ソフトウェア工学 64－15
（１９８９．２．３）

構造化分析手法の実時間システム向け拡張

竹中 一起
住友金属工業㈱ システムエンジニアリング事業本部

計算機システムの要求仕様を記述する方法の一つとして、図形表現を主体にした構造化分析手法（ＳＡ手法）が知られているが、事務処理分野を対象に開発されたものであるため、実時間システムの要求定義には適していなかった。近年、ＳＡ手法を実時間システム向けに拡張する試みがいくつか見られるようになってきたが、大規模かつ複雑な実時間プロセス制御システムの要求定義には充分とはいえない。そこで本稿では、要求定義という面からプロセス制御システムの特徴を明らかにした後、ＳＡ手法の実時間システムへの従来の拡張方法の問題点を指摘し、プロセス制御システムの要求定義に必要な機能を整理する。次に今回開発した新しい拡張ＳＡ手法の詳細について述べ、最後に例題によってその有用性を示す。

An Extended Structured Analysis Method for Real-time Systems (in Japanese)

Kazuki TAKENAKA
Sumitomo Metal Industries, Ltd.
1-3, Nishinagasu-hondori, Amagasaki, Hyogo, 660, Japan

Structured analysis (SA) has been widely used to describe requirement specifications of computer systems, but it is not well suited to define requirements for real-time systems. Recently, some proposals have been made to extend SA for real-time systems, but they are not sufficient for large and complex process control systems.

This paper first discusses the characteristics of process control systems in the view of requirements specification, and shows the flaws in other proposals and functionalities needed to define requirements of process control systems. A new extended SA method is described in detail, and finally its usefulness is shown by an example.

1．緒言

我々は、鉄鋼プロセス制御システムに代表される大規模かつ複雑な実時間プロセス制御システムの開発生産性を飛躍的に向上させることを目的として、システム開発の要求定義段階から、設計、製造、テストまでの工程を一貫して計算機支援する「リアルタイムシステム開発一貫支援システム」[1]の開発を進めている。

この開発一貫支援システムの要求定義段階では、理解し易さの重視という観点からデータフロー図をベースにした構造化分析手法[2]（以下では「ＳＡ手法」とも呼ぶ）を採用し、これを実時間システム用に拡張している。本稿では、要求定義という面からプロセス制御システムの特徴を明らかにした後、ＳＡ手法の実時間システムへの従来の拡張方法の問題点を指摘し、プロセス制御システムの要求定義に必要な機能を整理する。次に今回開発した新しい拡張ＳＡ手法の詳細について述べ、最後に例題によってその有用性を示す。

2．プロセス制御システムとその要求定義

2．1　プロセス制御システム

工場全体の生産や操業を管理するシステムにおいては、それを構成する計算機システムは、図１に示すような階層構造として構築される。構成要素となる各々の計算機システムの特徴を表１にまとめる。この中で今回の一貫支援システムで支援の対象とするのは、「プロセス制御システム」の開発である。要求仕様の定義という面から見た「プロセス制御システム」の特徴は以下に集約される。

①制御の対象が工場全体、あるいは操業ライン全体と言うように非常に大規模であるので、システムの状態を単一の状態遷移図等では表現しきれないことがしばしばである。すなわち有限状態機械でモデル化できないケースを想定しなければならない。

②非同期の複雑な外部信号を取扱うことができ、それに従って個々の処理を並列に実行できることが必要である。これに対処し、応用ソフトを作り易くするために、マルチタスクの概念が扱えるリアルタイムＯＳを基本ソフトとして搭載する。従って、要求定義手法の実時間対応への拡張にあたっては、リアルタイムＯＳの利用を前提に検討すれば良い。

③システムに要求される応答速度は、システム全体のレスポンスとして、おおよそ数十msから数百msのオーダであり、しかも個々の個別機能に関して指定時間内での処理の完了が厳しく要求されることは稀である。従って一部の例外を除き、時間制約に関する考慮は、要求定義段階ではあまり重要ではない（数ms以内での処理完了を要するものは、通常、下層のデジタル制御システムで対処される）。

これらは、同じ実時間システムと言っても、ロボットコントローラのような機器組込みシステムとは全く性格を異にするものである。

2．2　要求分析

システムの規模が大きくなればなるほど、要求仕様の定義は困難を極め、大規模システムの要求仕様定義にお

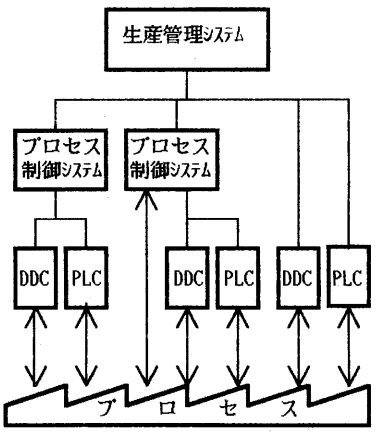


図１．計算機システムの構造

表１．計算機システムの特徴

	生産管理システム	プロセス制御システム	デジタル制御システム
ｼｽﾃﾑ化単位	生産工場単位	プロセス操業単位	設備／機器単位
主要機能	•大量データのオンライン処理 •オーダの投入順の策定	•プロセス操業データの高速処理 •操業や設備の統括制御	•シーケンス制御 •複数ループのＰＩＤ制御
応答速度	数秒	数十ms～数百ms	数ms
計算機	メインフレーム	ミニコン，高性能マイコン（専用機）	マイコン/ＣＰＵなし
ﾌﾟﾛｾｽ入出力	なし	直接/ﾃﾞｼﾞﾀﾙ制御ｼｽﾃﾑ経由	直接

いては、どこから手を付けてよいかすら分からないというケースも出てくる。そこで、システムに要求される機能を表2に示す7つのサブシステムに分解し、サブシステム単位に要求仕様の分析を階層的に進めていくことにする。これらサブシステム間の関連については、本稿でこれから述べる拡張SA手法の例題として、図5に示してある。なおここでは、システム全体の機能を計算機内部での処理の性格を基にサブシステム分割を行ったが、制御や管理の対象となるプロセスあるいは設備を基に分割してもよい。今回前者の分割方法を採用したのは、設計段階以降への接続がよりスムーズに行えると判断したからである。

表2．プロセス制御システムを構成するサブシステム

ｻﾌﾞｼｽﾃﾑ	内　　容
設定・制御	数式モデルや論理判断等によるアルゴリズムに基ずき、プロセス又はその制御装置へ直接あるいは間接に指示を与える。
操業管理	数式モデルや論理判断等によるアルゴリズムに基ずき、主としてオペレータに操業上／管理上のデータ提供や作業指示を行う。
トラッキング	プロセスの状態遷移、物流状況等を計算機内部で模擬し、その状況に応じて計算機内部の他の処理を起動する。
実績収集	データベース構築のため、外部入力データの収集や編集、加工等を行う。
プロセス信号入出力	プロセス情報の採集や指示出力のために、プロセス入出力装置に対して単純あるいは標準的な入出力処理を行う。
オペレータ入出力	オペレータへの情報提供や作業指示を行うと共に、オペレータの判断や指示を受ける等、MMIを司る。
上位伝送	上位に位置する計算機との間で、データの送受信を行う。

2．3　構造化分析手法の採用

前節で述べた要求分析は、SA手法で実施することにする。SA手法を採用したのは、形式的な厳密さと完全さをある程度犠牲にしているものの、データフロー図という図形表現で要求を定義していくことから、計算機については素人であるユーザ部門の人でも比較的容易に仕様書を理解できると判断したからである。

2．4　従来のリアルタイム拡張

ところで、SA手法はもともと事務処理システム向けに開発されたものであるため、実時間システムの要求定義に不可欠な制御やタイミングを記述することが困難であった。具体的に言うと、オリジナルのSA手法におけるデータフロー図では、データがいつ流れるのかとか、複数あるプロセスがどのような順番で、いつ起動されるのかといったことは記述できなかった。このようなシステムの制御やタイミングは、実現段階に至るまでは考慮する必要はないとする意見[3]もあるが、ここ数年、SA手法のもつ数々の長所を生かしながら、実時間システムの要求を記述できるような拡張がいくつか試みられている。その中では、Wardらによる拡張[4]とHatleyらによる拡張[5]の2つが代表的であり、我国においても立田の努力[6]により広く知られるようになってきている。

[Wardの拡張]

データフロー図を実時間用に拡張して変換図と名前を変え、システムの制御は状態機械で記述する。変換図においては、データの流れについても、その矢印の形で連続／不連続が区別され、制御信号が「信号イベント」として破線の矢印で表される。データ変換のプロセスは、状態機械として状態遷移図によって動作が表現される「制御変換部分」から、「活性化イベント」「不活性化イベント」と呼ばれる2種類の制御信号によって制御される。

[Hatleyの拡張]

実時間に関する要求を記述するために、制御フロー図と制御スペックの2種類の仕様書を新たに設けた。制御フロー図は、制御の流れを明示するためのものであり、データフロー図のデータフロー以外の部分をコピーしたものに、制御信号を破線の矢印で表現する。具体的な制御の内容は、制御スペックと呼ぶ別の図で表現する。制御を記述するためのモデルとしては、順序マシンと組み合せマシンの2通りがあり、ここに制御信号の組み合せと共に、どのプロセスをどういう順で起動するかを記述する。制御フロー図上の制御信号は、単に信号の流れを記述しているだけで、プロセスの起動を意味しているものではない。

これら両者の実時間向け拡張は、制御信号の記述を可能にしたこと、及びシステムの状態をモデル化しデータ変換プロセスの制御／起動を記述可能にしたという2点で大枠としては共通しているが、具体的な記述方法やその意味にはかなりの違いがある。そして、我々が今回検

討しているプロセス制御システムの要求仕様定義へ適用しようとすると、どちらの拡張方法とも一長一短であり、また不十分なところも多い。そこで我々は、これらの拡張方法を参考にしつつ、プロセス制御システムの要求定義を行うのに最適な拡張方法の検討を行った。次章では、ＳＡ手法の実時間プロセス制御システムへの拡張について述べる。

３．　ＳＡ手法の実時間システム向け拡張

３．１　プロセス制御システムの要求分析に必要な機能

プロセス制御システムの要求分析に必要な機能を、データと制御という２つの面から整理する。まずデータについては、

・温度や圧力、クレーンの位置のような「連続量」と、製造指示や操業実績のような「離散値」は、明確に区別したい。これらは実現段階において、その取扱いに大きな差異を生じるからである。従って、Wardの拡張手法と同様、データフローの矢印の数で区別することにする。

制御については、

・プロセス制御システムの機能を２．２で示したように７つのサブシステムに分割すれば、外部の状況変化や状態遷移をシステム内部で模擬する必要があるのは、「トラッキング」サブシステムに限られる。

・プロセス制御システムでは、工場全体とか製造ライン全体と言うように非常に広い範囲を対象とするので、「トラッキング」ではこれらの複雑な状態をモデル化できることが必要となる。Wardの拡張、Hatleyの拡張とも、システムの制御は状態機械でモデル化するが、状態機械では同時には最大一つしか加工しないような設備の内部状態を表現するのには適するものの、流れ作業をしている組立てラインのように同時に複数の製品を加工しているようなシステムを表現することは極めて困難である。そこでは、複数の状態の並列性や相互作用を考慮したり、あるいは状態に容量を考慮する必要があるからである。そこで、システムの状態を、状態機械（状態遷移図）だけでなくペトリネットでも表現できるようにする。

・その他のサブシステム内、あるいはサブシステム相互間の制御については、材料の搬入や加工の完了等、単なるイベント発生の通知程度で充分であり、システム状態の保存や制御信号の組み合せはほとんどの場合必要がない。

・「オペレータ入出力」や「上位伝送」では、外部からデータ（離散データ）を受信したタイミングで、該当の機能を実行することが要請される。従って、離散データの受信をデータ変換プロセスの起動要因としても取り扱うことができれば便利である。

・「実績収集」や「上位伝送」については、加工完了通知のような「トラッキング」からの制御指示と同様、５分毎、１時間毎といった周期起動も重要である。周期起動の概念は、Wardの手法、Hatleyの手法共に見られない。

・以上述べたように、プロセス制御システムにおいては、「トラッキング」サブシステムを除けば、制御信号の取扱いはそう複雑ではない。要求仕様書の可読性を重視すれば、データフローと制御フローは、一枚の図面上に共存している方が分かり易い。またプロセスの起動も、Hatleyの場合のように別の仕様書で表現するのではなく、この図面上に表現したい。

・Wardの方法では制御変換部分から出力される活性化／不活性化イベントでデータ変換部としてのプロセスが制御されるが、今回対象としているプロセス制御システムでは、これらはリアルタイムＯＳの機能により実現される。ここでは制御信号をプロセスの起動要因として取り扱うことにし、実現段階とのスムーズな接続も狙う。

時間制約については、設計段階において、タスク分割の際や、データベース／ファイルの構造や管理方法等の設計の際に検討すれば充分である。

３．２　構成要素

本稿の拡張ＳＡ手法では、以下の４種類の図面で要求仕様を記述するものとする。

- 変換図
- 要求辞書
- プロセススペック
- 制御スペック

最初の３つは、オリジナルのＳＡ手法に見られる「データフロー図」、「データ辞書」、「ミニ仕様書」を、各々今回のプロセス制御システム向けに拡張したものである。最後の「制御スペック」は、制御信号の合成を表現するために導入したものである。以下では、これら４種類の図面について、それを構成する要素に重点を置いて述べる。

３．２．１　変換図

変換図は、データフロー図を基に制御信号の取扱いが

できるように拡張したものである。変換図は、システムへの要求仕様が適当な粒度になるまで分析・詳細化を続け、階層的に定義していく。変換図で使用されるシンボルを図２にまとめる。

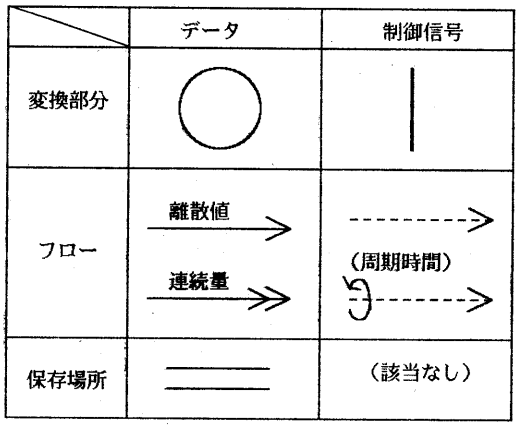


図２．変換図で使用されるシンボル

（１）変換部分

データの変換部分はシステムの機能を表現するもので、実線の円形で表す。ここでは、一個以上の入力データを、下の階層で定義されたデータ変換部分、あるいはプロセス仕様に基ずいて変換し、データフローとして出力したり、あるいは制御フローを生成したりする。データ変換部分への制御信号入力は、そのプロセスの起動要因を意味する。

制御信号の変換部分は、実太線の棒で表す。この変換部分への入出力は、破線矢印で表現される制御信号である。データフローが入出力されることはない。ここでは、どのような制御信号群から、どのような制御信号群が合成されるかということだけを明示する。具体的に、どのような信号の組み合せから、場合によってはシステムの現在状態も考慮して、どのような制御信号が合成されるかは、制御スペックで定義する。すなわち、制御信号の変換のみを定義するものであり、通常は「トラッキング」サブシステムの要求定義の場合にのみ使用される。

（２）フロー

フローは、矢印の付いた曲線で表す。

矢印が一つだけ付いた実線は、離散的なデータの流れを表す。離散的なデータの流れとは、加工指示や操業状況等、システム内で一つのまとまりとして取り扱ったり、送受信したりするデータであり、場合によってはプロセスを起動する要因になることも可能である。矢印が二つ付いた実線は、連続的なデータの流れを表す。連続的なデータとは、温度や圧力のようなシステムの物理量、あるいはクレーンの現在位置のように、なんらかの測定装置により、定常的に採取されているデータ、あるいはシステムが生成／指示する連続かつ可変の制御指示のようなものである。

矢印の付いた破線は、制御信号を表す。制御信号は、材料が加工設備に接近したことや、加工が終了したこと、すなわちシステム内で何かのトリガとなる条件が発生したことを別のプロセスに通知するためのもので、今回の拡張ＳＡ手法では、主としてデータ変換部分の起動に使用される。システム時計から周期的に発せられる制御信号は、起点側に回転矢印と周期時間を併記することによって表す。

なお、今回の拡張ＳＡ手法ではデータフロー、制御フローとも、フローの途中での合流や分岐は行わないものとする。

（３）保存場所

データの一時的な保存場所は、一対の平行な実直線で表す。これは、データベースやファイルを抽象化したものである。一つの図面上に同じ名前の保存場所が、複数個存在してもよい。これは、フローの交差を回避するためで、同名の保存場所は同一のものとみなす。

３．２．２　要求辞書

オリジナルのＳＡ手法におけるデータ辞書は、データフロー図上でデーフローとして現れた用語の定義を一定の規則にしたがって並べたものである。ここでの要求辞書は、オリジナルのデータ辞書と本質的には同じであり、変換図上に現れたデータフローと制御フロー及びデータ保存場所を詳細に定義したものである。

制御フローで、その起点が外界であるものに限り発生頻度を属性として持たせることができるようにする。これは、設計段階において時間制約を検討する際の基礎資料となる。

３．２．３　プロセススペック

プロセススペックは、オリジナルのＳＡ手法におけるミニ仕様書とほぼ同じで、階層的に定義した変換図の最下層に位置するデータ変換部分の機能を構造化言語等を使って記述する。ここで、制御信号の生成を記述できることが、今回の拡張点である。

３．２．４　制御スペック

変換図上における制御変換部分は、変換部分に対する制御信号の入力群と出力群を明示するだけで、そこでどの様な変換が施されるかということはブラックボックスとして取り扱われた。この変換の詳細仕様を定義するのが、制御スペックである。

この制御スペックの動作モデルとして何を選ぶかは開発しようとする応用システムの性格に左右されるが、今回の拡張ＳＡ手法では、以下の３種類を提供している。

なお、ここでは制御信号の変換のみを定義し、Hatleyの場合のように起動するプロセスを制御スペック上に記述することはしない。

・組み合せマシン

制御信号の出力が、現在の制御信号の入力値のみから決る、すなわち内部状態を持たないシステムをモデル化するためのもので、入力値に対する出力値の組み合せは、図３（ｂ）に示すデシジョン・テーブルで表現する。

・順序マシン

制御信号の出力が、現在の制御信号の入力値と以前のシステムの状態の両者から決る、すなわち内部状態を持つシステムをモデル化するためのもので、入力値に対する出力値の組み合せは、図３（ｃ）に示すような状態遷移図あるいは状態遷移マトリックスで表現する。

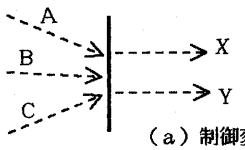
・ペトリネット

順序マシンでは表現できないほど複雑な、あるいは並列性を有するシステムのモデルは、図３（ｄ）に示すようなペトリネットで表現する。

３．３　構成規則

要求仕様を定義する段階では、その要求をどのように実現するかは考えるべきではない。しかしながら、要求を定義する手法の中に、要求仕様を実現に移す際の移行を容易にするようなメカニズムを組み込んでおくことは、システム設計以降への接続を容易にする。すなわち要求定義の手法にある程度の枠をはめることにより、設計段階以降への接続をスムーズに行うことは良い方法と考えられる。但し、これが度を過ぎると手法そのものの記述力を制限することになるので注意を要する。ここでは、要求仕様からシステムの内部構造（タスク関連図）を機械的に導出できるように、変換図のデータ変換部分に課した制約について述べる。

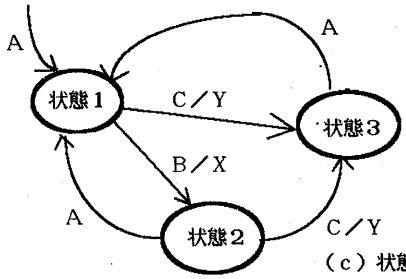
データ変換部分へのデータ及び制御信号の入出力を整理すると表３のようになる。



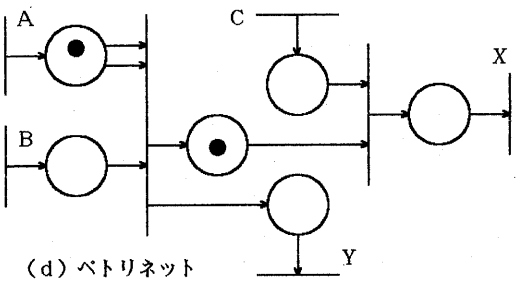
（ａ）制御変換部分

入力			出力	
A	B	C	X	Y
ON	ON	(don't care)	ON	ON
	OFF	ON	ON	OFF
		OFF	OFF	ON
OFF	ON	ON	OFF	ON
		OFF	ON	OFF
	OFF	(don't care)	OFF	OFF

（ｂ）デシジョン・テーブル



（ｃ）状態遷移図



（ｄ）ペトリネット

図３．制御スペックの表現

表３．各種フローの起点と終点

	入力フローの起点	出力データの行先
連続値データ	・データ変換部分 ・保存場所（データベース） ・外界	・データ変換部分 ・保存場所（データベース） ・外界
離散値データ	・データ変換部分 ・保存場所（ファイル／データベース） ・外界	・データ変換部分 ・保存場所（ファイル／データベース） ・外界
制御信号	・データ変換部分 ・制御変換部分 ・外界　・周期起動	・データ変換部分 ・制御変換部分 ・外界

これをプロセスの起動という観点から整理すると、プロセスの起動要因となり得るのは

・制御信号

・起点がデータ変換部分または外界の離散値データ

の２種類である。連続値データが起動要因になれないのは自明であろう。また保存場所からのデータ採集は、データ変換プロセスが主導権を握って処理する必要があるため、これも起動要因となることはできない。

データ変換部分はデータをどの様に変換するかを定義するものであるから、あるデータ変換部分へ入力されているデータは、その変換を行うために不可欠である。一方、制御信号はそのプロセスの起動タイミングを示すだけであるから、一つのデータ変換部分に複数の制御信号が入力されていても問題ない。それらは、ＯＲ条件として、制御信号がアクティブになるたびに、入力データの変換を施せば良い。但し、一つのデータ変換部分に、タスクの起動要因となれる離散値データが複数入力されている場合は問題である。なぜなら、ある起動要因によってプロセスが起動されたとしても、そのとき別の離散値データが利用可能でなければ、そのプロセスはデータの待合せの為に、処理を中断しなければならないからである。

そこで、データ変換部分への入力フローに関し以下の制約を課すことにする。

①一つのデータ変換部分に対して、制御信号は複数入力することができる。ただし、それらは独立した起動要因として扱われ、個々の信号の発生の度にプロセスが起動される。

②一つのデータ変換部分に対して、起動要因となれる離散値データの入力は最大１個に限る。これはデータの待合せに起因するプロセスの中断を排除するためである。

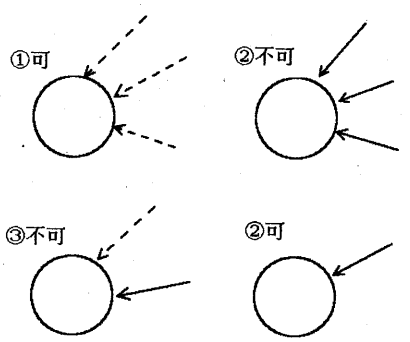


図４．データ変換部の構成規則

③一つのデータ変換部分に対して、起動要因となれる離散値データの入力と制御信号の入力が共存することはできない。これもプロセスの中断を排除するためである。

４．拡張手法による記述例

以上で説明した拡張ＳＡ手法の中で特徴的な変換図の記述例を図５に示す。これは、プロセス制御システムを構成する７つのサブシステムの相互関係を示したものである。図中で「データベース」と命名されたデータ保存場所は３ヶ所あるが、これらはすべて同じ名前を持つため、実際には一つのデータ保存場所を意味する。これは、データフローと制御フローの交差をできる限り避けるためである。

「トラッキング」については、１つの制御入力と３つの制御出力だけが記入され、データに関しては入出力がない。これは今回の応用システムでは、システムの内部状態が制御信号の合成（「外部状態変化」の制御信号がどの様な制御信号から構成されるかは要求辞書で表現される）のみで表現されることを意味する。これは、本稿の拡張手法の特徴の一つである。すなわち、この「トラッキング」の直下の変換図は、一つの制御変換部分だけからなり、そこでどのような変換が行われるかは対応する制御スペックに記述される。

「設定・制御」「上位伝送送信」では、各々２つの制御信号が入力されている。データ変換部分への複数の制御信号入力はＯＲ条件の起動要因として取り扱われるため、「上位伝送送信」は加工終了時に発せられる「操業実績送信依頼」のほか、１０分毎にも周期起動されることになる。マルチタスクＯＳのタスク起動／タスク間通信を、「機能コード＋データ」のメッセージにより行うとすれば、これらの制御信号は「機能コード」により識別することができる。

「上位伝送受信」や「オペレータ入力」については、制御信号の入力がない。これらは、唯一に制限された外部からの離散値データにより、プロセスの起動が行われる。

「データベース」へのデータ入出力が、ある場合は連続量として、ある場合は離散値として取り扱われているが、これはデータベースのある部分は連続値であり、ある部分は離散値であるためである。データベースの具体的な構造は、データフローや制御フローなどと同じく要求辞書で定義される。

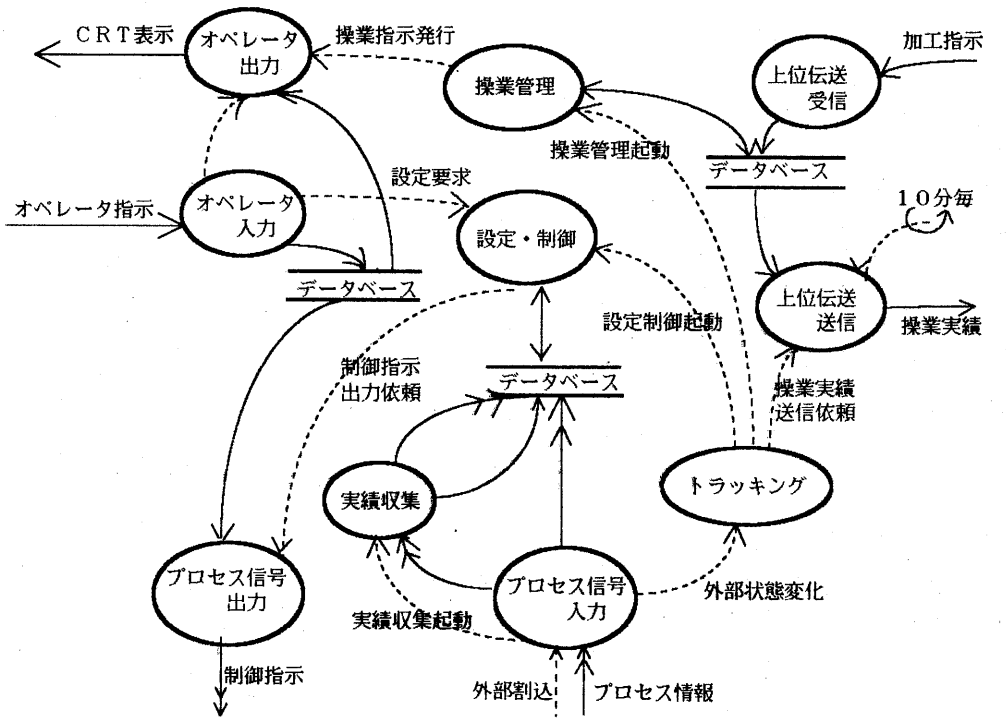


図5．変換図の記述例

5．結言

我々は、大規模かつ複雑な実時間プロセス制御システムの開発生産性を飛躍的に向上させることを目的として「リアルタイムシステム開発一貫支援システム」の開発を進めているが、この一環として事務処理システム向けに開発された構造化分析手法を、実時間プロセス制御システム向けに拡張した。本稿では、まず要求定義という面からプロセス制御システムの特徴を明らかにした後、従来の実時間システム向け拡張ＳＡ手法の問題点を指摘した。次にプロセス制御システムの要求定義に必要な機能を示し、今回開発した拡張ＳＡ手法の詳細について述べ、最後に例題によってその有用性を示した。

謝辞

日頃御指導頂く横井玉雄・システム技術開発部次長に深く感謝します。

[参考文献]

[1] 竹中、横井:「リアルタイムシステム開発一貫支援システムー基本構想ー」,情報処理学会第36回全国大会,1988.

[2] De Marco,T. Structured Analysis and System Specification. Yourdon Press,1978.

[3] Gomaa,H. A Software Design Method for Realtime Systems. Commun.ACM 27,9,1984.

[4] Ward,P.T.,et al. Structured Development for Real-Time Systems, Prentice-Hall/Yourdon Press, 1985.

[5] Hatley D.J. A Structured Analysis Method for Real-Time Systems, A Seminar for the Fall DECUS U.S. Symposium,1985.

[6] 立田:「蘇った構造化分析手法」,日経コンピュータ, No.152,1987. ほか