特集論文

監視制御システムのソフトウェア設計プロセス改善

長崎事業所　技術第１部　情報制御システム課
広瀬　敦規
杉内　寿

1. まえがき

当所では、監視制御システムのソフトウェア設計、ソフトウェア製作及びソフトウェア組合せ試験の工程を担当している。このソフトウェア開発においては、開発初期から参画しソフトウェア全体構成を熟知している担当者（以下、熟練者）が、変更点や変化点を確実に管理し、品質の高いソフトウェアを提供してきた。しかし、近年システム機能の高度化に対応するための増員や、熟練者の高齢化による世代交代の対策が必要となってきた。そこで、監視制御システムのソフトウェア開発経験が浅い担当者（以下、未経験者）でも容易にソフトウェア開発を担当できるようソフトウェア設計プロセスの改善を行ったので紹介する。

2. ソフトウェア設計プロセスの課題

従来のソフトウェア設計プロセスは、熟練者の経験と知識を前提としたプロセスであった。従来のソフトウェア設計プロセスを図1に示す。

熟練者は、監視制御システムを長年担当しており、各種システム仕様書（システム機能仕様書、画面仕様書、通信仕様書など）に点在するソフトウェア要求から、ソフトウェア全体構成とその振る舞い、ソフトウェア詳細仕様、ソースコードに至るまでを、一貫して熟知している。このため、従来のソフトウェア設計プロセスは、各種システム仕様書から直接改造項目ごとにソフトウェア仕様書を作成しており、ソフトウェア構造全体の文書化が不十分であった。また、機能間の関連についても、個別の資料となっていたため、変更箇所や影響範囲の評価は、熟練者の経験則と知識に委ねられていた。

このため、未経験者にとって従来のソフトウェア仕様書は、各種システム仕様書とソフトウェア仕様との関連性や、ソフトウェア全体構成を把握できない資料となっていた。また、未経験者が担当した場合、改造箇所の抽出漏れが発生しやすいプロセスとなっていた。人員増や技術伝承に対処するためには、以下の課題を解決する必要があった。

（1）未経験者でも、各種システム仕様書からソフトウェアの変更箇所を正確かつ漏れなく抽出できるソフトウェア設計プロセスへの改善

（2）未経験者でも、ソフトウェア全体構成が理解しやすい資料の充実

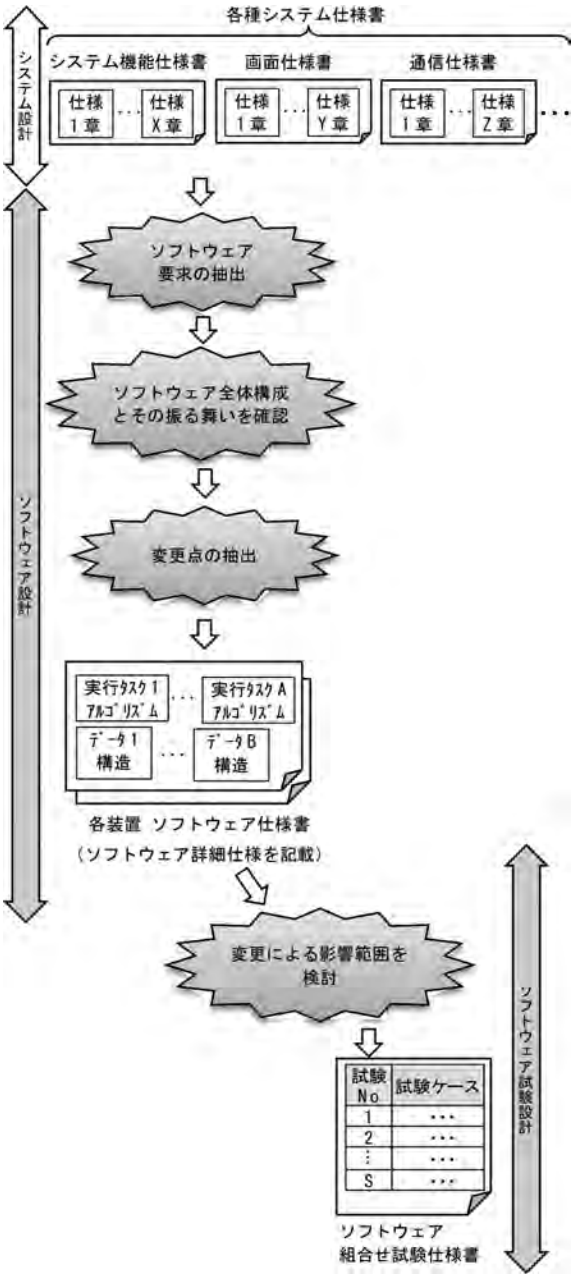

図1. 従来のソフトウェア設計プロセス

3. ソフトウェア設計プロセスの改善策

従来のソフトウェア設計プロセスは、熟練者の経験則と知識に依存していた。そのため、今回は、ソフトウェア設計プロセスをソフトウェア要求分析、ソフトウェアアーキテクチャ設計、ソフトウェア詳細設計の3プロセスで明確化し、ソフトウェア要求仕様書、ソフトウェアアーキテクチャ仕様書、ソフトウェア詳細設計書を各プロセスのアウトプットとした。さらに、トレーサビリティマトリクス表で、各仕様書間のトレーサビリティを確保した。プロセス改善を具体的に進めるため、過去案件を改善対象のモデルとして選定し、各仕様書を作成した。改善後のソフトウェア設計プロセスについて、図2を用いて説明する。

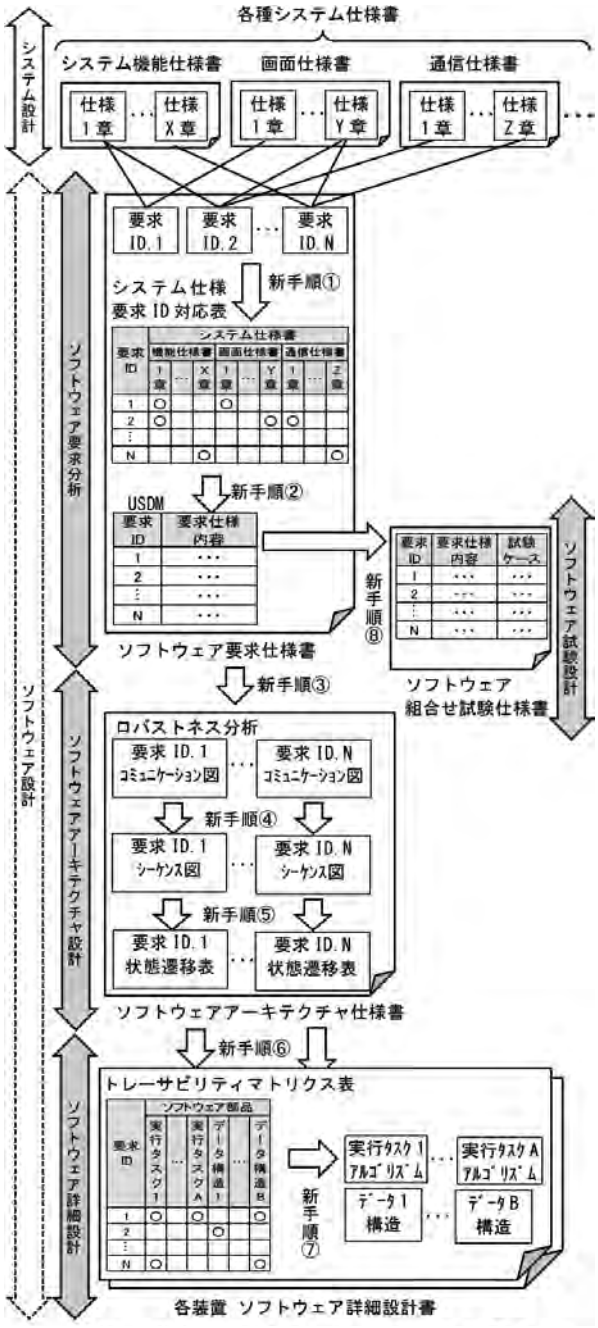

図2. 改善後のソフトウェア設計プロセス

(1) ソフトウェア要求仕様書の作成

各種システム仕様書には、ソフトウェア要求が点在している。未経験者でもソフトウェア要求を漏れずに抽出し整理できるようにするため、USDM(Universal Specification Describing Manner)(注1)をソフトウェア要求仕様書に適用した。

(a) ソフトウェア要求の仕分け(図2の新手順①)

トレーサビリティの起点とするため、各ソフトウェア要求にID(以下、要求ID)をつけ、システム仕様書の章と要求IDとを対応させる表を作成した。

(b) USDMを適用したソフトウェア要求仕様書の作成(図2の新手順②)

要求IDごとに、USDMを適用し仕様書化した。

なお、USDMの詳細な適用方法については4.1節で説明する。

(2) ソフトウェアアーキテクチャ仕様書の作成

ソフトウェア全体構成を視覚的に理解しやすくするため、また、仕様変更に対する影響範囲の妥当性を確認しやすくするため、ロバストネス分析によるコミュニケーション図及びシーケンス図をソフトウェアアーキテクチャ設計書に適用した。また、シーケンス図を補足するため状態遷移表を適用した。

(a) ロバストネス分析(注2)(図2の新手順③)

(i) 要求IDごとにロバストネス分析を行い、ソフトウェア部品化の基となる要素を定義した。

(ii) 各要素間の静的な依存関係をコミュニケーション図に記載した。

(b) シーケンス図作成(図2の新手順④)

ロバストネス分析で定義した要素間の動的な関係性が分かるよう、要求IDごとにシーケンス図へ展開した。

(c) 状態遷移表作成(図2の新手順⑤)

作成したシーケンス図について、異常時のシーケンスなど複数のシーケンスパターンが存在する場合は、その要求IDの状態遷移表を作成した。

なお、ロバストネス分析によるコミュニケーション図の詳細な作成方法については4.2節で説明する。

(注1) 要求や仕様を構造化し記載する手法。

(注2) 要求仕様より、システムのバウンダリ、コントール及びエンティティとなる要素を抽出し、要素間の関係性を検証する手法。

(3) ソフトウェア詳細設計書の作成

実装漏れの防止と影響評価のため、トレーサビリティマトリクス表をソフトウェア詳細設計書に適用した。

(a) トレーサビリティマトリクス表作成(図2の新手順⑥)

ロバストネス分析で定義した要素を、実行タスク、共通ライブラリまたはデータ構造の何れかのソフトウェア部品に対応させた。そして、ソフトウェア部品と要求IDとの依存関係を示すトレーサビリティマトリクス表を作成した。

(b) アルゴリズム設計(図2の新手順⑦)

トレーサビリティマトリクス表にてソフトウェア部品ごとに対応する要求IDを確認し、アルゴリズム設計やデータ構造の定義などを行った。その内容をソフトウェア詳細設計書に記載した。

(4) ソフトウェア組合せ試験仕様書の作成

(図2の新手順⑧)

試験ケースの抽出漏れを防ぐため、USDMをベースとし、要求IDごとに試験ケースを記述するようソフトウェア組合せ試験仕様書を変更した。

なお、試験仕様書の詳細については4.3節で説明する。

4. 適用手法の詳細説明

3章で適用した手法のうち、特に品質及び生産性向上を考慮し工夫した点について説明する。

4.1 USDMを適用したソフトウェア要求仕様書

ソフトウェア要求分析では、システム仕様書からソフトウェア要求を正確かつ漏れなく抽出し、簡潔に表現する必要がある。そのためには、ソフトウェア要求を分解し構造化することが望ましい。今回、USDMの適用に際し、USDMで定義する項目と記載内容は、表1のとおりとなる。

表1. USDMの記載内容

No	USDM項目	記載内容
1	要求	システムの利用者がシステムに求める要望を記載する。
2	理由	“要求”が発生した理由を記載する。
3	説明	“要求”に対する補足または制約等あれば記載する。
4	仕様	“要求”から発生した仕様を記載する。

また、監視制御システムに適用したUSDMの記載例を図3に示す。要求ID(REQ01)を例に仕様構造化を示す。

要求	REQ01	機器よりセンサ信号を入力し、入力データを変換し、監視画面に機器状態を表示したい。	
	理由	機器が設置された現場へ人が確認しに行かずとも、集中管理室で一括監視したい。	
	説明	客先より、監視エリアの地図上に機器の状態を表示するよう求められている。	
	仕様	REQ01-01	センサ入力:監視機器のセンサ信号をシステムに入力する。
		REQ01-02	入力変換:入力値を監視データへ変換する。
		REQ01-03	監視画面:監視画面の機器シンボル位置へ監視データを表示する。

図3.USDMを適用したソフトウェア要求仕様書の記載例

“要求”は、動詞を初めから終わりまで連鎖的に記載する。図3の要求REQ01を例にすると、「監視機器よりセンサ信号を入力し(動詞:仕様REQ01-01)、入力データを変換し(動詞:仕様REQ01-02)、監視画面に状態を表示したい(動詞:仕様REQ01-03)。」のように記載する。このように記載することで、要求から仕様へ分解しやすくなる。

“理由”及び“説明”は、“要求”の意味を理解しやすくするため、また“仕様”の根拠や妥当性を検証しやすくするために記載する。

“仕様”は、後のソフトウェア詳細設計書とソフトウェア組合せ試験仕様書とのトレーサビリティを確保しやすくするため、装置ごとやソフトウェア部品ごとに分割して記載する。

4.2　ロバストネス分析を適用したソフトウェアアーキテクチャ仕様書

今回、ソフトウェアアーキテクチャ設計において、ソフトウェア要求からソフトウェアを部品化し、ソフトウェアの全体構成を設計する手法として、ロバストネス分析を採用した。その手順を説明する。

(1) 要素の抽出

4.1節のUSDMにて定義した要求IDごとに、表2に示す要素を抽出する。この際、異なる要求ID間で役割が同じ要素については、共通化を行う。

表2. ロバストネス分析対応表

No.	要素	主な内容
1	バウンダリ	・ハードウェアインタフェース ・ユーザインタフェース （画面、ボタン、ランプ、ベルなど）
2	コントロール	・装置内部の情報処理機能 （入力変換、制御管理など）
3	エンティティ	・通信フォーマット ・共有データの構造体 （監視データなど）
4	アクター	・監視制御対象のハードウェア ・システムに関わる人 （管理者、利用者など）

(2) コミュニケーション図を作成

要素間の関係性を線で結びUML（Unified Modeling Language）の一つであるコミュニケーション図を作成する。コミュニケーション図は、要求IDごとに作成する。

作成例を、図4に示す。機器及び操作員をアクター、センサ入力及び監視画面をバウンダリ、センサ入力を監視データに変換する部分をコントロール、共有データである監視データをエンティティとして各要素に分解する。分解した各要素を要求に応じて配置し線で結ぶとコミュニケーション図となる。

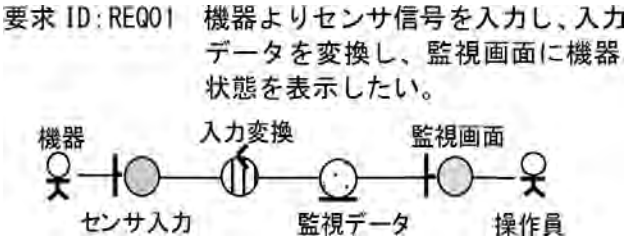

図4. コミュニケーション図の例

コミュニケーション図を作成することで、要求IDごとの要素を明確にすることができる。

4.3　ソフトウェア要求仕様書を活用したソフトウェア組合せ試験仕様書

ソフトウェア組合せ試験仕様書の作成において、図3のUSDMを適用したソフトウェア要求仕様書を活用した。

作成例を、図5に示す。USDMを適用したソフトウェア要求仕様書の“仕様”に対して右側に試験ケースと試験結果の記入欄を設けた。

図5. USDMを適用した試験仕様書の記載例

仕様に対して必ず試験ケースを割り当てられるため、ソフトウェア要求に対する試験ケースの抽出漏れがなくなる。

5. プロセス改善による効果

今回の改善により以下の効果を得られた。

(1) 各仕様書と要求IDを関連付けることにより、システム仕様からソフトウェア部品及び試験ケースまでのトレーサビリティを確保できた。

(2) USDMを適用したソフトウェア要求仕様書により、ソフトウェア要求の抽出漏れが発生しにくいプロセスになった。

(3) ロバストネス分析によるコミュニケーション図、シーケンス図及び状態遷移図を適用したソフトウェアアーキテクチャ設計書により、ソフトウェア全体構成が視覚的に理解しやすくなった。また、仕様変更に対する影響範囲の妥当性も確認しやすくなった。

(4) トレーサビリティマトリクス表を適用したソフトウェア詳細設計書により、実装漏れが発生しにくいプロセスになった。また、変更点に対するソフトウェア改造箇所の特定がしやすくなった。

(5) ソフトウェア組合せ試験仕様書とソフトウェア要求仕様書を連携させることで、試験ケースの抽出漏れが発生しないプロセスとなった。

以上より、未経験者でも容易にソフトウェア開発を担当できるプロセスへ改善できた。

6. 今後の展開

今回は、ソフトウェア設計プロセスを改善し、過去案件を選定の上、各仕様書を作成し検証を行った。今後、案件に適用し効果を確認し、更なる改善を図っていく予定である。なお、新規工事の受注が予定されているため、今回の改善内容を適用する。改造工事は新規工事で作成したものを流用し、変更点に対する修正のみを行うことで、プロセス改善による業務負荷増は抑えられる見込みである。

以下に今後の展開について記載する。

(1) 案件適用による効果の検証

今回改善したソフトウェア設計プロセスを案件に適用し、未経験者担当時の品質や生産性に与える効果について検証していく。

(2) トレーサビリティ対象範囲の拡大

プログラム製作工程へトレーサビリティ対象範囲を拡大する。ソフトウェア要求への追加変更に対する影響範囲を漏れなく迅速に特定可能とすることで、製作時間や実装漏れの削減を図っていく。

(3) トレーサビリティマトリクス表作成の効率化

改善後のソフトウェア設計プロセスでは多くの資料を作成するため、案件適用時は効率化を図る必要がある。特に、トレーサビリティマトリクス表は巨大な表となるため、作成に多大な時間を要する。そこで、トレーサビリティを容易に把握できるUML設計開発支援ツールとしてEnterprise Architect(注3)を導入し、その効果を検証していく。

過去の改造案件については、上記展開を踏まえて改善を行っていく予定である。

7. むすび

今後、監視制御システムは、より高度なソフトウェア要求が発生し、一層仕様が複雑化することが予想される。これに対応するため、新しいツールや設計手法などを積極的に取り入れ、改善活動を継続する所存である。

最後に、ソフトウェア設計プロセス改善にあたりご協力いただいた関係者の方々に深く感謝申し上げる。

(注３) UMLなどをモデリングするための設計開発支援ツール。

執筆者紹介

広瀬　敦規　　ヒロセ　アツノリ
2008年入社。監視制御システムのソフトウェア開発に従事。現在、長崎事業所技術第1部情報制御システム課。

杉内　寿　　スギウチ　ヒサシ
2011年入社。監視制御システムのソフトウェア開発に従事。現在、長崎事業所技術第1部情報制御システム課。