鉄鋼搬送ラインの自動制御システムにおける ソフトウェア再利用モデル

正員 高田 秀志*   非会員 松田 茂彦**
非会員 小林 潤**   正員 中田 秀男***
非会員 堀池 聡***

A Software Reuse Model for Automatic Control Systems of Steel Conveyance Lines

Hideyuki Takada*, Member, Shigehiko Matsuda**, Non-member, Jun Kobayashi**, Non-member,
Hideo Nakata***, Member, Satoshi Horiike***, Non-member

In recent years, as supervisory and control systems software for steel plants is getting more and more complicated, it has been the most important issue to build the system low cost and efficiently. As the programming environment for plant systems, the graphical language such as the function block diagram is widely utilized. This kind of language contributes to the improvement of software productivity, while the view point of reuse is focused on programs. In order to aim at further improvement, it is required to promote the reuse from the view point of plant level design.

In this paper, we propose an engineering model such that reusable parts is build from the view point of plant level design of plant equipment and operation functions. We also introduce the engineering tool based on the model. An evaluation result of partial application of the tool for a real plant system shows that 70% of developed parts can be estimated to be reused in the future application and half of inter-connections among software parts can be automated.

キーワード：ソフトウェア再利用，監視制御システム，エンジニアリングモデル

Keywords: Software Reuse, Supervisory and Control System, Engineering Model

1. まえがき

近年，製鉄業における品質要件の高度化や多品種小量生産への要求にともない，鉄鋼ラインの監視や制御を行うシステム（監視制御システム）のソフトウェアはますます複雑化している。一方で，システムベンダー側でも国際的な競争が激化し，この複雑化したシステムをいかに低コストで

効率的に構築できるかが最重要課題となってきた。

鉄鋼ラインの制御の種別を大きく分類すると，コイル等の物品の搬送を自動で行うための自動搬送制御，ライン上を流れる鉄板の速度や張力を調整する主幹制御，モータ等に対する冷却器を動作させるための補機制御に分けられる。このうち，主幹制御と補機制御は，ラインの性質に依存する部分が少ないため制御を標準化しやすいが，自動搬送制御については，機械の動作方式，隣り合う機械との干渉，ラインの運転要領などによって，制御が変化する度合いが大きい。そのため，低コストなシステム構築を実現するには，この自動搬送制御部分の生産性を向上させることが必須である。

制御システムの分野では，1980年代からリレー回路や電気回路を模した図的言語がすでに用いられていた⁽²⁾。その後，標準化活動が行われ，国際規格 IEC61131-3⁽³⁾ として制定されている。このような図的言語は，プログラミング言語の知識がなくても制御処理を記述できる，視認性が良いためデバッグ作業が効率化できるなどの特長がある。ま

* 京都大学情報学研究科

〒 606-8501 京都市左京区吉田本町

Graduate School of Informatics, Kyoto University

Yoshida Honmachi, Sakyo, Kyoto, 606-8501

** 東芝三菱電機産業システム（株）

〒 108-0073 東京都港区三田 3-13-16

Toshiba Mitsubishi-Electric Industrial Systems Corp.,

3-13-16, Mita, Minato-ku, Tokyo 108-0073

*** 三菱電機（株）先端技術総合研究所

〒 661-8661 兵庫県尼崎市塚口本町 8-1-1

Advanced Technology R & D Center, Mitsubishi Electric Corp.

8-1-1 Tsukaguchi Hon-machi, Amagasaki, Hyogo 661-8661

た、図的言語の一つであるファンクションブロック図では、典型的な処理をブロック化して再利用することも可能である。しかし、図的言語といえども、背後にある制御の論理は理解している必要があり、運転要領などのプラント仕様を図的言語へ詳細化するには、専門的な知識が必要である。

鉄鋼プロセス制御ソフトウェア技術の CAE(Computer Aided Engineering) 環境については、電気学会金属産業技術委員会による調査活動の結果が報告書としてまとめられ、上記のような図的言語によるプログラミングに加えて、運転方案等の上流設計を支援する環境の構築が提言されている⁽¹⁾。上流設計に対する取り組みとしては、プラントソフトウェアの効率的な要求定義方法⁽⁴⁾や、UML(Unified Modeling Language)を用いた方法⁽⁵⁾などが提案されている。また、再利用による生産性向上に対する研究としては、設備をオブジェクトとしてモデル化した EASIER モデル⁽⁶⁾が提案されている。このような方法を用いることにより、より上位の概念でシステムを構築することが可能となる。しかしながら、これらの方法は設備の構造をオブジェクトとして表現しているなど、あくまでもソフトウェアの視点からの上位概念であり、プラントの上位概念である機械設備や電気設備、運転方法の設計者にとっての視点とは合致しない。

本論文では、従来行われていたソフトウェアの視点からの部品化ではなく、制御システムの上位概念である設備や運転等のプラント設計の視点で部品を構築し、再利用を行うためのエンジニアリングモデルを提案する。これにより、制御システムの上位概念であるプラント設計の情報をプログラミングやシステム定義と連続性を確保して扱うことを可能にする。本論文で述べるエンジニアリングモデルの特長を以下に示す。

- 鉄鋼ラインにおける機械(コイルカーやロールなど)の単位で部品化を行う。部品は、その機械を構成するセンサやアクチュエータなどの機器情報、および、機械の機能を果たすためのプログラムを含む機能部品により構成する。これにより、ラインに存在する機械や機械の果たす機能の視点で部品を選択し、再利用することが可能になる。
- 機械を動作させる手順は対象のライン毎に違いが大きいことから部品化が難しいため、対象毎に記述することとするが、上位設計者による運転方法の記述に適した図的言語による記述を可能にする。
- 機械を動作させるのに必要な条件(インタロック)は、機械や材料が干渉しないか、運転モードは適切かなどといったプラント設計レベルの概念で記述することを可能にする。これにより、制御プログラムの専門家でもなくとも定義が可能になる。
- 再利用された部品と適用対象に応じて記述された機械動作手順やインタロックのための信号は、部品の構造に基づいた信号名称に基づいて自動接続することにより、部品の組み合わせ作業を効率化する。

上記のような特長を持つエンジニアリングモデルをツールとして実現し、実際の制御システムの構築に適用することにより、有効性の検証を行った。

本論文の構成は、以下の通りである。2章では、鉄鋼ライン監視制御システムの機能とそのエンジニアリング作業の概要を述べる。3章では、本論文で提案するソフトウェア再利用モデルについて述べる。4章では、本モデルに基づいて構築したエンジニアリングツールを紹介し、5章で適用の評価を行う。6章では、まとめと今後の展開について述べる。

2. 鉄鋼ライン制御システムとエンジニアリングの概要

2.1 鉄鋼ラインの制御 先に述べたように、鉄鋼ラインの制御は自動搬送制御、主幹制御、補機制御に分けられる。本論文ではこのうち、自動搬送制御に対するソフトウェアの再利用環境について検討を行う。

自動搬送制御とは、従来、人間が手動により操作盤を操作し、ライン上の各設備を動作させていたものを、コントローラに内蔵されたプログラムから自動的に行うようにしたものである。これにより、ラインの自動運転が可能になり、省人化を始めとして、生産能力や安全性の向上が図られてきた。

図1に、自動搬送制御の単純な例を示す。図に示すように、1. 制御プログラムからアクチュエータに対して動作信号が出力されると、2. 実際に動作が行われ、物が移動を始める。制御プログラムは、3. 移動限界を検出するセンサからの信号を周期的に監視し、反応すれば、4. アクチュエータへの動作信号を停止させる。実際の制御プログラムでは、数多くのアクチュエータやセンサを用いて、複数の動作を並列に実行したり、条件分岐などを行いながら、このような処理を行っている。

2.2 自動搬送制御のエンジニアリング 鉄鋼ラインのエンジニアリングは、大きく2つの過程に分けられる。第一の過程は、設備の構成や動作方法などのプラント仕様に関する機械的、電気的な設計であり、第二の過程は、各センサやアクチュエータ等の入出力信号の定義や制御プログラムの内容まで詳細化していくソフトウェア的な設計で

図1 自動搬送制御の例

Fig.1. An example of automatic conveyance control.

ある。

プラント仕様の設計では、まず、どの設備にどのようなセンサやアクチュエータが存在し、どのような役割をしているか、どのような仕様を持つものかなどを一覧にした機器リストの作成が行われる。また、運転員がどのような操作をした時に設備をどのように動作させるか、また、自動運転では設備をどのように動作させるかについて記述した運転方案と呼ばれる仕様書を作成する。

次に、その仕様書に基づき、プログラミングの作業が行われる。プログラミングの言語としては、国際規格 IEC61131-3 のようなものや、それに準ずる図的言語が多く使われている。代表的な図的言語としては、演算を行うブロック間の信号の流れを表現するファンクションブロック図、論理演算をリレー回路として表現するラダー図、処理や状態遷移の手順を表現する SFC(Sequential Function Chart) などが広く用いられている。

自動搬送制御システムの構築におけるこのような過程の問題点は、設備的な視点から動作を設計し、運転方案として策定するプラント設計者と、それに基づいてシステムとしての視点からプログラミング作業を行うソフトウェア実装者との間で、依然として紙文書に基づいた仕様の伝達が行われ、その結果、プログラミング作業の効率化が達成できないことである。このような状況を打破するには、設計者と実装者の双方が利用可能なエンジニアリング情報を整備し、プラント設計者とソフトウェア実装者の統合化を図る必要がある。すなわち、ソフトウェアの再利用の単位としてプラント設計者の視点を導入し、ソフトウェアが再利用できるような仕様の策定と、部品を用いた仕様の実装が両立できるような環境を構築することが有効であると考えられる。

3. ソフトウェア再利用モデル

本節では、先に述べたようなラインの上位設計の視点からソフトウェアの再利用を行えるようにするためのモデルを提案する。まず、モデルを構築する上での基本方針について検討を行い、その後で本モデルの定義を行う。

3.1 基本方針

3.1.1 制御システムエンジニアリングの側面 制御システムにおけるエンジニアリング作業の大きな特徴として、制御対象となる物理的な構造がソフトウェアと密接に関連することが挙げられる。すなわち、プラントにどのような機械設備が存在し、信号として扱うべきセンサやアクチュエータにどのようなものがあるかといった情報がシステム構築の出発点となり、ソフトウェアの機能や構成に影響を与えている。

制御システムのエンジニアリング作業が対象とする情報を整理すると、以下の 3 つの側面が相互に関連して存在していると考えられる。

- 機械設備の構成を表す物理的構造
- 機械設備に対して、それがどのような機能を果たして

いるかを示す機能的構造

- 機能を達成するためのソフトウェアを動作させるためのネットワークやコントローラ、タスク、I/O の構成などを表すシステム構造

従来より採られてきたエンジニアリングの手法は、プログラミング言語の規定や制御システムの構築方法に着目していたため、対象となるプラントの物理的な設備構造が統合的に扱われてきていない。制御システムの上位概念となるプラントの物理的な設備構造をプログラミングやシステム定義と連動して扱うことで、上位設計との連続性が確保され、エンジニアリング効率の向上につながると考えられる。

3.1.2 ソフトウェア構造の分析 鉄鋼搬送ラインにおけるソフトウェアの再利用を考えるに当たり、まず、どのようなソフトウェア構造を採ればよいかを決定する必要がある。これには、ドメイン分析の手法が有効である⁽⁷⁾。筆者らは、過去に構築したいくつかの代表的なラインのソフトウェアの分析を行ったり、実際のソフトウェア開発者らの意見を取り入れながら、どのような単位で部品化を行えばよいかの検討を行った。

その結果、図 2 に示すようなソフトウェア構造を採ることとした。まず、外界との作用を行うセンサやアクチュエータに対しては、入力機能や出力機能を実現するプログラムが必要である。このような機能には、例えば、センサに対するチャタリング防止機能が含まれる。次に、センサからの入力を用いてアクチュエータへの出力を決定するための制御機能が存在する。制御機能には様々なものが存在しているが、例えば、位置検出機能、速度計算機能などが挙げられる。これらの入力機能、出力機能および制御機能は、ライン中の機械に関連するものとして位置付けられる。

さらに、ラインを自動運転するために、複数の機械にまたがって、機械の動作を順序的に制御するシーケンス処理機能が存在する。例えば、コイル材をリールに装着したり、コンベア上を搬送したりするなどの動作は、機械を順に動作させることによって達成されている。この機能は、ラインを自動運転するために、各機械からの情報の取得や、機械への動作指令を行う。

3.1.3 機械単位での部品化 次に、どのような単位で部品を管理し、適用対象に応じて選択して利用するかが大きな問題となる。鉄鋼ラインにおいては、ライン上にどのような機械設備が存在しているかによって、監視制御に必要な機能が決定される。したがって、部品の単位としては、ライン上に存在する機械設備に対応してまとめるのが適切であると考えられる。

鉄鋼ライン上に存在する監視制御の対象となる設備としては、センサやアクチュエータといった、監視制御システムと信号を直接やりとりするものと、センサやアクチュエータを複数組み合わせ、何らかの機能単位でまとめあげたものがある。本論文では、前者を「機器」、後者を「機械」と呼ぶことにする。例えば、機器にはスイッチや電磁弁が相当し、機械にはコイルカーやコンベアが相当する。

図 2 部品化のためのソフトウェア構造

Fig. 2. Software parts structure.

図 3 機械単位での部品化

Fig. 3. Machine unit parts.

機械は、製造物の搬送や位置合わせなど、複数の機器との入出力を組み合わせで何らかの機能を果たしている。この機能は、監視制御システムのプログラムによって達成されているため、機械に関連する複数のプログラムをまとめて部品化することで、機械を再利用の単位とすることができる。

また、プログラムの中には、機械を構成している機器と入出力を行うものも含まれている。このため、機器情報も含めて部品の単位として管理する。このような部品を、本論文では「機械部品」と呼ぶことにする。

現実のプラントでの機械や機器が、どのように機械部品として定義されるかを示す例を図 3 に示す。図の例では、「モータ」や「センサ」という機器から構成される「コイルカー」という機械に対して、機械部品「コイルカー」を定義している。機械部品には、実際の機器に対して機器情報と入出力信号が定義され、また、コイルカーの機能を果たすために必要なプログラム群が定義される。

3・1・4 部品の選択と展開 機械部品は、鉄鋼ラインに存在し得る機械の種別や、同じ機械種別であっても制御方法が異なるものなどに分類し、作成する。システム構築時には、適用対象のラインに存在する機械の性質や制御の方法に応じて適切なものを選択する。選択された機械部品に含まれるプログラムや機器入出力の情報は一括して複製される。この複製は機械単位で行われ、複製された機械部品を、本論文では「機械インスタンス」と呼ぶ。

機械インスタンスとして複製される際に、機械部品に含まれるすべてのプログラムや機器情報が複製されるのでは

なく、適用対象に応じて必要なものを選択できるようにする。これにより、一つの機械部品を再利用できる機会を増やすことができる。

また、機械インスタンスには、機械部品には含まれていない、適用対象に特化したプログラムや機器情報を定義することを可能にする。

適用対象に応じてすべての機械インスタンスが抽出されると、制御システムとして必要なプログラムや機器情報をすべて抽出することができる。

3・1・5 部品間の接続 選択された機械部品に含まれているプログラムや機器情報は、それぞれが扱う信号を相互に接続することにより全体としての動作が可能となる。しかしながら、現実規模のシステムにおいて、すべての信号の接続を人手により指定することは現実的ではない。そこで、プログラムや機器情報の信号が持つ名称を用いて、必要な接続を自動的に行う仕組みを提供する。接続の判定には、単純に信号の名称のみを用いるのではなく、機械インスタンスとその中に含まれるプログラムや機器の構造を利用し、3・6 節で述べるような名称の決定規則を設ける。これにより、システム構築時の手間を削減することを可能にする。

3・2 機械部品 上に述べた基本方針に基づき、機械部品を定義する。機械部品は、部品化対象となる機械を構成する機器の入出力、および、機械の機能を実現するためのプログラムを持つ機能部品から構成される。

3・2・1 機器情報の定義 機器情報は、以下の属性により定義される。

- 機器の種別
- 機器の名称
- 入力信号の集合
- 出力信号の集合

入力信号および出力信号は、下記の属性により定義される。

- 信号の型
- 信号の名称

例えば、コイルカーを構成している、上下方向に直線運動可能で途中で停止可能な電磁弁 (DS3P 型) に対する機器

情報は、種別が「DS3P」、名称は「昇降」、入力信号としては「ビット」型の「上昇」および「下降」として定義される。

3・2・2 プログラムの定義 プログラムは、図4に示すように、入力信号の集合から何らかの演算をして出力信号の集合を得るものである。プログラムは以下の属性により定義される。

- プログラムの名称
- 入力信号の集合
- 出力信号の集合
- プログラムの実装（ファンクションブロック図など）

例えば、コイルカーに属する位置検出プログラムは、センサ等の入力信号と検出位置を表す出力信号、および、位置を特定するための演算を表すプログラムの実装により定義される。

3・2・3 機械部品の構成要素 機械部品は、機械の物理的な側面を表現する機器情報と、機能的な側面を表現する機能部品により構成される。機能部品は、図5に示すように、機械が果たす機能を実現するための入出力変数、プログラム、機器情報、信号間の接続を構成要素として持つ。

機械部品を構成する機器情報および機能部品は、複数の適用対象に共通に用いるものと、適用対象毎に選択して用いるものとを区別して定義する。機械部品を構成する要素は以下の通りである。

標準機器: 部品化の対象となる機械であれば、どの適用対象に備え付けられるものでも存在していると考えられる機器に対する機器情報。

標準機能: 部品化の対象となる機械であれば、どの適用対象に備え付けられるものでも存在していると考えられる機能。

選択機能: 部品化の対象となる機械について、適用対象によって選択的に利用する機能と、その機能の実現に必要な機器に対する機器情報。

図4 プログラム

Fig. 4. A program.

図5 機能部品

Fig. 5. A function parts.

機械部品は、以上の要素を一まとまりとして管理し、再利用の単位とする。

3・3 機械部品のインスタンス化 機械部品は、対象ラインに存在する機械設備に対応して選択され、インスタンス化される。機械インスタンスは、機械インスタンス名称を持つ。

機械部品のインスタンス化の方法を図6に示す。標準機器および標準機能は、機械インスタンスには必ず複製される。一方、選択機能は、機械インスタンスを定義する時に必要なもの、必要でないものを対象に応じて選択し、選択されたものだけが機械インスタンスに複製される。

また、対象ラインに特化して必要な機能が存在する場合には、特化機能を定義することができる。

機械インスタンスに定義された機器情報および機能により、機器の入出力信号、プログラムを抽出することができる。

3・4 シーケンス処理記述 機械部品に含まれる各機能は、その機械の中で閉じた機能を実現するためのもので、機械間をまたがる処理、例えば、ラインの自動運転として、ある機械を上昇させた後、別の機械を前進させるようなシーケンス的な処理は含まれない。また、このようなシーケンス的な処理は、適用対象の機械の構成やラインの運転方法によって異なる場合が多く、再利用の対象としない。

したがって、機械を順に動作させるためのシーケンス処理は、適用対象毎に記述するものとする。シーケンス記述のための言語としては、SFC(Sequential Function Chart)などが知られているが、本エンジニアリングモデルでは、運転方案の記述などに使われている記法に倣った表1および図7に示すような図的言語を定義する。

3・5 インタロック記述 インタロックとは、機械を動作させるために満たさなければならない条件を表したものである。例えば、機械を自動で前進させる場合には、前進先に障害となる他の機械が移動して来ていないか、非常停止ボタンが押されていないか、自動運転中かなどがイン

図6 機械部品のインスタンス化

Fig. 6. Instantiation of a machine parts.

表 1 シーケンス記述言語の記号の意味づけ

Table 1. Semantics of symbols of the sequence description language.

Symbol	Semantics
Action	Do a machine action
Condition	Wait for a condition satisfied
Branch	Branch according to a condition
AND	Wait for all merges
OR	Wait for one of merges

図 7 シーケンス記述言語の記号

Fig. 7. Symbols of the sequence description language.

タロックとなる。

インタロックは、プラント設計からの視点でとらえると、機械を動作させる場合の条件と、自動運転を行う場合の条件に分けることができる。本エンジニアリングモデルでは、下記のような機械や自動運転に対する下記の観点からインタロックを記述する。

●機械レベルインタロック

機械干涉: 機械を動かすことにより危険が及ばないか、壊れないか、他の機械と干渉しないか等の条件を表す。

運転モード: ラインの運転状態が機械を動かすことを許しているかの条件を表す。

●シーケンス処理レベルインタロック

材干涉: 自動で機械を動かすにあたり、材料が干渉しないかの条件を表す。

ホームポジション: 自動で機械を動かすにあたり、機械が所定の位置にいるかどうかの条件を表す。

運転モード: 自動で機械を動かすことを許しているかの条件を表す。

3・6 信号間の接続 これまでに述べた機械インスタンスの機能群、自動運転を記述したシーケンス処理、機械の動作条件にはそれぞれ入出力信号が定義されているが、これらを連携して動作させるために、それぞれが持つ入出力信号の接続を行う。接続は、信号名称に応じて自動的にを行い、自動で行えなかったものについては人手により接続を指定する。

信号間の自動接続処理は、抽出されたすべての機能の入力信号に対して、後述する規則によって定められる信号の名称が一致する出力端子が存在するかを検索し、その結果、1つのみ存在する場合にその出力信号を入力信号に接続す

図 8 プラント設計レベルでの再利用

Fig. 8. Reuse for plant design level.

るものである。

自動接続の可否を決定するための信号名称は、以下の規則により定められる。

- 機械インスタンスに含まれる機能のうち、標準機能および選択機能として定義されたものについては、定義されている各信号名称に機械インスタンスの名称を追加したものとする。
- 機械インスタンスに含まれる機能のうち、特化機能として定義されたものについては、定義されている各信号名称そのままとする。
- シーケンス処理およびインタロックに関する信号については、定義されている各信号名称そのままとする。

以上の規則により、例えば以下のような接続が自動で行われる。

- 機械部品「コイルカー」に標準機能として「出力回路」があり、その入力信号として「前進」というものが存在しているとき、この機械部品を用いた機械インスタンスの名称が「No.1 CC」として定義されている場合には、自動接続のための信号名称は「No.1 CC 前進」となる。一方、シーケンス処理の Action として「No.1 CC 前進」というものが定義されているとする。この場合、双方の名称が一致するので接続される。
- 機械部品「コンベア」を用いた機械インスタンス「No.1 Conv」に特化機能として「インタロック条件」があり、その入力端子として「No.1 CC 前進」が定義されている場合、自動接続のための信号名称はそのままの「No.1 CC 前進」が用いられる。これは、前述のシーケンス処理の Action 「No.1 CC 前進」と名称が一致するので自動接続される。

3・7 エンジニアリングフロー このようなエンジニアリング手法を採ることによる従来型プロセスとの違いを図 8 に示す。従来は、プラント設計者による機器、機能、操作方法などのプラント仕様に基づき、ソフトウェア設計

図 9 ツールの画面例

Fig. 9. Screen of the tool.

者が入出力信号，プログラム，タスク割り付けなどを行ってきた。これに対し，本論文で提案する手法は，機器の入出力信号，および，機能や操作方法に対するプログラムを機械単位で一体として部品として管理し，プラント設計情報を参照して部品を選択およびカスタマイズすることにより，部品に定義された入出力信号やプログラムの情報を自動生成しようというものである。

4. エンジニアリングツールの実装

ここでは，前節で述べた再利用モデルに基づいて構築したエンジニアリングツールの機能について述べる。本ツールは，鉄鋼システム向けプログラミング環境 MELSUCESES⁽⁸⁾ の上位ツールとして位置付けられ，MELSUCESES で作成したプログラムを機械部品として管理したり，機械インスタンスに定義された機能に含まれるプログラムを MELSUCESES にダウンロードしたりする機能を有している。

4.1 画面例 本ツールの画面例を図 9 に示す。画面左側のツリー表示は，機械部品や機械インスタンス，シーケンス処理の構造を階層的に表現したものであり，管理情報の操作はこのツリー（以下，プロジェクトツリーと呼ぶ）上で行う。画面右側は，部品の処理を図的に編集するためのファンクションブロック図エディタや，自動制御手順を編集するシーケンス図エディタが表示されている。

4.2 機械部品の定義 機械部品の例を，図 10 に示す。図の例は，鉄鋼ラインでコイルを搬送しリールに装着する「コイルカー」を機械部品として定義したものの一部である。図に示すように，標準機器としては，コイルカーを上下に動かすための電磁弁である「Lift (DS3P Valve)」，下降限界に達したことを検出するセンサーである「Lower Limit(PXS Sensor)」が定義されている。また，標準機能としては，電磁弁への出力「Valve Output」が定義されている。さらに，選択機能として，位置検出機能「Position Detect」が定義されているが，適用ラインに応じて，一位

図 10 機械部品の定義例

Fig.10. An example of definition of a machine parts.

置 (1 Position) か二位置 (2 Position) のどちらかが選択されるようになっている。それぞれの選択機能には，機能の実現に必要な機器「PXS Sensor」が定義されている。

機械部品に含まれる機能部品のプログラム記述には，前述の MELSUCESES を用いる。

4.3 機械インスタンスの生成 プロジェクトツリー上に機械インスタンスを追加するには，選択する機械部品と，機械インスタンスの名称を指定する。また，選択機能のうち，どれを用いるかを設定する。現状では未実装ではあるが，機械部品を定義する際に，選択機能間の依存関係（選択機能 A と選択機能 B は同時に用いることができない等）を定義して，機械インスタンスでの選択機能の選択を効率化することも考えている。

4.4 シーケンス処理の記述 シーケンス処理は，前節で述べた様式によりシーケンス処理エディタで記述する。シーケンス処理は適用対象毎に内容が異なる場合が多いが，典型的な処理も存在しているため，部品として記述しておき，適用対象毎に変更して使用することができるようになっている。

本エディタで記述したシーケンス処理は，運転方案図としても用いることが可能である。

4.5 インタロックの定義 インタロックの定義は，機械インスタンスに対する各機械動作や，各シーケンスに対して，前節で述べた「機械干渉」などの視点で動作に必要なとなる条件を指定することで行う。機械動作に対するインタロック定義の画面例を図 11 に示す。画面中の表の縦軸は，上昇や下降等の各機械動作を表し，横軸は，条件の指定に用いる信号名称を表している。表中の丸印は，縦軸に対応する機械動作を行うためには，横軸に対応する信号がオンでなければならないという条件を示している。

4.6 自動接続 抽出されたプログラムの信号間の自動接続処理は，各信号に対して前節で述べた自動接続用信号名称の決定規則に基づいて信号名を決定し，信号の名

図 11 インタロック定義の画面例

Fig. 11. Screen of the interlock definition.

称と信号の型が一致するものを接続するようにしている。

また、自動接続処理の際、機械部品に標準機能あるいは選択機能として登録されているプログラムの信号に対しては、通常は機械インスタンスの名称を付与することになっているが、自動接続される信号を増やすため、信号を定義する際に機械インスタンスの名称を付けないように指定できるようにした。これにより、例えば、ライン全体の運転モードのような機械間で共通に使われる信号についても自動接続の対象となる可能性を高めることができる。

4.7 多対一接続 機械インスタンスとして抽出された機能群の中に、例えば、ある機械を動作させる条件を示す信号が複数あったり、ある機械への動作指令信号が複数の機能部品から出力されている場合などの状況が考えられる。この場合、複数の出力信号に何らかの演算を施して入力信号に接続しなければならないが、この演算処理は特化機能として記述する必要があり、部品の再利用性として好ましくない。

そこで、ある入力信号に対して接続対象となる出力信号が複数存在する場合、出力信号を AND あるいは OR 演算した結果を入力信号に接続する多対一接続の仕組みを提供する。多対一接続の例を図 12 に示す。図の例では、「Function A」と「Function B」の出力信号が「Function C」の入力信号と意味的に一致し、接続したい場合を示している。この場合、ツール上で「AND 接続」を行うように指定すると、「Function A」と「Function B」の出力信号に対して AND 演算を施した結果を「Function C」の入力端子に接続するような実行プログラムが自動的に挿入される。

4.8 実行プログラムの生成 本ツールでのエンジニアリング結果は、前述のプログラミング環境 MELSUCSES に対してダウンロードされる。機械部品に定義されていたプログラムは MELSUCSES の言語で書かれており、選択されたプログラムがコピーしてダウンロードされる。シーケンス処理は、MELSUCSES のシーケンス処理記述言語プログラムに変換され、ダウンロードされる。インタロックは、MELSUCSES の持つインタロック表に変換されてダウンロードされる。自動あるいは人手により行われた信号間接続は、MELSUCSES 上のプログラム変数間の接続情報としてダウンロードされる。

MELSUCSES にダウンロードされたプログラムは、実行周期などの設定、コンパイル、タスク割り付け、I/O 割り

図 12 多対一接続

Fig. 12. An example of N:1 connection.

付けなどが行われた後、対象のコントローラに転送される。

5. 評価

本節では、これまでに述べたエンジニアリングモデルとそれに基づくツールを実際のシステム構築に部分適用した結果と、その評価について述べる。適用対象としては、コイルの搬送を行うコイルカーと呼ばれる機械設備の制御を選定した。

システムの生産性向上を評価するには、同じ対象に対して従来のエンジニアリング手法と本ツールを利用した手法とで工数を比較することが必要である。しかし、工数の測定は携わる人の技能などにも影響されるなど、現実には人や費用の面で困難である。そこで、本論文では、工数の削減に貢献する観点として、部品の再利用性および部品間の自動接続機能に対する評価を実施する。

5.1 再利用性の評価 今回の適用では、再利用性を考慮した部品群の構築と、それらの対象ラインへの適用を並行して行った。そのため、構築した部品群の再利用効果を厳密に測定することはできないが、構築したプログラムモジュールが再利用できるかどうかを元に、今後の適用においてどれくらいの効果が見込めるかの評価を行うこととする。

今回の適用でコイルカーの制御として構築したプログラムモジュールの単位としては、全部で 154 個であり、そのうち 32 個については、2 あるいは 3 つの中から選択的に利用することを仮定している。これらのプログラムをコイルカーに対する機械部品として、コイル搬送や高さ調芯等を行う機能部品に登録した。今回の適用対象に対して用いられたプログラムモジュールは全部で 137 個であり、そのうち同じメーカーの機械に対する再利用が見込まれるものは 97 個であった。これにより、再利用率としては約 70%が見込まれる。これらの機能別の内訳は、表 2 に示す通りである。

再利用が見込まれないプログラムモジュールは、操作系、機械動作条件等、ライン毎によって考え方が異なる部分が多くを占めている。このような部分についての再利用性を高めるには、ラインの運転方法などのプラント設計のレベルからの標準化が必要であると考えられる。

表 2 適用した機能部品

Table 2. Applied functional parts.

Function	Num. of Programs	Num. of Re-usable	Ratio
Auto	38	22	58%
Transfer	31	23	74%
Lift	45	37	80%
Others	23	15	52%
Total	137	97	70%

本ツールにおける再利用の方法は、従来から行われてきたプログラムモジュール単位の再利用ではなく、機械や機能といった単位で行われるのが特徴である。これは、再利用されるプログラムモジュールの選択にかかる工数の削減にも貢献すると考えられる。

5・2 自動接続機能の評価 部品群の構築に当たっては、機能や機械の名称に一貫性を持たせ、部品の入出力信号の名称にそれを用いることにより、自動接続機能を有効に利用できるように考慮した。

その結果、自動接続機能については、自動接続できた信号が 327、手動接続を要した信号が 316 であり、約 50% の信号に関して自動接続が行えている。自動接続処理に要した時間は、通常のパソコンで 10 秒程度であった。

これにより、従来人手により信号名の検索機能等を用いて行っていた接続作業の効率を大幅に改善できると考えている。

6. あとがき

本論文では、鉄鋼ラインの自動搬送制御領域のソフトウェア生産性を向上を目的として、上位概念である設備や運転の視点で部品を構築し、再利用を行うためのエンジニアリングモデルを提案した。また、本モデルに基づいたエンジニアリングツールを開発し、実際のラインに対するシステム構築に部分適用し、有効性を検証した。

本エンジニアリングツールは適用の初期段階であり、部品構築と対象ラインへの適用を同時に行っているため、本来の目的である生産性の向上の達成は道半ばであるが、今後、実適用を重ねて部品構築を進め、再利用性を向上させていきたいと考えている。

(平成 15 年 12 月 25 日受付、平成 16 年 6 月 30 日再受付)

文 献

- (1) 鉄鋼プロセス制御ソフトウェア CAE の現状と課題：電気学会技術報告，第 526 号 (1994-12)
- (2) 本位田真一・内平直志・中村英夫：「制御分野における自動プログラミング」，情報処理，Vol.28, No.10, pp.1398-1404 (1987-10)
- (3) IEC1131-3 ハンドブック：PLCopen JAPAN (1998)
- (4) 高橋正和・津田和彦：「プラントソフトウェアの効率的な要求定義法」，情報処理学会論議，Vol.42, No.3, pp.518-528 (2001-3)
- (5) H.J. Köhler, U. Nickel, J. Niere, and A. Zündorf: "Integrating UML Diagrams for Production Control Systems", Proc. of the International Conference of Software Engineering, pp.241-251 (2000)

- (6) F. Bonfatti, G. Gadda, and P.D. Monari: "An Improved Process for the Development of PLC Software", Proc. of the International Conference of Software Engineering, pp.400-410 (1997)
- (7) 田村恭久・伊藤 潔・杵島修三：「ドメイン分析・モデリング技術の現状と課題」，情報処理，Vol.35, No.10, pp.952-961 (1994-10)
- (8) 小島 透・渡部徹雄・干潟寛昭：「三菱統合一元化 SE システム」，情報処理学会研究報告 (ソフトウェア工学)，No.85-015 (1992)

高 田 秀 志 (正員) 1968 年生。1991 年京都大学工学部情報

工学科卒業。1993 年同大学院工学研究科情報工学専攻修士課程修了。同年三菱電機 (株) 入社。2001 年京都大学大学院情報学研究科社会情報学専攻博士後期課程修了。2004 年 1 月より、京都大学大学院情報学研究科 COE 研究員。産業用データベース、履歴データベース、教育システムなどの研究に従事。博士 (情報学)。情報処理学会，システム制御情報学会，ACM，IEEE CS 各会員。

小 林 潤 (非会員) 1968 年生。1991 年 3 月武蔵工業大学

電気電子工学科卒業。1993 年同大学大学院電気工学科修士課程修了。同年三菱電機 (株) に入社。現在 東芝三菱電機産業システム (株) に勤務。

松 田 茂 彦 (非会員) 1954 年生。1977 年 3 月大阪大学工学

部原子力工学科卒業。1979 年大阪大学大学院工学研究課前期課程修了。同年三菱電機 (株) に入社。現在 東芝三菱電機産業システム (株) に勤務。技術士 (情報工学部門)

中 田 秀 男 (正員) 1964 年生。1987 年 3 月京都大学工学部

情報工学科卒業。1989 年 3 月同大学院工学研究科情報工学専攻修士課程修了。同年三菱電機 (株) 入社。現在、同社先端技術総合研究所にて、ユーザインタフェース、視覚的プログラミングなどの研究開発に従事。2002 年広島市立大学情報科学研究科情報科学専攻博士課程修了。博士 (情報科学)。情報処理学会，ACM 各会員。

堀 池 聡 (非会員) 1957 年生。1982 年京都大学大学院工

学研究科数理工学専攻修了。同年三菱電機 (株) 入社。1986 年マサチューセッツ工科大学客員研究員。工学博士。分散実時間システムの研究に従事。IEEE，電子情報通信学会各会員。