社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2006－MPS－59 (14)
2006／5／26

状況制約下における並行モジュールのシーケンス構成論

岩田 健一，笹倉 万里子，山崎 進
岡山大学大学院自然科学研究科

概要

本論文では状況制約下における並行モジュールのシーケンスを求めるための数理モデルとしてモジュール計算系を提案する．この数理モデルでは，状況とモジュールという２つの構成要素を用い，状況と状況の間，モジュールとモジュールの間，状況とモジュールの間の３種類の制約を記述する事ができる．モジュール計算系は複数あって，並行動作が可能である．本論文ではこのモデルにおいてある状況からある状況への移動の経路と，そのときのモジュールのシーケンスを得る手続きを示す．このモデルでは３種類の制約を記述する事ができるので，現実社会の複雑な事象を容易に記述することができる．本論文ではその例として，製造業における部品の部品の収集組み立て問題を論ずる．

The Architecture of Concurrent Modules under Situational Conditions

IWATA Kenichi, SASAKURA Mariko and YAMASAKI Susumu
Department of Computer Science,
Graduate School of Natural Science and Technology, Okayama University

Abstract

In this paper, we propose a module calculus that gives a sequence of concurrent modules as a mathematical model. The model uses *situations* and *modules*, and is able to have three kinds of constraints that are between situations, between modules, and between a situation and a module. A module caliculus works with other caliculi concurrently. In this paper, we show a procedure that gives a way to get a path from a situation to another situation and a sequence of modules. We use three kinds of constraints, therefore the model has ability to describe complex phenomena in the real world naturally. We describe a problem of parts collection and composition in a productive area as an application.

1 はじめに

本論文では，状況による制約を与えた場合に，その制約のもとで解を与える並行動作可能なモジュールのシーケンスを求める事ができる数理モデルを提案する．

この数理モデルでは，状況と，ある状況において実行可能なモジュールの集合を定義する．また，状況と状況の間の制約，状況とモジュールの間の制約，モジュールとモジュールの間の制約をあたえる．このとき，状況の初期値と終了値，実行すべきモジュールを与えると，状況の列およびモジュールの列を得る事ができる．またこのときに，モジュールの列はひとつだけ得る事もできるが，複数の並行動作可能なものを得る事も可能である．

この数理モデルを利用すると，状況の初期値と終了値，その間に満たすべき実行モジュールを与えると，状況を遷移中に実行するべきモジュールの列であって，並行動作可能なものを得る事ができる．またおのおののモジュールの列について，状況の遷移の列を得る事ができる．

この結果の実際的な意味は様々に解釈が可能である．経営や事業活動の効率化が叫ばれて久しいが，製造業においてもトヨタカンバン方式 [1, 3] に代表され

る高効率な生産活動が追求されている．すなわち旧来の少品種多量生産から，多品種少量生産への転換である．多品種少量生産の現場においては，組立てに関しても短時間に生産する対象の製品が変わる事が多く，また，BTO (Build To Order:受注生産) の普及などにより，注文ごとに微妙に異なる組立てスペックへの対応を迫られる．このため，組立て現場への部品の供給や，そのために倉庫から部品を集める作業など，古くは大きな問題とは考えられなかった作業での効率化を計っており，その多くを人間の判断に依存している．

ここで，状況を生産者が位置を占める工場における区画とし，また，モジュールを生産する製品または利用する部品とすると，本システムにより，生産者の初期位置と終了位置，目的の生産物を与えると，システムにあらかじめ与えられた制約のもとにおける生産者の区画の移動の列と，ある区画における行動の列を得る事ができる．

すなわち，本論文で提案する数理モデルを利用すれば，今まで人間に頼っていた判断の少なくとも一部を自動化する事ができるため，複雑な作業が要求される少量多品種生産の現場における生産活動の助けとなるようなシステムを構築することができる．

2 数理モデル

数理モデルの定義を次に示す．モジュール計算系は

$$\Im = (P, \Sigma, Process, Situatoin, I)$$

で定義される．ここで，

(i) P はモジュールの集合．空モジュールを示す ϵ も含む．

(ii) Σ は状況の集合．

(iii) $Process$ はモジュールの集合 P 上のルールの集合．

(iv) $Situation : P \rightarrow \Sigma_p$ は関数．ただし，Σ_p は Σ の部分集合．

(v) $I \subseteq \Sigma \times \Sigma$ は制約関係．I は反射的推移閉包である．

$Process$ は空集合ではなく，その要素であるルールは以下の形で記述される．

$A : B_1, \ldots, B_n \quad n \geq 1$, または $A : \perp$ ここで $A, B_1, \ldots, B_n \in P$ である．$A : B_1, \ldots, B_n$ はモジュール A が $B_1, \ldots, B_n$ から構成されることを示している．$A : \perp$ は A が基本要素であることを示し，例えば，このモジュールが倉庫の中で見つかることを示している．

$Situation$ はあるモジュールに対する状況の集合である．直観的にはあるモジュール A は $Situation(A)$ の要素である状況に存在する．$Situation(\epsilon) = \Sigma$ とする．

I は状況の対の集合で，直観的には $(\sigma_i, \sigma_j) \in I$ は，状況 σ_i から σ_j に状況遷移できることを示す．

並行モジュール計算系は，モジュール計算系の組として定義される．

$$\Pi = (\Im_1, \ldots, \Im_n)$$

ここで $\Im_i = (P, \Sigma, Process_i, Situation_i, I)$ $(1 \leq i \leq n)$ はモジュール計算系である．

並行モジュール計算系における述語 $gather_{P_i}(p; \sigma_1, \sigma_2)$ $(1 \leq i \leq n)$ は図 1 で定義される．ここで $p \in P$ で $\sigma_1, \sigma_2 \in \Sigma$ であり，Γ, Γ_1, Γ_2 はモジュールの列 $B_1, \ldots, B_m$ $(m \geq 1)$ である．また，モジュールの列 Γ_1, Γ_2 の列はモジュールの列とする．

定義 1 意味関係 $SEM \subseteq \Sigma \times P^* \times \Sigma$ を次のように定義する．

(1) $SEM(\sigma_1; \epsilon; \sigma_2) \Leftrightarrow (\sigma_1, \sigma_2) = I$

(2) $SEM(\sigma_1; \gamma, x; \sigma_3) \Leftrightarrow \sigma_2 \in Situation(x)$, $(\sigma_2, \sigma_3) \in I, SEM(\sigma_1; \gamma; \sigma_2)$

定理 1 $gather_{P_i}(A; \sigma_1; \sigma_2) \Rightarrow \exists \beta \in P^*.\ SEM(\sigma_1, \beta, \sigma_2)$

証明は [2] を参考に与える事が出来る．

このとき，すべての再帰的に呼ばれた手続き $gather_{p_i}(Module; situation_1, situation_2)$ のうち，図 1 の式 (1), (2), (3) によって成り立ったものに関して，部品 $Module$ および初期状況 $situation_1$ と終了状況 $situaton_2$ を接続すれば，これは状況の遷移の列とモジュールの列であって，並行動作なものが得られる．この手続きは以下のように定義できる．

Procedure gather-path$(Module_k; \sigma_i, \sigma_j)$:
begin
　if 図 1 の (1) then
　　return σ, A, σ
　else if 図 1 の (2) then
　　return $\sigma_1, \sigma_3, A, \sigma_3, \sigma_2$
　else if 図 1 の (3) then
　　return $\sigma_1, \sigma_2, A, \sigma_2, \sigma_3$
end

(1) $\dfrac{(A:\bot)\in Process_i \quad \sigma\in Situation_i(A)}{gather_{P_i}(A;\sigma,\sigma)}$

(2) $\dfrac{(A:\bot)\in Process_i \quad \sigma_3\in Situation_i(A) \quad (\sigma_1,\sigma_3)\in I \quad (\sigma_3,\sigma_2)\in I}{gather_{P_i}(A;\sigma_1,\sigma_2)}$

(3) $\dfrac{(A:\Gamma)\in Process_i \quad gather_{P_i}(\Gamma;\sigma_1;\sigma_2) \quad \sigma_2\in Situation(A) \quad (\sigma_2,\sigma_3)\in I}{gather_{P_i}(A;\sigma_1;\sigma_3)}$

(4) $\dfrac{gather_{P_i}(\Gamma;\sigma_1;\sigma_2) \quad (j\neq i) \quad gather_{P_j}(A;\sigma_2;\sigma_3)}{gather_{P_i}(\Gamma,A;\sigma_1;\sigma_3)}$

(5) $\dfrac{gather_{P_j}(A;\sigma_1;\sigma_2) \quad (j\neq i) \quad gather_{P_i}(\Gamma;\sigma_2;\sigma_3)}{gather_{P_i}(A,\Gamma;\sigma_1;\sigma_3)}$

(6) $\dfrac{gather_{P_i}(\Gamma_1;\sigma_1;\sigma_2) \quad gather_{P_i}(\Gamma_2;\sigma_2;\sigma_3)}{gather_{P_i}(\Gamma_1,\Gamma_2;\sigma_1;\sigma_3)}$

図 1: 述語 *gather* に関する規則

3 応用

この章では，部品の収集組み立て問題に関する応用と，試験的な実装について示す．

3.1 部品の収集組み立て問題

部品の収集組立て問題について説明する．

すでに経営や事業活動の効率化が叫ばれて久しく，製造業においてもトヨタカンバン方式に代表される高効率な生産活動が追求されてきた．高効率な生産とは，すなわち旧来の少品種大量生産から，多品種少量生産への転換に他ならず，また，資本効率の向上の観点からは，余剰在庫の圧縮がもとめられる．

たとえば，大型計算機の組立てに関して考えてみると，その日の朝には一日の生産計画が決まっているのが通例である．最初の製品の組立て指示がプリントアウトされる．BTO によって顧客から指示された組立て仕様がもりこまれた，組立て指示が与えられるため，組立て毎に組立てる手順や製品が異なる．まずは組立て指示書をみて，必要な部品を倉庫から集めなければならない．すべての部品を一度に組立て現場に集めてくることは非効率であるため (部品の収集に時間がかかる，一時的な置き場所が必要になるなどの不利益がある)，必要とされる部品から先に集めなくてはならない．組立てにおいてどの部品が先に必要とされるかは，製品毎に異なるため，通常は組立て指示書では指示されない．ひとの判断で行うのである．通常は筐体が先に必要とされる．次は電源や，ディスクドライブなどになるであろう．

本論文で提案する数理モデルを利用すれば，今まで人間に頼っていた判断の少なくとも一部を自動化する事ができるため，少量多品種生産の現場における生産活動の助けとなるようなシステムを構築することができる．

3.2 実装システムについて

前記のシステムについて，試験的な実装を行った．実装は Java で行い，規模は 3600 行程度である．モジュール計算系をオブジェクトとして実装し，内部で再帰的に *gather* を呼ぶようになっている．また，*gather* の実行結果から，状況の遷移と実行モジュールの列を得るような仕組みとなっている．

図 1 の (4),(5),(6) の適用に関しては非決定的であるため，実装側で工夫する必要がある．今回の実装では，まず (6) によって，ひとつのモジュール系のみで解決を試み，成り立たなかった場合に (4), (5) を試行するようにした．このようにすると，モジュール系にもたせるルールを分割する事により，モジュール系ごとに役割を分担させる事ができる．

また，(2),(3) においては，$\sigma \in Situaion(A)$ となるような σ の探索が必要となる．これは $\sigma \in Situaion(A)$ となる σ が一意に決まらない場合，どれを選択するかは実装にまかされている．あまりかけ離れた位置の σ を選ぶと非現実的な解となるので，制約関係で定義されるもののうち，なるべく近いものを選ぶよう，広さ優先探索をして σ を求めるようにしている．すなわち (2) においては状況の制約関係 I によって移動が可能であって，かつ σ_1 に近いものから順に探し，はじめに見つかった $\sigma \in Situaion(A)$ について返す．(3) においては逆に σ_3 から逆向きに近いところを優先して探索し，一番始めに見つかった σ_2 を返す．

さらに上記で見たように，(2),(3) においては，状況の制約関係 I によって移動可能な状態の探索が必要になる．これも探索の順序は実装に依存する．上記にあうよう，広さ優先で探索するように実装している．

3.3 入力ファイルについて

状況やモジュールの集合，その間の制約条件などはファイルの形で与える．

- 状況の集合
- モジュールの集合
- モジュールと状況の間の制約
- モジュールとモジュールの間の制約
- 状況と状況との間の制約

3.4 実行例

下記のような状況とモジュールを与える．

状況の集合 $s1, s2, s3$

モジュールの集合 X, Y, Z

モジュールと状況の間の制約

$X : s1$
$Y : s2$
$Z : s3$

モジュールとモジュールの間の制約

$X :$
$Y : X$
$Z : X, Y$

状況と状況の間の制約

$s1 : s2$
$s2 : s1$
$s2 : s3$
$s3 : s2$

直観的な意味としては，図 2 に示したように，$s1, s2, s3$ という区画があり，互いに移動可能であって，区画 $s1$ には部品 X が在庫されており，区画 $s2$ では，部品 X を利用して部品 Y を組立てる事ができる．また区画 $s3$ では部品 X と部品 Y を利用して製品 Z を組立てる事ができる．

このとき，実行モジュールとして Z，初期状況として $s1$，終了状況として $s3$ を与えると，図 3 のような実行結果が得られる．意味は，区画 $s1$ で部品 X を 2 つ収集し，そのあと区画 $s2$ へ移動して，部品 X をひとつ使用して部品 Y を組立て，そのあと区画 $s3$ へ移動して，部品 X と部品 Y から製品 Z を組立てるというものである．

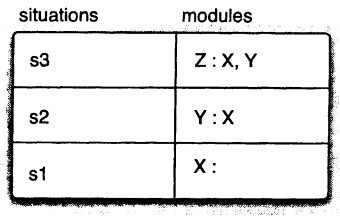

図 2: 実行例の状況とモジュール

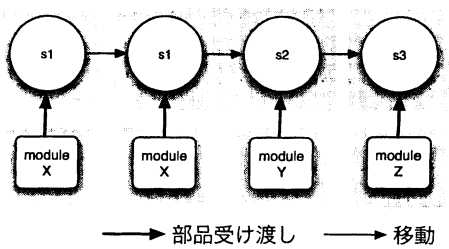

図 3: 実行結果

4 おわりに

本論文では状況制約下における並行モジュールのシーケンスを求めるための数理モデルとしてモジュール計算系を提案した．

また，その数理モデルを利用した応用として，並行部品収集組立て問題への適用を論じ，3 種類の制約を記述する事によって，現実社会の複雑な事象を自然に記述することができることを示した．

実世界の問題への応用に際しては，問題の規模が大きくなる事などから，高い実行性能が求められる．しかし本論文であつかった数理モデルには，解を求めるための経路が最短であるという保証はない．この点は実装で工夫すべき点であり，今後の課題である．

参考文献

[1] Kanban just-in-time at Toyota:management begins at the workplace, Japan Management Association, Productivity Press Inc.,1989.

[2] Yamasaki,S. and Sasakura,M., A calculus effectively performing event formation with visualization, Proc. of ISHPC-VI (CD-ROM), 2005.

[3] 大野 耐一, トヨタ生産方式—脱規模の経営をめざして, ダイヤモンド社, 1978.