

ワークセル・コントロール・システム

- モデルとユーザインタフェース -

6S-7

天明 崇 長谷川 雅樹

日本アイ・ビー・エム株式会社 東京基礎研究所

1. はじめに

ワークセル・コントロール・システムは、ワークセル内の生産機器を柔軟に制御することを目的とする統合ソフトウェアである。本システムは、ワークセルのプロセスを記述するためにプロセス・モデルを導入し、その視覚化をはかった。さらにワークセル内の生産機器の制御を記述するためにマシン・モデルを導入し統一的な記述により生産機器の柔軟な制御ができるようにした。この二つのモデルをシステム内で統一して扱える二階層プログラミングを実現し、実環境で使用するためのユーザ・インターフェースを構築した。

2. プロセス・モデル

ワークセル内の生産プロセスは、いくつかの作業プロセスに分割できる。この作業プロセスを以後タスクと呼ぶ。タスク間につながりを作り生産プロセスの流れを作成する。ワークセル内の生産プロセスの流れは、典型的な離散型の事象の系列と考えられ、あるタスクが実行を終了すると他のタスクが実行される。その実行形態の特徴は、非同期性にある。しかしながら現実の生産プロセスには、タスク間で同期をとる場合がある。このためには、タスク間につながりに同期をとるための機構を付け加えなくてはならない。この機構は、複数のタスクの終了を待って次のタスクを実行したり、あるタスク終了後に次の複数のタスクの実行を同時におこなう機構である。これはタスク間を制御が移動する条件と考えることができる。さらに、タスクの実行状態を示す機構を作ると、このモデルは、安全性かつ活性の性質を持つ制限された(多重枝なし、自己ループなし)ペトリネットと等価である。

本システムにおけるプロセスの記述法は、タスクをプレースで、タスク間を制御が移動する条件をトランジションで、タスクと条件のつながりをアークで表現する。タスクの実行状態は、トークンとして表す。プロセスの流れは、左から右に流れてゆく。

さらに現実のシステムでは、外部信号がタスクの実行を開始する条件となる場合がある。ここでいう外部信号とはセンサ、スイッチ、タイマ、カウンタ、サブシステムからの信号である。この外部信号がタスクの実行を開始する条件を、タスク間を制御が移動する条件に対して、外部条件とする。複数の外部信号ある場合は、それらの外部信号の論理値を論理演算し、その結果の論理値を、その外部条件の代表値とする。この外部条件を前述のペトリネット・モデルに加えることによってペトリネットの拡張がおこなわれる。

本システムにおける外部条件の記述法は、アローで表現しトランジションに付加する。

本システムで用いているプロセス・モデルの拡張ペトリネットGは次のように表される。

$$G = (P, T, I, O, M, E)$$

ここで、

$$P = \{P_1, \dots, P_i\} \quad i > 0 \quad \text{プレースの集合}$$

$$T = \{T_1, \dots, T_j\} \quad j > 0 \quad \text{トランジションの集合}$$

$$P \cap T = \emptyset$$

$$I : T \rightarrow P \quad \text{入力関数}$$

$$O : P \rightarrow T \quad \text{出力関数}$$

$$M : P \rightarrow \{0, 1\} \quad \text{マーキング関数}$$

$$E = \{E_1, \dots, E_k\} \quad k > 0 \quad \text{外部信号の集合}$$

である。

トランジションの発火によってプレース間を制御が移動していくわけである。Fig.1 は、発火規則を示すための拡張ペトリネットの図である。

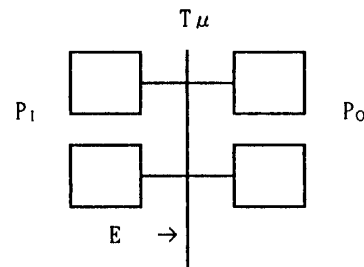


Fig.1 発火規則を示すための拡張ペトリネット

拡張ペトリネットにおける発火条件、トークンの生起、トークンの消滅は、次のようになる。

1) 発火条件式

$$T_{\mu}(t) = \bigwedge_{i=1}^M P_i(t) \wedge \bigwedge_{j=1}^N P_{o_j}(t) \wedge E(t)$$

2) トークンの生起式

$$P_{o_j}(t+1) = P_{o_j}(t) + T_{\mu}(t), \quad j=1 \dots N$$

3) トークンの消滅式

$$P_i(t+1) = P_i(t) - T_{\mu}(t), \quad i=1 \dots M$$

Fig.2 は、拡張ペトリネット・モデルによるプロセス・モデルの例である。

このモデルによって記述できるプロセスの種類は、連続プロセス及び並列プロセスである。連続プロセスとは、Fig.2 の中で TASK2 の実行後 TASK3 が実行がおこなわれる様に、順次タスクが実行されてゆくプロセスである。並列プロセスとは、Fig.1 の中で TASK4 と TASK5 が同時に実行される様に、二つ以上のタスクが並列に実行されてゆくプロセスである。この二種類のプロセスが記述できれば、複雑なプロセスは、これらの組合せで実現できる。またこのモデルでは、外部信号によるタスクの同期及びタスクの開始が、外部条件として取り扱える。このモデルを用いれば、現実の生産プロセスの記述は充分におこなえる。

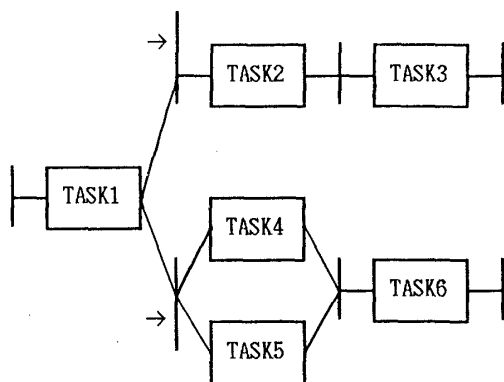


Fig.2 プロセス・モデル

3. マシン・モデル

ワークセル内の生産機器は色々あるが、ここでは柔軟な制御ができる生産機器であるプログラム可能な機器（プログラマブル・ロジック・コントローラ、ロボット、パーソナル・コンピュータ、等）を生産機器ユニットとしてモデルの対象にしている。さらに、単純な生産機器でもプログラマブル・ロジック・コントローラ、パーソナル・コンピュータを介して制御すれば、システムとしては柔軟な制御ができるので、これらもここで取り扱うモデルの対象の生産機器ユニットしている。これらの生産機器ユニットを以後マシンと呼ぶ。ここで定義したマシンの性質は、内部状態を保持し、制御のためのコマンドが規定され、さらに通信機能を持つので、このモデルは対象指向モデルとして取り扱うことができる。

Fig.3 は、マシン・モデルの例である。

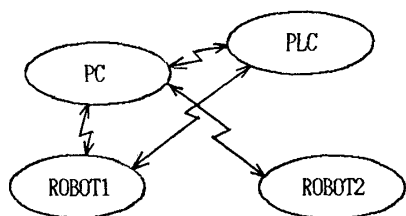


Fig.3 マシン・モデル

本システムにおけるマシンの記述法は、マシンをクラス・オブジェクトと設定し、マシンの制御コマンドはインスタンス・メソッドとする。マシンに論理名が付いたときインスタンス・オブジェクトとなる。マシンに新しい拡張タイプができたり、ユーザがコマンドを拡張した場合には、インヘリタンスを用いて、新しいマシンのクラス・オブジェクトを設定できる。

このモデルによって記述できるマシンは、前述の生産機器ユニットである。またマシンに対するインスタンス・メソッドの共有化をおこない、マシンの制御コマンドの統一化をおこなっている。本システムでは、マシンの論理名と統一コマンドを組合せて生産機器統一言語と呼んでいる。

4. ユーザ・インターフェース

ここで取り上げているプロセス・モデル及びマシン・モデルが、実際のワークセル内のプロセス及び生産機器

と対応するためには、ユーザの視点とモデルが共有化されなくてはならない。このために本システムでは、ワークセルの実環境とモデルの対応付け及びモデルの持つ共有変数は、環境テーブルに記述する。ユーザは、プロセス・モデルに従ってプロセスを記述し、次にプロセスの中のタスクを生産機器統一言語を用いて記述する。タスクの記述は複数のマシン・モデルによって記述することができるので複雑な作業も容易に記述がおこなえる。これによって本システムの特徴である、二階層プログラミング（初めにプロセスのプログラミングをおこない、次にタスクのプログラミングをおこなう）がおこなえる。

現在、本システムがパーソナル・コンピュータ上に提供する環境は、プログラミング環境、デバッグ環境、実行環境及びモニタ環境である。[1][2] プログラミング環境では、二階層プログラミングを実現している。プロセス・プログラミングにおいては、プロセス・モデルを視覚化してビジュアル・プログラミングで実現している。プロセス・プログラミングは、グラフィクス・エディタを用い、タスク・プログラミングにはテキスト・エディタを用いる。さらにマルチ・ウィンドウによってプロセス・プログラミングとタスク・プログラミングが同時におこなえる。またプロセス・プログラミングをおこなっているグラフィクス・エディタからタスク・プログラミングをおこなうテキスト・エディタを呼ぶことができる。

次にデバッグ環境、実行環境及びモニタ環境では、本システムはマシンごとに統一コマンドから生産機器固有の言語への変換ライブラリを持っているので、生産機器統一言語を用いてデバッグ、実行及びモニタがおこなえる。

5. おわりに

ここで提案しているプロセス・モデル及びマシン・モデルは、ワークセル内のプロセス及び生産機器の制御に関する概念を抽象モデル化したものである。ここでは、この二つのモデルに基づく、プロセス・プログラミングから生産機器統一言語を用いて記述するタスク・プログラミングまでを二階層プログラミングとしている。本システムでは、マルチ・ウィンドウを用いて二階層プログラミングのユーザ・インターフェースを実現した。さらに、プロセス・プログラミングの視覚化及び生産機器統一言語により、初心者から専門家までワークセルのプログラミングが容易になった。

参考文献

- [1] 天明 崇、長谷川 雅樹
『ワークセル・コントロール・システム (1) プログラミング』
情報処理学会第35回全国大会論文集, 1987
- [2] 長谷川 雅樹、天明 崇
『ワークセル・コントロール・システム (2) コントロール』
情報処理学会第35回全国大会論文集, 1987
- [3] 長谷川 健介、高橋 宏治、増田 良介、大野 秀嶺
『非連続生産システム制御のためのマーク流れ線図の提案』
計測自動制御学会論文集, 1984
- [4] 村田 智洋、薦田 憲久、松本 邦顕
『ペトリネットに基づく高機能MFA制御システム』
第9回計測自動制御学会システムシンポジウム, 1983
- [5] J.L.Peterson
『Petri Net Theory and the Modeling of Systems』
Prentice-Hall, 1981