

拡張有限状態機械で記述された協調計算プログラムとその実行系

2E-1 今城 広志 岡野 浩三 東野 輝夫 谷口 健一

大阪大学 基礎工学部 情報工学科

1 はじめに

グループウェアのような協調計算システムの設計法として、設計者はシステム全体としての動作(要求仕様)のみを記述し、その記述から、システムの各ノード(メンバー、計算機)の詳細な動作(動作仕様)を自動生成することが有効である。

我々は、有限個のレジスタを持つ拡張有限状態機械で記述された要求仕様と、入出力ゲートやレジスタの各ノードへの割当てを与え、要求仕様を複数のノードで協調して実現する際の各ノードの動作仕様を自動生成する方法を考案し、この処理系や各動作仕様の実行系、動作を視覚的に確認するツールを作成してきた。

従来の処理系では、ゲートの多重割当てには対応していなかったが、今回はこの点を改良している。

2 協調計算システムの要求仕様と動作仕様

2.1 要求仕様

設計者は、協調計算システムの要求仕様を有限個のレジスタを持つ拡張有限状態機械(EFSM)で記述する。

EFSMは、FSMに有限個のレジスタを加えたもので、図1のようにグラフで表すことができる。状態遷移を表す枝には遷移条件式、入出力動作、レジスタ更新式がラベルとして付けられている。入出力動作に関しては、例えば、 $a?x$ はゲート a から変数 x に値を入力することを、 $a!E(\dots)$ はゲート a へ式 $E(\dots)$ の値を出力することを表している。

EFSMのある状態において、その状態から出るすべての遷移の条件式を評価し、これが真となる遷移のうち1つが非決定的に選択され、入出力動作が実行される。その後レジスタ値が更新され、次の状態に移る。

2.2 ゲートやレジスタの割当て

設計者は要求仕様の他に、協調計算を実際には幾つのノードで実現するのか、また、ゲートやレジスタをどのノードに割当てるといった情報を与える。なお、同一のゲートやレジスタを複数のノードに割当ててもよい。同一のゲートが複数のノードに割当てら

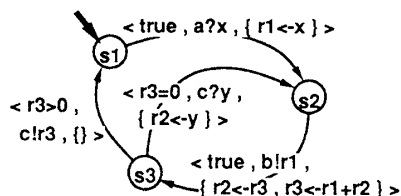


図1: EFSMの例

れている場合、そのゲートでの入力に関しては、複数のうちのいずれのゲートからも入力でき、出力に関しては、いずれのゲートからも出力される。

2.3 動作仕様

要求仕様とゲートやレジスタの割当てから、ノード間でレジスタ値などのメッセージを交換しながら全体として要求仕様を満足するように動作する以下のような各ノードの動作仕様を構成する。(a) 遷移の入出力動作に用いられるゲートをもつノードがその入出力動作を行なうようにする。1つの状態においては、各ノードに対して、そのような遷移が(複数)対応する。そこで、これらのノードのうちの1つが自分に対応された複数の遷移から実行可能な遷移を1つ選択するか、他のノードに遷移の選択権を譲る。選択権を得たノードが同様の動作を行なう。(b) レジスタの更新は、更新されるレジスタを持つ各ノードへ更新に必要なレジスタ値を送り、各ノードが更新式を計算し、行なわれる。(c) メッセージはレジスタ値の他、要求仕様に現れる全ての遷移を識別する遷移識別子を含む。これによってメッセージを受けたノードは、どの遷移が行なわれているかを知ることができる。

図2に要求仕様の1つの遷移系列と、これを実現している各ノードの動作を示す。図の破線は、ノード間のメッセージのやりとりを表す(破線横の r_1 などは、メッセージに含まれるレジスタ値)。各ノードの動作仕様は、各ノードの動作をEFSMで記述したものである。

遷移の選択に関して、図2では、ゲート a はノード1, 3に割当てられており、要求仕様の $a?x$ で、ノード1, 3のいずれかが $a?x$ を実行するが、この例では、最初のメッセージで、選択権をノード1がノード3へ譲っている。この例ではこの後ノード3が入力を実行しているが、選択権を再びノード1へ譲ることもある。

Distributed Computing Systems in Extended Finite State Machine Model
Hiroshi Imajo, Kozo Okano, Teruo Higashino and Kenichi Taniguchi
Dept. of Information and Computer Sciences, Faculty of Engineering Science, Osaka University, Toyonaka, Osaka 560, Japan

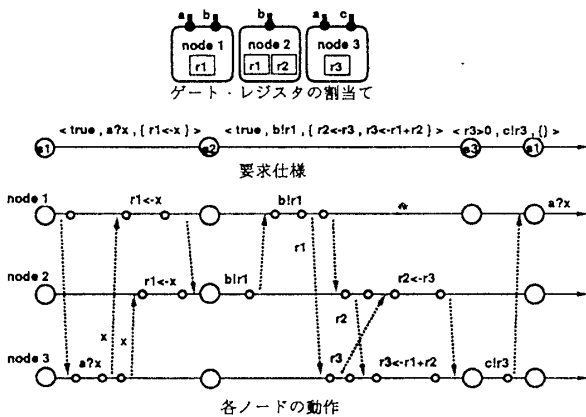


図 2: 各ノードの動作

3 協調計算システムの実現

我々は、要求仕様とゲートやレジスタの割当てから各ノードの動作仕様を自動生成するためのアルゴリズムを考案してきた^{(1),(2),(3)}。このアルゴリズムでは、要求仕様の1つの遷移ごとに、これを実現する各ノードの動作仕様の1つの遷移系列を生成する。

そこで今回、このアルゴリズムに従って、要求仕様などから各ノードの動作仕様を自動生成する生成系と、各動作仕様を実行し、全体として協調計算を実現する実行系を作成した。また、各ノードでの動作の様子を視覚的に確認するための表示系を作成した。

3.1 生成系

生成系は、要求仕様などから各ノードの動作仕様を自動生成するシステムである。

生成系に対する入力は、要求仕様 EFSM を定められたフォーマットに従って記述したテキスト (EFSM 型プログラム) のファイルの他、ゲートやレジスタの割当ての情報を書いたファイル、レジスタや入力用変数の型や構造を書いたファイル、及び要求仕様内で用いた関数を C 言語で書いたファイルの 4 つである。生成系は、これらの入力に対して、要求仕様 EFSM 型プログラムの構文チェックなどを行なった後、文献 (1), (2) の自動生成法を用いて、各ノードの動作仕様を構成し、各ノードの動作仕様 EFSM 型プログラムを記述したファイルを出力する。従来の生成系はゲートの多重割当てに対応していない。このため、要求仕様を表記上はゲートを多重割当てしていないように (自動的に) 変形⁽³⁾し、従来の生成系を用いるようにしている。

3.2 実行系

実行系は、動作仕様 EFSM 型プログラムを C プログラムへ変換する変換プログラムと、ノード間の通信

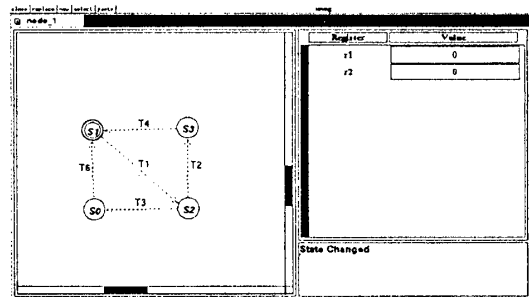


図 3: 表示系

を行なう際に用いる通信サーバから成る。

変換プログラムへの入力は、生成系によって得られた各ノードの動作仕様 EFSM 型プログラムで、出力は各ノードの C プログラムである。これらの C プログラムをコンパイルし、通信用ライブラリとリンクさせ、各ノードの実行プログラムを得る。各ノードの実行プログラムと通信サーバを並行して実行することによって、全体として要求仕様通りの協調計算が実現される。

3.3 表示系

1 つのノードの実行プログラム内で行なわれている状態遷移やレジスタの値などを可視化するために表示系を作成した。実行プログラムの実行の際、表示系を利用することができる。

表示系 (図 3) は状態遷移図、レジスタ表を常時表示しており、各ノードの動作に応じて状態遷移図やレジスタ値が変更される。

またレジスタの表示に関しては、数値・文字列以外にも静画像を表示したり音声で扱ったりして、簡単なマルチメディアに対応できるように現在改良中である。

4 おわりに

今後、協調作業関連の実用規模の例を考え、自動生成法の有用性を示すとともに、実行系に関して、多様なデータを扱えるように拡張したいと考えている。また、生成系の動作時間の計測も考えている。

文献

- (1) 岡野 浩三, 今城 広志, 東野 輝夫, 谷口 健一: " 拡張有限状態機械モデルを用いた分散システムの要求仕様から各ノードの動作仕様の自動導出", 情処学論, Vol.34, No.6, pp.1290-1301(1993-06).
- (2) T.Higashino, K.Okano, H.Imajo and K.Taniguchi: "Deriving Protocol Specifications from Service Specifications in Extended FSM Models", 13th IEEE Int. Conf. on Distributed Computing Systems, pp.141-148(May 1993).
- (3) 今城 広志, 岡野 浩三, 東野 輝夫, 谷口 健一: " 拡張有限状態機械を用いた協調作業向きの計算システム", 情処研報, Vol.93, No.58(93-OS-60, 93-DPS-61), pp.147-154(1993-07).