

ソフト仕様記述票を用いたソフト仕様書の記述手順の提案

Proposal on the development for a software specification using Software Specification Description Card

小林 展英^{1,2}, 宇都宮 浩之², 岡戸 真一郎², 岡本 惇一朗²

Nobuhide Kobayashi^{1,2}, Hiroyuki Utsunomiya², Shinichiro Okado², Junichiro Okamoto²

名古屋大学¹, 株式会社デンソークリエイト²

Nagoya University¹, Denso Create Inc.²

概要

要求仕様書からソフト仕様書を記述する工程は経験豊富な技術者が担当する 경우가多いが、その時に用いる記述知識は必ずしも明確にはなっていない。本稿では、ソフト仕様書の記述に関する知識をカード形式で表現したソフト仕様記述票とそれを活用する手順について提案する。また、実際のソフト仕様書の記述に適用した結果に基づいて、本稿で提案するカードの有効性について考察する。

Abstract

Expert software engineers are responsible for developing software specifications. However, their knowledge of describing specifications is not always clear. In this paper, a software specification is described using the Software Specification Description Card to visualize the knowledge. Additionally, the validity and the usage of the proposed card are discussed by applying to describing an actual software specification.

1. はじめに

顧客から提供された要求仕様書に基づいて作成するソフト仕様書に欠陥が混入した場合、要求仕様書に対する妥当性確認を実施する工程までの間にその欠陥を検出することは困難となる。開発工程毎に妥当性確認を実施する手法も提案されているが、開発現場の実情を考えると導入は容易でない。

ソフト仕様書に対する欠陥の混入は、要求仕様書そのものに欠陥が含まれた場合、あるいは要求仕様書に基づいてソフト仕様書を作成する過程で欠陥が混入した場合のいずれかに大別できる。これらの原因は、ソフト仕様書として記述すべき要素とその関係が明確に定義されていないことと、そのためにソフト仕様書を作成する手順や要求仕様書の妥当性を評価する方法が設計者の力量依存になっていることにある。

このため、本稿では、ソフト仕様書に記述すべき必要十分な要素とそれらの関係を明確化することにより、要求仕様書からソフト仕様書を作成する手順の可視化と要求仕様書の妥当性評価を容易にするソフト仕様書の作成方法を提案する。また、本提案を実際のソフト開発に適用した初期評価についても報告する。

2. 関連研究

2.1. EARS

EARS[2]では、要求仕様書中の記述を「あるイベントが発生した時に対応するレスポンスを応答する」という記述構造で表現できると提案している。また、イベントの種別は「偏在型」、「契機型」、「不測型」、「状態型」、「選択型」の5つの型で表現できるとし、自然言語で自由記述された文書の冗長性などを回避するた

めの制約言語として、〈契機〉と〈応答〉で構成されたそれぞれの型に対応した構文を定義している。また、その構文の有効性は名古屋大学などで評価されている[3][4][5]。しかしながら、要求仕様書とソフト仕様書との関連については述べていない。

2.2. AUTOSAR Software Component Template

AUTOSAR Software Component Template[7] (以降、SWCT と呼ぶ) では、車載ソフトを構成するアプリケーションの型として、AUTOSAR で定められたプラットフォーム上で稼働するために必要な設計要素とその関係が定められている。図 1 に SWCT に準拠したソフト構造の概要図を示す。また、SWCT に準拠した設計ツールを用いて車載ソフトを設計することで、アプリケーションの実装やプラットフォームのコンフィグレーション、およびアプリケーションとプラットフォームの統合まで自動化することが可能である。このため、SWCT には車載向けのソフト開発に必要な設計要素がすべて定義されていると言うことができる。しかしながら、要求仕様書からソフト仕様書に記述する設計要素を導出する方法については述べていない。

なお、SWCT は様々な車載ソフトへの対応が考慮されてきた結果、設計要素の数も多く、その関係も複雑になっている。本稿では、SWCT から基本的な設計要素として抽出した図 2 に示す範囲をソフト仕様書の記述内容とし、それ以外の設計要素はソフト仕様書の後工程で作成されるソフト設計書の記述内容に位置付ける。それぞれの要素の記述内容は表 1 を参照されたい (カッコ内の名称は SWCT 中の要素名)。

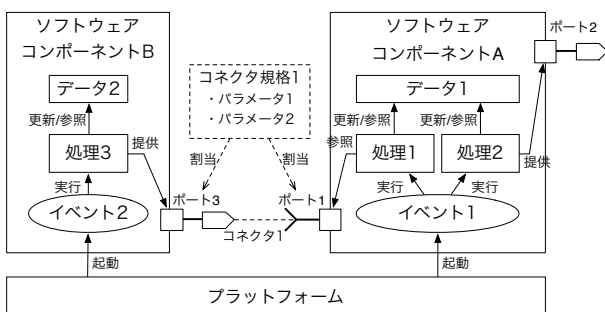


図 1 SWCT に準拠したソフト構造

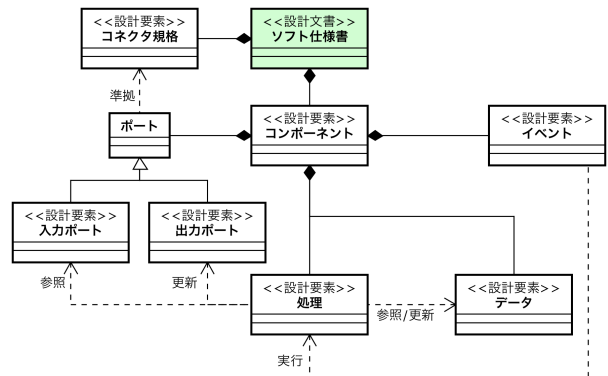


図 2 ソフト仕様書の構成要素の関係図

表 1 設計要素一覧

設計要素名		記述内容
コンポーネント (Software Component)		車載ソフトを構成するアプリケーションの定義を記述
ポート	入力ポート (RPort)	処理の実行に必要な情報の取得方法に関して記述
	出力ポート (PPort)	他アプリケーションが利用する情報の提供方法に関して記述
処理 (RunnableEntity)		入力ポートとデータから得た情報を使って出力ポートから提供する情報の演算規則を記述
データ (InnerRunnable Variable)		処理の実行完了後も保持しておくべき情報に関して記述
イベント (RTEEvent)		コンポーネントを実行する契機について記述
コネクタ規格 (PortInterface)		ポート間を流れる情報に関して記述 (同一のコネクタ規格が割付いたポート間のみ接続可)

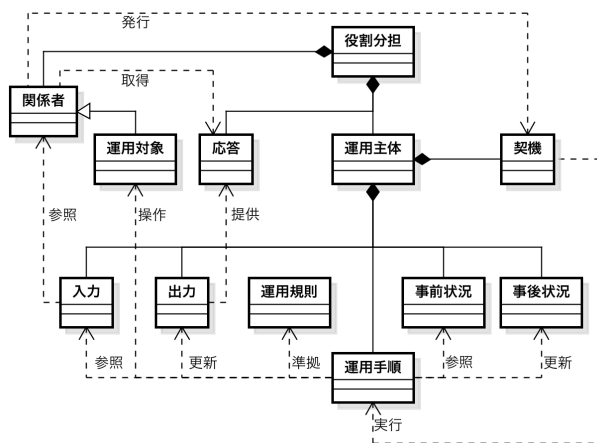
2.3. 運用活動票

運用活動票[1]では、システムを稼働させるために運用主体が実行すべき活動を明文化するために必要な要素として、要求 ID、運用主体、事前状況、運用対象、事後状況、契機、応答、運用手順、入力、出力、運用規則、関係者、役割分担の 13 要素を定めている。また、運用活動票を用いて運用知識を抽出する手順についても提案されている。しかしながら、車載ソフト開発への適用性の評価や実際に車載ソフト開発へ適用した試験結果は報告されていない。

3. ソフト仕様記述票

本章では、車載向けのソフト仕様書を記述するために、運用活動票に基づいて考案したソフト仕様記述票について説明する。

2.2 節に記した SWCT の設計要素をソフト仕様書に記述すべき必要十分な設計要素と仮定した場合、運用活動票がそれらの設計要素を網羅できていれば、ソフト仕様書の作成に対する適応性は高いと判断できる。運用活動票は、図 3 に示す構成要素とその関係を持っている。図中の<運用主体>を図 2 に示す SWCT のコンポーネントに位置付けると、それぞれの構成要素は表 2 のように対応付けられる。また、図を比較してみると、構成要素間の関係も等価表現となっていることが分かる。表中で対応付けられていない<関係者>と<運用対象>については、それぞれ記述対象のコンポーネントが処理を実行するために参照する情報を提供するコンポーネント、その情報に基づいて出力を生成する過程で操作するコンポーネントに対応付けることができる。



順 6 までを実施した後で、ソフト仕様記述票間の関係を包含、接続、無関係のいずれかに分類する。

【手順 8】接続関係にあるソフト仕様記述票の<入力>と<出力>が整合するよう追加、修正する。

【手順 9】包含関係にある上位と下位の運用活動票が、「上位=Σ下位」の関係になるよう修正する。

【手順 10】手順 9 まで実施した後で、仮記述が残っている箇所を要求設計者との質疑応答で解消する。

表 3 熟練者の設計成果との比較

評価項目	熟練者の設計成果	本稿の適用成果
検出した不明点の総数	18 個	20 個
・契機関連	4 個	4 個
・処理の対象範囲	2 個	1 個
・規則関連	2 個	4 個
・入出力関連	10 個	11 個

5. 適用評価

5.1. 適用対象

ソフト仕様記述票の有効性を評価するため、組込みシステムの共通問題[6]を要求仕様書に位置付けた実験を行う。組込みシステムは、設定されたタイミングに基づいて、センサから情報を読み出し、その情報を定められたルールに従って演算し、その結果をアクチュエータに出力する、という基本フローを有している。つまり、ソフト仕様記述票を適用して、この基本フローに欠陥がない組込みシステムを設計できれば、本記述票は有効であると判断できる。この前提に基づき、以下の評価項目で熟練者の設計成果と本稿の適用成果の比較実験を行う。

- (1) 基本フローの構成要素とその関係の合致性
- (2) 検出した要求仕様書に対する不明点

なお、(1)に関しては、熟練者の設計成果[10]と本稿の適用成果をともにシステミグラム形式に変換して基本フローを比較する。

5.2. 適用結果

5.1 節で定めた評価項目(1)については、要求仕様書に対する不明点に関する構成要素を除いて等価の結果を得ることができた。評価項目(2)については、単一の記述票で確認できる範囲に関して熟練者を上回る結果となった(規則関連、入出力関連)。この結果はカード形式の記述票を用いたことで、要求分析時に確認すべき点を常に意識して分析できた効果と予想される。一方、複数の記述票に記載された内容の不揃いさから気づく点については、熟練者の成果に及んでいない(処理の対象範囲)。この問題は、記述手順に複数の記述票を横並び比較する手順を加えることで解消できる。

5.3. 留意点

適用評価の過程で、以下の点についてソフト仕様記述票を記述する上での分かりにくさが判明した。

(1) 包含関係にあるソフト仕様記述票の記述方法

ソフト仕様記述票の間に包含関係が存在する場合、上位のソフト仕様記述票は下位のソフト仕様記述票の総括である必要がある。しかしながら、上位のソフト仕様記述票のみに記述されている内容が存在していた場合、その記述と下位のソフト仕様記述票を総括した内容が混在した状態になっていた。後工程で扱う必要のある記述箇所の特定が困難になるため、上位のソフト仕様記述票のみに存在する記述は、総括した記述を残して新たに起票した下位のソフト仕様記述票に移動することを指摘した。ただし、サブルーチンコールのように上位のソフト仕様記述票の一部を下位のソフト仕様記述票で記述する場合も想定されるため、総括と実体を容易に区別できる記述方法の検討が必要である。

(2) 明文化されていない契機の記述方法

EARS の偏在型に相当するようなく契機>が明文化されていない要求記述から<契機>を推察して記述する際、空欄、あるいは表現が不十分になる傾向があった。EARS の型と SWCT の<イベント>の種別を比較してみると、それぞれの特性から表 4 のように対応付けることができる。この対応関係を参考にして<契機>を推察するよう指示したところ、明文化されていない<契機>についても適切に推察できるようになった。なお、本稿が提案するソフト仕様記述票では、上記について運用活動票を拡張している(表 5 参照)。

表 4 EARS 型と SWCT イベントの対応関係

EARS 型	SWCT イベント
偏在型	<ul style="list-style-type: none"> ・定期時間経過 ・空き時間発生
契機型	<ul style="list-style-type: none"> ・他コンポーネントからのデータ受信 ・他コンポーネントからのトリガー受信 ・他コンポーネントからの機能呼出受信 ・データ送信完了 ・非同期型機能呼出の応答受信 ・初期化要求受信
不測型	<ul style="list-style-type: none"> ・データ受信異常発生 ・ハードにおけるデータ変換異常発生
状態型	<ul style="list-style-type: none"> ・状態遷移発生 ・状態遷移完了
選択型	(対応するイベントなし)

6. 議論

多くの車載ソフトの開発現場では、自由形式で記述された要求仕様書から AUTOSAR に準拠したソフト仕様書を記述する方法が熟練者の暗黙知になっている場合が多い。レビューを利用して暗黙知の伝承に取り組んでいる開発現場もあるが、熟練者の設計方法そのものが体系化できていないことも多く、ソフト仕様書の作成手順を可視化できていない問題があった。

本稿では、SWCT の構造と整合したソフト仕様記述票を用いることで、AUTOSAR に準拠したソフト仕様書に記すべき設計要素を明らかにした。さらに、その記述手順も定めることで、熟練者が暗黙知として有する設計知識を可視化することができた。

ただし、ソフト仕様記述票は、5.2 節の適用結果から分かる通り、要求仕様書に記述された<契機>の単位で内容が記述されているため、それらを統合したソフト全体の動きが整合しているか確認することは難しい。そのような場合には、処理主体、および入出力に記されたデータをノード、入出力の仕方をノードの関係に対応付けたシステミグラム[8][9]を作成することで容易に確認することができる (図 4 参照)。

また、本稿の後工程であるソフト構造の設計やテスト設計においてソフト仕様記述票をどのように活用する

かも重要なテーマである。前者はシステミグラムを用いた記述方法を考案しており、別途報告する予定である。後者は参考文献[11]で研究が進められている。さらに、本稿ではソフトの振る舞いに関する機能要求のみを対象としており、ソフトの構造や満足すべき特性に影響を与える非機能要求について考慮していない。それらを適切な表現で整理し、ソフト設計に反映する方法の考案も重要である。

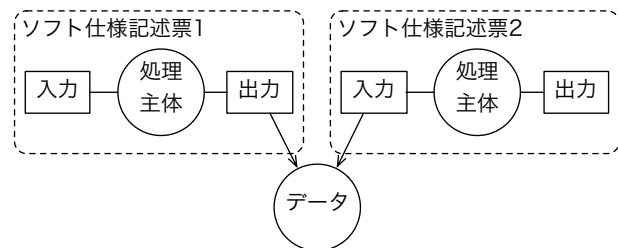


図 4 システミグラム変換イメージ

7. まとめと今後の課題

本稿では、ソフト仕様記述票を車載ソフト開発に適用した一例として、ソフト仕様書に記述すべき要素を網羅した SWCT とソフト仕様記述票の関係を明らかにし、車載ソフト開発への適応性が高いことを確認した。さらに、ソフト仕様記述票を用いたソフト仕様書の作成手順を提案し、その適用評価結果から熟練者が有する基本的な設計視点を網羅したソフト仕様書が作成できることを確認した。

なお、本提案が実際の開発現場で運用されるためには、現実的な工数で実践できることが重要である。今後の課題として、本提案を運用可能なツール環境を整備し、実際のソフト開発に適用した場合の生産性について評価を進めていく予定である。

参考文献

- [1] 山本修一郎, システム運用知識抽出法の提案, 人工知能学会 第七回知識流通ネットワーク研究会
- [2] Marvin, Alistair, et al, "Easy approach to requirements syntax (EARS).", Requirements Engineering Conference, 2009. RE'09. 17th IEEE International. IEEE, 2009.
- [3] Marvin, Alistair, and Wilkinson, Philip, "BIG

- EARS (The Return of “Easy Approach to Requirement Syntax”), 2010 IEEE 18th International Conference on Requirements Engineering (RE)
- [4] 要求テンプレートを用いた組込み共通問題の記述法についての研究, ディペンダビリティ手法適用のフロンティア 2015, 名古屋大学 山本研究室, 2015
- [5] 山本修一郎, 要求工学基礎知識 第 40 章 要求テンプレート, pp291-295, 名古屋大学, 2012.02
- [6] ソフトウェア工学の共通問題: 4. 組込みソフトウェア分野の共通問題の考え方, 情報処理 Vol54, No.9, pp.890-893, 2013, 平山雅之, 中本幸一
- [7] Software Component Template, AUTOSAR Release 4.2.1, http://www.autosar.org/fileadmin/files/releases/4-2/methodology-and-templates/templates/standard/AUTOSAR_TPS_SoftwareComponentTemplate.pdf
- [8] アーキテクチャ論(18)システミグラム, Computer report 52(10), 19-29, 2012 日本経営科学研究所
- [9] Boardman, J and B Sauser. 2008. Systems Thinking: Coping with 21st Century Problems. Boca Raton, FL: Taylor & Francis / CRC Press
- [10] システミグラムによる組込みシステム分野の共通問題の分析事例, 人工知能学会 第 15 回知識流通ネットワーク研究会, 小林展英, 岡戸真一郎
- [11] 大林英晶, 森崎修司, 山本修一郎, 要求記述表を用いたテスト項目抽出手法の適用評価について, 信学技報, vol. 114, no. 292, KBSE2014-36, pp.43-46, 2014 年 11 月

表 5 ソフト仕様記述票の記述例

仕様 ID	SWS_nn : 走行速度の取得					
処理主体	契機	処理手順		応答	処理対象	
走行状況 取得機能	<input checked="" type="checkbox"/> 偏在型 <input type="checkbox"/> 契機型 <input type="checkbox"/> 不測型 <input type="checkbox"/> 状態型 <input type="checkbox"/> 選択型	0.1sec 周期	<ul style="list-style-type: none"> 車軸回転数センサからパルス数を取得する パルス数から車輪回転数を計算する 車輪回転数と車輪直径から走行速度を計算する 		走行速度を出力する	車軸回転数センサ
事前状況		入力	出力		事後状況	
正常なパルス数が取得可能		・パルス数	・走行速度		走行速度が更新されている	
	規則	関連主体		役割分担		
	・ 走行速度計算式	プラットフォーム		0.1sec 周期イベントの発行		
		走行状況取得機能		走行速度の提供		
		危険運転警告機能		走行速度の取得		