

## モデルに基づくプラント制御ロジック生成支援システム

溝口祐美子、広本博史†、吉瀬仁志††、岩政幹人、伊藤聡、月本洋  
(株)東芝 研究開発センター、(株)東芝 府中工場†、(株)東芝 本社††

火力発電プラントの運転操作自動化ソフトウェアの仕様記述、運転操作ロジック図 (IBD) を自動生成するシステムについて述べる。従来は設計者がそのノウハウを用いて既存のプラントの IBD を修正してターゲットプラント用の IBD を作成してきた。この手法ではプラント運用上の各種条件を人間系で加味する必要があり、設計者はある一定以上の経験知識が必要であった。

本論文ではソフトウェア生産技術の1つである部品再利用技術を適用した IBD 自動生成システムを提案する。本システムではプラントの運転仕様を人間にわかりやすい形式で知識ベース化しているため、設計者はターゲットプラントにあわせて知識を修正することにより IBD の難型を容易に作成できる。

## A Model-based Generation System of Plant Control Logic

Yumiko Mizoguchi, Hiroshi Hiromoto †, Hitoshi Kichise ††,  
Mikito Iwamasa, Satoshi Ito, Hiroshi Tsukimoto  
Research & Development Center, Toshiba Corporation  
† Fuchu Works, Toshiba Corporation  
†† Principal office, Toshiba Corporation

This paper describes an automatic generation system of Interlock Block Diagrams (IBD) which are the control specification of power plants. Usually engineers design IBD of a target plant by modifying the IBD of the previous plant which is similar to the target. This method requires skilled engineers and the quality of the IBD depends on engineers' individual skill.

This paper proposes an automatic generation system of IBD applying technique of reusing software components. Because this system uses a human-friendly knowledge-base of the control specifications, engineers can easily generate the basic pattern of the target IBD by modifying the knowledge-base.

## 1 はじめに

火力発電プラントの運転操作自動化ソフトウェアは運転操作ロジック図 (IBD) と呼ばれる仕様記述を基に作成される。本論文では IBD の自動生成を支援するシステムを提案している。我々はこのシステムをソフトウェア生産技術の1つである部品再利用技術を適用して構築する。

プラントのロジック図 (IBD) を作成する一手法として過去のプラントの IBD を利用する方法がある。

この手法では設計者が作成するターゲットプラントの仕様に近い既存のプラントの IBD を選択し、プラントの仕様の相違 (例えばポンプの台数の変化など) に合わせて既存の IBD を直接修正する。

この手法は火力発電プラントのように運転操作方法や構造がパターン化されている場合に有効であり設計工数を削減することができるが、修正に対する個々の設計者に一定以上の経験知識が要求される。

## 2 アプローチ

我々は、プラントの運転操作方法や構造がパターン化している場合は IBD の記述も条件ごとにマクロ化した形式に整理することができる点に着目した。このようなマクロ化された IBD をテンプレートと呼ぶ。

我々の提案するシステムはテンプレートの部品をいくつか用意して、これら部品をプラントに合わせて具体化することで IBD を作成することを支援する。

また具体化に際しては知識ベースを利用し、熟練設計者のノウハウを知識ベースの中に取り込むことにより IBD の生産性の向上と品質の均一性の実現をはかっている。

ソフトウェア工学の立場から見るとテンプレートは抽象化されたソフトウェア部品に相当する。また本手法はソフトウェア部品を知識ベースを使って具体的なプログラムへ展開しているため、ソフトウェア部品再利用技術 [1] を運転操作ロジックの設計に適用していることになる。

同時に展開するための知識を人間に理解しやすく柔軟性のある形で表現することを試みる。人間に理解しやすい知識ベースで設計者を支援するアプローチの重要性は Fischer, Girgensohn らの論文 [2] で強調されている。

本論文では IBD 自動生成支援システムの一部であるタイミング条件展開システムの概要 (3節)、

IBD の抽象化部品として有効なテンプレートの表現方法 (4節)、抽象化された条件を具体的に展開するための知識ベースの構築方法および条件の展開方法を (5節)、人間に理解しやすく柔軟性のある知識ベースを提供することを階層操作展開表を用いたタイミング条件作成の仕組み (6節) を示す。

## 3 タイミング条件展開システム

本論文で提案するシステムの概要を図1に示す。タイミング条件展開システムは「解析・変換」、「タイミング条件生成」、「IBD 生成部」から構成される。

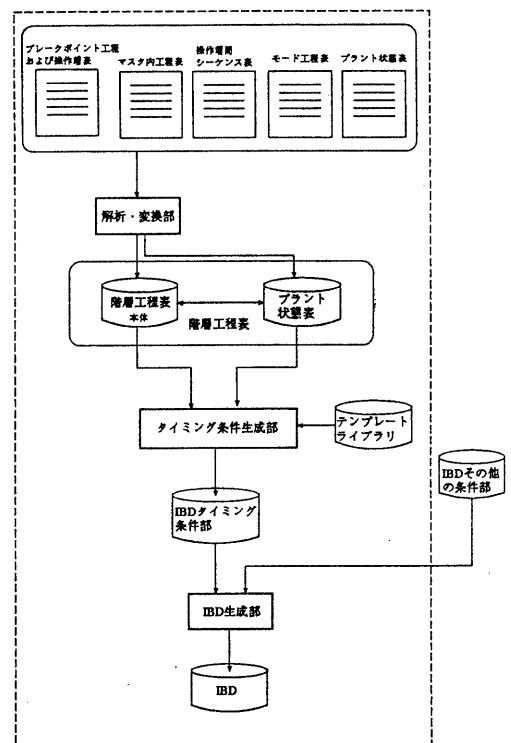


図1: タイミング条件展開システム概要

- 解析・変換  
入力 (プラントの運転操作タイミングに関する知識) を解析し階層操作展開図に変換する。階層操作展開図は本論文で提案する知識ベースであり6節で示している。
- タイミング条件生成  
IBD の操作の単位 (ポンプ起動、弁開など)

を操作端という。それぞれの操作端に対しテンプレートライブラリから適切なテンプレートを選択し知識ベース「階層操作展開図」の知識を用いて IBD タイミング条件部を作成する。ここで、タイミング条件はテンプレートの抽象化された条件の1つである。4節に示している。

- IBD 生成部

IBD タイミング条件部にタイミング条件以外の条件であるその他条件部を付加して IBD を作成する。

#### 4 テンプレート表現

テンプレートはロジックをマクロ化した IBD の型に対して入出力となる条件を抽象化した IBD の雛型である。図2に火力発電プラント IBD の一般的な基本形テンプレートを示す。図2に示す箱がロジックのマクロである。このマクロは箱の左側の条件が成立すると出力が成立し操作が実行されるロジックを表している。図2に示した基本形テンプレートでは抽象度が高いので操作端を関連性の高いグループに分けてそのグループに対して専用のテンプレートを用意する。このようなテンプレートをライブラリとして持つようにすることで汎用的にシステムを運用することができる。

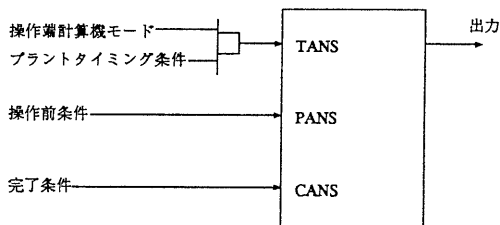


図 2: 基本形テンプレート

テンプレートは次の要素で構成される。

- 抽象化条件

抽象化された意味を持つ条件である。実際にある操作端の IBD を作成する時に知識を用いて適切な条件に具体化する。

- 固定ロジック

そのテンプレートを用いた操作端の IBD すべてに固定のロジックである。

- 固定条件

そのテンプレートを用いた操作端の IBD すべてに固定の条件である。

専用テンプレートの例として大型補機用テンプレートを図3に示す。大型補機は主にポンプなど大電力によって動かされる機器を指す。四角で囲んだ条件は大型補機の抽象条件であり、用意された知識によって作成したい操作端の IBD に適切のように具体化することができる。点線四角はマクロ化されたロジックであるとする。無印の条件およびロジックは固定条件、固定ロジックである。

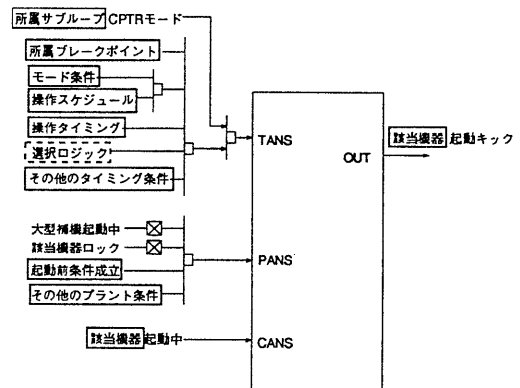


図 3: 大型補機用テンプレート

テンプレートをベースにして IBD を作成する場合、作成したい操作端 IBD のテンプレートとしてふさわしいものを選択する。

#### 5 テンプレートと展開知識

図3に示した大型補機用テンプレートを例にとりテンプレートのタイミング条件部の抽象化条件を具体化するのに必要な知識を示す。ここでは抽象化条件を具体化することをテンプレート展開という。

以下に大型補機用テンプレートのタイミング条件部の抽象化条件を具体化するために必要な知識をまとめる。

1. ブレイクポイントおよび操作端表

ブレイクポイントは運転操作を管理するためにプラントの起動/停止の流れを時系列的にいくつかのステージに分割したものである。ブレイクポイントおよび操作端表はブレイク

ポイントごとに起動する操作端をまとめた知識であり、IBD を作成する操作端を起動する時に成り立っている所属ブレイクポイントを具体化する。(図4参照)

工程 順序	ブレイク ポイント	操作
1	起動準備	循環水ポンプ起動
2	ユニット起動	A,B復水ポンプ起動
	!	!
	!	!

図4: ブレイクポイントおよび操作端表例

## 2. モード表

火力発電プラントはその起動時の状態のモード(コールド、ホットなど)によってクリーンアップ操作などの工程が異なる。クリーンアップ操作に関していえばモードはプラントの停止後の経過時間、真空を保持しているか真空を一旦破壊したかによって区別する。モード表はモードによる工程の実行/不実行を記述した知識であり、モード条件を具体化する。

## 3. 操作端間シーケンス表

操作端どうしの順番や関連に関する知識およびモードによる操作順序の違いに関する知識。

## 4. プラント状態表

プラントの状態と操作端の操作タイミングに関する知識。操作タイミングを具体化する。例えば操作端「並列シーケンス起動」を起動する条件として操作タイミング「主タービン回転数 > 3600- ε rpm」がある。

## 5. 操作端別条件表

ターゲットプラント固有の知識。音声指令など。その他のタイミング条件を具体化する。

ここでは大型補機用テンプレートを例にとったがその他の専用テンプレートを展開するのに必要な知識としてマスタ内操作展開表などがある。マスタは固定化された一連の操作時系列をまとめたものである。マスタ内操作展開表はマスタ用テンプレートで必要な工程を時系列にまとめたものである。

## 6 階層操作展開図

5節に示したようにテンプレートのタイミング条件展開するための知識の大部分はターゲットプラントの運転操作タイミングに関する知識である。運転操作に関する知識を以下に列挙する。

- ブレイクポイントおよび操作端表
- マスタ内操作展開表
- モード表
- 操作端間シーケンス表
- プラント状態表

これらの知識をユーザに示すことによってユーザは設計するプラントに対する理解を深めることができ、設計変更などの事態に容易に対応することができるようになる。表を作成する上での問題点としてプラントはモードによって工程が変化したり関連する操作が多数存在したりして必要な情報は複雑である。我々はこれらの情報を盛り込みかつ人間にとってわかりやすく変更に対して柔軟な表を考案した。この表を階層操作展開図と呼ぶことにする。

図7に階層操作展開図の例を示す。

階層操作展開図ではモードによって変化する操作順序、時間的なシーケンスだけでなく考慮しなくてはならない操作端との関係も示したい。以下に示す3つの工夫により人間にわかりやすく取り扱いが容易な表現を実現している。

### 1. 操作順序の階層化

工程は図5のように階層的に整理できる。また、機能によっては一連の操作を概念階層(部品(cold), 部品(warm)のように)を用いて部品としてまとめることができる。

### 2. 部品交換による操作順序変化への対応

部品の中身の要素や構成がモードによって変化する場合、図5に示すように部品を入れ換える(部品(cold), 部品(warm)を入れ換える)だけで異なるモードの操作順序を表現することができる。

### 3. 部品のターミナル化による接続関係の変化への対応

同階層の接続関係でモードによって操作が消えるような場合、消えた操作はInput/Outputをつなげる単なるターミナルとして取り扱う。この方法で前後の接続関係をいちいち変更する必要がなくなる。

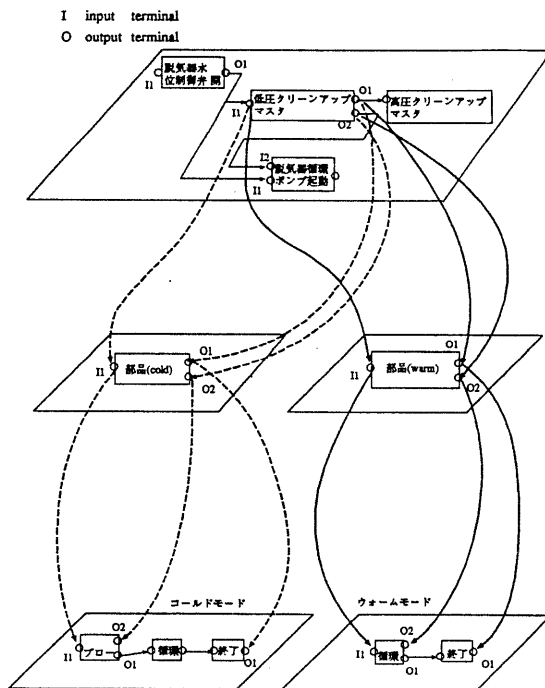


図 5: 操作順序の変化と部品変更

階層操作展開図は図 6 の形式で表現できる。

## 7 課題

- 本論文ではタイミング条件を展開する知識について論じたがその他操作前条件、完了条件を展開する知識が必要である。この条件を具体化するためにはプラントの構成や操作の意味に対する知識とこれらの知識を直接利用する処理が必要である。たとえばモデルベース推論技術 [3] が有効であると考えられる。
- 本論文ではテンプレートを知識を用いて具体化して IBD を作成する方法を提案してきたが知識の完全な構築は本質的に難しい。そこで事例ベース推論 (CBR) と融合し、過去の IBD を修正する方法が有効であると思われる。そのためには過去の IBD のどの条件を選ぶべきかを示す知識を洗いだし知識ベースを構築していく必要がある。

### フォーマット

名称	:
タイプ	:
指定条件	:
ダミー化条件	:
上位階層	:
下位階層	:
前操作	:
下層入力操作	:
後ろ操作	:
下層出力操作	:
プラント状態	:

図 6: 階層操作展開図表現形式

## 8 まとめ

火力発電プラントのように運転操作方法や構造がパターン化された対象は条件を抽象的に表現することでテンプレートを用いた部品再利用の考え方を適用することが可能である。これらの条件に対し知識を用いて IBD 作成支援を行なうことで、設計者の熟練度によらない IBD を得ることができる。

また展開のための知識をユーザにわかりやすく、変更に対して柔軟であるように整理できることがわかった。

この知識はプラントの運転方案に直接関連するので、ユーザはプラント操作に対して理解を深め変更があった時も容易に対処できるようになる。また、本論文で示した表は従来のプラント仕様のドキュメント等から容易に作成することができるので、知識獲得の負担も少ないと考えられる。

——→ 操作端どうしのシーケンス  
 - - - - 起動中

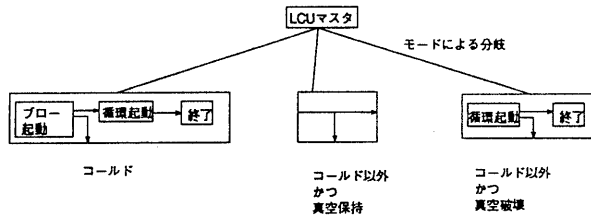
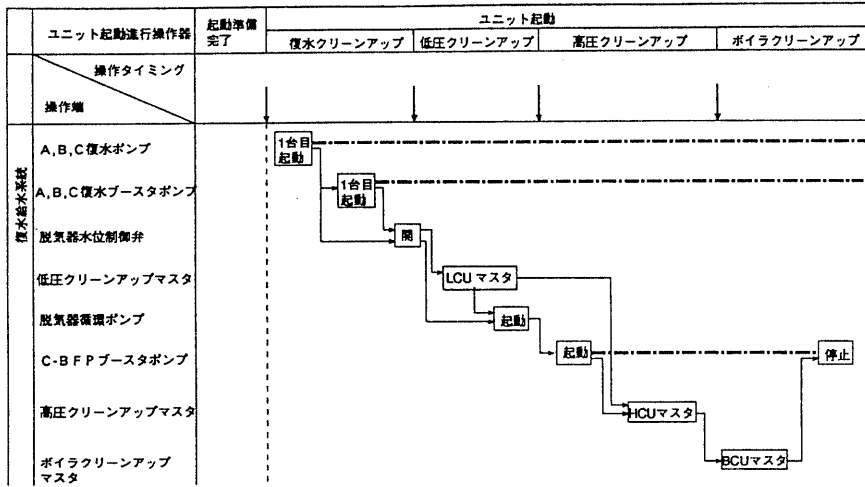


図 7: 階層操作展開図例

### 参考文献

- [1] S.Castano, V.De Antonellis. A Constructive Approach to Reuse of Conceptual Components. In *Proc.IEEE Advances in Software Reuse*, pages 19-28, 1993.
- [2] Fischer et al. Supporting Software Designers with Integrated Domain-Oriented Design Environments.*IEEE Transactions on Software Engineering*, Vol.18, No.6, June, 1992.
- [3] V.Sembugamoorthy, B.Chandrasekaran. Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems. In *Experience, Memory, and Reasoning*. J.I. Kolodner and C.K. Riesbeck.eds., Lawrence Erlbaum. Hillsdale.N.J., pages 47-73, 1986.