

医薬品製造設備における制御ソフトウェアの 効率的コンピュータ・バリデーション

高橋正和[†] 津田和彦^{††}

医薬品の製造では、設備運用の自動化が進んでいる。これにより、設備の頭脳たる制御ソフトウェアは、製造する医薬品の品質に大きな影響を与える。そのため、医薬品製造の管轄機関は、医薬品製造会社に対して、制御ソフトウェアが要求仕様と適合しており、適切な品質の医薬品を製造できることの保証を義務付けている。これをコンピュータ・バリデーションと呼ぶ。実務的には、制御ソフトウェアの各種仕様書の整備と網羅的な検証の記録により健全性を立証する。しかし、その作業が医薬品製造会社の大きな負担となっており、効率化が叫ばれている。本論文では、以下を実施することで効率的なコンピュータ・バリデーションの実施を可能にする。(1) 制御ソフトウェアの開発に適用可能なソフトウェア部品を準備する。(2) 設計仕様書間の関係をデータベース化して設計情報の展開を追跡できるようにする。(3) ソフトウェア部品に対応するテスト仕様書を事前準備して効率的な検証を実施する。上記手法を適用した結果、コンピュータ・バリデーションに要する時間が約 50[%] に削減された。

An Efficient Computer Validation Method for Pharmaceutical Facility Motion Control Software

MASAKAZU TAKAHASHI[†] and KAZUHIKO TSUDA^{††}

Recently, automatic operations of pharmaceutical facility have become popular. Consequently, Control Software (below CSW) create a negative impact to the quality of pharmaceutical products. Regulating authority forced validation to the pharmaceutical companies that CSW was developed according to the requirements and could produce appropriate quality drugs. This is called Computer Validation (CV). CV validates the CSW's soundness based on the development of the documents related to the CSW and exhaustive tests of CSW. But those works, especially development of documents and test results, have become a large work load. This paper proposes below effective CV processing. (1) Develop design specification templates, and software components that are applied to CSW development. (2) Create Database that defines relationship between design specifications, and enable to trace design information deployment. (3) Test CSW efficiently according to pre-established test specifications corresponding to software components. As the result of applying above the method to actual CSW developments, CV time lessens by 50[%] in comparison with conventional CV method.

1. はじめに

我々の周りには、様々な医薬品が存在し、健康の維持・病気の回復のために必要不可欠となっている。医薬品は人体に多大な影響を与えるため、その品質が適切であることは必須である。そのため、米国食品医薬局、英国医薬品庁、日本国厚生労働省等、各国の医薬品の製造を管轄する機関（以下、管轄機関と呼ぶ）は、

医薬品製造会社に対して、医薬品が適切な原材料・プロセス・製造設備で製造されており、製造された医薬品が適切な品質であることを立証し、管轄機関の承認を得るように義務付けている¹⁾。この立証作業をバリデーションと呼ぶ。

近年の医薬品は、自動化された設備で製造されるケースが大半であるため、製造設備の制御ソフトウェア（Control Software: CSW）の品質が、医薬品の品質に多大な影響を与える。そのため、管轄機関は医薬品製造設備の CSW に対してもバリデーションを実施するよう義務付けている。これをコンピュータ・バリデーション（Computer Validation: CV）と呼ぶ。CV は、「開発された CSW が顧客要求に基づいて開発

[†] 株式会社ギャラクシーエクスプレス
Galaxy Express Corporation

^{††} 筑波大学大学院ビジネス科学研究科
Graduate School of Systems Management, The University of Tsukuba

されていること」と「CSW の動作が適切であること」を文書・記録により立証する。しかし、CV を実施するには大量の設計仕様書・検証記録の作成が必要となるため、医薬品製造会社の大きな負担となっている。

本論文では、CV に対する医薬品製造会社の負担軽減を目的とし、CSW の開発効率を向上させる CV 指向 CSW 開発手法について提案する。CSW は、パーソナルコンピュータや個別の制御装置等（以下、CE と呼ぶ）に搭載され、C 言語や Basic 等で記述されるものと、Programmable Logic Controller に搭載され、ラダー図等で記述されるものに大別される。前者は規模の小さな医薬品製造設備の動作制御に、後者は規模の大きな医薬品製造設備の動作制御に用いられることが多い。本提案では、CE に搭載される小規模な CSW の CV の効率化について検討する。これは、CE に搭載される小規模な CSW 開発の方が、予算的・期間的に厳しいため、早期に問題解決する必要があるためである。なお、CSW 開発に関わるすべての作業は医薬品製造会社が行い、CV に必要な開発やセキュリティの規準は整備されているものと仮定する。

CSW は医薬品製造設備の制御という単一目的のために開発されたものであり、制御対象の機器も限定されるため、CSW を構成する機能には類似性がある。そこで、CSW を構成する機能をパラメータにより容易に特化できるソフトウェア部品として実現した^{2),3)}。ソフトウェア部品の仕様は、パラメータ化された部分以外は事前に確定しているため、設計仕様書やテスト仕様書のテンプレートは事前作成が可能である。

本提案では、この特徴を利用して、各種仕様書をパラメータの書き換えのみで作成できるようにした。さらに、CSW の機能関係や設計パラメータをデータベースに登録して管理した。これを用いてソフトウェア部品、パラメータ、各種仕様書、検証記録を一元的に管理することで、効率良く CV 作業を実施できるようにした。

2. バリデーションの概要と問題点

バリデーションは、医薬品製造プロセスに対するプロセス・バリデーションと医薬品製造設備の CSW に対するコンピュータ・バリデーションに大別できる。FDA (Food and Drug Administration) では、プロセス・バリデーションを「当該工程があらかじめ決められた規格、品質特性に適合する製品を恒常的に生産することを、高度に保証する証拠を文書で示すこと」と、コンピュータ・バリデーションを「当該のコンピュータシステムがあらかじめ決められた仕様（機能・性能）

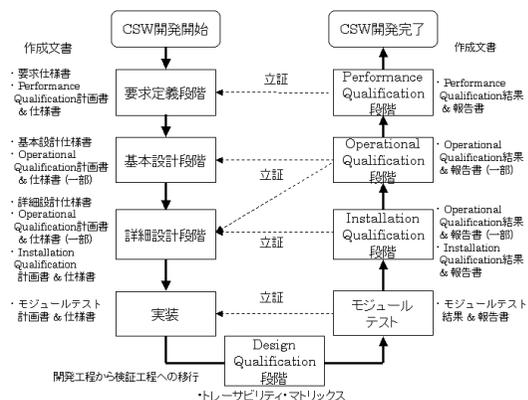


図1 コンピュータ・バリデーションの概要
Fig. 1 Outline of Computer Validation.

に従って確実に動作することを、高度に保証する証拠を文書で示すこと」と定義している⁴⁾。なお、本論文ではコンピュータ・バリデーションのみを対象とする。

2.1 コンピュータ・バリデーションの概要

CV については、各国の管轄機関が方針を提示したが、具体的な CV の実施方法が記述されていない。そのため、その解釈や実施方法が医薬品製造会社ごとに様々となり、医薬品製造会社・管轄機関ともに対応に苦慮していた。そこで、標準的 CV 実施手順として国際製薬技術協会 (International Society for Pharmaceutical Engineering: ISPE) の GAMP (Good Automated Manufacturing Practice) が提唱された⁵⁾。GAMP は各国の CV 実施方法を網羅しているので、それに則って CV を実施すれば、管轄機関から容易に認証が得られると考えられ、標準となった。

図1にCVの実施手順の概要と各段階における代表的な成果物を示す。図1の左側はCSWの開発工程を、右側は検証工程を表している。開発工程はウォーターフォール型の開発を前提としており、要求定義、基本設計、詳細設計、実装の各段階から構成される。検証工程は、モジュールテスト、Installation Qualification, Operational Qualification, Performance Qualification から構成される。さらに、両工程にわたって実施される Design Qualification がある。以下に各 Qualification について説明する。

Installation Qualification (以下、IQ) は、モジュールテスト済みの適切なソフトウェアが使用されていることを確認し、その結果を文書化することで実施する。本提案における IQ は、モジュールテスト済みの実装用ソフトウェア部品 (3.1 節で示す) を用いて CSW が実現されていることを確認し、IQ 結果報告書を作

成することとなる。

Operational Qualification (以下, OQ) は, CSW が詳細設計仕様書と基本設計仕様書で記述された機能を有していることを検証し, その結果を文書化することで実施する。本提案では, 詳細設計仕様書に対応する OQ は, 1 個の実装用ソフトウェア部品の機能 (たとえば入力機能等) の検証と OQ 結果報告書の作成となる。基本設計仕様書に対応する OQ は, 複数の実装用ソフトウェア部品を結合した機能 (たとえば, 入力, フィルタリング, 工学値変換の実装用ソフトウェア部品を結合した計測機能等) の検証と OQ 結果報告書の作成となる。

Performance Qualification (以下, PQ) は, CSW が要求仕様書で記述された機能と性能を有していることを検証し, 文書化することで実施する。PQ は, CSW を実際の医薬品製造設備に搭載し, 実際の医薬品製造プロセスを実施して, 要求と CSW が適合していることを検証する。

Design Qualification (以下, DQ) は, CSW の開発工程と検証工程を通じて, 要求仕様書, 基本設計仕様書, 詳細設計仕様書および CSW の間で機能がどのように展開されているのか, さらに, それらがどのテスト仕様書で検証されているのかを確認し, その結果を文書化することで実施する。このように設計結果が追跡できることをトレーサビリティと呼び, これを文書化したものをトレーサビリティ・マトリックスという。

2.2 コンピュータ・バリデーション実施上の問題点

医薬品製造会社は, 自社の医薬品製造設備に関わる CSW に対して, 個別に CV を実施している。しかし, 以下に示す理由により多大な負担となっている。

(1) 各種設計仕様書の作成

CV では, 2.1 節で述べたように大量の設計仕様書を作成する必要がある。それらは, 開発者が CSW ごとに新規に作成しており, 大きな負担となっている。さらに, CSW の開発完了後も, 変更にともない, 設計仕様書を改定する必要がある, 大きな負担となっている。

(2) 制御ソフトウェアの適合性確認

CV では, 開発された CSW が要求仕様書に適合していることを立証する必要がある。そのために, DQ を実施して, 要求仕様書—基本設計仕様書—詳細設計仕様書—CSW の間で, 要求がどの構成モジュールに展開されていくかトレーサビリティ・マトリックスを作成する必要がある。そのため, 開発者は大量の設計仕様書を調査し, それらの間の関連付けを行う必要がある, 大きな負担となっている。

(3) 網羅的テスト仕様書の作成と実施

CV では, 開発された CSW が各設計仕様書に適合しているか検証を実施し, それを記録として残すことが義務付けられている。さらに, 設計仕様書で記述されたすべての機能が漏れなく正確に CSW に組み込まれているか, 網羅的な検証を実施して確認する必要がある。しかし, CSW の構造は複雑であるため, 限られた期間と費用の中で網羅的なテスト仕様を作成し, 実施することは困難である。

3. CV 指向制御ソフトウェア開発手法の提案

本章では, 2.2 節に示す 3 つの課題を解決する手法について 3.1 節から 3.3 節に記述する。

3.1 ソフトウェア部品ベース開発による効率的な設計仕様書作成

各種設計仕様書作成の効率化は, ソフトウェア開発にとって大きな課題であり, 様々な手法が提案されている。定型フォーマットを用いて設計仕様書を作成する手法⁶⁾があるが, 開発したソフトウェアとの適合性確認を手作業で行う必要がある。また, 市販品としてソースコードを解析して詳細設計仕様書を生成するツール^{7),8)}が存在するが, 上流工程の設計仕様書を作成することができない。以上のように, 単一の手法を用いて, 要求定義から実装まで, すべての設計仕様書の作成効率を向上させることについては, 十分な成果が出ていないのが実情であった。

本論文では, 医薬品製造設備のドメイン分析を行うことで開発したソフトウェア部品と, その設計仕様書テンプレートを使用することで, CSW と設計仕様書を効率的に作成する手法を提案する。提案手法では, まずドメイン分析結果に基づいて CSW の機能を共通・類似・個別に分類する。共通機能については無修正で使用するソフトウェア部品と設計仕様書テンプレートを, 類似機能についてはパラメータによりカスタマイズして使用するソフトウェア部品と設計仕様書テンプレートを準備する。個別機能については標準化が困難なため, ソフトウェアと設計仕様書を新規に作成する。これらを用いた CV 指向開発環境 (CV Oriented Development Environment: CVODE) の概要を図 2 に示し, 各開発段階の作業内容を説明する。

なお, 通常は CV 実施前に医薬品の品質に与える影響度のリスク評価を実施して, 結果に応じた CV を実施するが, CVODE は最も CV 作業が多くなるリスク“高”に対応可能な構成とする。リスクが“中”と“低”の場合でも, リスク“高”と同様な CV を実施しておけば問題がない。さらに, CVODE を用いて CSW を

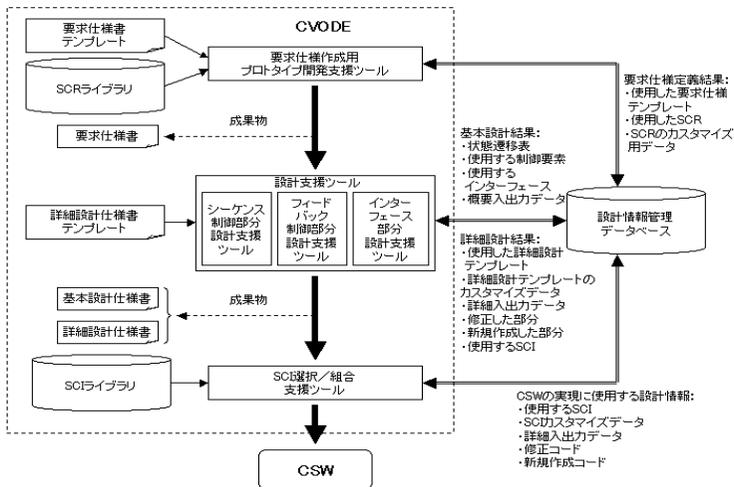


図2 CVODEの概要 Fig. 2 Outline of CVODE.

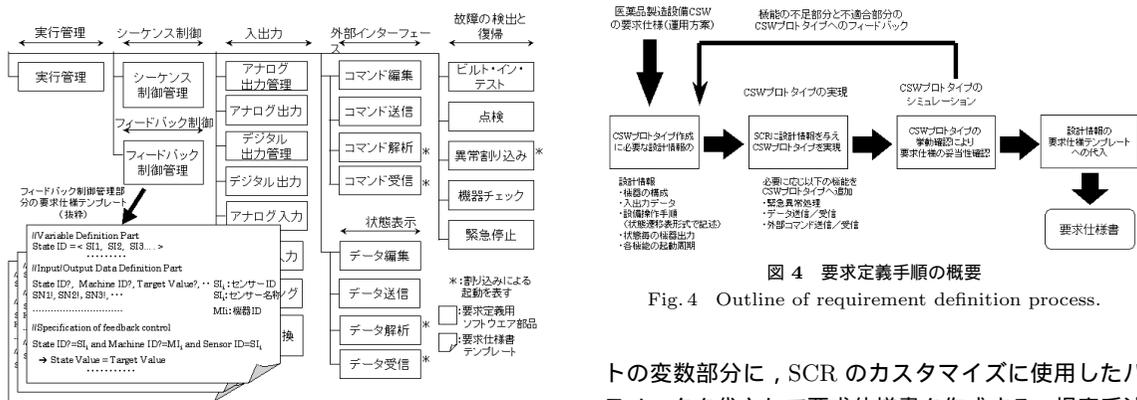


図3 SCRの概要 Fig. 3 Outline of SCR.

開発した場合には、CVに必要な開発仕様書は自動的に作成されるため、CVに要する時間に差異が生じない。以下に各段階におけるCV作業内容を記す。

(1) 要求定義段階

図3に示すドメイン分析の結果をもとに作成した要求定義用ソフトウェア部品 (Software Component for Requirement definition: SCR) を用いてCSWのプロトタイプを作成する。そして図4に示すように、要求とCSWの動作が一致するまで、SCRのパラメータを操作しつつ、繰り返し動作シミュレーションを実施する。ここで操作するパラメータは、機器の構成、入出力データ、設備操作手順等である。パラメータの操作および挙動確認シミュレーションは、要求仕様作成用プロトタイプ開発支援ツールを使用して行う²⁾。この作業の完了後、図3に示す要求仕様書テンプレ

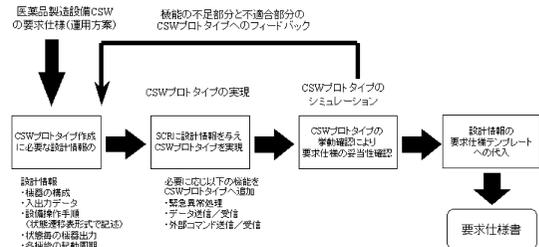


図4 要求定義手順の概要 Fig. 4 Outline of requirement definition process.

トの変数部分に、SCRのカスタマイズに使用したパラメータを代入して要求仕様書を作成する。提案手法では、要求を厳密に定義するため、要求仕様書テンプレートはZ言語で記述した。なお、使用したSCR、要求仕様書テンプレート、および、そのカスタマイズに用いるパラメータは、要求定義結果として設計情報管理データベース (詳細は3.2節で述べる) に登録し、CSWの開発および保守に使用する。

(2) 基本設計段階

図5に示すように要求仕様書の記述内容を機能に分割し、入出力データの概要を決定し、この結果を基本設計仕様書とする。要求仕様書の機能への分割には、Z言語が階層的に機能を記述できる特徴を用いる。機能分割と入出力データ決定は、設計支援ツールを使用して行う。これは、シーケンス制御部分を状態遷移表形式で記述するシーケンス制御部分設計支援ツール、フィードバック制御部分を比例・微分・積分等の制御要素を組み合わせて記述するフィードバック制御部分設計支援ツール、事前準備した一覧の中から使用する

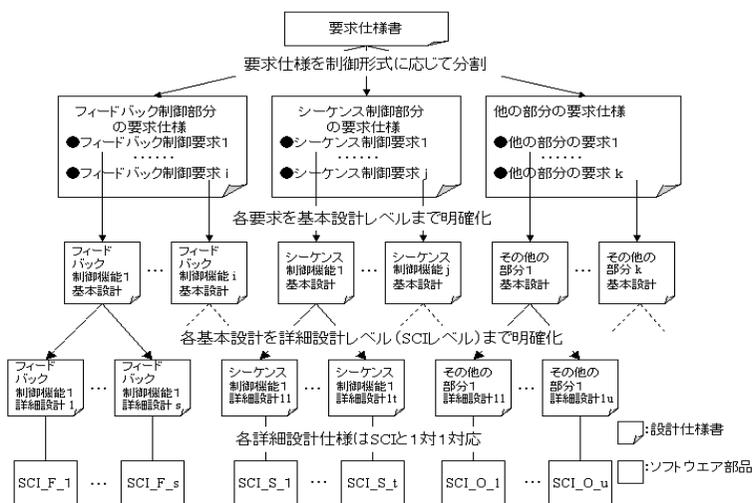


図 5 基本設計と詳細設計の概要
Fig. 5 Outline of basic and detailed design.

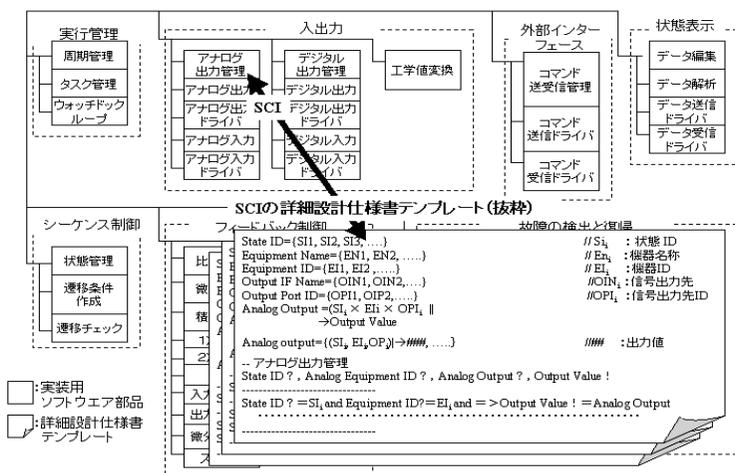


図 6 SCI の概要
Fig. 6 Outline of SCI.

インタフェースを選択するインタフェース部分設計支援ツールから構成される³⁾。なお、状態遷移表、使用する制御要素、使用する入出力インタフェース等は、基本設計結果として、設計情報管理データベースに登録し、CSWの開発と保守に使用する。

(3) 詳細設計段階

詳細設計は、図 5 に示すように基本設計仕様書を事前に準備した詳細設計仕様書レベルに分割・明確化することで行う。図 6 に示すように詳細設計仕様書は、(4) で後述する実装用ソフトウェア部品に対応するものであり、詳細設計仕様書テンプレートに CSW を特化するためのパラメータを代入して作成する。ただし、

CSW に固有な部分については、新規に詳細設計仕様書を作成する。使用した詳細設計仕様書テンプレート、そのカスタマイズに使用するパラメータ、新規に作成した詳細設計仕様書は、詳細設計結果として、設計情報管理データベースに登録し、CSWの開発に使用する。

(4) 実装段階

設計情報管理データベースに登録されている詳細設計結果と図 6 に示すドメイン分析結果をもとに開発した実装用ソフトウェア部品 (Software Components for Implementation: SCI) を用いて、図 7 に示す手順で CSW を実装する。はじめに、CSW 開発に利用する

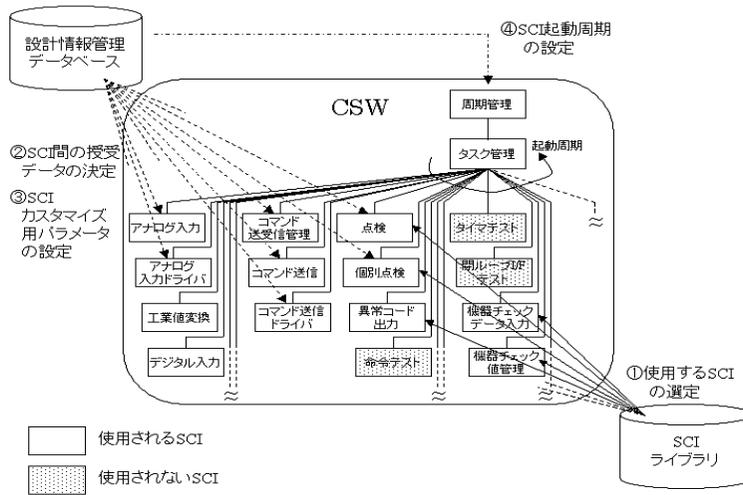


図 7 実装手順の概要

Fig. 7 Outline of implementation process.

SCI を選定し、その起動順序を決定する。2 番目に、SCI 間の授受データを決定する。3 番目に、詳細設計結果を用いて SCI をカスタマイズする。最後に、SCI の起動周期を設定して CSW を実現する。なお、SCI の起動順序・呼応関係の設定は、制御の応答時間や計算精度の制約から一意となるため、事前に定義した全 SCI の起動順序の中から、使用しない SCI を省くだけでよい。SCI の選定、カスタマイズ、および、組合せには SCI 選定/組合せ支援ツールを使用する。なお、個別機能部分については詳細設計結果をもとに新規にコードを開発する。

3.2 設計情報管理データベースによる効率的な制御ソフトウェアの適合性確認

ソフトウェアの適合性確認は、品質向上のために必須であり、効率化のために様々な手法が提案されている。Product Data Management System (PDMS) を用いて設計仕様書単位の適合性確認を実施する手法^{9),10)}があるが、より細かい要求や機能のレベルでの適合性確認を行うことは困難である。さらに、設計仕様書の PDMS への登録や設計仕様書間の関係付けは人手で行わなければならない、効率化が困難であった。以上のように、要求や機能のレベルで効率的な適合性確認を実施することについては、十分な成果が出ていないのが実情であった。

本論文では、3.1 節で述べた手法により、SCR、要求仕様書、基本設計仕様書、詳細設計仕様書、SCI、および、CSW の間で連続的な設計情報伝達が可能となった¹¹⁾。これらの設計情報を図 8 に示す設計情報管理データベースに蓄積し、その伝達経路を追跡する

ことで、要求と CSW の間での適合性確認を効率的に実施する手法を提案する。これは、ドメイン分析結果に基づいてソフトウェア部品と設計仕様書テンプレートを作成したことで、事前に CSW の機能とコードを決定することが可能となったためである。提案手法では、設計情報管理データベースに登録された要求仕様書、基本設計仕様書、詳細設計仕様書、SCI の対応関係を用いて、機能単位でのトレーサビリティ・マトリックスを自動作成する。

図 9 に設計情報管理データベースに登録された情報をもとに作成したトレーサビリティ・マトリックスを示す。これにより、開発者のトレーサビリティ・マトリックス作成の負担が減少し、適合性確認に要する負担が軽減する。なお、設計情報管理データベースには、開発に関わる全技術者が登録されており、個々に設計情報の投入・確認・承認・修正等の実施権限が与えられている。さらに、不正アクセスを防止するために、個々にアクセス権限が与えられており、そのアクセスはパスワードで制限される。また、アクセスの記録も作成される。

3.3 検証支援ツールによる効率的な網羅的テスト仕様書の作成と実施

ソフトウェア検証実施の効率化については、様々な手法が提案されているとともに、市販ツールも存在する。不適合を効率的に検出するための構成テスト仕様書を作成する手法¹²⁾は、効率的なテスト仕様書によりテスト数を削減できるが、個々のテスト時間を短縮することは困難であった。ソフトウェアを 1 行ずつ実行して不適合を探すツール¹³⁾は、デバックの効率向

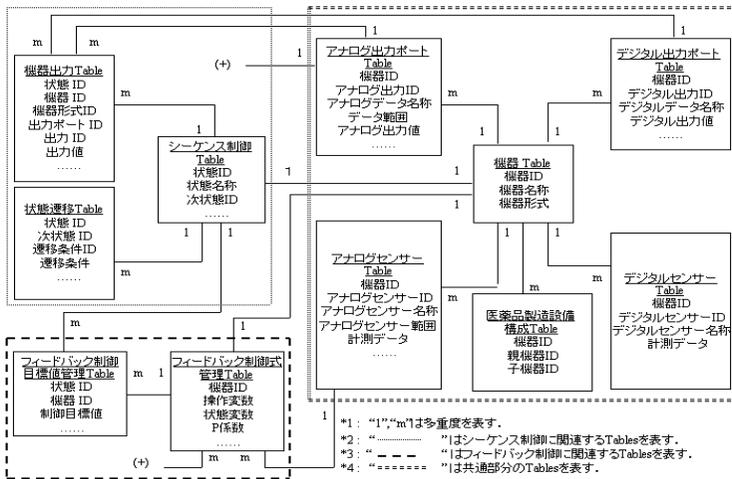


図 8 設計情報管理 DB の概要
Fig. 8 Outline of design information management DB.

要素ID	要素発行仕様	詳細発行仕様	SCIコード	モジュールテスト	IQ	OQ	PQ
同期化検証結果が可能なこと	実行管理	周期管理	Cycle Control	同期化テスト	Ver1.0.0.0
		タスク管理	Task Control	タスク管理テスト	Ver1.0.0.0
		ウォッチドックループ	WD Loop	WDループ管理テスト	Ver1.0.0.0
		アナログ入力管理	Analog Input Control	アナログ入力管理テスト	Ver1.0.0.0
アナログ入力	アナログ入力ドライバ	アナログ入力ドライバ	Analog Input Driver	アナログ入力ドライバテスト	Ver1.0.0.0
				アナログ入力ドライバテスト	Ver1.0.0.0
工学種変換	工学種変換	工学種変換	EJ Conversion	工学種変換テスト	Ver1.0.0.0
				工学種変換テスト	Ver1.0.0.0

図 9 作成したトレーサビリティ・マトリックス—抜粋
Fig. 9 Sample of traceability matrix — Extracted.

上は期待できるが、テスト仕様書の作成やテスト時間の短縮を実現することは困難であった。以上のように、実際の CSW 開発段階で問題となる網羅的なテストの実施とテスト期間の短縮という問題点を同時に解決することについては、十分な成果が出ていないのが実情であった。

本論文では、CSW を SCI ベースで実装する特徴を利用して、SCI 作成時に CV に必要となるモジュールテスト仕様書を事前作成して検証を実施しておくことで、モジュールテスト結果報告書の作成を効率化する。CV に必要となる各種仕様書を事前準備できるのは、SCI を再利用すること、SCI の起動順序・呼応関係が一意となることによる。ただし、OQ と PQ については、実際のプラントの運用手順や運転時間によって実施する必要があるため、これらは SCR や SCI のパラメータ部分となっており、事前実施は難しいため実装後に行う。図 10 に示すように、検証結果を CSW 構成管理データベースに登録して CSW の機能・テスト

仕様書・検証結果との関係を明確にすることで、CSW の CV 作業の効率化を実現する。なお、CSW 構成管理データベースについても、設計情報管理データベースと同様なアクセス管理を行う。

4. CV 指向の制御ソフトウェア開発環境の適用と評価

3章で提案した CVODE を医薬品製造設備用 CSW 開発に適用した。4.1 節に適用結果を、4.2 節に評価結果を述べる。

4.1 CVODE の適用結果

CVODE を 5 種類の医薬品製造設備用 CSW 開発に適用した。従来手法と CVODE を適用した場合における CV 各段階の作業時間の計測結果を表 1 に示す。さらに、従来手法と CVODE を適用した場合の作業時間の比較を図 11 に示す。

医薬品製造設備 A および B は原材料の計量、医薬品製造設備 C は打錠、医薬品製造設備 D および E は医薬品の製造プロセス制御を行う設備である。これらの設備の CSW の全開発段階について従来手法と CVODE を用いた開発手法で要する時間を計測し比較した。なお、従来手法とは設計仕様書・CSW・テスト仕様書を CSW ごとに新規作成する手法とする。

CVODE を適用することで要求定義段階では作業時間を従来手法と比較して平均 36.0[%] に、基本設計段階では作業時間を平均 45.7[%] に、詳細設計段階では作業時間は平均 49.9[%] に、実装段階では平均 34.8[%] に、モジュールテスト段階では平均 21.7[%] に、IQ 段階では平均 15.6[%] に、OQ 段階と PQ 段階では時間

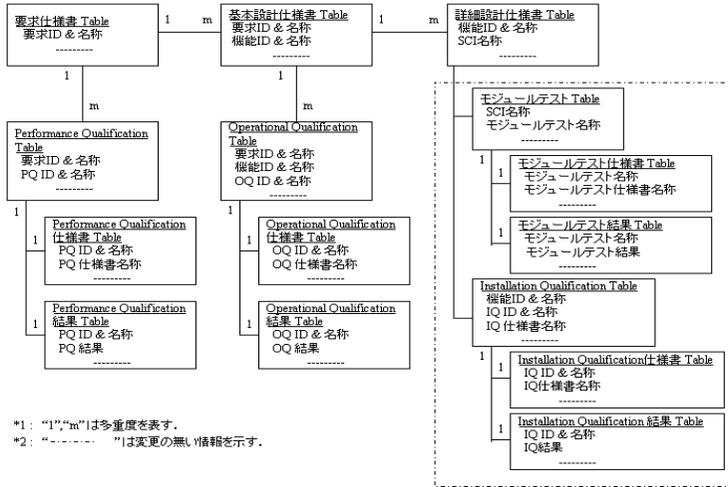


図 10 CSW 構成管理データベースの概要

Fig. 10 Outline of CSW configuration management database.

表 1 CVOICE 適用時の CSW の CV 作業時間結果

Table 1 CSW's CV time result using CVOICE.

設備名称	適用手法	要求定義	基本設計	詳細設計	実装	モジュール テスト	IQ	OQ	PQ	DQ	合計
A	提案	28	23	39	51	37	3	101	122	2	406
	従来	75	51	81	156	155	18	101	122	20	779
B	提案	22	12	18	37	34	3	68	84	2	280
	従来	61	26	38	107	165	18	68	84	17	584
C	提案	29	25	42	56	43	3	70	91	2	359
	従来	80	62	79	144	205	20	70	91	20	771
D	提案	19	11	28	42	43	2	83	102	2	332
	従来	55	24	55	125	201	14	83	102	18	677
E	提案	25	30	43	52	54	3	117	138	2	464
	従来	70	58	88	152	245	20	117	138	21	909

表中の数字の単位は [hours]

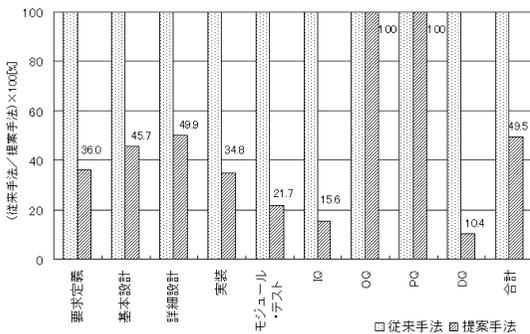


図 11 従来手法と CVOICE を用いた CV 作業時間の比較

Fig. 11 Comparison of working hours between conventional method and CVOICE.

変化なし、そして DQ 段階では 10.4[%] となった。以上より、全 CV 作業時間は 49.5[%] となった。

4.2 CVOICE の評価

CVOICE 適用時の CV 作業時間短縮の要因を検討する。

(1) 要求定義段階

要求定義に CVOICE を適用した場合、その作業時間は 36.0[%] に短縮できた。以下、この要因を分析する。要求定義段階では、「要求の列挙」、「要求の妥当性確認」、「要求仕様書の作成」を実施する。計測の結果、従来手法での作業時間比は、約 3 : 3 : 4 であった。CVOICE の適用により以下の効果が得られた。

- (a) 「要求の列挙」は、既存の機能の中から必要な機能を選択するだけとなるため、50[%] 程度の作業時間となった。
- (b) 「要求の妥当性確認」は、プロトタイプによる挙動確認を行うことで、文書による確認よりも効率が向上し、抜けによる後戻り作業も減少す

るため、50[%]程度の作業時間となった。

- (c) 「要求仕様書の作成」は、要求仕様書テンプレートに要求定義結果を代入するのみであるため、作業時間は、ほぼゼロとなった。

したがって、CVODEの適用により、要求定義段階の作業時間は30[%]程度になると考えられる。しかし、CVODEは新規に要求を定義する部分には適用できず、これまでの実績では適用可能な部分は要求定義全体の約90[%]である。新規部分については従来手法と差は生じない。以上の結果、CVODEの適用により、要求定義段階の作業時間は40[%]程度になると考えられる。この値は、表1の計測結果とよく一致している。

(2) 基本設計段階

基本設計にCVODEを適用した場合、その作業時間は45.7[%]に短縮できた。以下、この要因を分析する。基本設計段階では、「ソフトウェア機能の列挙」、「ソフトウェア機能ごとの入出力の決定」、「基本設計仕様書の作成」を実施する。計測の結果、従来手法での作業時間比は、約3:3:4であった。CVODEの適用により以下の効果が得られた。

- (a) 「ソフトウェア機能の列挙」は、プロトタイプから自動的に抽出でき、作業時間は、ほぼゼロとなる。
- (b) 「ソフトウェア機能ごとの入出力の決定」は、入出力データをソフトウェア機能の入出力に対応させるが、それらはすべて列挙されているため、50[%]程度の時間となった。
- (c) 「基本設計仕様書の作成」は、ソフトウェア機能を記述するが、必要な情報はすべて列挙されているため、50[%]程度の時間となった。

したがって、CVODEを適用した場合、基本設計段階の作業時間は35[%]程度になると考えられる。要求定義段階と同様な理由により、CVODEの適用可能な部分は基本設計の約90[%]である。以上の結果CVODEの適用により、基本設計段階の作業時間は45[%]程度になると考えられる。この値は、表1に示す計測結果とよく一致している。

(3) 詳細設計段階

詳細設計にCVODEを適用した場合、その作業時間は49.9[%]に短縮できた。以下、この要因を分析する。要求定義段階では、「基本設計結果のSCI詳細設計仕様書レベルまでの分割と詳細化」、「SCIごとの入出力の決定」、「詳細設計仕様書の作成」を実施する。計測の結果、従来手法での作業時間比は、約3:3:4であった。CVODEの適用により以下の効果が得られた。

- (a) 「基本設計結果のSCI詳細設計仕様書レベルま

での分割と詳細化」は、基本設計仕様書の分割目標がSCI設計仕様書と明確になっているため、作業時間は50[%]程度になった。

- (b) 「SCIごとの入出力の決定」は、各SCIの入出力データの対応付けを手作業で行うため、従来手法との間で作業時間の差は生じない。
- (c) 「詳細設計仕様書の作成」は、詳細設計仕様書テンプレートに詳細設計結果を代入するのみであるため、作業時間は、ほぼゼロとなる。

したがって、CVODEを適用した場合、詳細設計段階の作業時間は45[%]程度になると考えられる。要求定義段階と同様な理由により、CVODEの適用可能な部分は約90[%]である。以上の結果、CVODEの適用により、詳細設計段階の作業時間は55[%]程度になると考えられる。この値は、表1に示す計測結果とよく一致している。

(4) 実装段階

実装にCVODEを適用した場合、その作業時間は34.8[%]に短縮できた。以下、この要因を分析する。実装段階では、「CSWの構造の決定」、「CSWの開発」を実施する。計測の結果、従来手法での作業時間比は、約1:9であった。CVODEの適用により以下の効果が得られた。

- (a) 「CSWの構造の決定」は、設計情報管理データベースに登録されている設計情報を用いて機械的に決定できるため、作業時間は5時間程度の一定値となる。
- (b) 「CSWの開発」は、提案手法では「SCIを無修正で使用」、「SCIを修正して使用」、「新規にコードを開発」する部分に分類できる。今回、CVODEを適用した事例では、その比率は約8:1:1であった。

- 「SCIを無修正で使用」する部分は、設計情報管理データベースに登録されたパラメータを用いて特化できるため、作業時間は10時間程度の一定値となる。
- 「SCIを修正して使用する部分」は、経験上、修正量は全体の10[%]程度であることが分かっているため、「CSWの開発」の作業時間の10[%]を作業時間と考える。
- 「新規にコードを開発する部分」については、経験上、開発量は全体の10[%]程度であることが分かっているため、「CSWの開発」の作業時間の10[%]を作業時間と考える。

したがって、実装段階の作業時間は「15時間+従来手

法での実装時間の 20[%] 程度になると考えられる。今回の適用事例の場合、従来手法による実装時間は平均約 140 時間であり、15 時間は約 10[%] となるため、CVODE を適用した場合の実装段階の作業時間は 30[%] 程度となる。この値は、表 1 に示す計測結果とよく一致している。

(5) モジュールテスト段階

モジュールテストに CVODE を適用した場合、その作業時間は 21.7[%] に短縮できた。以下、この要因を分析する。

モジュールテストでは、「モジュールテスト計画書の作成」、「モジュールテスト仕様書の作成」、「モジュールテストの実施」、「モジュールテスト結果報告書の作成」を実施する。CVODE を適用した場合、これらの作業は SCI 作成時に実施することになるので、作業時間は、ほぼゼロとなる。しかし、実装段階の「SCI を修正して使用する部分」と「新規にコードを開発する部分」については新規にモジュールテストを実施する必要があるため、CVODE を適用した場合、モジュールテスト段階の作業時間は 20[%] 程度になると考えられる。この値は、表 1 に示す計測結果とよく一致している。

(6) IQ 段階

IQ に CVODE を適用した場合、その作業時間は 15.6[%] に短縮できた。以下、この要因を分析する。IQ では「使用されている CSW の確認」、「IQ 報告書の作成」を実施する。計測の結果、従来手法での作業時間比は、約 1:9 であった。CVODE の適用により以下の効果が得られた。

- (a) 「使用されている CSW の確認」は、使用されている CSW と構成管理データベースに登録されている CSW のバージョンが同一であることを確認すればよいため、作業時間は 1 時間程度の一定値となった。
- (b) 「IQ 報告書の作成」は IQ 報告書テンプレートに設計情報管理データベースに登録されているパラメータを代入するのみであるため、作業時間は 2 時間程度の一定値となる。

したがって、IQ 段階の作業時間は 3 時間程度の一定値になると考えられる。今回の適用事例の場合、従来手法での IQ 時間は平均約 18 時間であり、CVODE を適用した場合の IQ 段階の作業時間は 17[%] 程度となる。この値は、表 1 に示す計測結果とよく一致している。

(7) OQ 段階

OQ 段階では「OQ 計画書の作成」、「OQ 仕様の作成」、

「OQ の実施」、「OQ 結果報告書の作成」を実施する。これらの作業は、実際の医薬品製造設備の運用方案に基づいて実施されるため、事前に仕様書の準備や検証の実施をすることは不可能である。したがって、作業時間に変化は生じない。

(8) PQ 段階

PQ 段階では「PQ 計画書の作成」、「PQ 仕様の作成」、「PQ の実施」、「PQ 結果報告書の作成」を実施する。OQ と同様な理由により、作業時間の変化は生じない。

(9) DQ 段階

DQ に CVODE を適用した結果、その作業時間は 10.4[%] に短縮できた。以下に、この要因を分析する。DQ では、「設計仕様書の調査」、「トレーサビリティ・マトリックスの作成」を実施する。計測の結果、従来手法での作業時間比は約 1:1 であった。CVODE の適用により以下の効果が得られた。

- (a) 「設計仕様書の調査」は、設計情報はすべて設計情報管理データベースに登録されているため、作業時間は、ほぼゼロとなる。
- (b) 「トレーサビリティ・マトリックスの作成」は設計情報管理データベースと CSW 構成管理データベースに登録された設計情報をもとに、機械的に作成可能なため、作業時間は 2 時間程度の一定値となる。

したがって、DQ 段階の作業時間は 2 時間程度の一定値になると考えられる。今回の適用事例の場合、従来手法での DQ 時間は平均約 19 時間であり、CVODE を適用した場合の DQ 段階の作業時間は 11[%] 程度となる。この値は、表 1 に示す計測結果とよく一致している。

以上の結果、CVODE を適用した場合の CV 作業に必要な時間の削減率 ($TRTR$) は式 (1) で予測できる。ただし、 $IRTR$ は CVODE を適用した場合の CV 段階ごとの理論上の時間短縮率、 WT は従来方式を適用した場合の段階ごとの作業時間とし、 Σ は要求定義から DQ までのすべての段階での総和を示す。

$$TRTR = \frac{\Sigma(IRTR \times WT)}{\Sigma WT} \quad (1)$$

式 (1) を用いて「CVODE を適用した場合の CV 作業に必要な時間の削減率」を予測した結果 48.5[%] となった。この結果は表 1 から得られる結果の 49.5[%] とよく一致している。以上の結果、CVODE を CSW の CV に適用することで、従来手法と比較して CV 作業時間を約 50[%] に削減できることが確認できた。

5. おわりに

本論文では、CVOICE を適用した CSW の CV の効率的な実施方法について提案した。CVOICE を適用した結果、①要求定義から PQ までのトレーサビリティの確保、②網羅的な検証の実施、③CV 時間の短縮（従来比 50[%]）を実現することが可能となった。この結果、CSW の要求適合性と信頼性が向上し、適切な品質の医薬品が製造できるようになる。このため、医薬品製造メーカーとしては、効率的な CV の実施が可能となり、管轄機関からの承認を短期間・低コストで得ることが可能となると考えられる。

今後はソフトウェア部品群の充実を図るとともに、医薬品製造設備の高速シミュレータを準備し、OQ と PQ を効率的に実施する手法を検討していく。さらに、ソフトウェア部品と設計仕様書テンプレートの作成に要した作業量とその効果についての評価を行っていく。

謝辞 新エネルギー・産業技術総合開発機構 (NEDO) からの平成 14 年度業務委託「次世代輸送系システム設計基盤技術開発プロジェクト」により効率的なソフトウェア検証手法を研究しました。本論文は、この研究成果の一部をコンピュータ・バリデーションに適用し、有効性を確認した結果について記述したものです。この場を借りてお礼を述べます。さらに、本論文を作成するにあたり、内容をレビューし、適切なコメントをくださった石川島プラントエンジニアリング (株) の清水比奈子氏、金子聡氏にお礼を述べます。

参 考 文 献

- 1) 厚生省：コンピュータ使用医薬品等製造所適性管理ガイドライン，厚生省 (1993)。
- 2) 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，情報処理学会論文誌，Vol.42, No.3, pp.518-528 (2001)。
- 3) Takahashi, M. and Tsuda, K.: An Efficient Integrated Development Environment for Plant Control Software, *Trans. Institute of Electrical Engineers of Japan*, EIS, Vol.123, No.9, pp.1573-1586 (2003)。
- 4) 池田静男 (編)：GMP コンピュータガイドラインの実用的解釈の仕方と製薬企業の対応の実績，技術情報協会 (1997)。
- 5) GAMP forum: *GAMP Guide for Validation of Automated System in Pharmaceutical Manufacturing*, International Society for Pharmaceutical Engineering (1998)。

- 6) 土田泰治，辻潤一郎，田中 朗：定型文書作成支援システム，情報処理学会研究報告ヒューマンインターフェース，No.035-005 (2001)。
- 7) フリーソフトウェア：C 言語ドキュメント生成ツール Harbor C ver1.15。
<http://member.nifty.ne.jp/samidare/index.htm>
- 8) 富士通インフォソフテクノロジ：仕様書工房。
<http://www.ist.fujitsu.com/kobo/>
- 9) 高橋正和，津田和彦：PDM を用いたシステム開発文書の効率的な管理，情報処理学会秋期全国大会 (2000)。
- 10) Edwards, J. and Glibson, P.R.: Integrated multi-media computers in the execution of ISO9000 quality system requirements for document control and training, *Computers and Industrial Engineering*, Vol.32, No.3, pp.529-538 (1997)。
- 11) 高橋正和，津田和彦：シームレスなプラント動作ソフトウェアの開発，計測自動制御学会論文集，Vol.39, No.8, pp.749-758 (2003)。
- 12) 西 康晴，飯塚悦功：ソフトウェアを対象にした構成テストの設計，電子情報通信学会論文誌，Vol.J84-D-I, No.11, pp.1542-1552 (2001)。
- 13) 日本 IBM: Rational Test Real-Time。
<http://www-6.ibm.com/jp/software/rational/products/testrt/>

(平成 16 年 2 月 9 日受付)

(平成 16 年 10 月 4 日採録)



高橋 正和 (正会員)

1988 年立教大学理学部物理学科卒業。同年石川島播磨重工業 (株) 入社。2002 年 (株) ギャラクシーエクスプレス出向。1998 年筑波大学大学院経営システム科学専攻修了。2002 年同大学院博士課程修了。博士 (システムマネジメント)。電気学会，計測自動制御学会等の会員。



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 年筑波大学大学院ビジネス科学研究科助教授。電子情報通信学会，電気学会等の会員。