

4Q-02

カラーペトリネットで記述された ネットワークアプリケーション動作仕様記述からの Java プログラムの自動導出

坂上弘祐 岡野浩三 谷口健一

大阪大学 大学院基礎工学研究科 情報数理系専攻

1 まえがき

ソフトウェア設計において、複雑なプログラムが動作仕様から自動導出され、また、その動作仕様レベルにおける設計検証を形式的かつ自動的に行える事が望まれる [2, 5]. カラーペトリネット (CPN)[1] はネットワーク通信や同期制御を容易に表現でき、形式的検証も行いやすい. 本稿では、複数エージェントからなるクライアント・サーバシステム (ネットワークアプリケーション) の動作仕様を CPN で記述し、その CPN の記述から、実行系列をマルチスレッドとして自然に対応させた Java プログラムを自動導出する手法を提案する.

2 動作仕様記述

提案する自動導出法では、CPN デザインツール Design/CPN[3] で記述した各エージェントの動作仕様から、Java プログラム群を導出する. 動作仕様記述では、各状態を CPN のプレース、状態からの遷移を CPN のトランジションとして表す. また、CPN のフュージョンプレース (異なった位置にあるがトークンを共用できるプレース) の役割をネットワークによるデータの送受信とする.

なお、今回の導出方法では各部分ネットとして自由選択ネット [4] を使用する. また、トークンの型は論理型・整数型・実数型・文字列型・列挙型のみとする.

3 プログラム導出の概略

導出アルゴリズムでは、ペトリネット記述 (図 1) から独立した複数のタスクを抽出し (図 2)、それぞれのタスクを個々のスレッドに対応させる. このことにより Java のマルチスレッドプログラムを実現する.

ここで、各タスクはペトリネットの分岐のない連続

したトランジションの系列である.

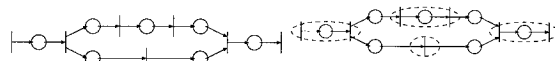


図 1: ペトリネット記述例 図 2: 独立したタスクの取り出し

このように取り出した個々のタスクを 1 つの Java スレッドクラスとして定義する. 変数参照などの同期制御のように、スレッド同士が連携する場合を除き、各スレッドは他のスレッドに関与せず、独立に実行する. タスク処理の要求があれば、その要求に対して、この定義したクラスのインスタンスを生成する. あるタスクから次のタスクに処理が移行する際、タスクが保持している情報を次のタスクに引き渡す. この変換方法により、複数のタスク処理の要求に対しても、それに応じた動的インスタンス生成が可能となる. すなわち、複数エージェントからなるクライアント・サーバシステムに対して、エージェントごとに各インスタンスを生成することにより、自然に処理することができる.

最後に、各スレッドをスレッドグループとしてまとめ、排他制御などを行い連携動作させながら管理するプログラムを導出する. 管理プログラムではスレッドグループ中のスレッドがすべて終了するまで実行を行う.

なお、入出力やネットワーク通信部に関しては、それぞれ入出力管理モジュールとネットワーク管理モジュールを用意しており、必要に応じて各モジュールを参照することで対応をはかっている.

4 導出プログラムの実装

このアルゴリズムに基づく変換系として、cpn-j を実装した. cpn-j は、CPN を入力とし、Java プログラムの導出を行なうトランスレータである. 入力は Design/CPN によって得られる XML テキストデータファイルを使用する. cpn-j は XML parser (Xerces Java Parser

Automatic Java code Generation from Network Application Descriptions in Coloured Petri-nets
Kousuke Sakaue, Kozo Okano and Kenichi Taniguchi
Graduate School of Engineering Science, Osaka University
Toyonaka-shi, Osaka, 560-8531, Japan

1.4.4) を利用し, Java (Java 2 SE 1.3.1) で実装した.

5 例題

cpn-j を簡単なオークションモデルに適用した. このオークションモデルでは各エージェント (売り手と買い手) 間でネットワークを介し, 金額のやりとりなどを行う. なお, クライアント側 (買い手) は複数存在し, サーバ側 (売り手) が複数のクライアント要求を処理する.

オークションのサーバ側の動作仕様を Design/CPN で実際に記述したものを図 3 に表す.

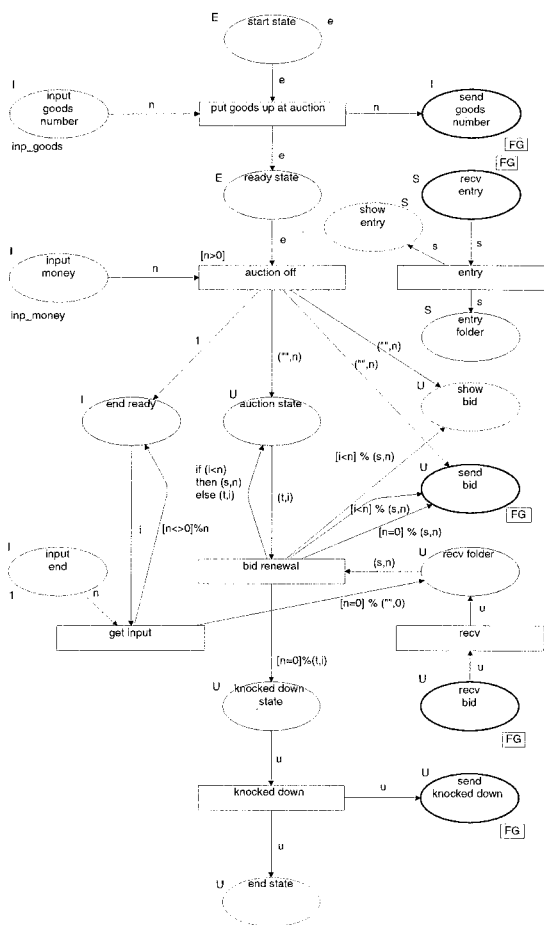


図 3: オークションのサーバ側の動作仕様

例題の動作仕様では, 複数のクライアント要求をサーバ側で各要求毎にスレッド処理を行う事で同時に要求

を処理する. また, タスク毎にスレッドとして並行処理 (この例題では入力の待ち受け部分とネットワーク通信によって設定・変更された金額を受け取って表示する部分) を実現する. この動作仕様を今回提案した導出系に適用させた結果は以下のとおりである.

- 動作仕様を記述したエージェント数: 2
 - ノード (プレース, トランジション) 数: 51
 - コネクタ (アーク) 数: 58
 - ML コード数: 15 行
 - 抽出されたタスク数: 11
- Design/CPN によって得られる XML テキストデータファイル: 約 3600 行
- 導出プログラムサイズ: 約 1500 行
- 導出時間 3 秒 (PentiumIII 500MHz, メモリ 512M, Linux)

この導出系を適用することで, 比較的単純なオークションモデルの CPN 記述から自動的に Java プログラムを導出できた. 今回の例では, Design/CPN ツールによる CPN のシミュレーションや分析結果をそのまま最終システムの実装に適用することができた. 本手法により, この規模の例であれば, 簡単な CPN の記述から実用的なプログラムが自動導出できることが確かめられた.

6 あとがき

本稿ではカラーベトリネットで記述された動作仕様から Java プログラムの自動導出法を提案した. また適用例題についても述べた.

提案手法により, CPN によるソフトウェア設計が可能となり, 高信頼性システムの設計が見込まれる. また, 実装における時間や負担の削減も期待できる.

今後の課題としては, 評価実験を行ない, 生成コードの効率化や簡略化を考えている.

参考文献

- [1] Kjeld H.Mortensen: "Automatic Code Generation Method Based on Coloured Petri Net Models Applied on an Access Control System," ICATPN2000, LNCS1825, pp. 367 - 386, 2000.
- [2] 山口弘純, 岡野浩三, 東野輝夫, 谷口健一: "レジスタ付きベトリネットを用いた全体動作仕様から分散動作仕様の自動合成とその応用," 電子情報通信学会技術研究報告, Vol. J80-A, No. 7, pp. 1064 - 1072, July 1997.
- [3] Jens Linneberg Rasmussen and Mejar Singh: "Designing a Security System by Means of Coloured Petri Nets," ICATPN1996, LNCS1091, pp. 400 - 419, 1996.
- [4] Marco Sgroi, Luciano Lavagno, Yosinori Watanabe and Alberto Sangiovanni-Vincentelli: "Quasi-Static Scheduling of Embedded Software Using Equal Conflict Nets," ICATPN1999, LNCS1639, pp. 208 - 227, 1999.
- [5] Mohammed Elkoutbi and Rudolf K.Keller: "User Interface Prototyping based on UML Scenarios and High-level Petri Nets," ICATPN2000, LNCS1825, pp. 166 - 186, 2000.