\_local}} \text{ getfile} . 0$ 
```

```
Administrator:  $\overline{\text{cfs}(\text{filename})} . \overline{\text{cfs}(\text{portname})} .$ 
 $\overline{\text{outname}} \text{ filename} . \overline{\text{inname}(\text{serverport})} .$ 
 $\overline{\text{serverport}} \text{ portname} . \overline{\text{serverport}} \text{ filename} . 0$ 
|  $\overline{\text{outname}(\text{getname})} . \overline{\text{inname}} \text{ where}[\text{getname}] . 0$ 
```

```
FileServer:  $\overline{\text{fileservport}(\text{port})} . \overline{\text{fileservport}(\text{filename})} .$ 
 $\overline{\text{port}} \text{ contents}[\text{filename}] . 0$ 
```

Client において *gui* は GUI との通信路を表す特別なポートである。*cfs remotename* は *cfs* ポート (Client-Administrator 間の通信路) ヘューザの要求するファイル名を送る動作を表す。*filenameLocal getfile* は *filenameLocal* というファイル名で *getfile* という *data* を保存する動作を表す。Administrator の *inname where[getfile]* は *getfile* という名前を引数とし *where* というライブラリを呼び出しその結果を *inname* というポートに出力をする動作を表す。その他同様に、各ポートを通じてポート名やデータを送受信することにより、ファイルダウンロードシステムとしての動作が表現されている。

3.2 π 計算から変換した Java プログラム

本研究で構築した変換系を用い、ファイルダウンローダの π 計算プロセスを Java のプログラムに変換した。ファイルダウンローダの π 計算式を変換する際、特別なポート *gui* からの入力はライブラリとして用意している GUI メソッドの呼び出しに変換する。また、3.1 項で導入した名前の区別に従いファイル出力やライブラリメソッドの呼び出しを行うコードを出力する。

変換系にはコンパイラ・コンパイラの一つである JavaCC を用いた [2]。また、変換によって得られる Java プログラムの通信部分には ORB の一種である HORB を用いた [3]。この変換では各 π 計算プロセスを 1 つのクラスと 3 つのメソッドに変換する。変換した Java プログラムの規模・構成を表 1 に示す。

表 1: 各プログラムの規模

構成数	Client	Administrator	FileServer
フィールド数	15	26	25
行数	76	111	107

3.3 実行に必要な補助的プログラム

π 計算プロセスから変換された Java プログラムだけでは実用的なプログラムとして動作できない。具体的には 3.1 項で述べたメソッド呼び出しをするためのメソッドを持つ Java プログラムと変換された各クラスを実行させる Java プログラムを以下の様にライブラリとして用意した。

Main メソッド: 各 Java クラスのインスタンスを生成し、インスタンスの 3 つのメソッドを起動する。

GUI メソッド: ユーザの要求するファイル名や保存する時のファイル名を入力するためのインターフェイス

DB メソッド: ファイル名と、そのファイルを保持するサーバーとの対応関係を管理するデータベース

入出力メソッド: Client や FileServer が実際にファイルにアクセスするためのメソッド

それぞれのメソッドの規模を表 2 に示す。

表 2: 各メソッドの規模

メソッド	行数
Main メソッド	8
GUI メソッド	138
DB メソッド	12
入出力メソッド	81

4 考察

本変換系を利用して実用的なシステムを構築するにあたり 3.3 項で述べた各メソッドを用意する必要があった。この中で Main メソッドはシステム毎に用意しておく必要があるが、GUI メソッド、DB メソッドや入力メソッドはダウンロード以外のシステムで再利用可能でもあると考えられる。

オブジェクト間通信に利用した Horb では、ファイル転送もオブジェクト参照渡しもすべて String 型の通信として統一的に処理され、データの種別に応じて型を区別する必要がない。このため、 π 計算の構文を拡張する事なくシステムが記述可能となり、容易に Java プログラムに変換が可能となる事が明らかになった。

5 結論

本研究の主要な成果は次の二点である。

1. 本研究では π 計算プロセスを Java のソースに変換する処理系を構築した。これにより信頼性の高いシステム開発が可能になると考えられる。
2. 実際にファイルダウンロードシステムを構築したことにより、実用的なシステムを構築するために必要となる補助的なプログラムやその利用方法を明らかにした。

6 今後の課題

本変換系は π 計算の構文規則の一部にしか対応していない。本変換系により様々なシステムを構築するには、今後その他の構文規則に対応する必要がある。

参考文献

- [1] J.Parrow: An Introduction to the π -Calculus, in J.A.Bergstra, A.Ponse, and S.A.smolka, eds., HAND-BOOK OF PROCESS ALGEBRA, NORTH-HOLLAND (2001)
- [2] 五月女健治: JavaCC コンパイラ・コンパイラ forJava, テクノプレス (2003)
- [3] 萩本順三: HORB ではじめる Java 分散オブジェクトプログラミング, 秀和システム (2003)