

セルコントローラ・IFCの制御言語設計

7T-1

江藤 博明 高田 博史

日本アイ・ビー・エム株式会社 東京基礎研究所

日本アイ・ビー・エム株式会社 製造技術開発

1 はじめに

我々はIFC (Integrated Flexible Controller) [1] というセルコントローラの開発を行なっている。IFCとはワークセルを開発する際の核になるコントローラで一体型で単一の言語による開発を可能にするものである。

従来の生産ラインにおけるワークセルはロボット、エアバルブ、DI/DOセンサー、オペレーションパネル等の多くの機器の複合体である。これらの機器の制御には独自の制御言語の習得が必要である。また複数の機器間で同期をとったり相互排除を行ないながらリアルタイムに制御しなければならない。すなわちワークセルの開発ではハードウェア設計（機器間の結線）とその設計に応じたソフト開発は不可分の関係にある。これらの状況からシステム開発やメンテナンスは困難なものになっている。

一体型コントローラであるIFCではC言語による単一開発と本稿で述べるグラフィカルな言語を持ち、特に本稿で述べるシステムでは1) ハードウェア依存部分の分離および2) アプリケーションと言語の対応を明確にすることを目的に設計した。

コントローラの対象とするアプリケーションは同期を取りながら逐次的に処理していくもの（プロセス型）とイベントに応じた処理をするもの（イベント型）およびその複合型に大別できる。従来の製品 [3][4] ではどのアプリケーションでも対応できるように設計されているため、記述の単位が原始的命令になりがちでアプリケーション記述の部品としての役割分担が不明確であった。

本稿ではプロセス型アプリケーションを対象にする。アプリケーションをしぼるにより言語設計をアプリケーションの立場から検討できるようになる。以下ではプロセス型アプリケーションの立場から、アプリケーションを表現するための用語について整理を試み（節2）、その結果得られた用語にそって言語仕様を決定する。（節3）言語仕様ではセル装置固有の要件の一つ安全性についても取り上げる。

2 用語の定義

ここではプロセス型アプリケーションで使用される用語について検討する。ここではワークセル内で扱っている対象を製品、それに組み付ける物を部品と呼んでいる。

• 工程、小工程

ワークセル内で行なっている処理はそのワークセルが扱っている製品に対する処理であり、部品の組みつけや調整などの処理の連続として実現される。この処理の連続を工程と定義した。工程には製品を連続して投入可能なものと処理が終るまで投入できないものがある。連続して投入可能な工程も分割していくと一つの製品のみを処理する工程へと分割できる。この一つの製品のみを処理する工程を小工程と呼ぶ。すなわち工程は小工程の集まりである。

• 作業場、投入条件

ワークセル内で行なわれている処理は部品の組みつけや調整などの処理の連続であり、製品なり部品に着目すると処理に応じた場所がある。これを作業場と呼ぶ。個々の処理で同一の作業場を利用することもあるし別の作業場へと移動を伴うこともあるが、それぞれの処理はそれを開始するための前提条件を持っている。この条件を投入条件と呼ぶ。

• 作業

作業場における処理の一つ一つを作業と呼ぶ。作業は入出力を伴う処理の単位であり、DIDOへの出力、RS232Cからの入出力、ロボットへの指令等である。

• 工程の歩調合わせ

ワークセル内で行なわれている処理は製品に関する工程、その製品に組みつける部品に関する工程、製品をテストする工程等のように複数の工程から構成される。この複数の工程はワークセル内で処理する製品に合わせて処理している。このような同期処理を歩調合わせと呼ぶ。

パイプライン的に処理される工程も工程の歩調合わせで説明づけられる。パイプライン的な処理とは連続している小工程においてそれぞれの小工程が歩調を合わせて次の小工程へと製品を送り出すという処理形態を指す。この場合には連続する小工程の終りと始まりで工程の

歩調合わせをしていると考える。

• 工程の合流

ワークセル内で行なわれている部品の組みつけ処理を工程の合流と捉える。組み付けられた部品に関する工程は組みつけ後になくなり、製品の工程が続く。一般にワークセルの内部処理では幾つもの工程が生成されるが最終的にはワークセルから出力される製品に関する工程に合流する。

• 工程の分岐

工程の中で処理対象の型、エラー情報等の状況により後処理に条件分けが生じる場合を工程の分岐と捉える。工程は製品の流れであるから分岐では必ず一つだけの工程に流れることになる。そこで工程の分岐は排他的なものであると分析した。

3 言語仕様

この節では言語仕様について説明する。アプリケーション定義は入出力の定義と工程の定義の二つから構成される。入出力定義はアプリケーション本体からの入出力の分離という目的で設計した。工程定義は節2で抽出した用語を実現する機能部品として設計した。

3.1 入出力の定義

入力定義は各種のデバイスと工程定義内で参照される名称の関連づけのために入力名、デバイス名、ポート番号および入力ディレイからなる。この定義によりアプリケーションプログラムとハードウェアの依存関係が分離される。

同様に出力定義は出力名称、安全回路条件および出力処理からなる。安全回路条件の役割は出力の直前に安全回路（非常停止ボタン等）の監視をすることである。ワークセルアプリケーションでは安全に対する配慮が重要視されている。このような重要度にあわせた記述方法の導入がメンテナンスに効果を発揮する。

3.2 工程の定義

工程とは節2で分析した構成要素からなる。それを階層関係で表現すると図1の左側で表される。これは左側の項目は右側の項目から構成されることを表している。また工程同士の関係として工程の歩調合わせ、工程の分岐や工程の合流が用いられる。サンプルとして実際のセル装置の処理の一部を書き移してみた。

動作を簡単に説明する。工程それぞれは並列に動作するものである。工程の最初の作業場の投入条件が真となるとその工程は活動状態になる。このとき作業場に定義された作業を順次実行する。作業の実行は右に記された文字列は入出力として定義されたものであり、そこで定

義された作業を実行する。次の作業場への実行遷移は次の投入条件によって決定される。条件が真となるまで工程は待機状態になる。投入条件部品の右に記された文字列は条件名称であり別途入力の論理演算として定義する。入力状態を常時監視し条件が真となった時に投入条件以下の作業が実行される。工程は最後の作業場を抜けるとその工程の最初の投入条件の待機状態になる。

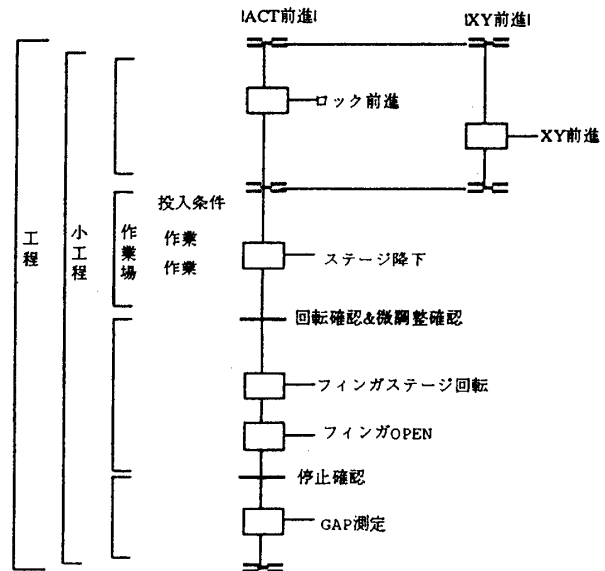


図1: 工程の構成と記述サンプル

4 まとめ

本稿ではセルコントローラにより実現するアプリケーションの立場から、アプリケーションを表現するための用語整理とコントローラへの要件を検討し、制御言語の機能部品を設計した。これよりライン制御に代表されるプロセス型アプリケーションでは、アプリケーションレベルの言葉でプログラム記述できるようになった。また部品の役割は機能的に直交しているためアプリケーションのブレークダウンに選択の幅が狭まることで、プログラムからアプリケーションへの対応づけが容易になった。

参考文献

- [1] Mutsumi Asahi, et al, "Integrated and Flexible Controller", 3rd International Conference on CIM 95, 1995.
- [2] John M. Vlissdes, Mark A. Linton, "Unidraw: A Framework for Building Domain-Specific Graphical Editors", ACM Transactions on Information Systems, Vol.8, No.3, 1990.
- [3] "MELSAP-II:三菱シーケンサプログラミングマニュアル"
- [4] "HIDIC-S10 カタログ"