

## 機械制御用ソフトウェア開発支援システム

## 4 J-5 一 仕様記述とプログラミング方法 一

小野哲司, 越智俊昭

住友重機械工業(株)

## 1. はじめに

当社では従来より、ソフトウェアの開発効率の向上、高品質なソフトウェアの生産を目的とした機械制御用ソフトウェアの開発支援システムに取り組んでいる。1986年には、サーボ制御を対象としたBBPS<sup>1)</sup>を開発し成果を挙げている。今回、シーケンス制御にまで拡張した開発支援システム‘BASP (Building block Architecture for Sequence and servo Programming)’を開発したので報告する。本システムは、製鉄機械の制御システムを対象としたものであり、サーボ制御機能とシーケンス制御機能を含む制御系全体を総括的に扱い、ソフトウェアの機能設計からソフトウェア製作、機器組み込みまでを支援するものである。その中で今回は特に、仕様記述方法とプログラミング方法について報告する。

## 2. 支援システムの特徴

支援システムの特徴をアプリケーションの開発手順に沿って以下に示す。

- ① 機能記述方式：フローチャート形式とブロック線図によりサーボ制御とシーケンス制御を含む制御機能仕様を記述できる枠組みを与えた。
- ② 簡易プログラミング言語：機能記述から一意的にコーディング可能なコマンド群を用意、ソフトウェア仕様が不要な手順を実現。
- ③ 実行形式のソフトウェア自動生成：マルチタスクOSの基で動作するC言語のソフトウェアに翻訳するプログラムジェネレータを用意。
- ④ デバッガ：データのモニタや制御パラメータの調整等をオンラインで行えるデバッガを準備し、デバッグ・調整効率の向上を図る。

## 3. 機能記述とプログラミング方法

## 3. 1 対象とする機械制御システムの特徴

製鉄機械の制御機能の特徴的な項目を以下に示す。

- ・ 上位コントローラからの指令や外部信号に基づいて制御モードの切り替えを行う‘シーケンス制御機能’と、機械の位置や圧力、速度等マイナーループ制御を行う‘サーボ制御機能’が複雑に絡み合った構成をとっている。
- ・ シーケンス制御機能は比較的単純な動作順序の制御とインタロックの制御が中心である。
- ・ 複数制御軸の同時実行やその間の協調等、並列処理的要素が必要とされる。

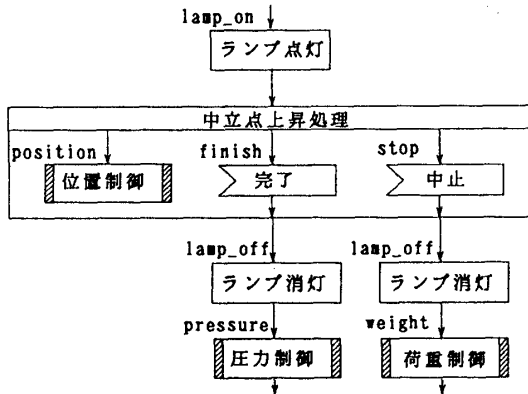
## 3. 2 機能記述とプログラミング

サーボ制御機能はブロック線図の形式(従来のBBPSの方式)で、シーケンス制御機能はフローチャート形式で記述することとした。次の制御機能の本記述方式で記述した例を図1に、またプログラミング例を図2に示す。

- 1) ランプを点灯し、‘中立点上昇処理’を行う。
- 2) ‘中立点上昇処理’は、機械を位置制御しながら次の事象を待つ。
  - a. ‘完了’事象の発生(例えばリミットスイッチON)
  - b. ‘中止’事象の発生(例えば非常停止スイッチON)

- 3) '完了' 事象発生後はランプを消灯して圧力制御に切り替える。
- 4) '中止' 事象発生後はランプを消灯して荷重制御に切り替える。

シーケンス制御記述



サーボ制御記述

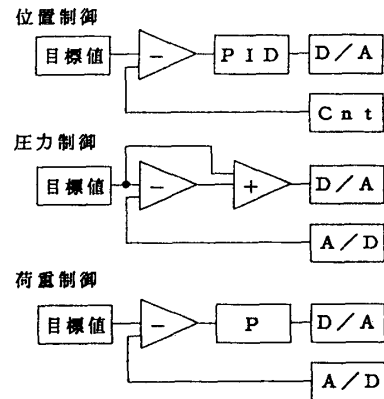


図1 システム表記例

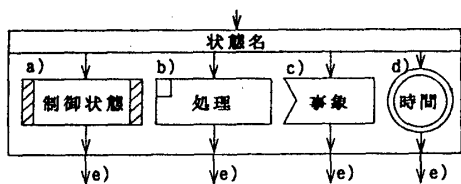
```
lamp_on();
@Alt(
  :::@Cycle:Cycle_1( position );
  :::@Event:finish;    lamp_off();  @Cycle:Cycle_1(pressure);
  :::@Event:stop;     lamp_off();  @Cycle:Cycle_1(weight);
)
```

図2 コーディング例

3. 3 機能要素

通常のフローチャートの記述要素に加え、機械制御特有の要素を新しい記号として定めた。特徴的な点について以下で説明する。

- ・シーケンス制御機能の中にサーボ制御機能の記述を可能とした。
  - ・次の2種類の並列処理記述要素を開発した。
    - Para : 複数制御軸の待ち合わせ等、同期をとって次の処理へ移行。
    - Alt : どれかの事象発生やタイムアウトによって次の処理へ移行。
- なお、Altで記述が可能な要素を図3に示す。



- a) 指定した制御を実行してその終了を待つ
- b) 指定した処理を実行してその終了を待つ
- c) 指定した事象を待つ
- d) 指定した時間を待つ
- e) どれかが終了した時点で次の処理へ移行

図3 Altで記述可能な要素

4. まとめ

本システムにより、次の効果が望める。

- ・視覚的に分かりやすい機能記述方式により設計時の効率が向上する。
- ・ソフトウェアを自動生成する事で高品質、高効率な開発が可能となる。
- ・機能記述と対応したコマンドを用意する事で要求品質に忠実な制御用ソフトウェアが生成できる。

5. 参考文献

1) 村上亨、太田朝也 “ビルディングブロック方式による制御用ソフトウェア開発システム”、情報処理学会第35回全国大会