プラントソフトウエアの効率的要求定義法

高橋正和† 津田和彦†

本論文では，プラントの動作制御や運転支援を行うソフトウエア（PS: Plant Software）の開発を目的とした「プラントソフトウエアの要求定義支援システム（PRDS: Plant software Requirements Definition System）」を提案する．PRDS は PS 要求定義確認用プロトタイプのソフトウエア部品（PSC: Plant Software Components），要求定義支援ツール（RDT: Requirements Definition Tool），動作確認用シミュレータ（MRS: Motion Review Simulator）から構成される．PRDS を用いた PS 要求定義プロセスは，はじめに RDT を用いて要求定義を行い，それをパラメータとして PSC に与えて目的 PS の要求を特化する．次に，MRS を用いて PS の動作確認を行い，要求の妥当性を確認する．さらに，要求が妥当となるまで，パラメータ修正を繰り返し，PS に対する要求を洗練していく．提案手法の性能を確認するためいくつかの事例に対して PRDS を用いた PS の要求定義を行った．その結果，従来の一般的な開発手法であるウォーターフォール型 PS 開発プロセスと比較して開発時間を 30〜37 [%] に短縮できることを確認した．

The Efficient Method of Plant Software Requirements Definition

Masakazu Takahashi† and Kazuhiko Tsuda†

This paper proposes “the Plant Software (below PS) requirements definition support system (PRDS: Plant software Requirements Definition System)” which aimed at the development of the PS that is used for motion control and operation support of the plant. PRDS is composed by the software components to build PS prototype (PSC: Plant Software Components), the support tool to define the PS requirement (RDT: Requirements Definition Tool), and simulator which confirms the PS’s motion (MRS: Motion Review Simulator). In the PS requirements definition process which PRDS was apply for, at first the requirements of PS is defined by using RDT, and that is given to PSC as a parameter, and a target PS is specified. Next, the motion of PS is confirmed by using MRS, and the validity of the requirement is confirmed through that. Furthermore, until a suitable requirements can be defined, a parameter is revised many times, and a requirements of PS are refined. To confirm the performance of proposed method, requirements definition of PS using PDRS is done in some cases. Consequently, the reduction of development time becomes 30–37 [%] in comparison with usual water-fall type PS development process.

1. はじめに

化学製品の製造や材料の加工等を行う機械設備をプラントと呼ぶ．プラントは，ニーズの多様化や動作の精密化にともなって，機能や性能が高度化してきた．そのため，手動運転が困難となり，コンピュータによる動作制御や操作支援が必要となっている．プラントの動作制御や操作支援を行うソフトウエアをプラントソフトウエア（以下，PS: Plant Software）と呼ぶ．

通常，プラントの開発ではプラントごとに顧客とプラント製作会社の間で使用目的，機能および性能等を決定する．そして，それに基づいて設計と製作が行われる．したがって，それに付随する PS も同様の開発形態となる．このような開発形態のソフトウエアを二者間契約型ソフトウエアと呼ぶ．二者間契約型ソフトウエアでは，開発に先立って顧客と製作会社の間でソフトウエアに対する機能や性能に対する要求を明確化して合意をとる．そして，その要求に基づいてソフトウエアを開発する．つまり，開発開始時点で決定した要求に基づいてソフトウエアの開発が行われるため，要求を正確に定義することが重要となる．ところが，近年の PS 開発では，プラントの複雑化や高度化にともない，下記の問題が生じている．

- プラントの運用方法の決定が遅れるため，PS 開発の開始時点で PS への要求が明確化できない．
- プラント本体の設計仕様が不確定なため，変更が生じ，PS への要求の変更が生ずる．

† 筑波大学大学院企業科学専攻
Tokyo Graduate School of Systems Management, The University of Tsukuba

- プラント設計者の PS への要求が十分検討されていないため，不整合や漏れがある．
- PS 開発者の技術レベルがまちまちなため，不適切な設計が行われたり，検討項目に抜けが生じたりする．
- 顧客要望により PS 完成間際に機能追加がされ，開発した PS と追加した PS の間で矛盾が生じる．

この結果，信頼性の低下，開発期間の長期化，開発コストの増加等の問題が発生している．原因は，プラントの機能・性能が高度化や多様化したため，従来のような書類ベースの調整を通して要求定義をする方法では矛盾や漏れが生じ，要求を正確に定義することが困難になったためである．なお，ここでは要求定義を「顧客やプラント設計者から製作目的，運用方法，機器図面，計測制御図面等の情報を収集してプラント機器の挙動や制約条件を決定すること」と定義する．これは，顧客やプラント設計者には，PS 内部の挙動を知る必要がないためである．PS 内部の挙動は基本設計段階以降で決定する．

ところで，要求定義の方法は，すべてのソフトウエアの開発対象（以下，ドメイン）に適用可能な方法と特定のドメインに対して適用可能な方法に大別される．すべてのドメインに対して適用可能な方法には構造化分析設計[1)]やオブジェクト指向分析設計[2)]があり，それをシステム化した CASE[3)]がある．これらは，多くの支援ツールが市販されている等の利点がある．一方，すべてのドメインへの適用を可能とするため，開発者が定義しなければならない項目が多く，作業が煩雑になる等の欠点がある．特定のドメインに対して適用可能な方法にはドメイン分析・モデリング[4)]，プロトタイピング[5)]等がある．これらは，開発対象ドメインに特化しているため，適用が容易であり必要な情報のみを定義すればよく，開発効率が良い等の利点がある[6)]．さらに，繰返し利用による効率向上も期待できる．その一方，ドメイン分析・モデリング等の事前準備や内容の理解に時間を要する，プロトタイプ作成が開発の負荷となり期間を圧迫する等の欠点がある．

しかし，これらの手法は，いずれもカスタマイズが複雑であり，コンピュータの専門的な知識を持たない顧客やプラント開発者が使用することは困難であった．そのため，PS 開発にあまり利用されなかった．ところで，後述するように PS のドメイン分析・モデリングをした結果，類似した構造や機能を有していることが判明した．そこで，その類似性に着目し，PS 要求確認用プロトタイプを作成するためのソフトウエア部品を作成する（以降，PSC: Plant Software Components）．この PSC を利用することで PS 要求確認用のプロトタイプを短時間で作成する．作成したプロトタイプの挙動は，動作確認用シミュレータを用いて確認する．不適合があれば，プロトタイプを修正する．この挙動確認作業と修正作業を繰り返し実施することで，要求を洗練していく．ところで，要求定義段階では，ソフトウエアの機能や構造を明確にすることが重要であり，詳細なカスタマイズは必要としない．この性質を利用して，本論文では，PSC のカスタマイズ範囲を限定する代わりにカスタマイズの容易性と要求内容の可視性を重視した PSC を開発する．

2. PSC の開発

本章では一般的なソフトウエア部品作成手順[7)]により PSC を開発した結果について報告する．

2.1 PS ドメインの定義

本論文で取り扱う代表的なプラントを**表 1** に示す．約 30 種類のプラントのハードウエアとソフトウエアについて比較した結果，以下の特徴が得られた．以降，次の特徴を持つソフトウエアを PS と再定義する．

(1) PS の構成に関する特徴

- リアルタイム性（応答や処理時間）が要求される．
- 一定時間間隔（周期）で繰り返し同じ処理をする．
- 計測データや動作指示をトリガにして処理を行う．
- PS 構成はプラント機器構成に応じて変化する．

(2) PS の機能に関する特徴

- 機能実行管理，機器動作制御，機器とのデータ入出力，外部とのインタフェース（以下，IF），状態表示，異常時処理の各機能を持つ．
- 機能実行管理は PS のタスクの実行を管理する．
- 機器動作制御はプロセス制御とフィードバック制御（以下，FB 制御）に大別される．
- 機器とのデータ入出力は，アナログとデジタルの入出力に大別される．入力はセンサーからのデータ計測，出力は機器への動作の指示である．
- 外部との IF は，動作指示用の制御盤やネットワークへの接続に大別される．
- 状態表示は，画面，ペンレコーダー等に計測したデータを表示する．
- 「異常時処理」は，機能縮退，冗長系の切替え，緊急停止等がある．

2.2 PS ドメインモデルの作成

PS はプラントの動作制御や運転支援が目的である．そのため，PS ドメインモデルを作成するには，対象となるプラントの機器と機能の両方を検討する必要がある[8)]．

表 1 代表的なプラントの例
Table 1 Examples of typical plants.

プラントの名称	プラントソフトウエアのタイプ	代表的な機能
航空エンジン	動作制御	・エンジン出力のフィードバック制御 ・中央制御コンピュータとのデータ交換
発電用ガスタービン	動作制御	・ガスタービン出力のフィードバック制御 ・自動機器点検と故障発生時の安全化
	状態監視	・データ収集、表示および蓄積 ・故障診断と故障対策ガイダンス
貯蔵タンク（液化天然ガス、石油 等）	動作制御	・貯蔵物の流量のフィードバック制御 ・設備の準備や運転のためのプロセス制御 ・操作室からの動作指示の受信やデータの表示
物流倉庫	状態監視	・出庫の指示 ・在庫の表示と入庫指示 ・操作室からの動作指示の受信やデータの表示

表 2 代表的なプラントの機器構成（一部抜粋）
Table 2 Machine configuration of typical plants (partly extracted).

プラントを構成する機器の名称 / プラントの名称	アナログ値を入力出力する機器			オン／オフ信号を入力出力する機器			コマンドやデータの授受をする機器		
	温度勾配炉	流量調節弁	…	電磁弁	各種電源	…	操作盤	データ表示装置	…
宇宙ステーション搭載曝露環境基盤技術実験装置	○		…	○	○	…		○	…
航空エンジン制御装置		○	…	○	○	…			…
発電用タービン制御装置		○	…	○	○	…	○		…
貯蔵用プラント（タンク）加圧制御装置		○	…	○	○	…	○		…
貯蔵用プラント流量調節装置		○	…	○	○	…	○		…

はじめにプラントの機器について検討する．機器の動作制御を行うには，機器の種類に応じた PS を使用する必要がある．しかし，機器の種類は，100 種類以上に及ぶうえ，次々に新しい機器が開発されるため，全機器を列挙することは不可能である．しかし，PS 要求定義をするためには，機器の詳細な動作制御の情報は必要としないため，機器を PS 要求定義が可能な程度まで抽象化して最小限の種類でプラント構成を記述できるようにする．必要最小限の機器を明確にするため，主要プラントの機器構成を**表 2** に示す．その結果，機器が取り扱う信号に着目することで，**図 1** に示す 3 つのグループに分類できた．

次に PS 機能について検討する．代表的な PS 機能構成を**表 3** に示す．PS は個々の機器の動作制御をする必要があるため，機器の種類の影響を受けるが，機能は共通であることが分かる．そこで，機器が 3 種類に分類できる性質を用い，入出力，動作制御，外部との IF の各機能を抽象化し，共通化する．その結果，PS の機能を以下の 6 種類に大別する．

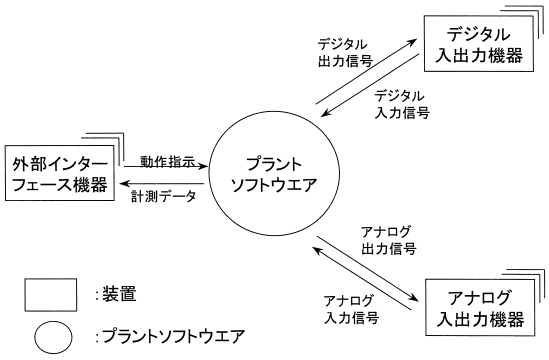

図 1 標準的な機器を用いたプラントの記述
Fig. 1 Plant description using standard machine.

(1) 実行管理機能

実行管理機能は，PS の各機能を適切な起動方法，順序，周期で起動する．起動方法は，周期的な繰返し起動（以下，周期起動）と割込みによる起動（以下，割込起動）に大別される．周期起動する機能は，動作制御機能，入出力機能，外部表示機能および異常処理の

表 3 PS の機能構成（一部抜粋）
Table 3 PS's function configuration (partly extracted).

機能の名称 / プラントの名称	タスクの実行管理	プロセスの実行管理	機器に出力する操作量を計算する			機器の対して動作指示や計測信号の入出力をする			
	タスク管理	プロセス管理	フィードバック制御	アナログ制御	デジタル制御	アナログ制御	デジタル制御	アナログ制御	…
宇宙ステーション搭載曝露環境基盤技術実験装置	○	○		○	○	○	○	○	…
航空エンジン制御装置	○		○	○	○	○	○	○	…
発電用タービン制御装置	○	○	○	○	○	○	○	○	…
貯蔵用プラント（タンク）加圧制御装置	○		○	○	○	○	○	○	…
貯蔵用プラント流量調節装置	○		○	○	○	○	○	○	…

一部（定常点検）である．割込起動する機能は，外部機器との IF と異常処理の一部（緊急停止）である．

(2) 動作制御機能

動作制御機能は，決められた手順で機器の操作をするプロセス制御と，プラントの状態を目標状態に近づけるための出力計算をする FB 制御に大別される．

プロセス制御は，使用目的がプラントごとに異なるため類似点がない．しかし，ある動作が完了するまで次の動作を行わないというプロセス管理の方法は同様となる．通常，プロセスは状態遷移表やブロックダイヤグラムを使って記述される．

FB 制御は，機器ごとに最適な制御をするため，詳細な制御ロジックの類似点がない．しかし，要求定義段階では詳細な制御ロジックは必要とならない．したがって，入力データ（制御量）と制御目標から出力データ（操作量）を計算する基本機能は同様となる．

(3) 入出力機能

入出力機能は，機器に対する出力や，センサーからデータを入力する．プラントを構成する機器は 100 種類以上あるが，前述の議論によりアナログ入出力機器とデジタル入出力機器に大別される．これらの機器には，データ入出力のタイミングや操作方法の違いがあるが，要求定義段階では，その差異は重要でない．したがって，アナログおよびデジタルの信号を入出力する基本機能は同様となる．

(4) 外部機器との IF 機能

外部機器との IF 機能は，制御盤やネットワーク経由で接続された制御室等との間で動作指示を授受する．プラントで用いられる IF には RS-422，GPIB，MIL-STD1553B，LAN 等がある．通信プロトコルは，送信する動作指示の頻度や通信データ量等により，規格で定められたものが採用される．したがって，外部機器との IF には類似点が少ない．しかし，前述の議論により，要求定義段階では IF や通信プロトコルの正確な模擬は必要とならない．

(5) 状態表示機能

状態表示機能は，機器から入力したデータを端末等に表示する．状態表示の方法としては，数値データ，トレンドグラフ，メータ表示等の方法がある．状態表示の方法やレイアウトは顧客の要求や表示画面の大きさにより決定されるため，類似点は少ない．

(6) 異常処理機能

異常処理機能は，機器から入力したデータを用いて定常点検，機能縮退および緊急停止等をする．

定常点検では，計測データが安全基準内にあるか確認し，故障コードを出力する．点検の方法は類似しているが，入力データの種類，安全基準および故障コードについては類似点はない

機能縮退では故障した機器を使用しないでプラントを稼働させる．これは機器の構成に影響を受けるためプラント間で類似点はない．

緊急停止では機器の安全化を行い，プラントを停止させる．これは機器の構成に影響を受けるためプラント間で類似点はない．

2.3 PSC の適用方法

上述の PS ドメインモデルを作成した結果，プラントの持つ個々の機能は，同一な部分，類似した部分，多様な部分の 3 種類に分類できることが明らかになった．

同一な部分は，PS ごとの相違がほとんどないため，無修正で使用する PSC として事前準備する．類似した部分は，外部から PS の固有情報をパラメータとして与えることでカスタマイズする PSC（以降，パラメータ PSC）として事前準備する．多様な部分は，入出力データのインタフェースのみを規定した PSC（以

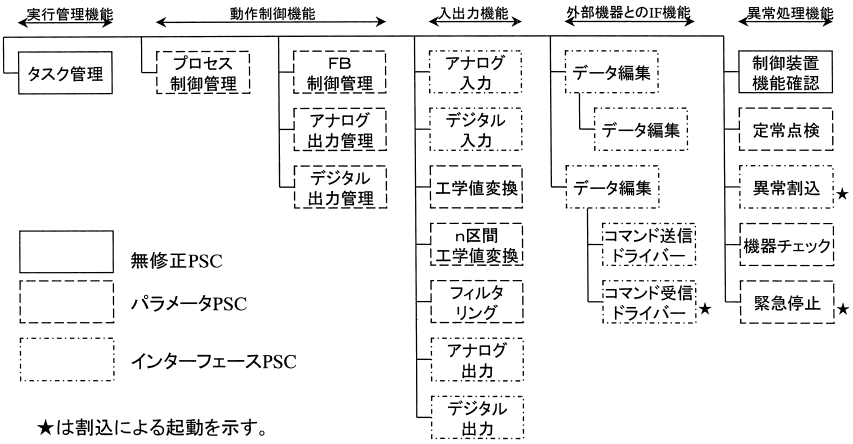

図 2 PSC の一覧
Fig. 2 List of PSC.

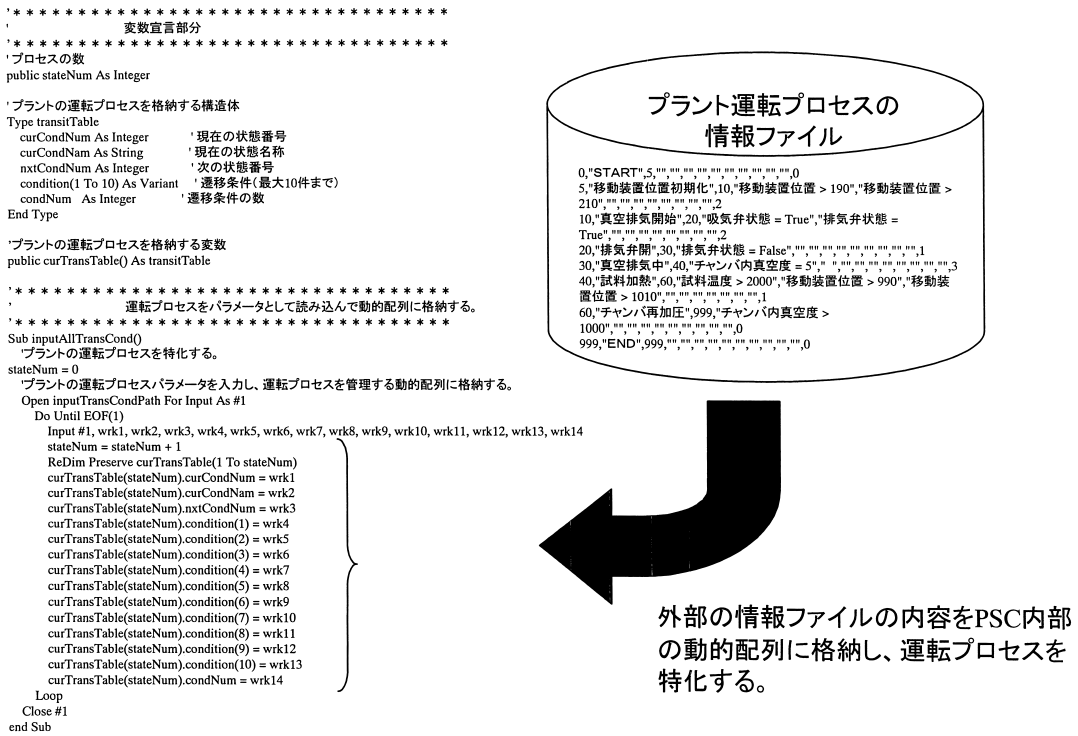

図 3 代表的な PSC のソースコード
Fig. 3 Typical PSC's source codes.

降，インタフェース PSC）として事前準備する．インタフェース PSC 内部の処理部分は，PS プロトタイプの作成時に記述する．

以降，上記の 3 種類のソフトウエア部品を PSC（Plant Software Component）と再定義する．PSC は適用対象を PS 要求定義用プロトタイプの作成に限定することで，その個数や機能を限定した．また適用対象を広げるため，容易にカスタマイズできるようにした．

ところで，機能自体が異なる部位には，入出力，外部機器との IF，状態表示等がある．これらは PS の要求を定義するには必須な部位ではないため，スタブモジュール等を作成して代用することが可能である．

ドメインモデルに従って作成した PSC の構成を**図 2** に示す．図 2 中の実線四角形は無修正で使用する PSC，点線四角形はパラメータ PSC，一点鎖線四角

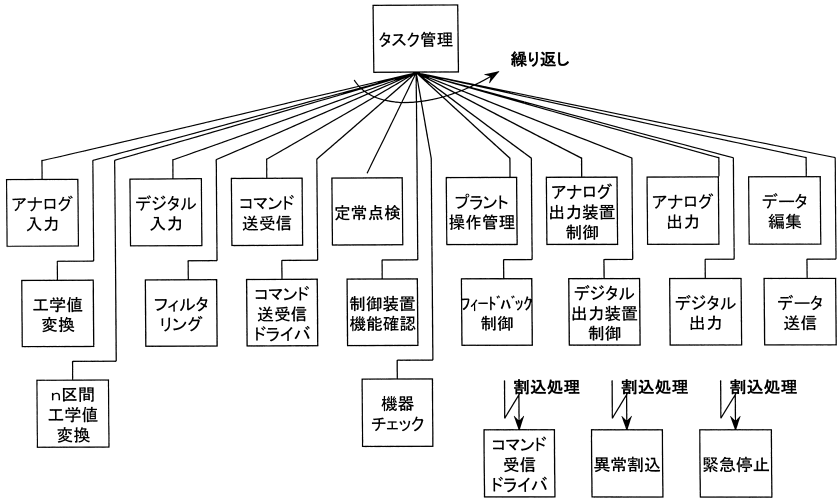

図 4 PSC の起動順序
Fig. 4 Execution sequence of PSC.

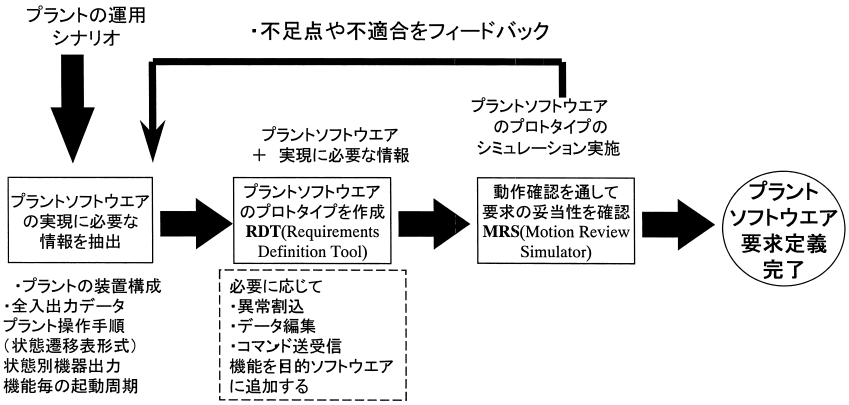

図 5 PSC ベースの PS の要求定義
Fig. 5 PSC based PS requirement definition.

形は，インタフェース PSC を表す．なお，状態表示機能は，市販ツールを適用することが可能との判断から PSC 化を見送った．**図 3** にパラメータ PSC のソースコードの例を示す．

最後に各 PSC の起動順序を決定する．PS の代表的な状態（アイドル，通常，動作指示受信，異常，緊急停止の各状態）について PSC 起動シーケンス図を記述し，その結果をもとに各 PSC の起動順序を決定した．**図 4** に決定した PSC の起動順序を示す．

3. PSC を用いた要求定義プロセス

本章では，PSC を用いた要求定義プロセスとそれを効率的に行うための支援ツールについて提案する．

3.1 要求定義プロセス

本論文で提案する PSC ベースの要求定義プロセスを**図 5** に示す．本プロセスでは，PS プロトタイプを稼働させ，挙動を確認する．不具合が確認されれば要求定義した内容へのフィードバックをする．この作業を繰り返すことで，要求を顧客やプラント設計者のイメージに合致したものへと洗練していく．この要求定義プロセスを実現するための要件を以下に示す．

- 要求定義が容易にできること．
- プロトタイプが容易に作成できること．
- プロトタイプの挙動を可視化できること．

これらの要件を満足させるには，PSC を特化するための固有情報を定義するツール，プロトタイプの挙動を可視化するツールが必須である．試作したツールについて 3.2 節および 3.3 節で紹介をする．以降，PSC とこれらのツール群を総称して PRDS（Plant software Requirements Definition System）と呼ぶ．

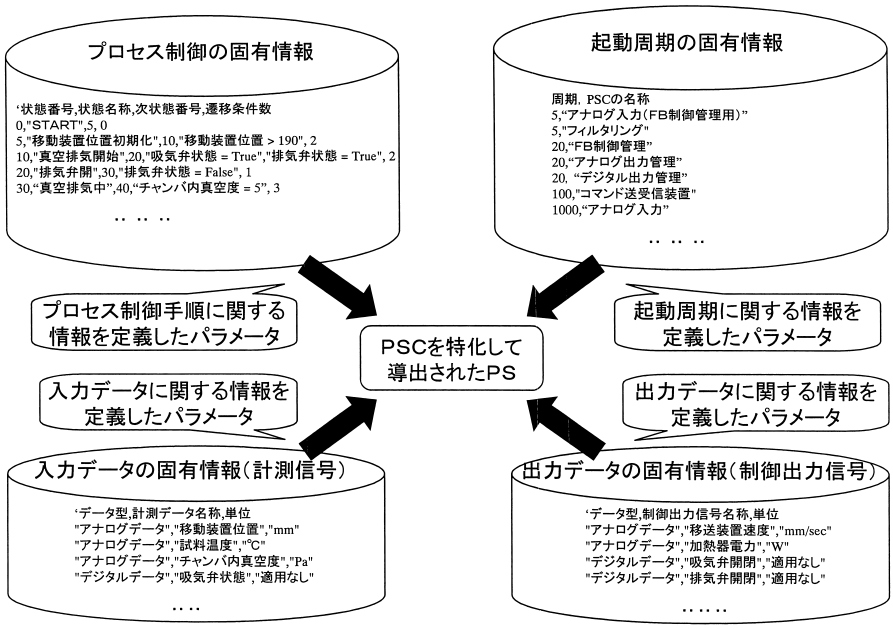

図 6 個別ファイルの関係
Fig. 6 Relations between every information files.

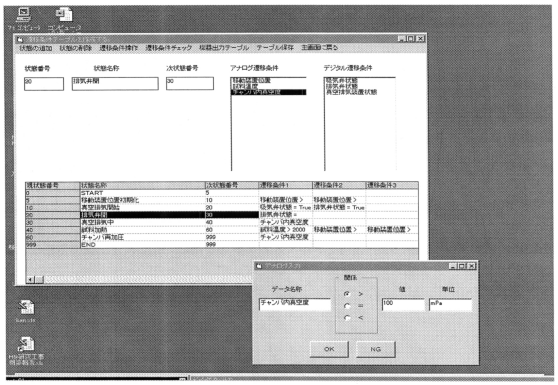

図 7 RDT の画面
Fig. 7 Screen of RDT.

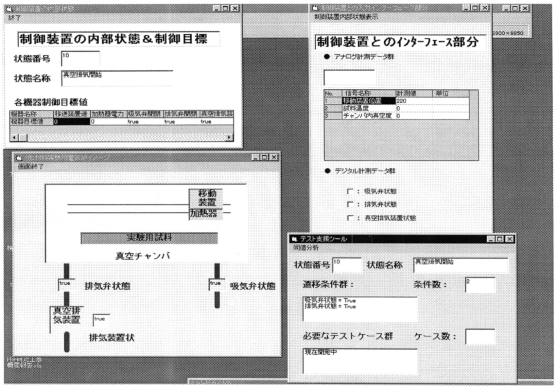

図 8 MRS の画面
Fig. 8 Screen of MRS.

3.2 要求定義支援ツール

図 5 に示す RDT（Requirements Definition Tool）とは，PS を導出するために必要な入出力データ，プロセス制御，起動周期等の固有情報を定義するためのツールである．入出力データは，プラントの入力である計測信号と出力である制御信号を定義する．プロセス制御は，プラントの運転手順を状態遷移表形式で定義する．起動周期は，各 PCS の起動間隔を定義する．RDT では，上記情報を個別ファイルで出力する．個別ファイルの関係を**図 6** に示す．RDT では，要求の確認を容易にするため，プロセス制御を状態遷移表形式で入力することとした．例として RDT の画面を**図 7** に示す．

さらに，PS 開発効率を向上させるため，固有情報を用いて要求仕様書を自動作成する仕組みを作成した．本ツールは固有情報を事前準備した要求定義書のテンプレートにはめ込んで，要求定義書を作成する．

3.3 動作確認用シミュレータ

図 5 中に示す MRS（Motion Review Simulator）とは，作成した PS プロトタイプを実際のプラントを模擬した画面上で稼働させる動作確認用シミュレータである．**図 8** に示す MRS 画面において，PS の挙動をビジュアルに確認することで，要求の矛盾やタイミング上の不適合等を見つけ出す．見つけ出された不適

合は，RDT を用いて修正する．この修正作業と MRS を用いた確認作業を繰り返し実行することで，PS の要求を顧客や設計者のイメージに合致したものに洗練していく．

一般にプラントの動作表示画面は，市販のツールを適用することで，容易に作成できる．プラントの動作表示画面には，表示データとしてプラントの内部変数が必要となるため，内部データを取得する関数を準備し，外部から利用できるようにした．これにより，MRS と画面表示ツールの独立性が増し，要求に応じてデータ表示画面をカスタマイズすることが容易になった．

4. PSC の適用と評価

本章では，5 つの事例に対して PSC と PS 要求定義プロセスを適用した結果について報告し，評価する．

4.1 適 用 結 果

はじめに PSC の適用結果を**表 4** に示す．総 PSC 数は，PS の作成に適用した PSC の個数，適用 PSC 数は動作制御，異常処理等の各 PS の間で共通化できない PSC の個数を指す．

次に PSC を用いて作成した PS プロトタイプのコード再利用率を**表 5** に示す．再利用率は下記のとおり定義する．なお，LOC（Lines of Code）は，行数を表す[9]．

再利用率 = (再利用した LOC)
/(再利用した LOC+新規作成 LOC).

次に各 PS を作成した際，パラメータ PSC のカスタマイズに用いたパラメータ数を**表 6** に示す．プロセス制御数は，プラントの運転手順の段階数を表す．入出力データ数は，プラントの計測信号と制御信号の数を表す．固有パラメータ LOC とは，前述のパラメータ PSC をカスタマイズするために記述したパラメータの総 LOC である．

最後に，要求定義時間に対する実験結果を**表 7** に示す．なお，要求定義時間の定義は次のとおりである．

要求定義時間 = 要求分析から要求定義書を
作成するまでに要した時間.

要求定義時間は，従来の開発プロセスを適用した場合の開発時間（類似の PS の開発実績）と比較した．PSC と要求定義プロセスの適用例として，事例 A の PS 構成を**図 9**，プロセス制御概要を**図 10** に示す．

4.2 総 合 評 価

4.2.1 PSC の再利用率に関する評価

PSC の再利用率について再利用 LOC のばらつき，新規作成 LOC のばらつき，再利用率，固有情報数のばらつきの 4 つの視点から検討と評価をする．

表 4 PSC 適用結果
Table 4 Result of application of PSC.

	総PSC数	適用PSC数
A	21	13
B	13	5
C	16	8
D	12	4
E	19	11

表 5 再利用 LOC と新規作成 LOC の関係
Table 5 Relations lines number of reusable codes and new created codes.

	再利用 LOC	新規作成 LOC	再利用率［%］
A	425	32	92. 9
B	298	27	91. 7
C	345	34	91. 0
D	278	25	91. 7
E	408	38	91. 5

表 6 入出力データとプロセス制御の数と固有パラメータの関係
Table 6 Relations between number of in/output data and process control, and lines of charactaristic parameters.

	入出力ﾃﾞｰﾀ数	プロセス制御数	固有ﾊﾟﾗﾒｰﾀLOC
A	21	25	105
B	27	14	82
C	10	6	44
D	9	8	45
E	12	20	82

表 7 PS 要求定義時間に関する結果
Table 7 Results of the time of PS requirements definition.

	本ﾌﾟﾛｾｽ [hour]	従来ﾌﾟﾛｾｽ [hour]	本ﾌﾟﾛｾｽ／従来ﾌﾟﾛｾｽ[%]
A	42	134	31
B	28	76	37
C	19	56	34
D	22	61	36
E	35	118	30

(1) 再利用 LOC のばらつきの検討

再利用 LOC を算出する理論式について検討する．PS は，基本 PS と適用 PS から構成されている．基本 PSC とは，各事例で共通に使用されているタスク管理，プラント操作管理，装置制御（3 種類），データ編集，制御装置機能確認，定常点検の 8 個の PSC を指す．適用 PSC は各事例で共通化できない PSC を指す．PSC 適用結果は表 4 のとおりである．したがって，再利用 LOC は式 (1) で記述できる．なお，集計の結果，基本 PSC の総 LOC は，201 行であった．

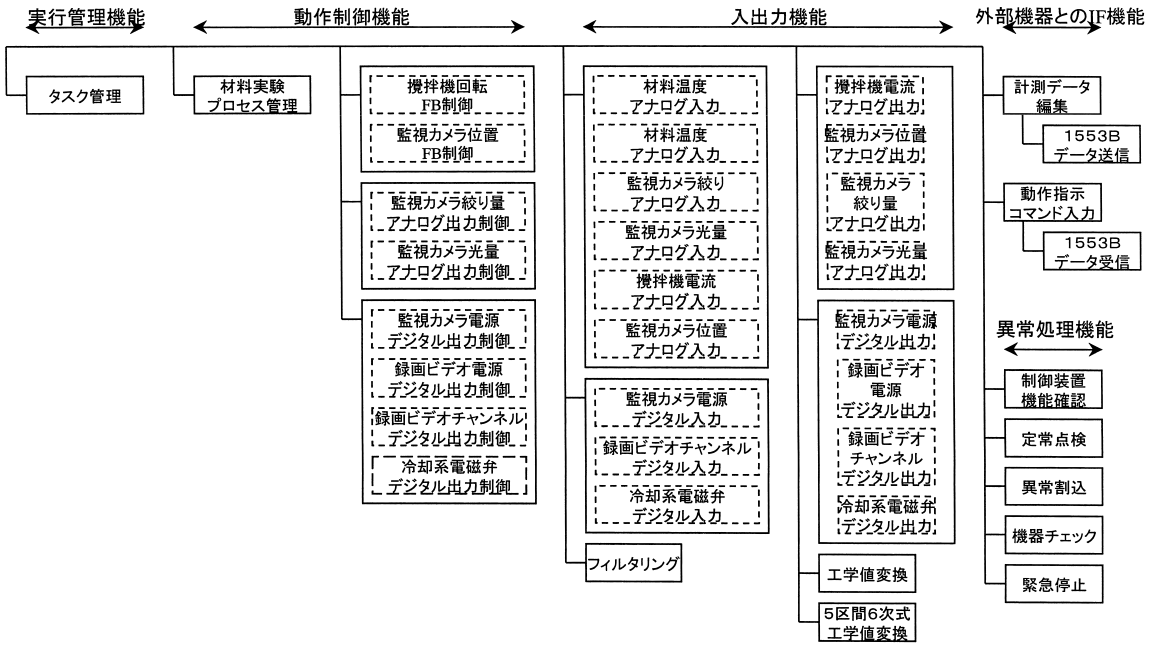

図 9 事例 A の PSC 構成
Fig. 9 PSC configuration of the case A.

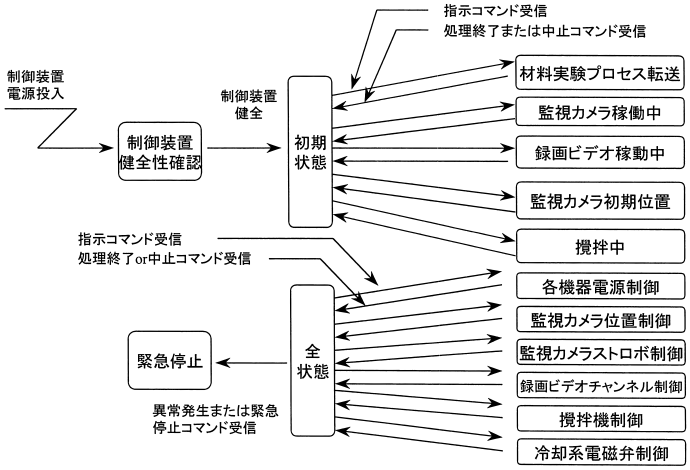

図 10 事例 A のプロセス制御
Fig. 10 Process control of the case A.

再利用 LOC = 基本 PSC の総 LOC（定数）
+ 適用 PSC の総 LOC (1)

次に適用 PSC の総 LOC について検討する．適用 PSC は主にプラントの入出力を模擬する PSC であるため，要求定義段階では詳細な機能を実現する必要がない．したがって，プロトタイプを作成する場合には入出力と簡単な応答を模擬する適用 PSC（一般にはスタブと呼ばれる）を作成して適用する．そのため，適用 PSC の LOC は使用した適用 PSC の個数に比例する．適用 PSC は，サブルーチン定義，変数宣言，入力データ変換，出力データ変換，応答計算の 5 つの部分から構成されている．過去の事例における各部位の LOC は，最大 25 [LOC]，最小 17 [LOC]，平均 20.4 [LOC] であり，標準偏差は 2.7 [LOC] であった．平均 LOC に比べて標準偏差は小さいので，平均 LOC を適用 PSC の LOC として採用する．したがって，再利用 LOC は式 (2) に示すとおりとなる．

再利用 LOC = 201 + 20.4 × 適用 PSC の数 (2)

式 (2) を用いて事例 A〜E の再利用 LOC を計算すると 466.2 [LOC]，303.0 [LOC]，364.2 [LOC]，282.6 [LOC]，425.4 [LOC] となる．一方，実際の再利用 LOC は表 5 のとおりであり，式 (2) から算出した LOC との誤差は 10 [%] 未満である．この数字は，見積誤差としては許容範囲である．したがって，式 (2) は再利用 LOC の計算に適用可能である．

(2) 新規作成 LOC の検討

事例 A〜E では，プロセス制御の遷移条件に追加があり，PSC に新規コードを追加した．結果は表 5 のとおりである．一般に新規作成 LOC は個別の要求に応じたカスタマイズとなるため，新規 LOC を予測することは難しい．しかし，過去の事例における新規作成 LOC を調査した結果，最大 39 [LOC]，最小 28 [LOC]，平均 33.8 [LOC] であり，標準偏差は 3.9 [LOC] であった．新規開発 LOC は，プログラム全体の 10%程度であり十分小さいので，この誤差は，見積りという目的においては特に問題とならない．よって，平均 LOC を新規作成 LOC とする．

(3) 再利用率の検討

再利用 LOC は式 (2) で記述できるため，再利用率は式 (3) で記述できる．

$$\text{再利用率} = (201 + 20.4 \times \text{適用 PSC 数}) \Big/ (201 + 20.4 \times \text{適用 PSC 数} + 33.8) \quad (3)$$

式 (3) を用いて事例 A〜E の再利用率を計算すると，93.2 [%]，90.0 [%]，91.5 [%]，89.3 [%]，92.6 [%] であった．一方，実際の再利用率は，表 5 のとおりである．式 (4) との誤差は 3 [%] 以内でありよく一致している．したがって，再利用率は式 (3) で計算できる．

(4) 固有情報 LOC の検討

固有情報とは，パラメータ PSC をカスタマイズするためのプロセス制御，入出力，起動周期のパラメータであり，固有情報 LOC は固有情報の行数である．プロセス制御のパラメータは，すべてのプロセスに対して「次のプロセス」,「遷移条件」,「その際のプラント機器への出力」を定義する．今回開発した PSC では図 6 に示すように各項目をそれぞれ 1 行で記述するため，遷移あたりの記述行数は 3 行となる．入出力データは，図 6 に示すようにプラントの計測データと機器制御信号を 1 行に 1 種類ずつ記述する．したがって表 6 の入出力データ数と一致する．起動周期は，図 6 に示すように各 PSC ごとに 1 行で記述する．したがって，表 4 に示す事例 A〜E の総 PSC 数と一致する．したがって，固有情報 LOC は遷移の数に比例する．したがって，固有情報 LOC は式 (4) となる．

$$\text{固有情報 LOC} = \text{総 PSC 数} + \text{入出力データ数} + 3 \times \text{遷移の数} \quad (4)$$

固有情報は RDT を用いて定義した内容から機械的に作成される．そのため，式 (4) により正確に計算することができる．

4.2.2 PS の要求定義時間に関する評価

はじめに提案する要求定義プロセスにより，要求定義時間が削減される理由について検討する．一般に，要求定義作業は要求の列挙，要求の妥当性確認，要求定義書の作成に大別される．

要求の列挙では，プラントの機器構成，PSC の起動周期，プロセス制御等の明確化をする．従来プロセスでは，経験の少ない開発者は何を定義すべきか試行錯誤していたため時間を要していた．本プロセスで定義すべき項目を明確化したことで，試行錯誤が減少し，PS 開発時間が短縮した．ところで列挙する項目の中でプラント機器構成と PSC の起動周期は，設計情報から容易に特定できる．したがって，要求定義の時間差は，プロセス制御の複雑さにより生ずる．プロセス制御の複雑さは，図 10 から明らかなように，遷移の数に比例する．したがって，要求の列挙時間は遷移の数に比例する．

要求の妥当性確認では，要求をウォークスルーやプロトタイプにより確認する．従来プロセスでは，文書ベースで確認をしていたため，チェックもれが生じた．その結果，後戻りが生じ，時間を要していた．本プロセスにより PS の挙動確認が容易となり，チェック漏れが減少した．さらにパラメータ PSC を採用したことでプロトタイプ作成時間が減少した．ところで，一般に状態遷移モデルの遷移確認を行う場合には，全遷移パターンを確認しなければならない．遷移パターン数は，状態遷移の分岐数に依存する．したがって，状態遷移の確認に要する時間は，全遷移パターンの遷移数の総和に比例する．しかし，今回の適用対象となる PS のプロセス制御は，固有の目的に特化しており，単一の目的のために使用されるプロセスであるため，状態遷移の分岐はほとんどない．そのため，PS のプロセス制御の妥当性確認時間は遷移数に比例する．

要求定義書の作成では，要求から要求定義書を作成する．RDT の使用により，固有情報を定義するだけで，要求仕様書が作成可能になった．この作業は，機械的に実施されるため，開発者の作業はない．

以上の議論により，要求定義時間は，状態遷移数に比例することが分かった．式 (5) に記述する．

PS 開発時間
＝プラントの機器構成定義の時間
＋起動周期の定義に要する時間
＋要求の列挙と確認の単位時間 × 状態遷移数
(5)

上記定数が明らかになれば，式 (5) により要求定義時間を予測できる．ところで，上記の定数の中で要求の列挙と確認の単位時間は，個人の能力に依存しており，様々である．よって，本論文では 5 つの適用事例から計算し，11.8 [時間] および 1.22 [時間] と仮定した．したがって，式 (5) は式 (6) のよう記述できる．

要求定義時間 $= 11.8 + 1.22 \times$ 遷移の数 (6)

式 (6) を用いて事例 A〜E の要求定義時間を計算すると 42.3 [時間]，28.9 [時間]，19.1 [時間]，21.6 [時間]，36.2 [時間] となる．一方，実際の要求定義時間は表 7 のとおりであり，式 (6) との誤差は 5 [%] 以下となり，ほぼ一致している．したがって，式 (6) は要求定義時間の計算に適用できる．

最後に本プロセスと従来プロセスによる要求定義時間を比較すると，従来の 30〜37 [%] であり，平均 33 [%] となった．本数値はドメインの特徴により異なるため，一概に比較することは困難である．しかし，PS ドメインにおいては，PS 要求定義に要する時間を 1/3 にすることができ，本論文で提案する PSC と要求定義プロセスが有効であることが確認できた．

5. おわりに

本論文では，要求定義用 PSC，要求定義プロセス，支援ツールについて解説した．これらを適用することで，従来の 1/3 の時間で PS 要求定義を実現できるようにした．この結果，以下のことが明らかになった．

- PSC は PS の要求を定義するのに有効であり，PS の構成や機能を表現するのに適している．
- 本プロセスと RDT，MRS のツールは，適合性の高い要求を短時間で定義するのに有効である．
- 要求定義時間を誤差 5 [%] で見積り可能にした．

これらのことから，PS 開発の正確な時間見積りやコストダウンが実現され，プラント製作会社の競争力が向上した．さらに競争力を向上させるためには，PS 開発の全プロセスを効率化する必要がある．そのためには，定義した要求を基本設計等のプロセスに漏れなくシームレスに引き渡す仕組みを構築することが課題となる．

参考文献

1) Demarco, T.: *Structured Analysis and System Specification*, Yordon Inc. (1978).
2) Booch, G., Rumbaugh, J. and Jacobson, I.: *Unified Modeling Language Semantics and Notation Guide 1.0*, Rational Software Corporation (1997).
3) Matsumoto, M., et al.: Specifications reuse process modeling and CASE study-based Evaluations, *COMPSAC91, Proc. 15'th annual international computer software & applications* (1991).
4) 伊藤　潔, 杵嶋修三, 田村恭久, 廣田豊彦, 吉田裕之：ドメイン分析・モデリング，共立出版 (1996).
5) Martin, J.: *Rapid Application Development*, Macmillian Publishing Company (1991).
6) 名取万里，加賀谷聡，本位田真一：ドメイン分析に基づく仕様再利用法，情報処理学会論文誌，Vol.37, No.3, pp.393–408 (1996).
7) Tracz, W. and Coglianes, L.: *Domain-Specific Software Architecture Engineering Process Guidelines*, IBM Federal System Company, ADAGE-IBM-92-02 (1992).
8) DeBaud, J. and Schmid, K.: A Systematic Approach to Derive Scope of Software Product Lines, *ICSE'99*, Los Angeles CA, pp.34–43 (1999).
9) 松本正雄（編）：ソフトウエアのモデル化と再利用，共立出版 (1996).

(平成 12 年 2 月 1 日受付)
(平成 12 年 12 月 1 日採録)

高橋　正和（学生会員）

1988 年立教大学理学部物理学科卒業．石川島播磨重工業（株）に勤務．1998 年筑波大学大学院経営システム科学専攻修了．現在，同大学院博士課程在学中．計測自動制御学会，経営情報学会等会員．

津田　和彦（正会員）

1986 年徳島大学工学部情報工学科卒業．同年三菱電機（株）入社，1991 年住友金属工業（株）入社．1994 年徳島大学大学院工学研究科システム工学専攻修了．工学博士．1998 年筑波大学大学院経営システム科学専攻助教授．自然言語理解，データベース，ヒューマンコンピュータインタラクションナレッジマネージメントに興味を持つ．電子情報通信学会会員．