

プラント制御ソフトウェア設計における仕様獲得方法

笹氣 光一 後藤 和之 中山 康子
(株)東芝 研究開発センター

本稿では、プラント制御ソフトウェアの仕様を獲得するインターフェイスについて述べる。設計者が持つプラントに関する知識を知識ベース化することにより、この知識ベースを参照してプラントの機器構成には明示されていない機器同士の関係を理解し、プラントの運転手順を表すフローチャート形式の制御仕様を自動生成するインターフェイスを開発した。これにより、設計者にとって理解しやすい形式でプラントの要求定義を記述することが可能になり、仕様入力の手間を大幅に軽減した。

An Interactive Specification Acquisition Method for Software Design

Kouichi SASAKI, Kazuyuki GOTOH, and Yasuko NAKAYAMA

Research and Development Center,
Toshiba Corporation

This paper describes a specification acquisition method used in PlantBASE, an automatic software design system for plant control. In this method, a plant model is first provided with the knowledge about behaviors and functions of each machine composing a target plant. By referring to this knowledge, PlantBASE recognizes the physical structure of the target plant from the specifications given in the form of tables and diagrams and generates a flowchart of machine operations, thereby reducing the cost of describing the control sequence for the plant.

1 はじめに

ソフトウェア設計を支援する CASE ツールや自動プログラミングシステムにおいては、計算機が理解できるレベルの詳細な仕様を設計者が正確に入力する必要がある。したがって、これらのシステムによってソフトウェアの生産性が向上する一方で、設計者の仕様入力のコストがかかって増大するという問題が生じる。この仕様入力のボトルネックを解消するためには、設計者が記述しやすい形式で仕様を入力できるようにしなければならない。そのため、計算機に仕様を理解する能力を持たせる必要があり、ヒューマンインタフェースが果たす役割は大きい。

このようなヒューマンインタフェースに着目し、設計者にとってわかりやすい形式で記述した要求定義から、システムが要求する詳細な仕様を獲得する種々の方法が提案されている。例えば、グラフィカルな制約言語を用いて手順を記述する方法 ([1])、アイコンアニメーションを用いる方法 ([2])、ネットワーク形式の図表を用いる方法 ([3])、自然言語の要求をフレームモデルで解釈する方法 ([4]) などが提案されている。

これに対して筆者等は、プラント制御ソフトウェアの設計支援システム ([5],[6],[7],[9],[12]) において、設計者が機器構成の表と機器の配置図面によって要求定義を与えると、プラント機器の運転手順を表すフローチャートを自動的に生成する仕様獲得インターフェイスを実現した。

以下、要求定義から仕様を獲得する際の問題点とそれを解決するためのアプローチ、さらに、仕様獲得インターフェイスの実現方法について述べる。

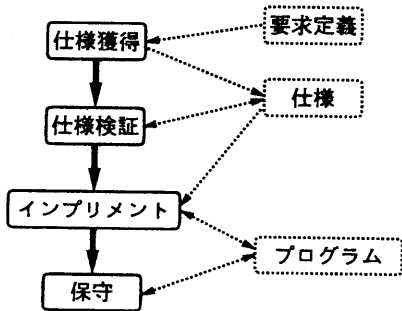


図 1: ソフトウェア設計工程

2 制御ソフトウェアの仕様獲得

2.1 設計工程における仕様獲得

一般に、ソフトウェアは図 1 のような工程に従って設計される。

設計者は、要求定義からソフトウェアの仕様を獲得し、その仕様をもとにプログラムをインプリメントする。要求定義からの仕様獲得はソフトウェア設計工程の最上流の工程である。プラント制御ソフトウェア設計において、図 1 の要求定義、仕様は、それぞれ以下に示す形式となる。

要求定義は、

- (1) プラントを構成する機器の名称とその属性を記述した表形式の設備リスト (図 2)
- (2) 機器間の位置関係、接続関係について記述した図面であるプロセスフロー図 (図 3)
- (3) 運転の時間や、並列している機器の運転の順番などについて記述した運転方案の 3 種類の仕様として記述される。

名称	補機	運転モード	運転操作	機器状態	異常状態
1-1号除塵機		現場/中央 単独/連動 手動/自動	正転操作 停止操作	準備完了 正転 終点位置	過負荷 過トルク
1号搬出機		単独/連動	正転操作	正転	過負荷 蛇行 非常停止
	洗浄弁	常時	開操作	全開	

図 2: 設備リストの例

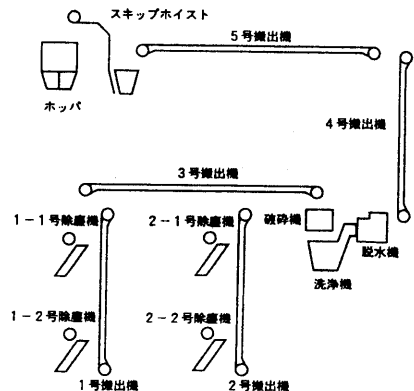


図 3: プロセスフロー図の例

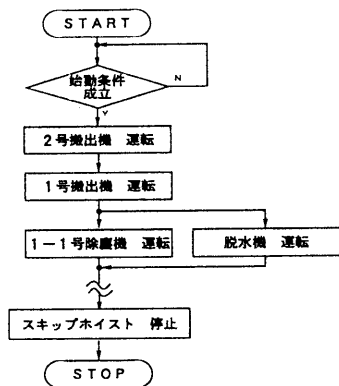


図 4: 運転手順シーケンスの例

これらの要求定義から獲得される制御仕様は、フローチャート形式の運転手順シーケンス(図4)である。

従来、設計者は要求定義からプラントの構成と機器同士の関係を理解し、プラントの状態の変化をイメージしながら運転手順シーケンスを記述していた。しかし、対象となるプラントが大規模化、複雑化すると機器同士の関係の組み合わせが増大し、その結果、並列に運転する機器が存在したり、一つの機器に複数の運転方法が存在するという状況が発生し、運転手順シーケンスの記述が困難になる。

この問題を解決するために、要求定義に記述されたプラントの構成を自動的に理解することにより、運転手順シーケンスの生成を支援する。

2.2 要求定義理解の問題点

要求定義からプラントの構成を理解する過程において、プロセスフロー図の自動的な理解が困難であるという問題がある。プロセスフロー図は機器間の接続関係と、接続された機器間での処理対象の流れる方向を示す上流下流関係を与えるものであるが、これらの関係をプロセスフロー図から獲得するのは、以下の理由により容易ではない。

(1) 接続関係獲得の困難さ

機器同士の接続には

- 結線や配管などの物理的な部品によって接続されていることにより成立する接続(物理的接続関係)
- 機器間での処理対象の受け渡しによって成立する接

続(機能的接続関係)

の2種類がある。プロセスフロー図においては、物理的に接続していない機器間に機能的接続関係が成立する場合でも、この関係は図上には明示されないことが多い。

例えば、図5では図中に矢印で示したような接続関係が成立するが、この矢印は通常は図面上には明示されない。

(2) 上流下流関係獲得の困難さ

また、プロセスフロー図は建築用CADで描かれた設計図面とは異なり、実際のプラントにおける機器の厳密な位置(座標)を表示することよりも、プラントを構成する機器の全容を設計者がイメージしやすいように表示することに主眼を置いて作成されている。このため、機器同士の物理的な位置関係の方向(上下、左右)が常に統一されているとは限らない。図5の例でも、プラント内に存在する機器すべてを図面上に表示するために、物理的な上下関係の図面内における方向が統一されていない。このため、例えば「図面上でより左にある機器がより上流の機器である」というような座標に依存した推論では処理対象の流れを推論することは難しい。

2.3 アプローチ

このように、プロセスフロー図の図面上の情報だけではプラントの機器関係を理解することは難しい。しかし、プラントの設計者は曖昧な形式で与えられているプロセスフロー図から機器同士の位置関係や接続関係を読み取ることができる。我々はこのプロセスフロー図の認識過程において、プラントの構成機器の機能に関する知識が利用されていると考えた。機器の機能に関する知識は、従来の研究では仕様の検証に利用されていた([9], [10], [11])が、本稿ではこの知識を仕様の獲得に応用し、接続関係を推論する方法を提案する。

3 システムの構成

我々の開発したプラント制御ソフトウェア設計支援システム PlantBASE の構成を図6に示す。

PlantBASE の特徴は、プラントの概念に関する知識をプラントモデルという知識ベースに記述し、この知識を利用してプラント制御ソフトウェアの設計工程を一貫して支援する点にある。プラントモデルは対象分野となるプラントに関する一般的な知識を記述した一般モデルと、

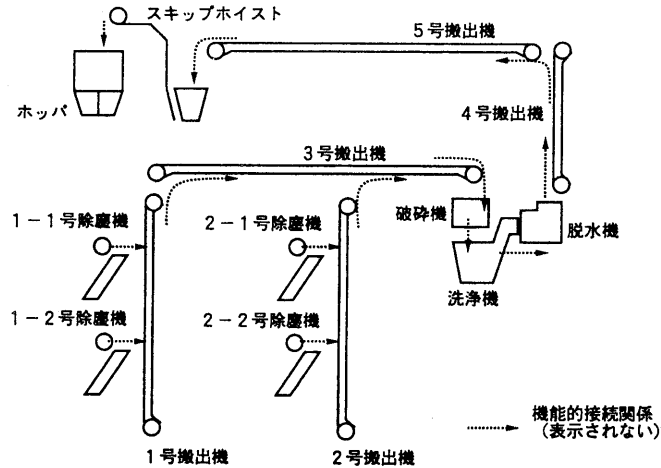


図5: プロセスフロー図と機能的関係の例

一般モデルに記述した概念を実在のプラントのレベルにまで詳細化した詳細モデルの2種類のモデルと、機器の動作順序の制約条件などを記述した運転知識から構成されている。

一般モデルには、対象分野のプラントを構成する機器の名称、それぞれの機器モードや動作、その動作によって引き起される状態、機器の有する機能が一般的な概念間の関係として表現されている。詳細モデルには実際のプラントの機器の構成と、それぞれの機器の名称、動作、状態、運転モードなどの属性に加えて、接続関係、連動関係などの機器同士の関係が記述される。

プラントに関する知識をこのような形で表現することにより、機器構成の異なる同一分野のプラントに対して、一般的な知識が再利用可能となる。

プラントモデルは、オブジェクト指向データモデルで表現されており、一般モデルはクラスオブジェクトに、詳細モデルはインスタンスオブジェクトにそれぞれ相当する。

PlantBASEの入力は、設備リストと、プロセスフロー図の2種類の仕様であり、出力は、プラントの制御仕様、すなわちフローチャート形式の運転手順シーケンスである。生成された運転手順シーケンスをプログラム展開知識により展開することにより、プラント制御用のプログラマブルコントローラ(PC)の制御プログラムが生成できる[5]。

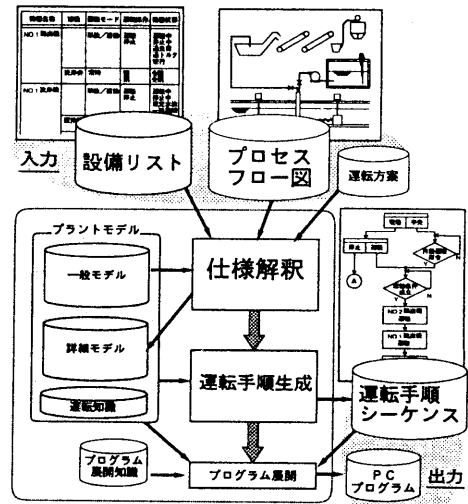


図6: PlantBASEのシステム構成

制御仕様の生成は、(1) 要求定義を解釈して一般モデルから設計対象のプラントに特化した詳細モデルを導出し、(2) 詳細モデルと運転知識をもとに制御仕様を生成する、という手順で行なわれる。要求定義を解釈することにより一般モデルから詳細モデルを導出することが仕様の獲得に相当する。

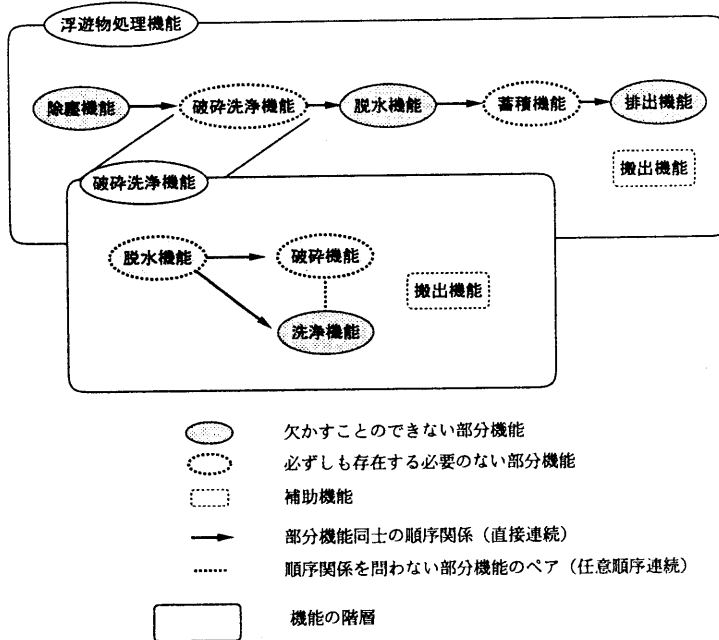


図 7: 機能系統知識の表現例

4 仕様獲得方法

4.1 プラントの機能系統知識

プラント設備は、ある機能を果たすことを目的として設計されており、ある設備の達成すべき機能は、その設備を構成する機器の機能を組み合わせるものとして表現することができる。プラントモデルでは、プラントの機能は、部分機能によって階層的に表現されており、各階層において、部分機能同士の関係も記述されている。この階層状に記述された機能に関する知識を、機能系統知識と呼ぶ。

プラントモデルのうち、一般モデルに記述する知識は、分野内のプラントについて汎用であることが要求される。PlantBASEのモデルでは、この汎用性を満足させるために、各階層における部分機能を以下の2つのタイプに分類した。

- 上位の機能を達成するために欠かすことのできない部分機能
 - 必ずしも存在する必要のない部分機能
- また、必ずしも存在する必要のない部分機能の中でも、

他の部分機能を達成する機器の間で処理対象を受け渡すための機能(例えば「搬出機」の「搬出機能」)はある階層内の機能の順序関係の任意の位置に任意の回数挿入してもかまわない。このため、一般モデルでは、このような機能を特に「補助機能」として分類している。

各階層において、部分機能は以下の2つの順序関係によって関係づけられる。すなわち、

- 部分機能同士の接続順序を定義する「直接連続」
- 「直接連続」が成立する部分機能のペアの中で、順序を入れ換えても構わないペアを与える「任意順序連続」

である。

プラントにおける機能系統知識のイメージを表示したのが図7である。機能に関する知識をこのように階層的に表現することにより、ある機能を達成するための部分機能としての役割を何通りにも記述することができる。例えば、図7の例では、「浮遊物処理機能」「破碎洗浄機能」それぞれの階層に部分機能として「脱水機能」が属している。しかし、この「脱水機能」は前者の中では「欠

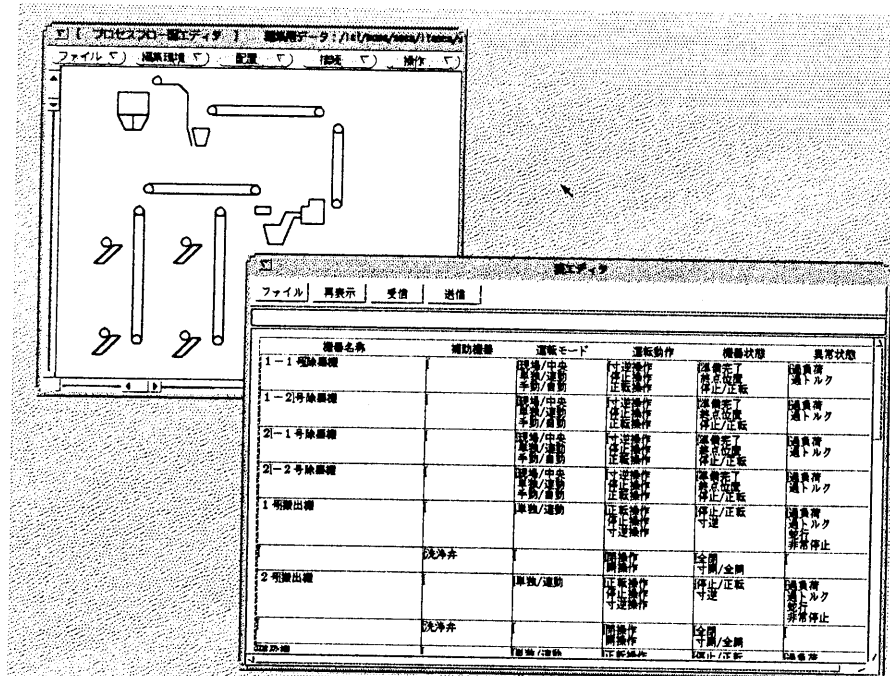


図 8: 仕様入力環境の画面例

かすことのできない部分機能」であるが、後者の中では「必ずしも存在する必要のない機能」として表現されている。このように同じ名称の機能に各階層で異った役割を割り充てることができるので、プラントの機器の機能同士の接続関係を多様な対象プラントに対応できるように表現することができる。

4.2 機器構成の獲得手順

図 8 に PlantBASE での仕様入力環境の画面例を示す。設計者は設備リストの入力ツールと、プロセスフロー図の入力ツールを用いて、設計対象となるプラントの要求定義を入力する。入力された仕様をもとに、PlantBASE は以下の手順で詳細モデルの導出を行なう。

(1) インスタンスの生成

PlantBASE は設備リストで与えられた機器の名称を解釈し、一般モデルに記述された知識を用いることにより機器のクラスを同定し、機器インスタンスを生成する([12],[13])。

ある属性の値を入力しなかった場合でも、一般モデル

のクラスオブジェクトのデフォルト値を参照することにより、これらの値を補充することができる。

(2) 機能的接続関係の候補の生成

前の過程で生成された機器インスタンスに対応する機能と、一般モデルに記述された機能系統知識の順序関係をもとに、この順序関係に矛盾しない機器同士の機能的接続関係の候補を生成する。

図 9 は、図 5 により仕様を与えられるプラントについて機器「脱水機」と機能的に接続できる機器候補の生成の手順について示したものである。

手順 (1) で、設備リストの解釈により機器「脱水機」のインスタンスが生成されており、その「機能」属性値は「脱水機能」となっている。

まず機能系統知識を参照することにより、脱水機能と接続しうる機能を決定する。図 7 の機能系統知識では、「浮遊物処理機能」の部分機能である「脱水機能」は「蓄積機能」と「直接連続」により関係づけられている。また、「補助機能」である「搬出機能」は「脱水機能」と

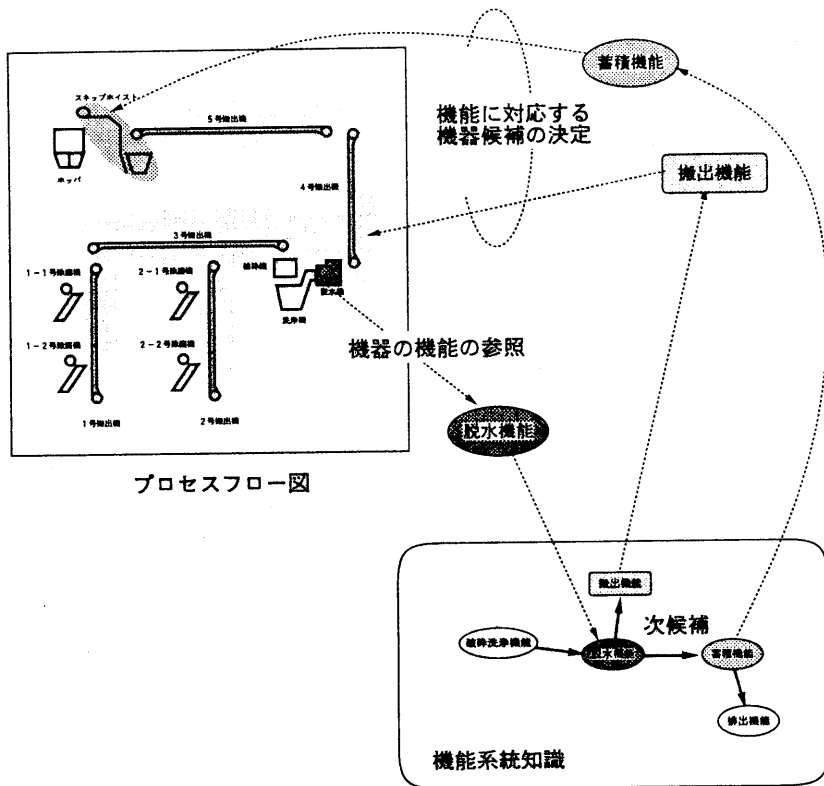


図9: 機能的接続関係を満足する機器候補の決定手順

接続することができる。

次に、これらの機能の候補を「機能」属性値として持つ機器を検索する。図5の仕様においては、「蓄積機能」は「スキップホイスト」の、「搬出機能」は「1～5号搬出機」の「機能」属性値である。

したがって、この例において「脱水機」と機能的に接続しうる機器の候補として「1～5号搬出機」および「スキップホイスト」が挙げられる。

(3) 図形情報を用いたの候補の評価

手順(2)で生成された接続関係の候補それぞれについて、PlantBASEは機器同士がプロセスフロー図上で座標的に近いかどうかについて評価を行なう。機能の観点と図形上の位置関係の観点、双方で接続可能であると判断された機器系列の候補が存在した場合、その系列内の機器間に接続関係が成立すると推論する。

図5の例では、ステップ2で求めた「脱水機」と機能的に接続可能な機器のうち、図面上で最も近い位置にある「4号搬出機」が「脱水機」の下流機器であると推論される。

4.3 制御仕様の生成

このような手順を経ることにより導出された詳細モデルをもとに、PlantBASEは以下のように対象プラントの動作手順を生成する。

プラントモデルには機器の運転手順に関する制約の知識が記述されている。この制約とは、例えば、「プラントの起動時には、上流の機器を動作させるには、下流の機器が起動していなくてはならない」といったものである。この、運転手順に関する知識を、詳細モデルに記述された機器同士の上流下流関係に適用し、運転手順を自動的に生成する。

5 まとめ

表と図面の形式で与えられたプラントの要求定義をプラントの機器に関する知識を用いて解釈し、機器の運転手順を示すフローチャート形式の制御仕様を自動的に生成するインターフェイスを実現した。このインターフェイスにより獲得した制御仕様をもとに、プラント制御ソフトウェアを自動的に生成することができる。

設計者の知識をシステムに持たせることにより、設計者にとってわかりやすい形式で記述した要求定義をシステムが理解し、仕様獲得が可能になった。これにより設計者の仕様入力負担を大幅に軽減できる。

エンジニアリング分野においては、設計者の意図をシステムに反映することが重要であり、これを実現するためには、知識処理を応用したヒューマンインターフェイス技術が不可欠である。

参考文献

- [1] M.R. Blackburn, "Using expert systems to construct formal specifications", *IEEE EXPERT*, SPRING, 1989, pp. 62-74
- [2] R. St-Denis, "LGV: A domain knowledge validation environment", *Comput. & Graphics* Vol. 14, 1990, pp. 311-320
- [3] M.K. Hook et al., "A knowledge based specification system for ASDSPN", *Proc. second IEE National Conf. on Telecommunications*, 1989, pp. 144-147
- [4] 大西, "要求仕様データベースシステムの開発", *信学技報* Vol. 94 No.219, 1994, pp. 1-8
- [5] 中山他, "対象モデルに基づくシーケンス制御ソフトウェアの設計支援", 第38回情報処理学会全国大会, 1989, 4G-1
- [6] Y. Nakayama et al., "Model-based automatic programming for plant control", *Proc. IEEE CAIA-90*, 1990, pp. 281-287
- [7] H. Mizutani et al., "A knowledge representation for model-based high-level specification", *Proc. IEEE Conf. CAIA-91*, 1991, pp. 124-128
- [8] H. Mizutani et al., "Automatic programming for sequence control", *Proc. IEEE Conf. IAAI-92*, 1992, pp. 315-331
- [9] S. Itoh et al., "Model-based explanation of specifications for sequence control", *Proc. IEEE Conf. CAIA-94*, 1994, pp. 261-267
- [10] V. Sembugamoorthy et al., "Functional representation of devices and compilation of diagnostic problem-solving systems", J.L. Kolodner et al. (eds.), *Experience, Memory and Reasoning*. Lawrence Erlbaum; Hillsdale, NJ, 1986, pp. 47-73
- [11] B. Chandrasekaran et al., "Functional representation as design rationale", *IEEE Trans. on Computing*, January, 1993, pp. 48-56
- [12] 中山他, "プラントモデルに基づく制御ソフト設計支援 PlantBASE(1)-(3)", 第49回情報処理学会全国大会, 1994, 6J3-5
- [13] 後藤他, "モデルに基づくプラント制御ソフトウェア設計支援 PlantBASE - 一般モデルの構築 -", 人工知能学会研究会 *SIG-KBS-9403-4*, 1995, pp. 25-32