

# アクティビティモデル間の変換に基づくFAシステムのモデリングフレームワーク — シーケンス制御システムの制御プログラミングへの応用 —

旭川工業高等専門学校 ○戸村 豊明 北海道大学工学部 金井 理, 岸浪 建史

## 要 旨

本研究は、制御プログラミング、離散事象シミュレーションといったさまざまなアプリケーションで共通利用可能な部分を持つ作業、制御対象のアクティビティモデルをフレームワークとして提案し、モデル間の相互関係と変換過程を明らかにすることを目的としている。本報では、状態遷移システムとしてみなせるシーケンス制御システムにおいて、制御プログラムのアクティビティモデルを導入し、モデル間の相互関係、変換過程から制御プログラムを開発する手法に関して報告する。

### 1 はじめに

FAシステムのモデル化は、制御プログラミング、離散事象・機器動作シミュレーション等のある目的ごとに固有のモデル化が行われている。アプリケーションごとにモデルへ要求される情報はそれぞれ異なるが、作業、制御対象に関しては、アプリケーションに依存しない共通部分が多いと考えられる。しかし、現状ではその共通部分についてもモデル間の互換性がなく、かつ共通利用がはかられていない。

そこで本研究では、異なるアプリケーションで共通利用可能な部分を持つ作業モデル、制御対象モデルをフレームワークとして提案し、その手段として、アクティビティモデル、状態遷移モデルを IDEF0, Petri net で表現し、NC 工作機械システムを例にモデル間の相互関係と変換過程を明らかにした<sup>[1]</sup>。本報では、PLC、アクチュエータ、センサからなるシーケンス制御システムを取り上げ、そのフレームワークはどのように書けるのか、モデルのアプリケーションとして制御プログラミングを考えた時、どのようにフレームワークを利用可能であるのかについて議論する。

### 2 アクティビティモデルから状態遷移モデルへの変換

FAシステムの作業、制御対象、制御プログラムのアクティビティはそれぞれ、ワーク、制御対象、解釈器の状態遷移を伴う事で進行される。本研究では、アクティビティのモデル化ツールとして IDEF0, 状態遷移のモデル化ツールとして Petri net を適用する。IDEF0, Petri net の表記法を図1に示す。IDEF0における全入出力は、Petri netにおける状態Placeへ置き換えられる。アクティビティの開始は、入力Placeから状態PlaceへのTransitionの発火、アクティビティの終了は、状態Placeから出力PlaceへのTransitionの発火であると考ええると、IDEF0からPetri netへの変換が可能となり、構成要素間の対応関係は表1のように表される<sup>[2]</sup>。

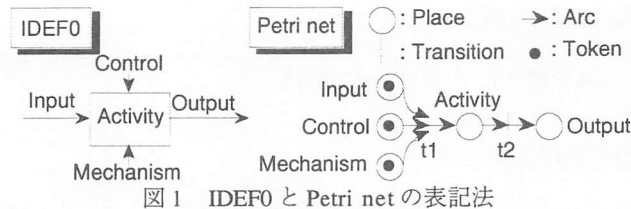


図1 IDEF0とPetri netの表記法

表1 IDEF0とPetri netの構成要素間の対応関係

IDEF0	Petri net
Activity	Place and two Transitions
Input	Place
Control	Place
Mechanism	Place
Output	Place

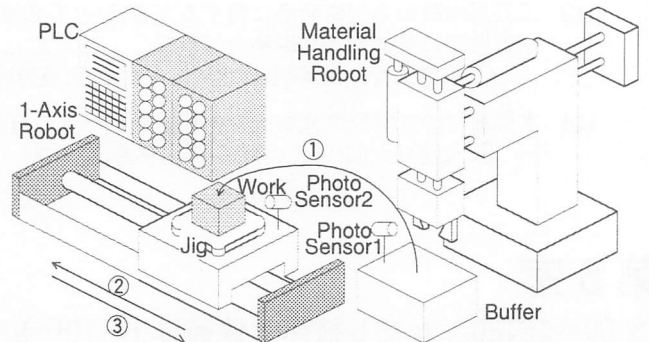


図2 本報で取り上げるシーケンス制御システム

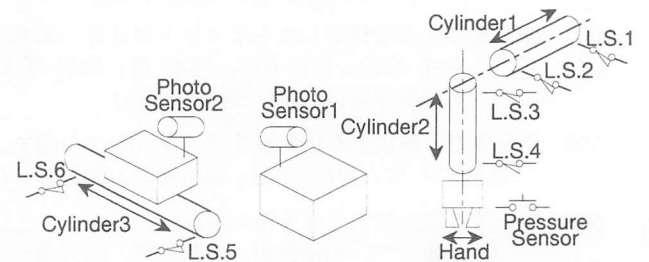


図3 本報で取り上げるシーケンス制御システムの回路系

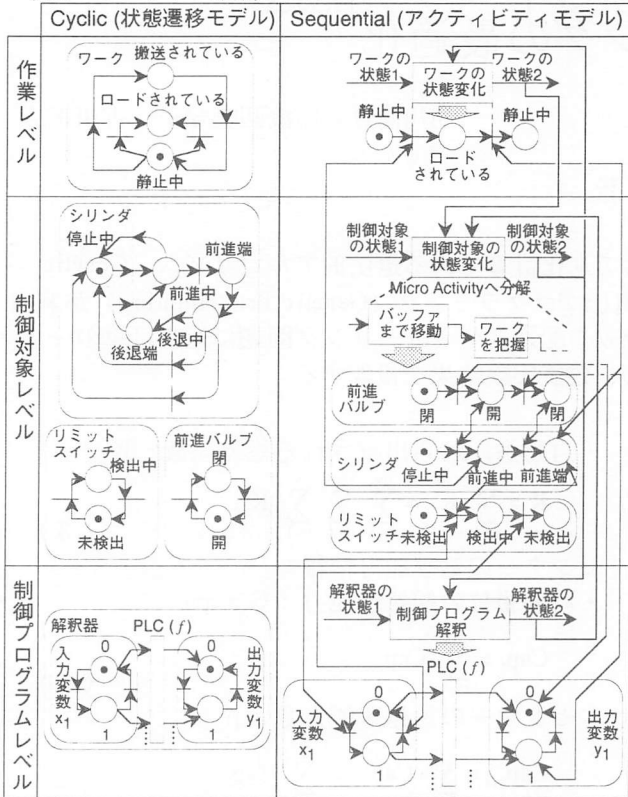
### 3 シーケンス制御システムモデルのフレームワーク

図2のようなPLC、単純動作のロボット、各種センサからなるシーケンス制御システムを対象とした時、このモデルのフレームワークは、作業、制御対象、制御プログラムレベルの3階層に分割する事で、表2のように表される。表2の左側は各レベルにおけるオブジェクト固有の振る舞いを、循環的な状態遷移モデルとして表現したもので、特に制御対象レベルでは、図3のような回路系の構成要素が組み込まれる。表2の左側は、モデル化対象となるFAシステムの種類に応じて、状態遷移モデルの差し替えが可能である。

表2における各レベルのアクティビティモデルは、左側の循環的な状態遷移モデルを時系列状に引き延ばした状態遷移モデルへ変換可能である。但し、制御対象レベルでは、複数の機器の状態遷移が並行して起こる場合がある。

各レベルにおけるモデル同士の因果関係は、隣り合うレベルにおけるモデル同士の状態遷移の関係を表している。例を挙げれば、作業レベルと制御対象レベルにおいて、ワークはセンサの状態遷移を引き起こし、アクチュエータはワークの状態遷移を引き起こすという関係である。

表2 シーケンス制御システムモデルのフレームワーク



4 フレームワークを利用したシーケンス制御プログラミング

表2のフレームワークを利用したシーケンス制御プログラミングにおけるモデル変換手順は、図4のようになる。これは、表2の作業レベルから制御プログラムレベルまで順番にアクティビティのモデル化を行いながら、モデル間に因果関係を定義する事に等しい。

図2のシーケンス制御システムを対象とした作業レベルのアクティビティのモデル化の具体例を図5に示す。図5において、機器構成、要求作業がどのような記述で与えられるのかは、本報では議論しない。図5のアクティビティ①に対応する制御対象レベルのアクティビティのモデル化の結果を図6に示す。アクティビティモデルにおいて、センサはControl、アクチュエータはMechanismとして入力される。また、センサの状態はアクティビティの開始・終了のための条件を表し、そのアクティビティ実行時のアクチュエータの状態遷移が因果関係に従って、他のオブジェクトへ伝搬する。さらに、図6のアクティビティ②に対応する制御プログラムレベルのアクティビティのモデル化結果を図7に示す。図7のアクティビティモデルにおいて、入力変数の状態遷移は、制御対象の状態遷移により引き起こされ、出力変数の状態遷移は、制御対象の状態遷移を引き起こす。

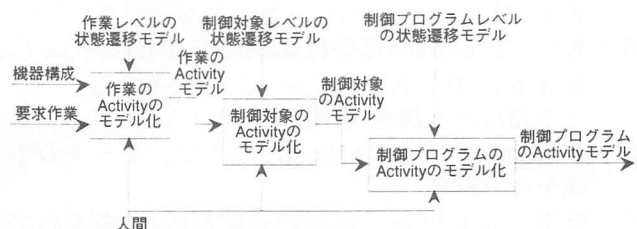


図4 フレームワークを利用した制御プログラミング用のモデル変換手順

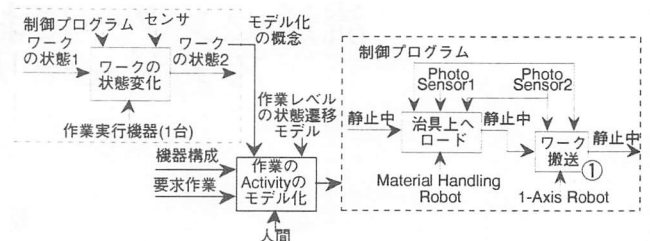


図5 作業のアクティビティのモデル化の具体例

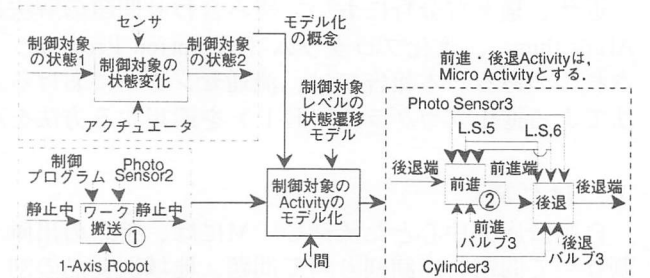


図6 制御対象のアクティビティのモデル化の具体例

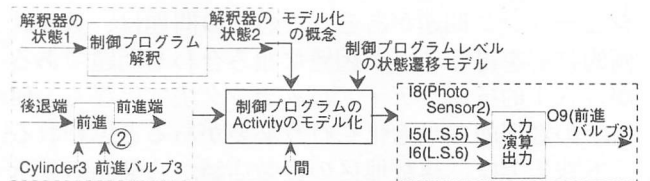


図7 制御プログラムのアクティビティのモデル化の具体例

表2のように各レベルのアクティビティモデルをSequentialな状態遷移モデルへ変換し、モデル間の因果関係を定義すれば、作業レベルにおける各アクティビティを実行するための入出力変数とその値の組み合わせが明らかになるので、入力変数を論理的に組み合わせ、その出力変数値を得られるようなPetri netモデルを、表2の制御プログラムレベルにおける状態遷移モデルの関数修飾 Transition<sup>[1]</sup> PLC(f)へ実装すればよい。

5 結論と今後の課題

本報では、シーケンス制御システムを例に、FAシステムのモデルのアプリケーションで共通利用可能な作業モデル、制御対象モデルのフレームワークを提案し、モデル間の相互関係と変換過程を明確化してから、アプリケーションとしてのシーケンス制御プログラミングへのフレームワークの利用方法について議論した。

今後は、フレームワークから得られたモデルを制御プログラミング、事象駆動シミュレーション等のアプリケーションへ実装する事で、フレームワークの有効性を明らかにしたい。

参考文献

[1]戸村, 金井, 岸浪: FA 制御対象のフレームワークに関する基礎研究, 1997年度精密工学会秋季大会学術講演会講演論文集, 1997(発表予定).  
 [2]Kzyszt of Santarek, Ibrahim M.Buseif: Modelling and design of flexible manufacturing systems using SADT and Petri nets tools, Proceeding of 13th International Conference on Computer-Aided Production Engineering(Proc. of 13th C.A.P.E.), warsaw, pp.385-392, June, 1997.  
 [3]電気学会編: シーケンス制御工学, オーム社, 1992.