

ステークホルダ関心事に基づく ソフトウェア要求仕様書構成モデルの設計方法

蛸島 昭之^{†1†2} 青山 幹雄^{†1}

概要：ソフトウェア開発に関わるステークホルダ間で要求を適切に定義し共有するためにはソフトウェア要求仕様書 (SRS: Software Requirements Specification) の構成モデルが必要となる。既存の SRS 構成モデルには ISO 29148 が提案する SRS テンプレートなどがあるが、それらは汎用的なため、ドメイン固有の要求を表現するには不十分である。本稿では、既存の SRS テンプレートへドメイン固有の要求を表現するための要求項目を追加することで、ドメインに特化した SRS の派生モデルを設計する体系的な方法を提案する。提案方法では、ドメインの特性に関するステークホルダの関心事から参照モデルに追加すべきドメイン固有の要求項目を導出する。提案方法を自動車ソフトウェアのドメインに適用し、その有効性を評価する。

キーワード：ソフトウェア要求仕様書，構成モデル，SRS テンプレート，インスペクション

A Design Method for Construction Model of Software Requirements Specification Based on Stakeholders' Concerns

AKIYUKI TAKOSHIMA^{†1} MIKIO AOYAMA^{†1}

1. はじめに

ソフトウェア技術の発達により、ソフトウェアによってのみ実現できる付加価値（製品やサービス）が増加している。それに伴い、ソフトウェアに対する社会からの要求が高度化かつ多様化している。第一著者の関係する自動車産業分野では、自動運転機能の実現へ向けた製品開発が要求の高度化を後押ししている。さらに、コネクテッドカーと呼ばれるインターネットへの接続機能を備えた自動車の普及に伴い、これまで自動車分野では存在しなかったプライバシー要求やセキュリティ要求といった新たな種類の要求が発生し、要求が多様化している。このように高度化するソフトウェアの開発を所定の予算と期間を満たしつつ完了するには要求工学 (RE: Requirements Engineering) の活用が不可欠である[12]。なかでも多様な要求を表現できるソフトウェア要求仕様書 (SRS: Software Requirements Specification) の品質がソフトウェアの品質、開発コスト、開発期間 (QCD) を満たすうえで重要であることが知られている[21]。ソフトウェア開発に関わるステークホルダ間で要求を適切に定義し共有するためには SRS の構成モデルが必要となる。既存の SRS 構成モデルには ISO 29148 が提案する SRS テンプレートなどがあるが、それらは汎用的なため、ドメイン固有の要求を表現するには不十分である。

2. 研究課題

SRS の品質を確保するには、SRS に記述されるべき内容の構造、すなわち目次構成、が SRS の作成に先立ち事前に定義されている必要がある。多くの文献によって SRS 構成モデルが提案されているが、それらは汎用的で特定のドメインでそのまま利用するには不適切であるか、特定のドメイン用途に特化しているため他のドメインへは適用できないなどの問題がある。そのため、あるドメインに特有の要求を表現できる SRS 構成モデルを設計することが課題となっている。また、自動車ソフトウェアのように多くのステークホルダに関わるドメインでは、SRS 構成モデルは多様な要求を表現できなければならない。この多様な要求を表現するための要求項目のメタモデルを定義し、体系的にドメイン固有 SRS 構成モデルを設計する方法を提案する。

以上の理由から、本稿では下記の3点を研究課題とする。

(1) 要求項目メタモデルの提案

開発対象ソフトウェアのステークホルダがドメインに対して持つ関心事に基づき要求項目のメタモデルを定義し、多様な要求と SRS 構造の対応関係を明らかにする。

(2) ステークホルダの関心事に基づくドメイン固有 SRS 構成モデル設計方法の提案

要求項目メタモデルの構造に基づき、体系的に SRS 構成モデルを設計する方法を提案する。

(3) 自動車ドメイン固有 SRS 構成モデルの提案

提案方法に基づき、自動車ソフトウェアの多様なステークホルダの要求を表現できる SRS 構成モデルを提案する。

^{†1} 南山大学大学院
Nanzan University

^{†2} 株式会社デンソー
DENSO CORPORATION.

3. 関連研究

ソフトウェア要求仕様書 (SRS) の構成モデルとしてもっとも多く参照されているものの 1 つに国際規格の IEEE 830-1998 [6]がある。IEEE 830 はすでに廃版となっており後継規格の ISO/IEEE 29148-2011 によって置き換えられている [8]。SRS 構成モデルを定めた組織標準として、欧州宇宙機関 (European Space Agency) の定めた ESA software engineering standards (ESAPSS-05-0 Issue 2) [4], アメリカ航空宇宙局 (National Aeronautics and Space Administration) が定めた NASA Software Documentation Standard (NASA-Std-2100-91) [10], アメリカ国防総省 (United States Department of Defense) が定めた Software Requirements Specification (DI-IPSC-81433A) [3], などがある。

標準規格以外では, Robertson らがコンサルティング経験から得た知見をもとに Volere Requirements Specification Template を提案している[14]。Wieggers らも著書の中で SRS のテンプレートを提案している [18]。また, 国内では情報サービス産業協会 (JISA) の要求工学 WG により編纂された要求工学知識体系 (REBOK) が SRS テンプレートを提案している [9]。著者によるこれらの SRS 構成モデル間の対応関係を表 1 に示す。

国際規格や組織標準として提供されている複数のソフトウェア要求仕様書テンプレートの中から, 自組織の標準テンプレートを作成するための適切な参照テンプレートを選択する基準を定義し, 5 つのテンプレートに対する評価を行った研究[5]や, 医療機器ソフトウェアに特化したソフトウェア要求仕様書[17]の提案など, SRS 構成モデルに関する研究がある。しかし, 参照テンプレートを拡張し, 自動車などの特定ドメインに適した SRS 構成モデルを設計する方法は確立していない。

4. アプローチ

ステークホルダはソフトウェアに対して各自のパースペクティブ (立場) に起因する異なる関心事を持ち, それらの関心事の違いが要求の多様性を生み出す。したがって, 多様な要求を表現できる SRS 構造モデルの設計にはステークホルダのドメイン特性に対する関心事を捉えることが必要である。本稿では, ステークホルダのドメイン特性に関する関心事に着目し SRS 構造モデルを設計するアプローチをとる。また, 提案方法を用いて, 自動車ソフトウェア向けの SRS 構造モデルを設計することで, 提案方法の有効性を評価する。

表 1 SRS 構成モデル間の対応関係

Table 1 Relation between SRS Constructive Models

ISO/IEC/IEEE 29148:2011	REBOK	IEEE Std. 830-1998	NASA DID P200	DI-IPSC-81433A	ソフトウェア要求 第3版
1. Introduction	1. はじめに	1. Introduction	1.0 Introduction		1. はじめに
1.1 Purpose	1.1 ソフトウェアの目的			1.2 System overview	1.3 プロジェクトスコープ
1.2 Scope	1.2 ソフトウェアの適用範囲	1.2 Scope			2. 概説
1.3 Product overview	2. 概要	2. Overall description			2.1 プロダクトの概要
1.3.1 Product perspective	2.1 ソフトウェアの展望	2.1 Product Perspective			5.1 ユーザーインターフェイス
		2.1.1 System interfaces			5.3 ハードウェアインターフェイス
		2.1.2 User interfaces			5.2 ソフトウェアインターフェイス
		2.1.3 Hardware interfaces			5.4 通信インターフェイス
		2.1.4 Software interfaces			
		2.1.5 Communications intrfaces			
		2.1.6 Memory constraints			
		2.1.7 Operations			
		2.1.8 Site adaptation requirements	5.6 Site Adaptation	3.6 Adaptation requirements	
1.3.2 Product functions	2.2 ソフトウェアの機能概要	2.2 Product functions			
1.3.3 User characteristics	2.3 ユーザ特性	2.3 User characteristics			
1.3.4 Limitations	2.4 制約	2.4 Constraints	5.5 Implementation Constraints		
? Apportioning of requirements	2.6 ソフトウェアで用いている定義, 用語, 略語	2.6 Apportioning of requirements	7.0 Partitioning for Phased Delivery	3.18 Precedence and criticality of requirements	
1.4 Definitions	1.4 ソフトウェアに関連する資料	1.3 Definitions, acronyms, and abbreviations	9.0 Glossary		付録A: 用語集
2. References	1.5 ソフトウェアの概要	1.4 References	2.0 Related Documenatation	2. Referenced documents	1.4 参考文献
3. Specific requirements	3. 詳細要求	3. Specific Requirements		1.3 Document overview	
3.1 External interfaces	3.2 外部インターフェイス要求	3.1 External interfaces	4.0 External Interface Requirements	3.3 CSCI external interface requirements	5. 外部インターフェイス要求
3.2 Functions	3.3 ソフトウェア機能	3.2 Functions		3.2 CSCI capability requirements	3. システムフィーチャ
3.3 Usability Requirements	3.4 性能要求	3.3 Performance requirements	5.2 Perfomance and Quality Engineering Requirements		6.2 性能
3.4 Performance requirements	3.5 論理データベース要求	3.4 Logical database requirements		3.5 CSCI internal data requirements	
3.5 Logical database requirements	3.5 設計制約	3.5 Design constraints	5.5 Implementation Constraints	3.12 Design and implementation constraints	2.4 設計と実装の制約条件
3.6 Design constraints	3.4 ソフトウェア品質要求	3.5.1 Standards compliance		3.11 Software quality factors	6. 品質属性
3.7 Software system attribtes		3.6 Software system attributes			
		3.6.1 Reliability			
		3.6.2 Availability			
		3.6.3 Security	5.4 Security and Privacy Requirements	3.8 Security and privacy requirements	6.3 セキュリティ
		3.6.4 Maintainability			
		3.6.5 Portability			
3.8 Supporting information		4. Supporting Information			
4. Verification				4. Qualification provisions	
5. Appendices	2.5 前提条件と依存	4.2 Appendices	11.0 Appendices	Appendices	
5.1 Assumptions and dependencies		2.5 Assumptions and dependencies	8.0 Abbreviations and Acronyms		2.5 前提条件と依存関係
5.2 Acronyms and abbreviations				1.3 Document overview	1.1 目的
		1.1 Purpose			
		4.1 Table of contents and index			
				1. Scope	
				1.1 Identification	1.2 文書の表記法
				3. Requirements	
	3.1 システム特性 (System mode)			3.1 Required states and modes	
				3.2 CSCI Requirements	
				3.4 CSCI internal interface requirements	
				5.1 Process and Data Requirements	4. データ要求
					4.1 論理データベース
					4.2 データフロー/シグナリ
					4.3 レポート
					4.4 データの獲得, 整合性, 保持, 廃棄
			5.3 Safety Requirements	3.7 Safety requirements	6.4 安全性
			5.7 Design Goals		2.2 ユーザインタフェース特性
				3.9 CSCI environment requirements	2.3 稼働環境
				3.10 Computer resource requirements	6.1 ユーザインタフェース
				3.13 Personal-related requirements	6.x [その他]
				3.14 Training-related requirements	7. 国際化要求と現地化要求
				3.15 Logistics-related requirements	8. その他の要求
				3.16 Other requirements	付録B: 分析モデル
				3.17 Packaging requirements	
			6.0 Traceability to Parents Design	5. Requirements Traceability	
			10.0 Notes	6. Notes	

とステップ (2) で展開したドメイン特性とパースペクティブから、ステークホルダがドメイン特性に関して持つ関心事を展開する。ステップ (5) では、まず初めにドメイン特性に関する関心事と参照 SRS 構成モデルの目次項目の対応付けを行う。続いて、参照 SRS 構成モデルとの対応関係が見つからなかったドメイン特性に関する関心事に対して、それらの関心事に関する要求を表現するための目次項目を定義する。ステップ (6) では、新たに定義した目次項目を追加することで参照 SRS 構成モデルを拡張し、ドメイン固有の SRS 構成モデルを設計する。以降の節では、自動車ソフトウェア向け SRS 構成モデルの設計を例に各ステップの詳細について説明する。

5.3 (1) 参照 SRS 構成モデルの選択

SRS 構造モデルのすべてを独自に定義するには工数を要するので、自組織に適した既存の SRS テンプレートを選定し利用する方法を提案する。ベースとなる SRS テンプレート (以下、参照 SRS 構成モデルと呼ぶ) の選定には[17]で提案されている評価基準を利用する。[17]による 5 つの SRS に対する評価結果を表 2 に示す。

表 2 SRS テンプレートの評価基準と評価結果

Table 2 SRS Evaluation Criteria and Evaluation Results

評価基準	IEEE 830-1984	ESA PSS-05-0	NASA DID-P200	JPL D-4005	DOD MIL-STD-498
独立性	Medium	Medium	High	High	High
統合性	Mdeium	Medium	High	High	High
正確性	High	Low	High	High	High
一般性	Not for large project	Not for large projects	Not for small projects	Suits all projects	Not for small projects
構成	High	Low	Low	Medium	Medium to high
項目の完全性	High	Medium	High	High	High
ビューの完全性	Satisfiers all users	Satisfies only testers	Satisfies design and testers	Satisfies all users	Satisfies all users
変更容易性	Adequate	Inadequate	Inadequate	Adequate	Adequate

自動車ソフトウェア開発では下記の理由から「一般性」「ビューの完全性」「変更容易性」の 3 つの評価基準が重要となる。

- (1) 規模の異なる大小のプロジェクトが存在することから、評価基準「一般性」がすべてのプロジェクトに適合する (Suits all projects) と評価されていること
- (2) 多様なステークホルダが SRS の読者となることから、評価基準「ビューの完全性」がすべてのユーザ (のニーズ) を満足する (Satisfies all users) と評価されていること
- (3) 派生開発が主流のため変更容易性が重要であること

から、評価基準「変更容易性」が十分である (Adequate) と評価されていること

3 つの評価基準をすべて満たすのは JPL D-4005 のみであるが、この規格は現在一般には公開されていない。JPL D-4005 に次いで自動車ソフトウェアへの適合度が高いのは「一般性」と「変更容易性」の評価基準を満たす IEEE 830-1998 と DOD MIL-STD-498 の 2 つである。自動車ソフトウェア開発では国際規格への準拠が強く求められるので、最終的に IEEE 830-1998 が参照 SRS 構成モデルとして最適であると結論した。しかし、IEEE 830 はすでに廃版となり後継規格の ISO/IEC/IEEE 29148-2011 で置き換えられている。ISO 29148 が提案する SRS 構造は、細部の違いはあるが基本的に IEEE 830 を踏襲しているため、本稿では ISO 29148:2011 で定義されている SRS の構造を自動車 SRS 構造モデルの参照モデルとした。

5.4 (2) ドメイン特性の展開

5.4.1 製品領域の定義

自動車を構成するカーエレクトロニクス製品は多様であり、複数の製品領域に分類することができる。文献により製品領域の分け方には差異があり、広く合意された分類は存在しない。本稿では、文献[11][12][15][16]が定義する製品領域を参考に、次の 5 つの製品領域に分類した。

- (1) パワトレ&シャシー
- (2) ボディー
- (3) アクティブセーフティー
- (4) インフォテインメント
- (5) 通信インフラ

これら 5 つの製品領域のうち、パワトレ&シャシードメインはパワートレインとシャシー、ボディードメインは利便快適とパッシブセーフティーをそれぞれサブ製品領域に持つ。各製品領域の説明と製品例を表 3 に示す。

表 3 カーエレクトロニクス製品領域

Table 3 Domains of Car Electronics Products

製品領域	説明	製品例	
パワトレ&シャシー	車両の駆動力に関する製品	エンジン制御ユニット	
	車両の走行制御に関する製品	電動パワーステアリング	
ボディー	利便快適	車室内の利便性と快適性に関する製品	カーエアコン
	パッシブセーフティー	衝突時の乗員の安全を確保するための製品	エアバッグ
アクティブセーフティー	衝突事故を回避または損害を軽減するための製品	自動ブレーキ	
インフォテインメント	乗員へ情報と娯楽を提供するための製品	カーナビゲーション	
通信インフラ	車両と車外のサービスを接続するための製品	データ通信モジュール	

5.4.2 ドメイン特性展開表の作成

上記で定義した製品領域を入力にドメイン特性の展開を行う。本研究で調査した範囲内では、カーエレクトロニク

ス製品領域による特性の違いを網羅的に扱った文献は存在しないため、製品領域の定義で参照した文献[11][12][15][16]を参考に、製品領域からドメイン特性への展開を行った。その結果を表 4 に示す。

表 4 ドメイン特性展開表

Table 4 Domains Characteristics Deployment Table

ドメイン特性	製品ドメイン						
	パワトレ&シャシー		ボディー		アクティブセイフティ	インフォテインメント	通信インフラ
	パワー トレイン	シャシー	利便快適	ハッチブ セイフティ			
制御アルゴリズム (連続制御)	○	○			○		
離散イベント処理 (離散制御)			○	○			
データ処理					○	○	○
車内ネットワーク上での相互運用性	○	○	○	○	○	○	○
車外システムとの相互運用性						○	○
ハードリアルタイム	○	○		○	○	○	○
ソフトリアルタイム			○			○	○
安全性要求	○	○		○	○		
セキュリティ要求						○	○
プライバシー要求						○	○
信頼性要求	○	○		○	○		○
可用性要求	○	○		○	○		
法規遵守	○	○		○	○		○
ユーザインタフェース (メカニカル)			○			○	
ユーザインタフェース (GUI)						○	
適合 (キャリブレーション)	○	○			○		
可変性	○	○	○	○	○	○	○

5.5 (3) ステークホルダのパーспекティブの展開

5.5.1 ステークホルダの特定と抽象化

SRS が要求を定義する対象のソフトウェアに関わりを持つステークホルダを特定する。ステークホルダの特定方法にはオニオンモデル[2]やペルソナ[13]を利用した方法など多くの特定方法がある。

自動車ソフトウェアの開発では、3年から5年程度の長い開発ライフサイクルを通して多くのステークホルダが関わるため企業情報システムの開発に利用される方法をそのまま適用することは困難である。本稿では、簡略化した自動車開発のライフサイクルに沿って主要なステークホルダを特定した。

特定したステークホルダは、同種の関心事を持つグループをステークホルダクラスとしてまとめることで抽象化する。ステークホルダの特定と抽象化の結果を表 5 に示す。

5.5.2 パerspекティブ展開表の作成

ステークホルダクラスからパーспекティブを展開する。パーспекティブとは、ある特定の読者が SRS に対して取る立場である。そのため、異なるステークホルダは SRS に対して異なるパーспекティブを持つ。例えば、取得者は SRS に記述される「要求を出す (要求する)」というパーспекティブを持つ。また、表 5 に挙げたステークホルダクラスのうち整備士のように直接は SRS を参照しないステークホルダクラスもある。しかし、整備士はテクニカルライタが SRS を入力に作成したサービスマニュアルを

表 5 自動車ソフトウェアのステークホルダ

Table 5 Stakeholders of Automotive Software

開発ライフサイクル	ステークホルダ	ステークホルダクラス
製品企画	完成車メーカ	取得者 (顧客)
	上位Tierサプライヤ	
製品開発	プロジェクトマネージャ	プロジェクトマネージャ
	要求エンジニア	要求エンジニア
	システム開発者	開発者
	ソフトウェア開発者	
	ハードウェア開発者	
	ビルド担当者	
	システムテスタ	テスタ
	ソフトウェアテスタ	
	ハードウェアテスタ	テクニカルライタ
	ユーザマニュアル作成者	
サービスマニュアル作成者		
量産	生産技術者	製造部門
	品質保証技術者	
販売	自動車セールス	自動車セールス
	整備士	整備士
利用	自動車オーナー	ユーザ
	商用車両運転手	
	品質管理技術者	品質管理部門

参照して車両の点検や修理を行う。そのため、整備士は SRS を間接的に参照していると解釈し、「点検・修理する」を整備士のパーспекティブとして設定した。各ステークホルダクラスに対して主要なパーспекティブ (1 次パーспекティブ) と 2 次的なパーспекティブを展開した結果を表 6 に示す。

5.6 (4) ステークホルダのドメイン特性に関する関心事の展開

これまで展開したドメイン特性とステークホルダクラスのパーспекティブから、ステークホルダクラスが持つドメイン特性に関する関心事を展開する。関心事の展開には各ステークホルダクラスの 1 次パーспекティブのみを利用する。ドメイン特性とステークホルダクラスのパーспекティブからドメイン特性に関する関心事を展開する理由は、ステークホルダが自身のパーспекティブを通してドメイン特性を見ることで、そのドメインに関する関心事が

表 6 パerspекティブ展開表

Table 6 Perspective Deployment Table

	パーспекティブ										
	要求する	管理する	変更する	設計・実装する	テストする	説明する	製造する	販売する	点検・修理する	利用する	品質管理する
取得者 (顧客)	P	S			S						S
プロジェクトマネージャ	S	P									
要求エンジニア			P								S
開発者				P	S						S
テスタ					P						S
テクニカルライタ						P					
製造部門					S		P				
自動車セールス								P			S
整備士									P		
ユーザ										P	
品質管理部門											P

凡例
 P: 1次パーспекティブ S: 2次パーспекティブ

明確になるためである。そのため、パースペクティブの異なるステークホルダ同士では関心事も異なる。

関心事を展開する際には、展開のためのガイドとして自動車ソフトウェアのような組込みソフトウェアでは[1]の関心事チェックリスト、企業情報システムソフトウェアでは Zachman フレームワーク[19]のような視点をガイドとして利用することを提案する。例えば[1]には時間的な特性 (Temporal characteristics) に関するチェックリストとして以下の質問が提案されている。

- (1) センサとアクチュエータの間に遅延はあるか？
- (2) フィードバックループには応答時間の制約はあるか？
- (3) 通信チャンネルには応答時間の制約はあるか？

開発者のパースペクティブ「設計・実装する」からドメイン特性「ハードリアルタイム」を見る際に上記のチェックリスト (2) や (3) を参照することで、「応答時間」というドメイン関心事の引き出しがガイドされる。同様にすべてのドメイン特性に対してドメイン関心事を展開した結果を表 7 に示す。

5.7 (5) ドメイン特性に関する関心事から SRS 目次項目への展開

ドメイン特性に関する関心事から SRS 目次項目への展開 (対応付け) を行う。SRS 目次項目への展開の目的は、ステップ (1) で選定した参照 SRS 構成モデルが取り扱えるドメイン関心事を識別し、取り扱えないドメイン関心事に関する要求を表現するための追加の目次項目を選定することである。

参照 SRS 構成モデルが関連する要求を表現できるドメイン関心事を識別するには、参照 SRS 構成モデルとして選定した SRS テンプレートが提供する記述ガイドを利用する。一般的な記述ガイドには目次項目ごとに記述すべき内容の説明が含まれるため、その説明から参照 SRS 構成モデルにドメイン関心事を取り扱える目次項目が含まれるかを判断する。表 8 の上部 (緑色の部分) がドメイン関心事の参照 SRS 目次項目への展開結果を表している。

続いて、参照 SRS 構成モデルでは関連する要求を表現できないドメイン関心事に対して、参照 SRS 構成モデル以外の SRS テンプレートから目次項目を追加するか、独自に目次項目を定義し参照 SRS 構成モデルを拡張する。追加の目次項目を選定する際には、関連研究に挙げた SRS テンプレートなどが利用できる。

5.8 (6) 参照 SRS 構成モデルの拡張

SRS 構成モデル設計プロセスの最終ステップとして、ステップ (5) で展開した追加目次項目を参照 SRS 構成モデルへ追加 (mix-in) することで拡張し、新たな SRS 構成モデルを設計する。(1) から (6) のステップから成る SRS 構成モデルの設計プロセスに従い設計した自動車ソフトウェアの SRS 構成モデルを図 4 に示す。

6. 評価

5 章で設計した自動車ソフトウェアの SRS 構成モデル (図 4) を分析し、本稿で提案する SRS 構成モデル設計方法の有効性を評価する。

図 4 では、設計した SRS 構成モデルをクラス図で表現している。SRS クラス以外の個々のクラスは目次項目に相当し、SRS クラスは他のクラスのコンポジションとして表現される。白い背景色のクラスが参照 SRS 構成モデル (ISO 29148) から継承した目次項目、青い背景色のクラスが参照 SRS 構成モデルへ追加した目次項目をそれぞれ表している。

章レベルへの追加項目は単なるプレースホルダのため、節と項のレベルへの追加項目にのみ着目すると、節レベルへの追加項目は「ソフトウェアのモード」「データ辞書」「可変性要求」「標準と法規への適合要求」「国際化要求と現地化要求」である。これらの目次項目は、多くの国と地域へ出荷されるため多くのバリエーションを持ち、安全性確保のための業界標準や法規への適合が求められるといった自動車ドメインの特性を捉えている。一方、項レベルへの追加項目はすべて自動車ソフトウェアが満たすべき非機能要求であり、自動車ソフトウェアの特性を捉えている。

以上の結果から、提案方法に基づき設計した自動車ドメイン固有 SRS 構成モデルは、上位 (節) のレベルではドメインの特性に起因する要求、下位 (項) のレベルではドメインに属するソフトウェアに求められる非機能要求を表現するための目次項目が導出されたと言える。

7. 考察

7.1 SRS 構成モデル設計方法の有効性

多様なステークホルダから要求を獲得し、それらを表現するためには、適切な構造を持った SRS 構成モデルが必要である。関連研究でも述べたように、SRS 構成モデルを提案する多くの文献があるが、その多くは経験的に重要と思われる要素 (目次項目) を列挙した汎用的なもので、特定のドメインへそのまま適用するには不適切である。しかし、特定ドメイン (自組織) に適した SRS 構成モデルの設計に関する研究は少ない。複数の SRS 構成モデルからベースとなる参照 SRS 構成モデルを選択するための基準を提案する研究はあるが、選択した参照 SRS 構成モデルから派生 SRS 構成モデルを設計する方法については未確立である。

本稿では、既存研究では未確立であった参照 SRS 構成モデルに対する過不足を明らかにし、拡張するための体系的な方法を提案した。自動車ドメインに対して提案方法を適用し有効性を評価した。適用結果から、ドメインそのものの特性に関する要求、ドメインに属するソフトウェアに関する要求をそれぞれ表現するための目次項目の追加により参照 SRS 構成モデルが自動車ドメイン向けに拡張されたことが確認された。ドメイン全体の特性と、そのドメインに属するソフトウェアの特性という 2 つの要因に起因する

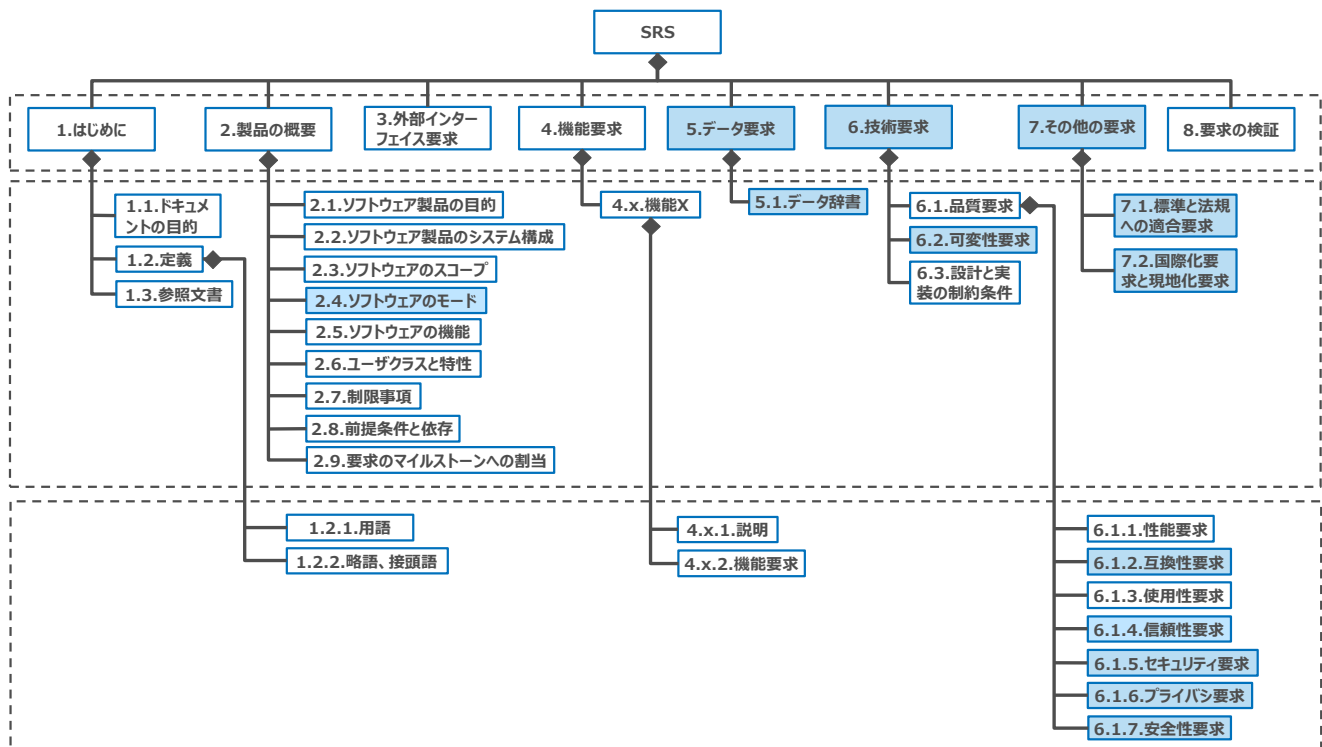


図 4 自動車ドメイン固有 SRS 構成モデル

Figure 4 Automotive-Domain-Specific SRS Construction Model

要求を表現するための目次項目が特定できていることから、本稿で提案する設計方法は有効であると考えられる。

7.2 自動車ドメイン固有 SRS 構成モデル

本研究で調査した範囲では、自動車ソフトウェアのための SRS 構成モデルを提案した研究はない。本稿の提案方法に基づき設計した自動車ドメイン固有 SRS 構成モデルは、著者の所属する以外の自動車ソフトウェア開発組織にとっても有用であると考えられる。

8. まとめ

本稿では、ステークホルダのドメイン特性に対する関心事に基づき要求項目のメタモデルを定義した。さらに、要求項目展開表によって体系的にドメイン固有 SRS 構成モデルを設計する方法を提案した。自動車ドメインに対し提案方法を適用し、ドメイン固有 SRS 構成モデルを体系的に設計できることが確認できた。

今後、実際の製品開発に適用することで、設計した SRS 構成モデルの妥当性を評価する。また、自動車以外のドメインにも提案方法を適用し、設計方法の有効性を評価する。

参考文献

- [1] D. Aceituna, Survey of Concerns in Embedded Systems Requirements Engineering, SAE Int'l J. of Passenger Cars-Electronic and Electrical Systems, Vol. 7, No. 1, May 2014, pp. 1-7.
- [2] I. Alexander, A Taxonomy of Stakeholders, Human Roles in System Development. Issues and Trends in Technology and Human Interaction, 2006, pp. 25-71.
- [3] DoD, DI-IPSC-81433A, Data Item Description: Software Requirements Specification, Dec. 1999.
- [4] ESA Software Engineering Standards, ESA PSS-05-0 Issue 2 Revision 1, European Space Agency, ESA Pub. Div., 1994.
- [5] E. A. Giakoumakis and G. Xylomenos, Evaluation and Selection Criteria for Software Requirements Specification Standards, Softw. Eng. J., Vol. 11, No. 5, 1996, pp. 307-319.
- [6] IEEE, Std. 830-1998: IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998.
- [7] ISO/IEC, ISO/IEC 25010:2011, Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)- Systems and Software Quality Models, 2011.
- [8] ISO/IEC/IEEE, ISO/IEC/IEEE 29148:2011 - Systems and Software Engineering - Life Cycle Processes - Requirements Engineering, ISO, 2011.
- [9] JISA REBOK 企画 WG, 要求工学知識体系 第1版, 近代科学社, 2011
- [10] NASA, NASA-STD-2100-91, NASA Software Documentation Standard, NASA, Jul. 1991.
- [11] R. Oshana and M. Kraeling, Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications, Newnes, May 2013.
- [12] A. Pretschner, et al., Software Engineering for Automotive Systems, Proc. of FOSE 2007, IEEE, May 2007, pp. 55-71.
- [13] J. Pruitt and T. Adlin, The Persona Lifecycle: Keeping People in Mind throughout Product Design, Morgan Kaufmann, 2006.
- [14] S. Robertson, and J. Robertson, Mastering the Requirements Process, 2nd ed., Addison-Wesley, 2006.
- [15] J. Schäuffele and T. Zurawka, Automotive Software Engineering 2nd ed., SAE International, Sep. 2016.
- [16] M. Staron, Automotive Software Architecture, Springer, 2017.
- [17] H. Wang, et al., Envisioning a Requirements Specification Template for Medical Device Software, Proc. of PROFES 2014, LNCS Vol. 8892, Springer, Dec. 2014, pp. 209-223.
- [18] K. Wiegers and J. Beatty, Software Requirements, 3rd ed., Microsoft Press, 2013.
- [19] J. A. Zachman, A Framework for Information Systems Architecture, IBM Syst. J., Vol. 26, No. 3, 1987, pp. 276-292.

