車載システム開発における
ディペンダビリティ保証手法に関する研究

小林 展英

目次

1. 序論

1.1. 本研究の目的

従来の自動車業界では，運用後の振る舞いが固定化できることを前提とした自動車単体で実現されるシステムを中心に品質保証に取り組んできた．しかしながら，今後の自動車業界は，様々な機器が有する情報を連携させることで，自動運転をはじめとする高度なシステムの実用化を目指している．この分野におけるシステムの障害は，ユーザの生命に対する危険，およびユーザの個人情報の流出をもたらし，大きな社会問題を招く．また，このようなシステムの多くは，自動車単体で実現されるのではなく，図 1-1 に示すように，自動車が置かれた状況に合わせて動的に着脱される他車，IoT 機器，社会インフラのような他システム群，実世界の状況に合わせて常に進化する知識群，さらにそれら膨大な情報群を効率的に扱う人工知能，などと連携することで実現される．これらのシステムは運用開始後も進化することに価値があり，従来の自動車業界が品質保証の前提としてきたシステム特性とは大きく異なっている．このように，今後の自動車業界では，品質に対する価値観が互いに異なるシステムが相互に連携して一つのシステムを実現していくこととなり，こうした状況は，前述した問題の解決をさらに難しくしている．

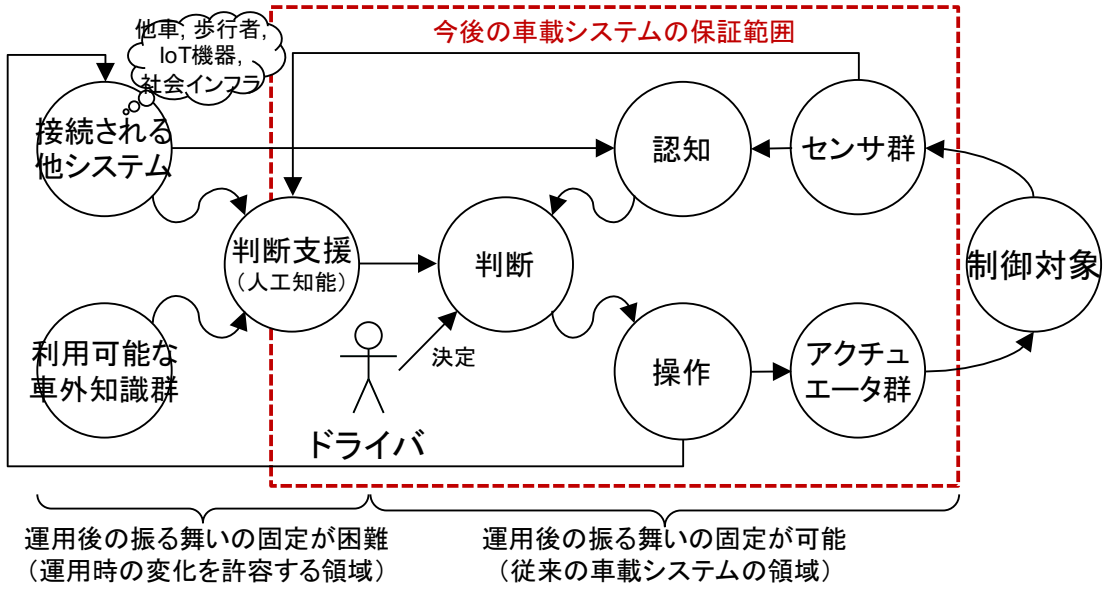


図 1-1 今後の車載システムの構成

このような状況下で, システムのディペンダビリティを保証する手段として O-DA (Open Dependability through Assuredness)[1]が注目されている．O-DA は，The Open Group で標準化されたオープンシステムに対するディペンダビリティ品質保証フレームワークであり，開発活動は TOGAF (The Open Group Architecture Framework)[2]に基づいている．O-DA の特徴は，TOGAF に基づいて作成した設計成果の品質状況をアシュアランスケースを用いて確認し，その結果を関係者と常に合意形成できている点にある．アシュアランスケースは，議論の前提条件を明らかにし，その前提条件に基づいて議論を構造的に分解して記述できる文書である．システムのディペンダビリティに関する議論にアシュアランスケースを使用することで，異なる価値観を有したステークホルダ間の前提条件を揃え，議論内容を正しく共有することが可能となる．なお，セーフティに関するアシュアランスケースをセーフティケース，セキュリティに関するアシュアランスケースをセキュリティケースと呼ぶ．

しかしながら，車載システム開発における従来のアシュアランスケースの研究には，表 1-1 に示す 5 つの不足が存在している．本研究では，これらの不足に対応する上で解決が必要となる 5 つの課題に取り組む．

表 1-1 従来の研究におけるアシュアランスケースの不足

ID	不足内容	課題
不足①	車載システム開発で利用されているセーフティ分析手法とセーフティケースを統合する方法が考慮されていない.	従来分析手法と統合したセーフティケース作成法の導入（1.1.1 節参照）
不足②	セキュリティケース作成法の有効性が車載システム開発において確認されていない.	車載分野に対するセキュリティケース作成法の有効性確認（1.1.2 参照）
不足③	セーフティ要求とセキュリティ要求が背反した場合など要求が対立した際の解決手法が考慮されていない.	品質特性の異なる要求の対立問題解消法の導入（1.1.3 参照）
不足④	車載システム開発にゴール指向手法を適用する際に有用な参照モデルとの関係が議論されていない.	車載システム開発向け参照モデルの活用法の導入（1.1.4 参照）
不足⑤	セーフティ，セキュリティ以外の要求の保証にも適用できるアシュアランスケースの統一的な作成手法が存在していない.	品質特性に基づくアシュアランスケース作成法の導入（1.1.5 参照）

1.1.1. 課題①：従来分析手法と統合したセーフティケース作成法の導入

現在, 車載ソフトウェア開発では, HAZOP（Hazard and Operability Studies）, FTA（Fault Tree Analysis）といった分析手法を用いて安全性を分析し，その結果に基づいて車載ソフトウェアの開発を進めている．ISO26262 の本格導入を想定すると，これに加えて開発した車載ソフトウェアの安全性を第三者に納得してもらうためのアシュアランスケースの作成が必要になる．アシュアランスケースの作成には，記述品質の安定化を図るために，D-Case などの図式言語の採用が期待されるが，従来の D-Case 作成法では，HAZOP，FTA の分析結果を証拠として用いる，という分析過程が反映されない単純で間接的なガイドラインしか存在していなかった．このため，開発現場では具体的な安全分析結果に基づいた説明が間接的になるという問題があった．この問題を解決するためには，HAZOP，FTA と D-Case を対応づけるとともに，その手順を提示する必要がある．このため，本提案では，外部と内部の視点，分析と確認の視点に基づいて，

HAZOP, FTA, D-Case を対応づけた新たな組み合わせ手法を提案する．また，本手法を適用した実験結果に基づき，その有効性を確認する．

1.1.2. 課題②：車載分野に対するセキュリティケース作成法の有効性確認

モバイルサービスを保証対象としたセキュリティケースの効果的な作成法は考案されているが[3]，その手法を車載分野のサービスに適用した際の有効性について議論していない．車載分野のサービスは，AUTOSAR[4]が策定したアーキテクチャに基づいたシステム上で実現されるため，前述した研究成果を適用する際に，そのアーキテクチャをどのように表現するか考案する必要がある．さらに，考案した表現方法を用いて記述した車載分野の事例に対して，前述のセキュリティケース作成法を適用し，その有効性と他分野のシステムへの適用性について確認する必要がある．

1.1.3. 課題③：品質特性の異なる要求の対立問題解消法の導入

ディペンダビリティは，セーフティ，セキュリティのように品質特性の異なる要求で構成されるため，それらの要求間で対立問題が発生する可能性を含んでいる．この問題を解消するためには，それらに対する達成度を定量的に評価できる手法が必要となる．ゴール指向分析法は，セーフティとセキュリティのような非機能要求の比較に導入されている．また，ゴールに属性を付与したいくつかの手法が非機能要求の定量的な評価方法を提案している．本提案では，NFR フレームワークのソフトゴールに定量的な重みを付与した SIG (Softgoal Interdependency Graphs) を用いて，非機能要求を定量的に評価する手法を提案する．また，重み付きソフトゴールの適用結果に基づき，非機能要求の定義，および設計方針案の選択方法を説明する．さらに，車載ソフトウェア開発分野への適用事例を用いて，本手法が自動車業界においても有効であることを確認する．

1.1.4. 課題④：車載システム開発向け参照モデルの活用法の導入

車載システム開発において必要とされる知識は，車載ソフトウェアの大規模，複雑化に従って大幅に増加している．このため，一人のエンジニアが独力で開発全体の知識を備えることは非常に困難な状況となっている．本提案では，7 人の侍フレームワーク[5]に基づいて熟練エンジニアの知識を可視化したメタモデル

を構築する方法を提案する．さらに，そのメタモデルをプロダクトラインにおける標準ソフトウェア資産の評価に適用した結果を説明し，その結果に基づいて提案手法の有効性を確認する。

1.1.5. 課題⑤：品質特性に基づくアシュアランスケース作成法の導入

O-DA を運用するためには，セーフティ要求，セキュリティ要求と同様に，ディペンダビリティを構成する様々な品質特性の要求に対して，アシュアランスケースを作成する必要がある．本提案では，品質特性に基づいて統一的にアシュアランスケースを生成する手法を提案する．さらに，車載ソフトウェア開発のレビューに適用した実験結果に基づいて，その有効性を確認する．

1.2. 本研究の内容

本研究は，O-DA 運用の要となるアシュアランスケースの実用化に関する研究を中心としている．O-DA が保証対象する品質特性はディペンダビリティと呼ばれ，車載分野では，セーフティ，セキュリティがディペンダビリティを構成する主要な品質特性となる．本研究では，これら 2 つの品質特性を統合的に扱い，最終的にアシュアランスケースを用いて，その品質を保証する手法について述べる．

1.3. 本研究のアプローチ

本研究では，1.1 節で設定した課題に対して次のようにアプローチする．3 章では，課題①に対応するために，従来から用いられている安全分析手法と統合したセーフティケース作成法として，「D-Case を用いた安全分析結果の説明手法」を提案する．4 章では，課題③に対応するために，重み付けしたソフトゴールを用いて品質特性の異なる要求の衝突問題を定量的に評価する「非機能要求の定量評価手法」を提案する．5 章では，課題④に対応するために，熟練エンジニアの設計知識を形式知化し，ゴール指向分析で活用する方法として，「7 人の侍フレームワークを用いた標準ソフト資産の評価知識」を提案する．6 章では，課題⑤に対応するために，セーフティ，セキュリティに特化することなく統一的にアシュアランスケースを生成する方法として，「SPRME を用いたアシュアランスケース作成手法」を提案する．図 1-2 に上述した内容の関係図を示す．なお，本研究では，課題②に対応するセキュリティケース作成法に関して述べられて

いない．しかし，この部分については実施範囲と相互に依存が少なく、分けて考えても問題がない．この部分については今後の課題として扱う．

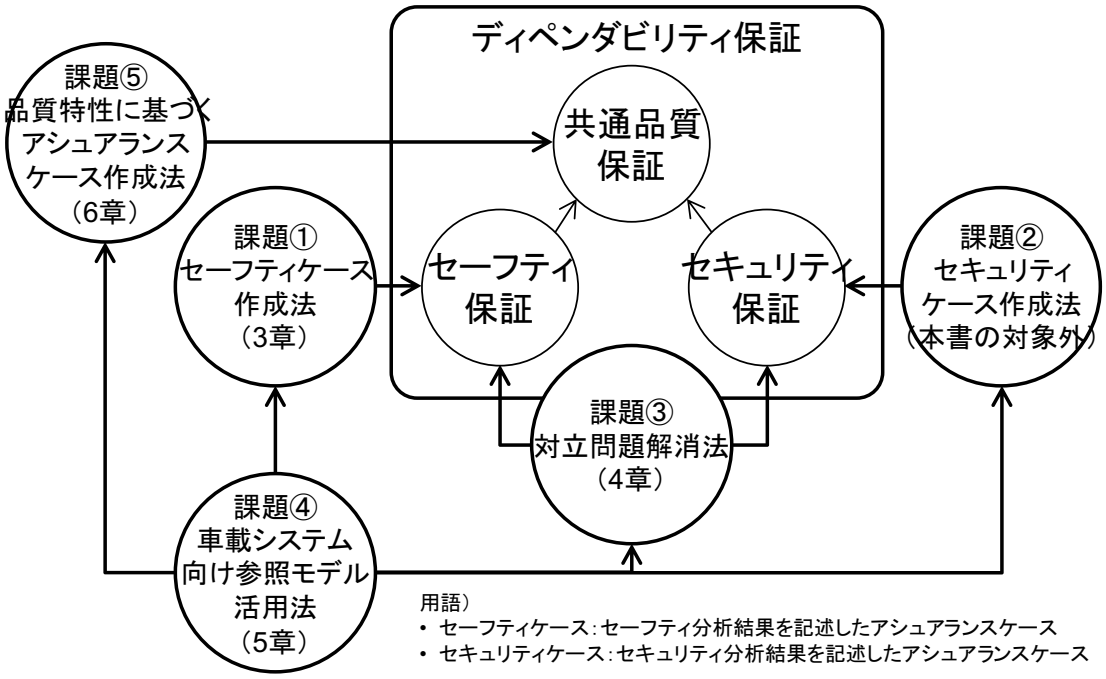


図 1-2 本論文における各章の位置づけ

2. 関連研究

本章では，2.1 節で本研究の背景となる O-DA (Open Dependability through Assuredness) の関連研究について述べる．さらに，1.3 節で述べた提案毎に対応する関連研究を 2.2 節以降で述べる．

2.1. O-DA (Open Dependability through Assuredness) に関する研究

2.1.1. O-DA の概要

The Open Group が発行した O-DA[1]は，大規模で複雑なシステムのディペンダビリティを達成するために，高保証性 (Assuredness) の概念を導入している．高保証性とは，保証対象となるシステムのアーキテクチャの実装が，保証すべき指定された要件を満たしていることを確信するために，満足な水準の証拠が提供されていることをシステムのステークホルダが合意している状態であり，各ステークホルダはその合意に責任を持つ必要がある．O-DA は，図 2-1 に概観を示す TOGAF[2]に基づいて考案されたディペンダビリティ保証フレームワークであり，O-DA を用いた開発では，保証対象に起こり得るリスクの対策状況をアシュアランスケースを用いて網羅的に確認し，ステークホルダ間で合意形成を図る．これにより，保証対象が高保証性を有していることが保証される．

O-DA の現場導入を支援する研究成果としては，山本によって O-DA テンプレートが考案されている[6]．このテンプレートは，システムアーキテクチャ評価サービスを事例としたサービス構想書の雛形を提供しており，その適用性の高さは，実際の車載ソフトウェア開発企業におけるソフトウェア品質評価サービスの構想書を立案した事例において確認されている[7]．

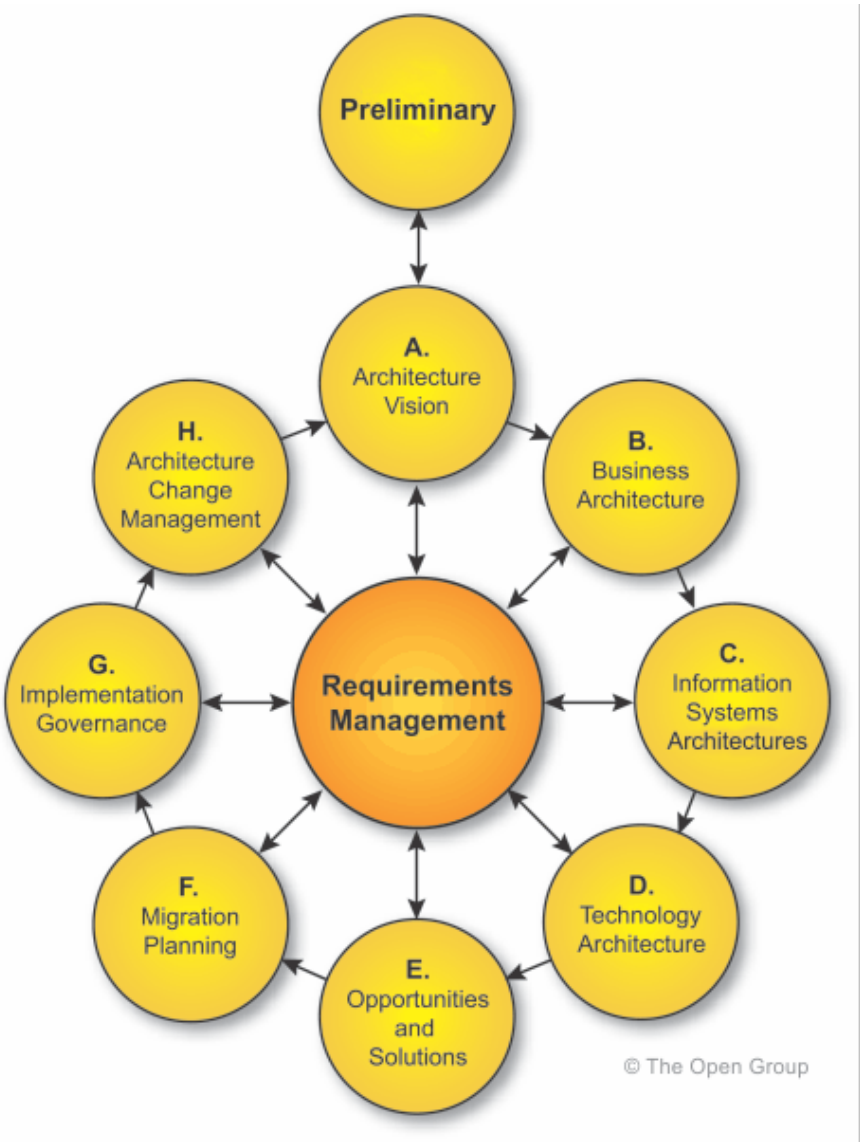


図 2-1 TOGAF が提供する開発フレームワーク
（出典：オープングループ http://www.opengroup.org）

- Preliminary：
  プロジェクトの準備活動
- A. Architecture Vision：
  スコープ，制約，期待，ステークホルダを定義，事業環境を確認
- B. Business Architecture：
  ビジネスの現行と目標アーキテクチャを定義，差異分析
- C. Information Systems Architecture：

情報システムの現行と目標アーキテクチャを定義，差異分析

- D. Technology Architecture：
  技術の現行と目標アーキテクチャを定義，差異分析
- E. Opportunities and solutions：
  実施計画，展開手段，要素を定義して移行アーキテクチャ構築
- F. Migration Planning：
  費用対効果分析，リスク分析に基づき移行実施計画を詳細化
- G. Implementation Governance：
  アーキテクチャ移行計画を管理，実装結果を確認
- H. Architecture Change Management：
  アーキテクチャの事業目標適合性を継続的監視，変更管理
- Requirements Management：
  全行程で要求確認を実施

2.1.2. アシュアランスケースの概要

アシュアランスケースは，保証対象となるシステムのディペンダビリティを保証するために作成される文書である[8]．アシュアランスケースが普及した背景には，1988 年に発生した死者 167 名を出した北海油田事故などの深刻な事故があり，そのような事故を背景としてエビデンスに基づく議論の重要性が認知されるようになってきている[9]．また，自動車業界においてアシュアランスケースが必要となる背景として，運用開始後も常に進化し続ける人工知能，システム構成が動的に変化する IoT 機器との連携など，従来存在しなかった特性を有するシステムの登場が予見される点が挙げられる．このようなシステムを保証するためには，保証の前提条件を明らかにした上で，保証内容を構造的に分解し，適切な粒度になった保証内容を事実に基づいて確認できる必要がある．つまり，こういった内容を関係者と正しく合意形成するために用いられるアシュアランスケースは，保証の前提条件，保証内容の分解の仕方などが正しく整理されて記述できていることが期待される．このような記述要件を満足するための記述法として，後述する GSN (Goal Structuring Notation) などの記述言語が提案されている．

2.1.3. GSN (Goal Structuring Notation)の概要

GSN[10]は Tim Kelly によって提案されたゴール指向のアシュアランスケース記述法であり，GSN COMMUNITY STANDARD VERSION 1[11]として広く知られている．また，アシュアランスケースの議論内容がリアルタイムに確認できることを目的として，DEOS から GSN をベースとした D-Case が提案されている[12], [13]．GSN では，表 2-1 に示すノードを用いて主張したい命題の妥当性を説明する．最上位の Goal は，Context で分解根拠を示された Strategy を介して下位の Goal に分解される．Solution は，Goal の主張内容が実際に対策されていることを確認した記録を定義し，最下位の Goal に紐付ける．これにより，最下位の Goal の妥当性が証明され，その積み上げで最上位の Goal の妥当性が証明される．

表 2-1 GSN で使用できる基本的なノードと関係

ノード名	記号	説明
Goal	主張	相手と合意形成したい「対象が達成すべき状態」を示す。 例)「システムは安全である」
Strategy	説明	上位の主張を下位の主張に分解する説明を示す。 例)「～に従って説明する」
Context	前提	主張や説明が基づく相手と合意済みの情報を示す。 例)「リスク分析の結果得られたハザードのリスト」、 「ソフトウェアの構造」など
Solution	証拠	主張が達成できていることを示す証拠（証跡） 例)「テスト報告書」、「運用記録」
Undeveloped	◇	まだ具体化できていない主張や説明であることを示す （未定義要素）。
SupportedBy	◀——	主張が説明に従って分解され証拠に至る流れを示す。
InContextOf	◁——	主張や説明が基づく前提への紐付けを示す。

2.2. セーフティケース作成法に関連する研究

HAZOP は，外部視点で観察可能な分析対象のパラメータに対して，多い，少ないといった正常な状態からの逸脱を導き出すガイドワードを組み合わせることでハザードを抽出する分析手法である[14]. 様々な分野に適用可能な手法として有効性が確認されているが，分析結果が対策されたことの証跡との対応づけは定義していない．また，FTA は，分析対象にとって望ましくない事象を故障木と呼ばれるゴールツリーのトップイベントに設定し，分析対象の構造などに基づいてサブイベントに分解することで，その事象を引き起こす分析対象の内部要因を特定する手法である[15]．サブイベントへの分解には，AND 分解，OR 分解を用いることができ，分解されたサブイベント間の関係性を表現することが可能である．ただし，HAZOP 同様に分析結果の対策に関する証跡との対応づけは定義していない．D-Case は，GSN（Goal Structuring Notation）[16]を拡張した記法であり，トップゴールに記述した主張を戦略ノードの記述内容に従ってサブゴールに AND 分解し，さらにその過程で用いた基準等の情報を前提ノードで表現しながら，最下位となるサブゴールの妥当性を証跡と関連付けたエビデンスノードで説明する，という流れでゴールツリー形式のアシュアランスケースを記述する．また，システム運用時の状況変化に対応することを想定したモニターノードなどを有している点が特徴である．その他，ノードの接続関係の自由度が高い GSN の表記規則を部分的に制約することで，第三者の誤解釈を避ける規則が実装されている[17]．表 2-2 に上述した HAZOP，FTA，D-Case の比較表を示しておく．

表 2-2 比較表：HAZOP, FTA, D-Case

名称	利用工程	視点	証跡との対応付け
HAZOP	分析	外部	不可
FTA	分析	内部	不可
D-Case	確認	外部	可

分析手法である HAZOP, FTA との組み合わせに関する研究としては, 文献[18], [19]がある．これらの文献では安全分析における HAZOP，FTA の位置付けと他手法との組み合わせについて述べているが，D-Case との関係については述べていない．一方，D-Case との組み合わせについては，文献[20]で HAZOP との関

係を述べている．この手法では，シーケンス図に基づく D-Case の作成法を提案し，物品購入のシーケンス図に対して HAZOP によるリスク分析に基づいて D-Case が作成できることを明らかにしている．しかし，FTA との関係については考慮していないので，システムのアーキテクチャに対する内部リスクを考慮できていない点と対策の網羅性についての確認が十分でない点に課題がある．文献[21], [22]で FTA を証拠に用いた D-Case の構造が示されているが，具体的な FTA の分析結果との組み合わせについては述べられていない．さらに，文献[23]では規格化されている知識モデル群との組み合わせ，文献[24]では SysML をはじめとする設計手法との組み合わせが述べられているが，HAZOP，FTA との関係については述べていない．

安全分析結果に基づいて D-Case を作成する手法に関する研究としては，説明構造[25-27]や説明の分解[28]に焦点を当てたパターンについて研究されているが, D-Case の作成方法については述べられていない. 文献[29], [30]では D-Case の作成手順に関して述べているが，HAZOP，FTA との関係は述べていない．また，文献[31]においてモデル情報に基づいた D-Case の統一的な作成方法が提案されているが，HAZOP，FTA の分析結果を事例とした適用評価は行われていない．

自動車業界が必要とする安全分析環境としては，システムの利用者が期待する正常な動作の逸脱状態をハザードとして抽出し，その発生につながるシステム内部の要因の特定と対策を立案し，さらにその過程の妥当性と対策状況を第三者に説明できる必要がある．上記課題は，HAZOP，FTA，D-Case の組み合わせで解決が期待できるが，HAZOP，FTA を用いた分析時にどのような情報の記録が必要で，D-Case においてどのように利用されるべきかが明らかにされていない．本研究では，上述した 3 つの手法の組み合わせ方について 3 章で説明していく．

2.3. 対立問題解消法に関連する研究

ゴール指向要求定義手法を拡張して，定量的な属性を付与した手法はいくつか存在している．表 2-3 に定量的な属性付きゴール指向要求定義手法をゴールグラフ，属性，目的，手法の視点で比較した結果を示す．

表 2-3 属性付きゴール指向定義手法

Approach	Goal graph	Attribute	Purpose	Method
QA-NFR	NFR SIG	satisfice, contribution, criticality metric	Architecture evaluation	satisfice propagation
SGW	NFR SIG	Decomposition, contribution, achievement weight	Architecture evaluation, conflict resolution	Weight propagation, Tabular calculation
AGORA	Goal graph	Preference, contribution	Conflict recognition	attribute expressions for calculating the values
FBCM	Goal tree	Correlation coefficient KPI	Goal dependency management	statistical analysis of correlation coefficient for KPI values
IGEPM	Goal tree	Contribution, validation, achievement KPI	Dynamic goal graph improvement for business change	Goal selection based on KPI based business attributes
GQM	GQM tree	KPI	Quantitative goal satisfaction condition	Metric based Question evaluation

QA-NFR (Quantitative Assessment using NFR approach) [32]と SGW (Soft goal using weight) [33] は，NFR フレームワークの SIG (Soft goal Interdependency Graph)を用いて定量的にソフトゴールを評価する．QA-NFR は，Subramanian らによって提案されたアーキテクチャに対するセーフティ要求とセキュリティ要求の定量的な分析手法である．セーフティ要求とセキュリティ要求のそれぞれにラベルを定義することで，アーキテクチャを定量的に評価することが可能となる．SIG は，セキュリティとセーフティに関するソフトゴールの定義に利用され，SIG 作成に用いるソフトゴールは，NFR ソフトゴール (Non-functional requirements Softgoals)，操作ソフトゴール (Operationalization Softgoal)，理由ソフトゴール (Claim Softgoal)で構成される．対象システムに要求される品質特性は，NFR ソフトゴールで明文化され，下位のソフトゴールに分解することで詳細化される．操作ソフトゴールは NFR ソフトゴールを満足化 (Satisfice) するアーキテクチャを定義し，NFR ソフトゴールと対応付けて貢献関係を定義する．また，SIG における意思決定や相互依存関係の根拠は，理由ソフトゴールによって表現される．さらに，操作ソフトゴー

ルが親となるNFRソフトゴールを満足化する度合いをラベル値の伝搬規則を用いて数値化することで，対象システムに採用したアーキテクチャの妥当性を定量的に確認することができる. 伝搬規則は, SIG 上に定義された満足度, 貢献度, 臨界度を用いて，操作ソフトゴールの属性を SIG 上のソフトゴールに伝搬する方法を定めている. 満足度はソフトゴールの達成状況, 貢献度は上位のソフトゴールに対する下位のソフトゴールの貢献状況，さらに臨界度はソフトゴールの臨界状況の定義に使用される．仮に，NFR ソフトゴール間で衝突が発生した場合，定量化されたアーキテクチャの評価値を判断根拠として解決を図るべきである.

しかしながら, QA-NFR はソフトゴールの分解時に下位のソフトゴール間の定量的な関係について考慮していない．この課題に対して，山本は SGW (Softgoal Weight)を提案している[33]．本手法は，分解の属性として下位のソフトゴール間の優先度を定義することで，セーフティとセキュリティのような相互関係のあるソフトゴールのトレードオフを考慮することができる. しかしながら, SGW は重みの伝搬を計算する表形式を提案しているが，その表を用いた計算方法を明確に定義していない.

AGORA (Attributed Goal-Oriented Requirements Analysis) は，ステークホルダ間の要求の衝突を明らかにするために，ゴールに対する属性として優先度と貢献度を定義する[34]．また，評価値を計算する属性の計算式が提供されている．貢献度は上述した NFR フレームワークと似ているが，AGORA はソフトウェアアーキテクチャの評価について考慮していない.

FBCM (Fact Based Collaboration Modelling) [35]は，KPI 値に基づいてゴール間の相関係数を統計的に分析することによって目標となるゴールツリーを見直す方法を提案している．FBCM はゴール分割に対して統計的な証拠を提供することができる.

IGEPM (Incremental Goal Evolution Process Methodology) [36]は，ビジネス環境の変化に基づいてゴールグラフを継続的に改善する方法を提案している. IGEPM は，ビジネスプロセスの実際の活動から観察すべき KPI 値を収集し，その属性は次のように定義される.

- Cnt (y, x)は，親ゴール x に対する子ゴール y の貢献値を意味する.
- Vld (x)は，ルートゴールからパス x に対する最小の貢献値を意味する.

・ Acv (x, c)は，ゴール x の KPI 値 c が達成している場合を 1 とし，それ以外の場合を 0 とする．

IGEPM は，上記属性を用いた定量的な実績に基づいて，適切なタスクゴールを選択する方法を提供する．

GQM [37]はゴールの達成度を評価するための指標を特定する方法である．GQM はゴール層，クエスチョン層，メトリクス層から構成される．GQM はソフトゴールの分割を考慮していないため，ゴール間で依存関係のある属性を扱うことができない．

FTA は，望ましくない事象をゴールツリーの最上位に定義し，それを下位のゴールに詳細化していくことで，その発生要因を抽出する．FTA では事象の発生確率を定義することができるが，それ以外のゴールの属性は明確に定義していない．GSN は，トップゴールに主張する命題を定義し，その命題の前提となる事実に基づいて分割された下位のサブゴールとその確認記録によって命題の妥当性を説明する．しかしながら，FTA 同様にゴールの属性については明確に定義していない．

OCTAVE と ATAM はアーキテクチャ品質を評価するために提案されている．OCTAVE (The operationally critical threat, asset, and vulnerability evaluation) [38]は，脆弱性を評価するためのチェックリストを提供している．しかしながら，OCTAVE は，セーフティ要求とセキュリティ要求の間に存在する衝突問題の解決について考慮していない．ATAM (Architecture Trade off Analysis Method) [39] は，ユーティリティーツリーを使用してアーキテクチャに大きな影響を与える重要な要因を記述するシナリオベースの方法を提供する．ATAM は，セーフティ要求とセキュリティ要求を分析するために品質のトレードオフを分析する方法を提供しているが，ユーティリティツリーにおいて属性を扱っていない．

2.4. 車載システム向け参照モデル活用法に関連する研究

自動車業界では，車載システム開発活動で作成される開発成果の記述言語として EAST-ADL，AUTOSAR が標準化されている[40-42]．これらの標準規格は，主に欧州企業で実際の開発プロジェクトに適用されており，EAST-ADL は車載システム開発活動における全ての設計要素とその関係を標準化している．一方，

AUTOSAR は，全ての車載ソフトウェアが必要とする共通機能を提供するソフトウェアプラットフォーム，製品依存の要求を実現するソフトウェアコンポーネントの規格，およびそれらを利用した開発を支えるプロセスとツールチェーンに関して標準化を進めている[43-47]．しかしながら，標準化された仕様書は膨大な数が存在し，記述内容も複雑である．このため，これらの仕様書から標準ソフトウェア資産の評価基準に相当する可変点に関連した情報を抽出することは容易ではない．また，EAST-ADL や AUTOSAR を用いた可変点の表記法[42]や様々な設計手法[48], [49]が提案されているが， 車載システム開発活動全体を踏まえた視点で議論されていない．一方，参考文献[50-53]では，自動車企業の事例を含んだプロダクトライン開発のライフサイクルについて議論している．しかしながら，7 人の侍フレームワーク[5]のような参照モデルとの関係を考慮していないため，標準ソフトウェア資産の評価基準を抽出する方法は熟練者の経験に依存せざるを得ない．その他，参考文献[54]では，プロダクトラインの品質に関連する設計要素について議論しているが，車載システム開発への適用については考慮していない．

2.5. 品質特性に基づくアシュアランスケース作成法に関連する研究

Goal Structuring Notation (GSN)はアシュアランスケースの記述に適しており[16]，その記述に用いる標準的なパターンが提案されている[25]，[55]，[56]．しかしながら，それらはアシュアランスケースの記述に要求される具体的な手順を説明していない．また，メタモデルを用いてアシュアランスケースの記述品質を制御する手法[57]，[58]や設計成果からアシュアランスケースへ変換する手法[59]，[60]が提案されている．しかしながら，これらの手法はソフトウェアレビューへの適用について考慮していない．山本らは SPRME に基づいたアシュアランスケースの統一的な作成手法を提案[31]しているが，ソフトウェアレビューにおける有効性を確認していない．

3. D-Case を用いた安全分析結果の説明手法の提案

本章では，1.1.1 節で述べた課題①を解決する方法として，従来から用いられてきた安全分析手法と D-Case を統合した安全分析結果の説明手法を提案する．

3.1. はじめに

自動車業界では，ISO26262[61]に対応した車載ソフトウェア開発手法の確立が急務の課題となっており，安全分析の領域については，従来から利用されている HAZOP，FTA が採用される可能性が高い．従来から行われてきた安全分析は，「利用者視点でのハザード抽出」，「開発者視点でのハザード要因の抽出」，「開発者視点での対策確認」の 3 つの工程に大まかに分けることができる．一方，ISO26262 を想定した安全分析では，「利用者視点での対策確認（第三者への説明）」を加える必要があり，この工程では上述した 3 つの工程のつながりや各工程で行われた判断の根拠を可視化することが求められる．文献[62]の調査結果からも分かる通り，HAZOP，FTA は分析手法として利用実績も高く，分析手法として有用であるが，これらの手法だけでは第三者への説明時に必要となる情報を可視化することは困難である．また，従来の開発現場における安全分析では，HAZOP，FTA の分析結果，およびその対策の確認結果をアシュアランスケースとして統合する手順が明文化されておらず，さらにその記述方法も自由記述の自然言語が採用されているため，アシュアランスケースの記述品質が分析者のスキルに大きく依存していた．

本章では，「利用者視点での対策確認（第三者への説明）」に対して D-Case を採用し，上述したそれ以外の 3 つの工程で用いられる手法と対応づけたアシュアランスケースの作成方法を提案する．また，D-Case の自由度の高さに起因する記述品質の問題を，HAZOP，FTA と組み合わせることで解決する方法についても考察する．さらに，本手法を簡易的なヘッドライト制御システムに適用し，その結果に基づいた本手法の有効性について説明する．

なお，本節以降では，3.2 節で本章の提案手法について説明する．さらに，3.3 節で提案手法を適用した実験結果について述べ，3.4 節で実験結果に基づいた本手法の有効性について考察する．最後に，3.5 節でまとめと今後の課題について述べる．

3.2. D-Case を導入した安全分析手法

3.2.1. 研究仮説

車載システムに対する安全分析の結果を分析作業の当事者ではない第三者が納得するためには，安全分析を支える工程間の関係，およびそれぞれの工程の分析過程が妥当であり，さらに導出された結果がシステムに対して確実に反映されていることを確認できる必要がある．このために可視化すべき情報としては以下が挙げられる．

- 工程毎の分析結果の関係を示した全体像の情報
- 網羅的な分析結果を導くために用いた判断の情報
- システムに対する分析結果の実施状況の情報

従来から安全分析で利用されている HAZOP，FTA では，分析の網羅性の根拠を示すことができないため，分析者の用いた知識が暗黙知化し，第三者が確認できないという問題があった．また，それぞれの分析結果の関係を可視化し，安全分析全体を確認する記法，および分析結果がシステムに対して実施されたことの証跡を確認する記法も提供していない．このため，従来の手法では，HAZOP，FTA だけでは不足する上記情報を可視化するために，自由記述の自然言語文章を作成することで補っている．しかしながら，自由記述された文章は，分析結果を段階的に説明する過程で網羅性に不備が生じるなど，作成者のスキルに依って記述品質が大きく変動するため，不足する情報が必ずしも正しく補えるとは限らない．

本提案では，HAZOP，FTA に D-Case を組み合わせることで，上述した課題を解決する手法を 3.2.2 節で提案する．D-Case は，抽象度の高い主張を明文化された客観的な記述に基づいて網羅的に分解して具体化する記法であり，主張の特性や主張の妥当性を証明する根拠についても可視化が求められる．この制約に従って安全分析結果の説明内容を記述することで，自由記述の自然言語文章と比べて，説明内容の網羅性や客観性に関して安定した記述品質が期待できる．ただし，D-Case だけでは根拠の選択などに自由度が高く，現場運用するには限界がある．このため，本手法が提案するように，安全分析における利用手順が明文化された HAZOP，FTA と組み合わせることで，この課題の解消も期待

できる．

なお，従来手法と提案手法の相違点を表 3-1 に示す．分析工程ではどちらも HAZOP，FTA が採用されているが，従来手法では HAZOP，FTA の利用手順が現場の分析者に依存しており，確認工程で必要となる情報が HAZOP，FTA を用いた分析工程で考慮されていない可能性がある．一方，提案手法では，D-Case 作成時に必要となる情報が明文化されており，HAZOP，FTA，対策確認表の作成過程で必要な情報を確実に記録することができる．

上述した従来手法の問題点が提案手法によって解決できることを，以下の研究仮説として設定し，3.4 節にて研究仮説の達成状況を実験結果に基づいて確認する．

- 研究仮説 1：提案手法が従来手法よりも，安全分析全体を確認する上で有効である．
- 研究仮説 2：提案手法が従来手法よりも，分析過程で用いた判断根拠を確認する上で有効である．
- 研究仮説 3：提案手法が従来手法よりも，ハザード対策の組み込み状況を確認する上で有効である．
- 研究仮説 4：提案手法は，D-Case の記述品質に関する問題を解消する上で有効である．

3.2.2. 提案手法

第三者への説明責任が求められる ISO26262 への対応を想定した安全分析では，3.1 節で述べた 4 つの工程とそれぞれの工程で用いる手法の標準化が不可欠となる．本節では，それぞれの工程に対して表 3-1 の提案手法を割り当てた図 3-1 に示す安全分析手法の構成と分析手順について説明する．本提案は，安全分析工程を「分析と確認」,「外部と内部」の視点で分割した上で，それぞれの工程の特性に合わせて最適な手法を割り当てている．このように,「分析と確認」,「外部と内部」という 2 つの視点で，HAZOP，FTA，対策確認表，D-Case を組み合わせた統合手法が本提案の特徴である．各工程の詳細について次節以降で説明する．

表 3-1 安全分析工程と採用手法

工程	従来手法	提案手法
利用者視点でのハザード抽出 （外部分析）	HAZOP	HAZOP
開発者視点でのハザード要因の抽出 （内部分析）	FTA	FTA
開発者視点での対策確認 （内部確認）	自然言語文章 （自由記述）	対策確認表
利用者視点での対策確認 （外部確認）	自然言語文章 （自由記述）	D-Case

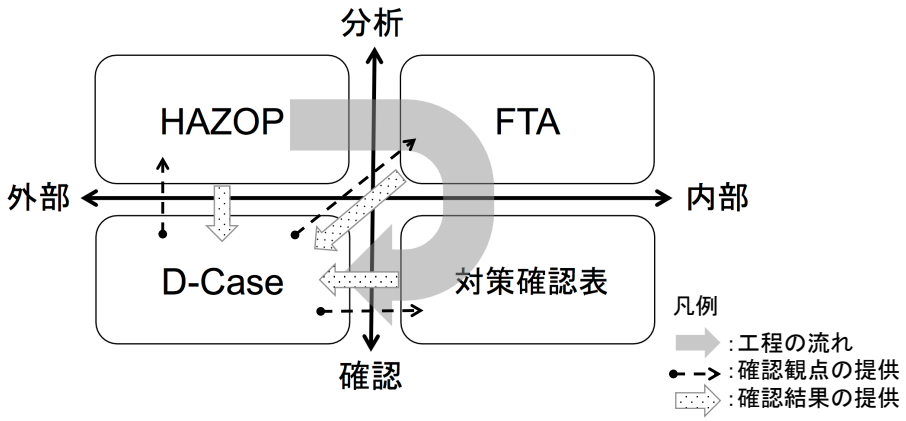


図 3-1 提案手法の構成

3.2.2.1. 利用者視点でのハザード抽出

本工程では，安全分析の対象となるシステムが外部環境に及ぼす可能性のあるハザードを抽出する．ハザードはシステムと外部環境の間に位置するパラメータの逸脱状態として抽出する．具体的な手順を以下に記す．

(1) システムと外部環境の間に存在するパラメータを明らかにする
(2) HAZOP 分析に用いるガイドワードを IEC61882 の定義から選定する（ない，多い，少ない等）[63], [64]
(3) パラメータとガイドワードの組み合わせからパラメータの逸脱状態を抽出する

3.2.2.2. 開発者視点でのハザード要因の抽出

本工程では，3.2.2.1 節で抽出したハザードを引き起こす要因を抽出する．ハザードを引き起こす要因は，FTA を用いて以下の手順で抽出する．

(1) FTA のトップイベントに 3.2.2.1 節で抽出したハザードを設定する
(2) 安全分析の対象となるシステムに関する設計情報（システム構成など）に基づいてサブイベントに分解する
(3) 分解に用いる設計情報がなくなるまで分解を繰り返す
(4) 最下位のサブイベントに対して想定される故障モード要因を明らかにする
(5) 最下位のサブイベントと故障モード要因の組み合わせからハザード要因を明らかにする

3.2.2.3. 開発者視点での対策確認

本工程では，3.2.2.2 節で抽出したハザード要因を解決するための対策を定義する．対策は与えられている設計情報に基づき，その要因と関連付けられているハードウェア，ソフトウェアを明らかにした上で，最適な箇所での実装方法を表形式でまとめる（本表を対策確認表と呼ぶ）．

3.2.2.4. 利用者視点での対策確認

3.2.2.3 節までの工程で作成した分析結果に基づき，第三者への説明に用いるアシュアランスケースを D-Case 形式で作成する．

(1) トップゴールに「対象となるシステムは安全である」を設定する．
(2) トップゴールの主張を 3.2.2.1 節で行った HAZOP 分析結果に基づいて分解する．分析結果を導出した根拠として HAZOP 分析に用いたパラメータとガイドワードを関連付ける．
(3) ハザード対策の安全性を 3.2.2.2 節で行った FTA 分析結果に基づいてハザード要因に至るまで分解する．分解の根拠として，FTA 分析で分解に用いたシステム構成と故障モード要因を関連付ける．
(4) ハザード要因への対策の安全性を 3.2.2.3 節で作成した対策確認表に基づいて分解する．対策箇所の網羅性の根拠として，ECU ソフトウェア構成を

関連付ける.

(5) 最下位のゴールに記された対策内容が実際のシステムに組み込まれていることを確認できる証拠を紐づける.

3.3. 手法の適用実験

本提案では，被験者に与えた情報の形式の差が理解に与える影響を比較するために，従来手法，提案手法それぞれを用いて作成したアシュアランスケースの比較実験について述べる.

3.3.1. 実験対象

本節では，実験対象となるヘッドライト制御システムについて定義する.

3.3.1.1. ヘッドライト制御システム

ヘッドライト制御システムが提供する機能を表 3-2 に示す．また，ヘッドライト制御システムの構成を図 3-2 に示す．図中の凡例に示されている通り，ヘッドライト制御システムは大きく分けてセンサ，ECU（Electronic Control Unit の略称)，アクチュエータの 3 分類で構成されており，外界の情報をセンサで読み取り，その情報に基づいて ECU がアクチュエータの制御方法を計算し，その結果に従ってアクチュエータが外界に作用する，という流れで前述の機能が実現される．なお，本実験における分析範囲を限定するため，本システムの前提条件として，センサ，アクチュエータは故障することはないものとする.

表 3-2 機能一覧

ID	機能
1	ユーザの ON 操作でヘッドライトを点灯する
2	ユーザの OFF 操作でヘッドライトを消灯する

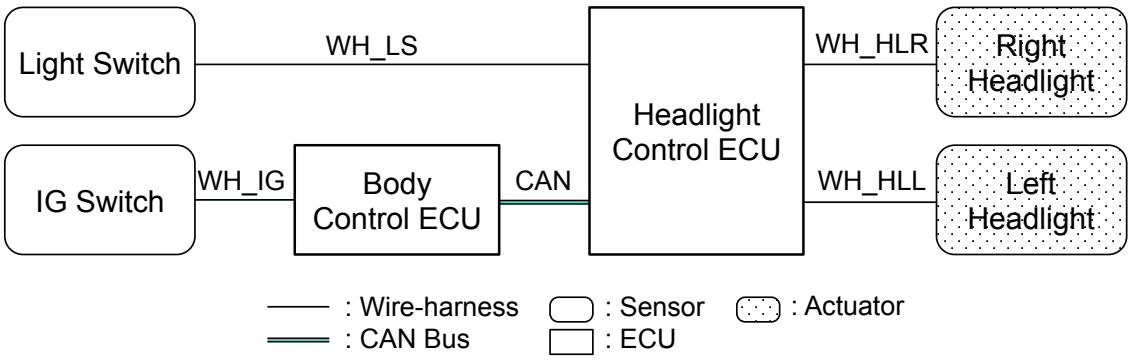


図 3-2 システム構成

3.3.1.2. ヘッドライト制御 ECU ソフトウェア

図 3-3 にヘッドライト制御 ECU のソフトウェア構成を示す．ボデー制御 ECU もアプリケーション層のコンポーネントが異なるのみで同様の構成を有するものとする．なお，本実験では，システムレベルの安全分析に焦点を置くため，ECU のハードウェア構成要素の定義は省略し，ハザード対策を組み込むソフトウェア構成要素のみ定義する．

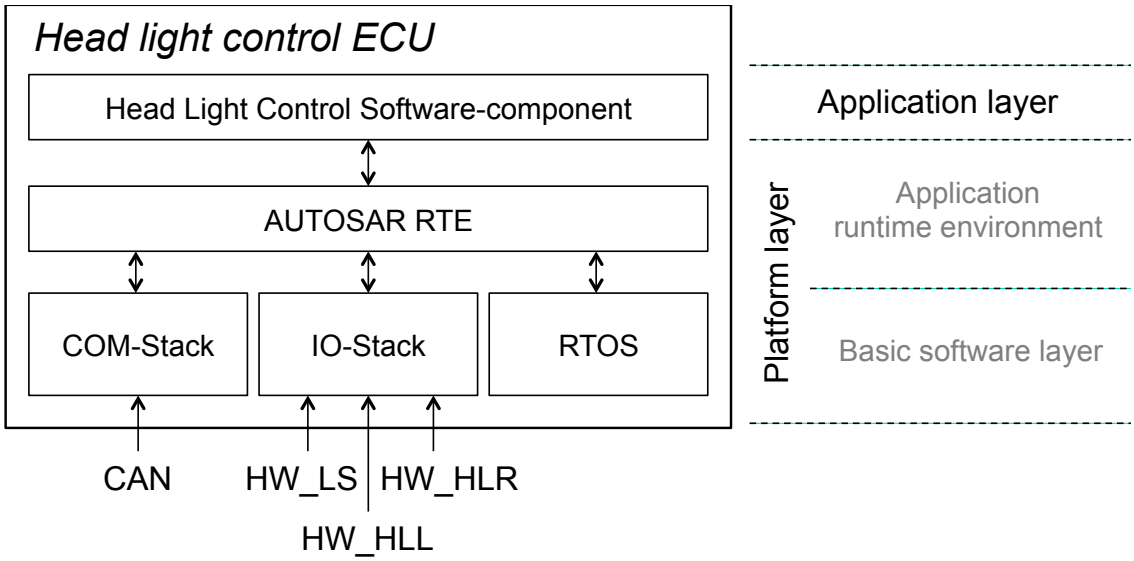


図 3-3 ソフトウェア構成

3.3.2. 実験内容

比較対象となるアシュアランスケースは，ともに HAZOP，FTA，D-Case を利用した経験のある同一のエンジニアが作成した結果であり，従来手法が 3.3.3 節の全文，提案手法が図 3-4 に対応する．従来手法における開発者視点での対策確認（内部確認）の結果は，表 3-1 に示した通り自然言語文書となるが，本実験では被験者が FTA の分析結果との対応づけを確認し易くするために，対策確認表を採用している．実験手順としては，第三者がアシュアランスケースの妥当性を判断する際に，その根拠として確認が必要となる内容を表 3-3 に示す質問形式とし，その回答時間と正答率を計測する．なお，3 年以上のソフトウェア開発経験を有した技術者を対象として，従来手法を先に実施するグループ（被験者 A1～A5）と提案手法を先に実施するグループ（被験者 B1～B5）に分けて実験を行う．

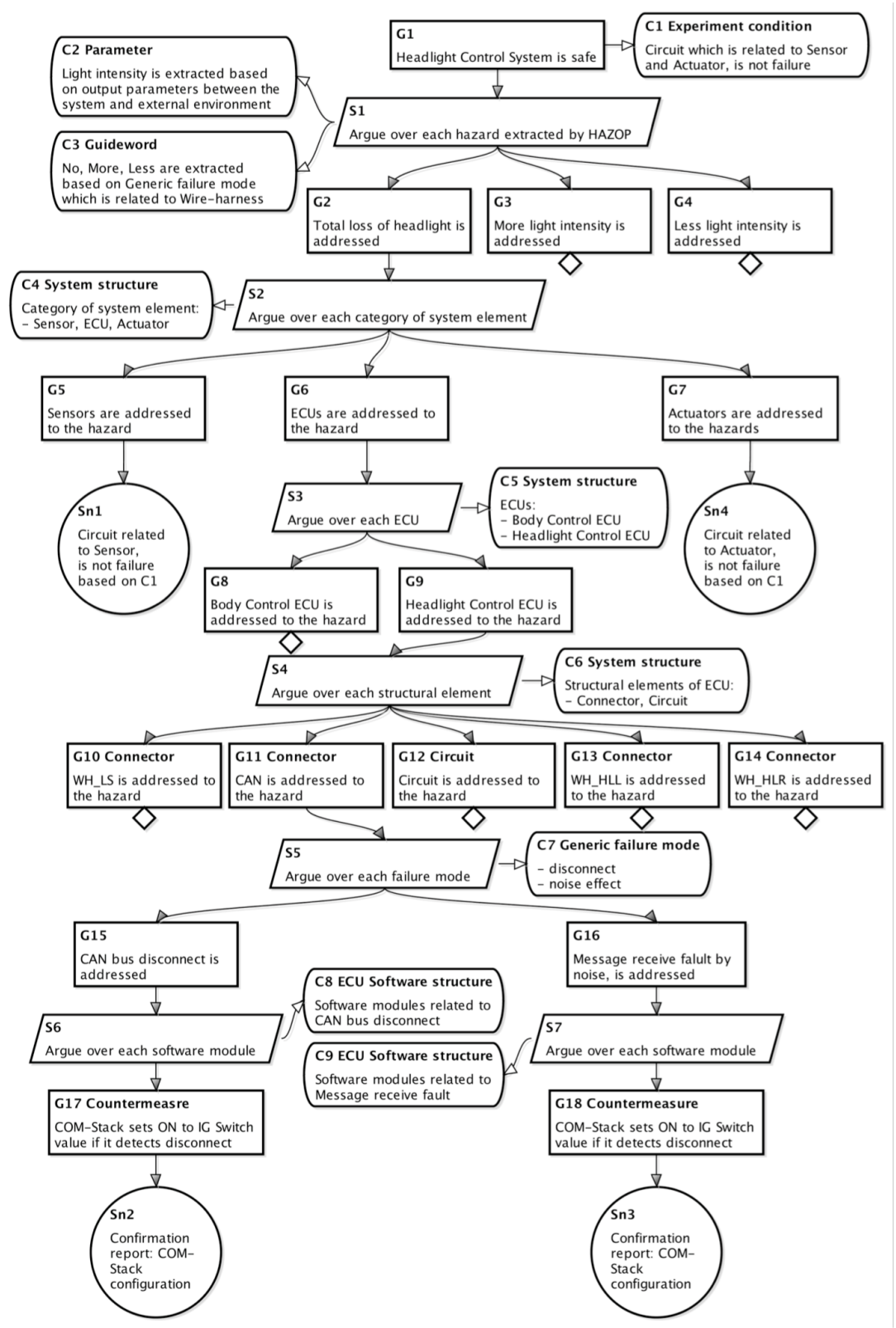


図 3-4 D-Case を用いたアシュアランスケース

表 3-3 質問内容

ID	質問内容
Q1	HAZOP 分析の対象となるパラメータを抽出する際に用いた設計情報の記述箇所を述べよ.
Q2	HAZOP 分析のガードワードを選定する際に用いた設計情報の記述箇所を述べよ.
Q3	FTA 分析のトップイベントの分解が ECU に対してのみ行われている理由の記述箇所を述べよ.
Q4	FTA 分析の最下位イベントに相当するハザード要因を抽出する際に用いた設計情報の記述箇所を述べよ.
Q5	ハザード要因への対策箇所を特定する際に用いた設計情報の記述箇所を述べよ.

3.3.3. 従来手法を用いたアシュアランスケース

本書では，ヘッドライト制御システムの安全性分析結果を説明する．システムの詳細は 3.3.1 節を参照されたい．

3.3.3.1. ハザード抽出

本実験では，HAZOP 分析の対象とするパラメータとして，システムと外部環境（他車，歩行者等）の間に存在するパラメータであるヘッドライトの光量を採用する．また，ガイドワードには No, More, Less を選定した．上記 2 つの判断結果と、その組み合わせに基づいて抽出されたハザード一覧の記録を表 3-4 に示す．なお，本実験では ID.1 のみを対象として以降の実験を進める．

表 3-4 HAZOP 分析結果

ID	Parameter	Guideword	Hazard
1	Light intensity	No	Total loss of headlight
2		More	More light intensity
3		Less	Less light intensity

3.3.3.2. ハザード要因の抽出

3.3.3.1 節で抽出した ID.1 のハザード「Total loss of headlight」が設定されたトップイベントを図 3-2 のシステム構成で定義された設計情報に従って分解した．その上で最下位のイベントに対して，表 3-5 に示した故障モード要因を組み合わせることで末端の基本イベントとしてハザード要因を抽出した（図 3-5 参照）．

なお，実際の開発現場では，故障モード要因には分析時の知見として資産化されているものが再利用されるが，本実験では簡易化のため表 3-5 の内容に留めている．知見の数は異なるが，分析手順は同じであり，実験の本質に影響はない．

表 3-5 故障モード要因

Structural element	Generic failure mode
Wire-harness	disconnect
	noise effect
ECU	failure

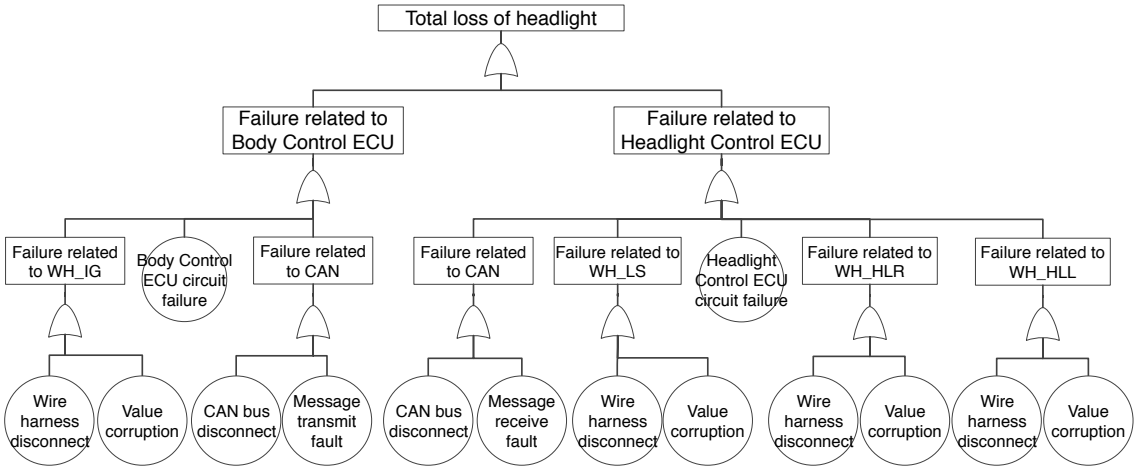


図 3-5 FTA 分析結果

3.3.3.3. 対策確認

3.3.3.2 節で導き出したハザード要因を解消するための対策を表 3-6 の内容に従って定義した．

表 3-6 対策確認表

ID	Device	Failure mode	Countermeasure
1	Body Control ECU	WH_IG: Wire harness disconnect	Circuit sets ON to IG Switch value
2		WH_IG: Value corruption	IO-Stack provides preventing the chattering
3		Body Control ECU hardware breakdown	Headlight Control ECU monitors and addresses
4		CAN: CAN bus disconnect	Same as above
5		CAN: Message transmission fault	Same as above
6	Headlight Control ECU	CAN: CAN bus disconnect	COM-Stack sets ON to IG Switch value
7		CAN: Message receive fault	Same as above
8		WH_LS: Wire harness disconnect	IO-Stack sets ON to Light Switch value
9		WH_LS: Value corruption	IO-Stack provides preventing the chattering
10		Headlight Control ECU hardware breakdown	Body Control ECU monitors and indicates the caution
11		WH_HLR: Wire harness disconnect	Circuit addresses in Right Headlight
12		WH_HLR: Value corruption	Circuit provides preventing the chattering in Right Headlight
13		WH_HLL: Wire harness disconnect	Circuit addresses in Left Headlight
14		WH_HLL: Value corruption	Circuit provides preventing the chattering in Left Headlight

3.3.4. 提案手法を用いたアシュアランスケース

提案手法を用いたアシュアランスケースを図 3-4 に示す．トップゴールは安全分析の手順に沿って以下の順で分解している．文中のラベルは図 3-4 のノードラベルに対応している．

(1) HAZOP 分析の結果に基づきトップゴールを分解（G2, G3, G4）
(2) システム構成：機器分類（センサ，ECU，アクチュエータ）に基づき分解（G5, G6, G7）
(3) システム構成：機器（ボデー制御 ECU，ヘッドライト制御 ECU）に基づき分解（G8, G9）
(4) システム構成：機器の構成要素（コネクタ，電気回路）に基づき分解（G10, G11, G12, G13, G14）
(5) 故障モード要因（ハーネス：断線，ノイズ，電気回路：故障）に基づき分解（G15, G16）
(6) ECU ソフトウェア構成に基づき分解（故障モードに関連するソフトウェアモジュールのみ）（G17, G18）

なお，本実験ではヘッドライト制御 ECU の CAN bus に関する対策の説明のみ全て掲載し，それ以外は省略している（該当箇所は D-Case の Undeveloped に相当）．

3.3.5. 実験結果

3.3.2 節に従って比較実験を行った結果を図 3-6，図 3-7 に示す．結果詳細は表 3-7 を参照されたい．

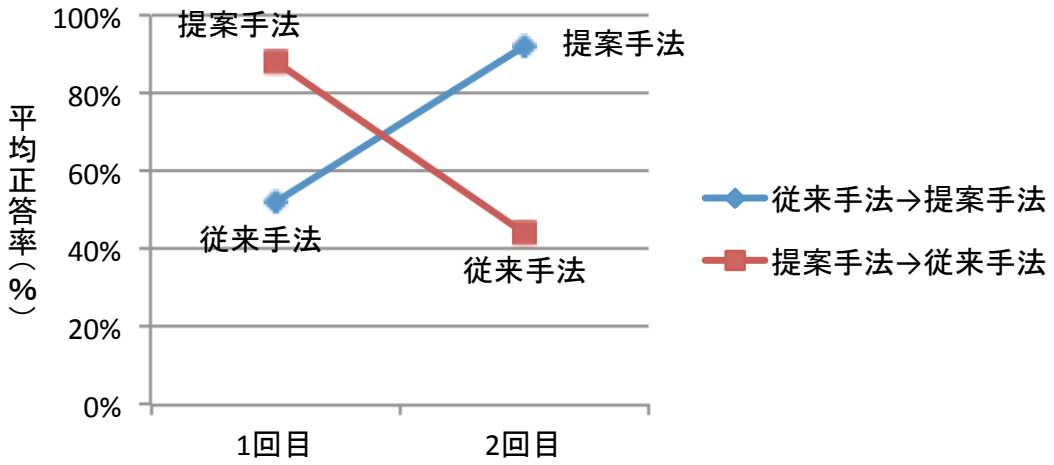


図 3-6 従来手法と提案手法の比較：正答率

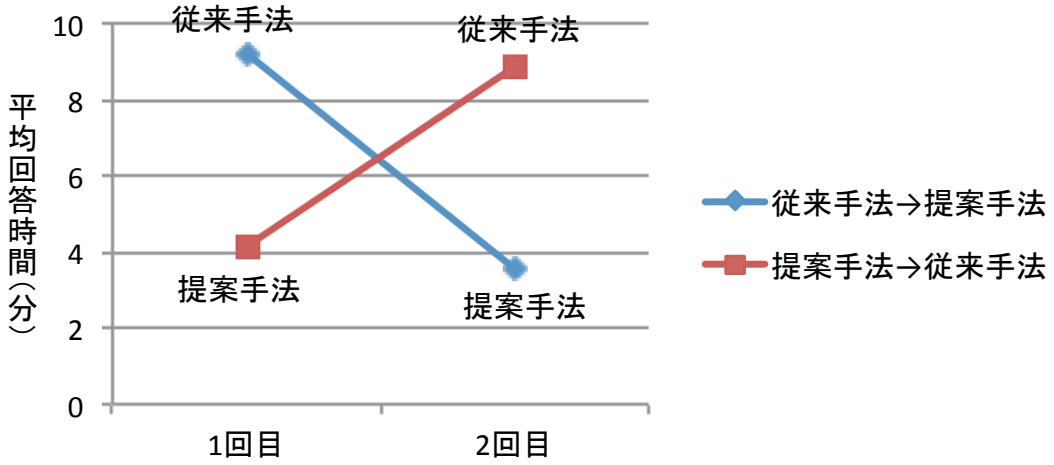


図 3-7 従来手法と提案手法の比較：回答時間

表 3-7 従来手法と提案手法の比較結果（上段：従来手法，下段：提案手法）

被験者		従来手法						
		回答時間	正答率	正答率内訳				
				Q1	Q2	Q3	Q4	Q5
従来手法→提案手法	A1	8m30s	40%	○	×	×	○	×
	A2	16m00s	60%	○	×	○	×	○
	A3	4m00s	60%	○	×	○	○	×
	A4	7m30s	60%	○	×	○	○	×
	A5	10m00s	40%	×	×	○	○	×
	平均	9m12s	52%	80%	0%	80%	80%	20%
提案手法→従来手法	B1	13m30s	40%	○	×	○	×	×
	B2	5m00s	40%	×	×	○	○	×
	B3	7m30s	60%	○	×	○	×	○
	B4	8m00s	40%	○	×	○	×	×
	B5	10m10s	40%	○	×	○	×	×
	平均	8m48s	44%	80%	0%	100%	20%	20%

被験者		提案手法						
		回答時間	正答率	正答率内訳				
				Q1	Q2	Q3	Q4	Q5
従来手法→提案手法	A1	4m25s	80%	○	○	×	○	○
	A2	4m44s	100%	○	○	○	○	○
	A3	1m28s	100%	○	○	○	○	○
	A4	4m00s	80%	○	○	○	○	×
	A5	3m00s	100%	○	○	○	○	○
	平均	3m30s	92%	100%	100%	80%	100%	80%
提案手法→従来手法	B1	5m20s	100%	○	○	○	○	○
	B2	2m00s	100%	○	○	○	○	○
	B3	4m30s	80%	○	○	○	○	×
	B4	4m00s	80%	○	○	○	×	○
	B5	5m00s	80%	○	○	○	×	○
	平均	4m12s	88%	100%	100%	100%	60%	80%

3.4. 考察

3.4.1. 研究仮説 1 の実証

従来手法では, HAZOP, FTA それぞれ単独での利用方法は定義されていたが, その分析結果の統合手順は定められておらず，自由記述の自然言語でアシュアランスケースを記述する分析者に依存した記述品質となっていた. 一方, 提案手法では，HAZOP，FTA の分析結果だけでなく，その分析過程で用いられた判断基準を含めて D-Case 上で可視化すべき情報を明らかにし, それらの情報の統合方法を D-Case の作成手順として定義している. この結果，それぞれの分析工程が適切に履行されていることの確認を含めて，安全分析全体を示すことが可能となる. また，本手法を用いることで，分析に用いた手順や判断基準が明文化できるため，それらを再利用可能な知識として蓄積する効果も期待できる. 表 3-8 に示した被験者の意見からも，説明の流れや構造が可視化され (O1, O2)，かつ説明の単位も適切に部品化されたことで (O3, O4)，従来手法と比べて理解が容易になっていることが分かる. 本結果から研究仮説 1 は実証できたと言える.

表 3-8 D-Case に対する被験者の意見

ID	被験者の意見
O1	上から順に読めば分析結果を順序立てて理解できるのが良い
O2	確認時に注目する必要のある領域が絞られるので読みやすい
O3	説明が適切に部品化されるため全体理解が容易である
O4	ノード内の文章が長文でないので理解しやすい
O5	判断の根拠が前提ノードにまとまっているので見つけ易い

3.4.2. 研究仮説 2 の実証

HAZOP，FTA の分析過程で用いた判断の根拠は，従来手法では自由記述の自然言語文章として記録されるため，文章の品質に依っては記述箇所の特定が困難となる場合が発生する. このため, 第三者の理解に要するコストが高くなってしまう. 一方，本章で提案した D-Case をアシュアランスケースに用いた場合, 前提ノードで示した判断の根拠が確認の過程で適切に紐付けられて確認できる.

3.3.5 節の実験結果においても，従来手法の正答率が 50%程度であることに対して，本提案の正答率は 90%前後を得ている. 従来手法の質問 Q2，Q5 の正答

率が低い原因を被験者に確認したところ，分析結果の記述箇所から該当する記述箇所を参照する記述が明文化されておらず，分析結果から分析者の意図を類推して該当箇所を探す必要があるためであった．一方，提案手法で不正解のあった質問 Q3 について被験者に確認したところ，該当箇所である C1 への参照関係がノード Sn1, Sn2 の文中に記述されており，そこまで踏み込んで確認できなかったことが原因であった．また，質問 Q4，Q5 は，ハザード要因という用語が D-Case 上で明記されていないため，該当箇所を誤解釈したことが原因であった．なお，従来手法と提案手法の実施順序に依らず，提案手法の正答率が高かった理由としては，表 3-8 の被験者の意見にも挙げられている通り，質問に対する関連箇所を絞り込むことが容易であり，さらに D-Case の記法上，分析時に用いた判断に関する記述は前提ノードに記されていることが明白であるため，その特定が容易であったことが挙げられる．回答時間についても従来手法と比べて半分以下の時間で確認できており，研究仮説 2 は実証できたと言える．なお，回答時間の短縮は，表 3-8 の O5 に挙がっているように，質問内容の回答となる判断の根拠が前提ノードに集約して表現されていた効果と予想される．また，O1 に挙がっているように，D-Case は分析結果の説明が上から順に構造化されるため，質問の関連箇所の特定が容易であったことも一因と予想される．

3.4.3. 研究仮説 3 の実証

実験結果の図 3-5 が示すように，研究仮説 3 で設定したハザード対策の組み込み状況の確認は，FTA だけでは表現できない．これに対して D-Case では，最下層のゴールノードで対策を記し，その対策が組み込まれていることの確認記録を記した証拠ノードと紐付けることで，組み込み状況を可視化できている図 3-4 の G17 と Sn2，および G18 と Sn3 の関係に相当)．上記結果から，本提案手法は研究仮説 3 を実証できたと言える．

3.4.4. 研究仮説 4 の実証

D-Case の記述品質は作成者のスキルによって左右されることが分かっている [65]．この問題は，D-Case の十分な作成手順が提示されていないことに原因があり，例えば，HAZOP，FTA を D-Case のエビデンスに用いる，という曖昧な手順しか提示されていなかった．このため，安全分析に D-Case だけを導入しても，外部視点でのハザード抽出や内部視点でのハザード要因の抽出手順が不明

確な D-Case を作成する可能性が高く，開発現場での運用が困難であった．提案手法では，HAZOP，FTA を用いた分析工程の手順を定義し，その過程で用いた判断基準に基づいて D-Case を作成する手順を提案している．提案手法に従って作成した D-Case は，表 3-7 に示した通り，90%前後の正答率を得ており，高い記述品質を保つことができている．この結果から，本章の提案手法は研究仮説 4 を実証できたと言える．

3.4.5. 本提案の限界

本手法の有効性は，本章で示した事例のみに基づいて評価しており，複数の事例に基づいた統計的な検証はしていない．対象となるシステムの規模や複雑さといった条件の異なる複数の事例に適用し，本提案と同様の効果が得られることを確認する必要がある．また，本提案では作成したアシュアランスケースに関する理解度の実験のみを行っており，作成効率に関する従来手法との比較が必要である．

3.5. 結論

本章では，「分析と確認」ならびに「外部と内部」の視点で HAZOP，FTA，対策確認表，および D-Case を組み合わせた安全分析手法を提案した．本手法を採用することで，第三者の確認時に不可欠となる情報を可視化した上で安全分析結果を説明でき，さらに，簡易的なヘッドライト制御システムへの適用事例に基づいて，その有効性を確認した．また，本提案で採用した HAZOP，FTA，D-Case は，それぞれ単独では課題を抱えていたが，それらを緊密に組み合わせることで互いの課題の解消を図り，第三者が正しく理解できるアシュアランスケースの作成手順を確立することができた．今後の課題として，実際の車載システムへの適用事例を増やし，その結果に基づいて本手法の有効性を統計的に検証していく必要がある．さらに，HAZOP や FTA の分析結果を準形式化し，その記述規約に基づいて D-Case を自動生成する仕組みを考案していくといった発展が考えられる．本仕組みが実現されれば，エンジニアは自身の分析結果を第三者の視点で確認することが可能となる．

4. 非機能要求の定量評価手法の提案

本章では，1.1.3 節で述べた課題③を解決する方法として，NFR フレームワークのソフトゴールに重み付け属性を持たせることで，対立する非機能要求のトレードオフを定量的に評価する手法を提案する．

4.1. はじめに

ソフトウェア工学では，ゴールツリーに対する 3 種類の活用手法が考案されている．NFR フレームワークにおける Softgoal Interdependency Graph (SIG)は，非機能要求の構造を表現するために利用されている[18]．Fault Tree Analysis (FTA) [66]は，ルートに定義された故障をその原因に至るまで分解する故障木を提供している．Goal Structuring Notation (GSN) [67] [27]は，ルートに定義された命題を議論の前提となる情報に従ってサブゴールに分解していき，末端のサブゴールにそれを実証できる証拠を紐付けることで，最上位の命題を説明するゴールツリーを提供する．これらの手法は，ゴールに関して定性的な議論は提供しているが，定量的な議論は提供していない．例えば，NFR フレームワークの SIG は，品質要求を定義し，それが評価対象となるシステムアーキテクチャによって満足されていることを確認する際に利用されている．しかしながら，NFR フレームワークでは，特性の異なるソフトゴールは分離して評価されるため，セキュリティとセーフティに関するソフトゴールを統合的に評価することができない．つまり，セーフティとセキュリティのソフトゴールが相反する場合の衝突問題を解決することが困難である．

本章では，NFR フレームワークが提供する SIG を拡張し，ソフトゴールに重み付けした提案手法の有効性について議論する．重みの値は，ソフトゴールの分割毎に割り当てられ，ソフトゴール間の優先度を明確に定義する．提案手法では，これらの優先度を用いて特性の異なるソフトゴール間の衝突問題を解決する．本節以降の構成を以下に示す．

4.2 節では，SIG に定量的な重みを導入した提案手法について説明する．4.3 節では，車載向けプラットフォームのアーキテクチャ候補の評価に対して，本提案手法を適用した自動車業界における事例を説明する．4.4 節では，本事例に基づき，提案手法の有効性と限界について議論する．4.5 節では，本章の結論と今後の課題について説明する．

4.2. 重み付きソフトゴール

4.2.1. 基本コンセプト

重み付きソフトゴールの基本コンセプトを次のように定義する．SIG G を< g_0, S, O, D, P_w, C_w, A_w>と定義する．S を G に含まれるソフトゴールのセット，g_0 を S の中で特別な要素であるルートゴールとする．O は操作ソフトゴールのセット，D は S と O の要素の依存関係を定義する．P_w はソフトゴールの分割に対する優先度の重みを定義する．ソフトゴールを下位のソフトゴールに分割する場合，分割の重みのラベル P_w=<W_1,…,W_k>を親となるソフトゴールの名前に付け加える．k を下位のソフトゴールの数とし，W_i は $\Sigma_{i=1,\ldots,k} W_i$=1 を満足するように定義される．$C_w$ は上位と下位のソフトゴール間の貢献度の重みを定義する．SIG には肯定と否定の貢献があり，肯定と否定の貢献の重みはそれぞれ+N と-N とする．N は 1，あるいは 2 であり，1 と 2 はそれぞれ弱い貢献と強い貢献を意味する．肯定と否定の貢献は，関連線のスタイルで表現し，実線，破線がそれぞれ肯定と否定の貢献を示す．A_w はソフトゴールの達成した重みを定義する．操作ソフトゴールの達成した重みは最初から与えられている．ソフトゴール A_w(g)の重みは，次のように操作ソフトゴールの達成した重みによって計算される．なお，Child(g)は g のサブゴールのセットを表現する．

$$A_w(g) = \Sigma_{h \text{ in } Child(g)} P_w(h)\ C_w(h)\ A_w(h)$$

4.2.2. 事例

提案手法を説明するために，ID メディアの実現方式を比較した SIG を図 4-1 に示す．RFID (Radio Frequency Identification) と Bar code は，SIG の最下層の操作ソフトゴールとして定義されている．ルートのソフトゴールは，Information flexibility, Input operability, Cost を示すソフトゴールに分解されている．ソフトゴール Information flexibility は，diversity, capacity, ID reusability, modifiability を示すソフトゴールに分解され，ソフトゴール Input operability は，simultaneous reading, readability, pollution tolerance を示すソフトゴールに分解されている．RFID に対する非機能要求の達成値は，次のように計算される．

$$(1/6+1/3+1/3+1/6)/4 + (1/3+1/3+1/3)/2 + (-1)/4 = 1/4 + (1)/2 - 1/4 = \mathbf{1/2}$$

同様に Bar code に対する非機能要求の達成値は次のように計算される．

(-1/6-1/3-1/3-1/6)/4 + (-1/3-1/3-1/3)/2 + (1)/4 = -1/4 + (-1)/2 + 1/4 = **-1/2**

上述の計算値が示す通り，ID メディアを実現する方式として，RFID が Bar code よりも優れていることが分かる．

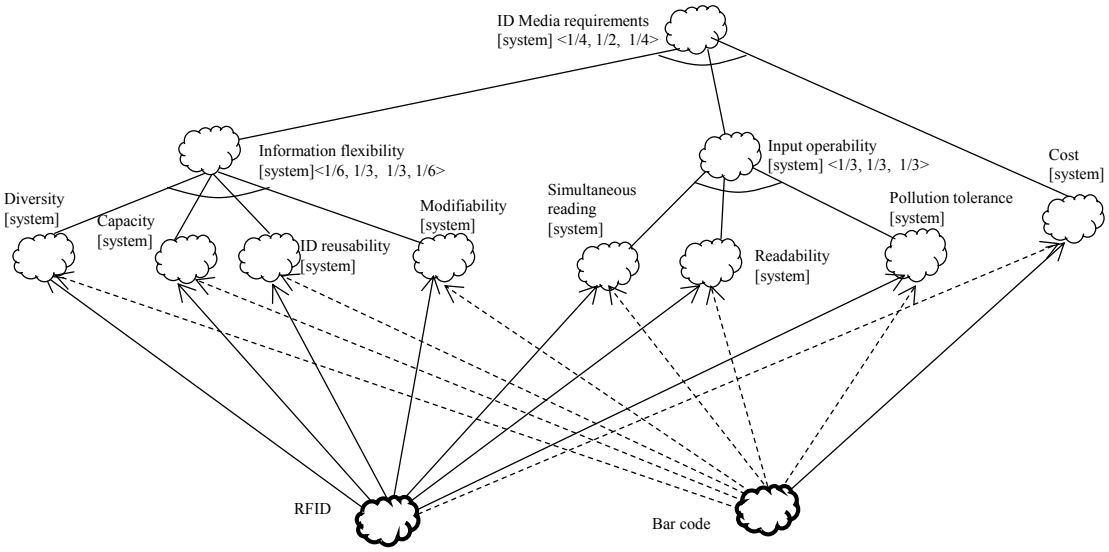


図 4-1 ID メディアの実現方式に関する評価

なお，図 4-1 における最下位の NFR ソフトゴールに貢献度の重みが割り当てられていないことが重要である．なぜなら，ソフトゴール RFID とソフトゴール Bar code は，異なる実現方式となる操作ソフトゴールであり，最下位の NFR ソフトゴールは重み値のリストを有していないからである．

4.2.3. 表形式を用いた評価値の計算方法

評価値は，表形式で計算することができる．表におけるソフトゴール g の位置を (x_g, y_g) とし，x と y はそれぞれ表の行と列の位置を示すこととする．B(g, i) をソフトゴール g の i 番目の最下位のゴール数とし，i 番目の下位のソフトゴールの位置を$(x_{(g,\ i)}, y_{(g,i)})$とする．$X_{(g,\ i)}$と $y_{(g,\ i)}$は次の式で計算することができる．

$$x_{(g,\,i)} = 1+(x_g - 1)+\Sigma_{k=1,\,i-1} B(g,\,k)*2$$
$$y_{(g,\,i)} = y_g + 2$$

最初の行は見出しを記述する．ソフトゴール g に対する貢献度の重みは(x_g +1, y_g)で表現される．ソフトゴール g に対する優先度の重みは，列 y_g + 1 に示される．操作ソフトゴールに対する貢献度の重みは表の最終列で定義される．例えば，SIG の深さを N とすれば，SIG に対する表の最終列は 2N + 1 となる．ソフトゴール g に対する貢献度の重み Wa(g)は，次の式で計算できる．

$$Wa(g) = V(x_g + 1,\, y_g) = \Sigma_{k=1,\,i} V(x_{(g,\,k)} + 1,\, y_g + 2)*V(x_{(g,k)},\, y_g + 1)$$

V(x, y)は，対応する表要素(x, y)の値であり，$V(x_{(g,\,k)}+1,\, y_g+2)$はソフトゴール g の k 番目の下位のソフトゴールに対する貢献度の重みを意味する．$V(x_{(g,\,k)},\, y_g+1)$は，ゴール g の k 番目の分解に対する優先度の重みを意味する．

表 4-1 は，図 4-1 で示した SIG に対する RFID を表形式で評価した結果である．一番左の列はトップのソフトゴールに対応する．同じ列の次の行の値は選択した方式に対する評価値を合計した結果である．2 番目の列は選択したトップゴールの分解に対する優先度の重みを示している．RFID がラベル付けされた 6 番目の列の値は，NFR ソフトゴールに対する貢献値を示している．同様に Bar code の評価結果を表 4-2 に示す．

表 4-1 表形式を用いた RFID の評価結果

SIG decomposition structure					RFID
ID Media requirements 0.50000	1/4	Information flexibility 1	1/6	Diversity 1	1
			1/3	Capacity 1	1
			1/3	ID reusability 1	1
			1/6	Modifiability 1	1
	1/2	Input operability 1	1/3	Simultaneous reading 1	1
			1/3	Readability 1	1
			1/3	Pollution tolerance 1	1
	1/4	Cost -1			-1

表 4-2 表形式を用いた Bar code の評価結果

SIG decomposition structure					Bar code
ID Media requirements -0.50000	1/4	Information flexibility -1	1/6	Diversity -1	-1
			1/3	Capacity -1	-1
			1/3	ID reusability -1	-1
			1/6	Modifiability -1	-1
	1/2	Input operability -1	1/3	Simultaneous reading -1	-1
			1/3	Readability -1	-1
			1/3	Pollution tolerance -1	-1
	1/4	Cost 1			1

4.3. ECU アーキテクチャ評価事例

AUTOSAR (Automotive Open System Architecture) [4], [68] は，ECU (Electronic Control Units) アーキテクチャの標準化を推進しており，車載アプリケーションが搭載されるソフトウェアプラットフォームの仕様も策定している．こうした背景を踏まえて，今後の ECU が採用するアーキテクチャを大別すると，図 4-2 に示すように，AUTOSAR に完全準拠したアーキテクチャ，あるいは自社独自のプラットフォームに基づいたアーキテクチャの 2 案が想定される．本節では，この 2 つのアーキテクチャの有効性を定量的に評価する．以下に評価の前提条件を示す．

- 他社から AUTOSAR 仕様に準拠した市販プラットフォームを購入できる．
- 自社独自のプラットフォームは AUTOSAR に準拠していない．
- 既存ECU ソフトウェアは自社独自のプラットフォームを用いて開発されている．
- AUTOSAR 準拠プラットフォームは自社独自のプラットフォームの機能を全て網羅できている．

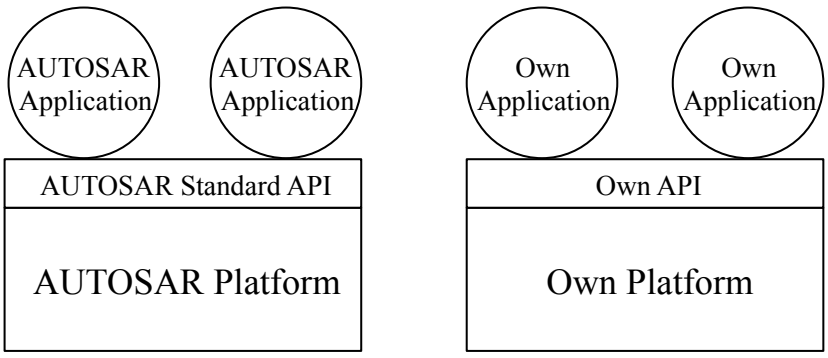


図 4-2 車載ソフトウェアに採用するアーキテクチャ案

車載ソフトウェア開発に携わるエンジニアは，様々な要求に対応できる優れたソフトウェアアーキテクチャの選択を常に迫られている．しかしながら，多くのエンジニアは最善の選択肢を定量的な証拠に基づいて選択する方法論を持っておらず，その選択は経験則に基づき定性的に判断している．このため，最

適なアーキテクチャの選択は，経験の浅いエンジニアにとって非常に難しい判断となっている．この課題を解決するために提案手法を適用した事例を以下に説明する．

図 4-3 に既存の ECU ソフトウェアをベースとして，新たな ECU ソフトウェアを開発する際のソフトウェアアーキテクチャ方式の比較結果を示す．最上位のソフトゴールは，performance, reusability, cost, modifiability に分解されている．新たな ECU ソフトウェアに採用するアーキテクチャは，様々な顧客からの要求に対応できる必要があり，アプリケーション層の開発に注力できる環境を提供する必要があるため，reusability と modifiability が重要となる．このため，NFR ソフトゴールの重み付けは<1/6, 1/3, 1/6, 1/3>とする．また，さらに下位の NFR ソフトゴールの重み付けは全て均等であるものとする．上記前提に基づき，Full AUTOSAR system に対する非機能要求の評価値は以下の式で計算できる．

$$(-1/2 - 1/2)/6 + (1/2 + 1/2)/3 + (-1/2 - 1/2)/6 + (1/3 + (1/2 + 1/2)/3 + (-1/2 + (1/2 + 1/2)/2)/3)/3$$
$$= -1/6 + 1/3 - 1/6 + (1/3 + 1/3 + 0)/3 = \mathbf{2/9}$$

一方，Full Own system に対する評価値は以下の式で計算できる．

$$(1/2 + 1/2)/6 + (-1/2 - 1/2)/3 + (1/2 + 0)/6 + (1/3 + (-1/2 + 1/2)/3 + (1/2 + (-1/2 + 1/2)/2)/3)/3$$
$$= 1/6 - 1/3 + 1/12 + (1/3 + 0 + 1/6)/3 = \mathbf{1/12}$$

上記結果から，Full AUTOSAR system が Full Own system より優れていることを定量的に判断することができる．表 4-3，表 4-4 に Full AUTOSAR system，Full Own system を表形式で評価した結果を示す．

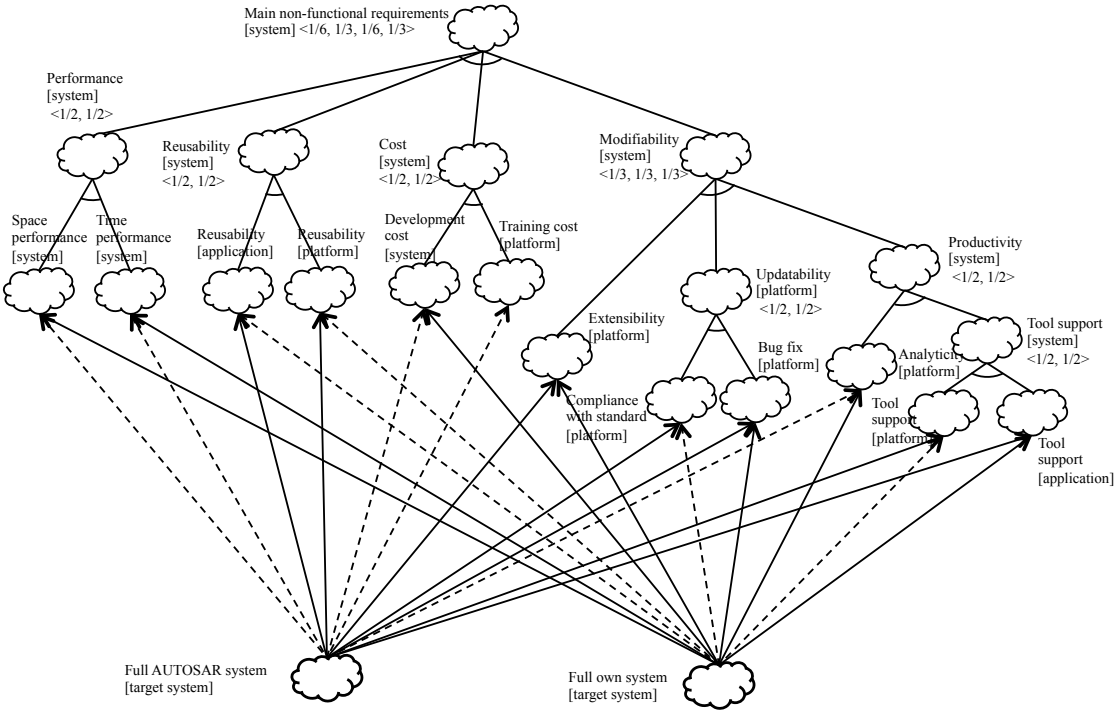


図 4-3 ECU ソフトウェアのアーキテクチャ評価

表 4-3 Full AUTOSAR system 評価結果

SIG decomposition structure									Full AUTOSAR system
Main non-functional requirement 0.22222	1/6	Performance [system] -1.00000	1/2	Space Performance [system] -1.00000					-1
			1/2	Time Performance [system] -1.00000					-1
	1/3	Reusability [system] 1.00000	1/2	Reusability [application] 1.00000					1
			1/2	Reusability [platform] 1.00000					1
	1/6	Cost [system] -1.00000	1/2	Development cost [system] -1.00000					-1
			1/2	Training cost [system] -1.00000					-1
	1/3	Modifiability [system] 0.66667	1/3	Extensibility [platform] 1.00000					1
			1/3	Updatability [platform] 1.00000	1/2	Compliant with standard [platform] 1.00000			1
					1/2	Bug fix [platform] 1.00000			1
			1/3	Productivity [system] 0.00000	1/2	Analyticity [platform] -1.00000			-1
					1/2	Tool support [system] 1.00000	1/2	Tool support [application] 1.00000	1
							1/2	Tool support [platform] 1.00000	1

表 4-4 Full Own system 評価結果

SIG decomposition structure									Full Own system
Main non-functional requirement 0.08333	1/6	Performance [system] 1.00000	1/2	Space Performance [system] 1.00000					1
			1/2	Time Performance [system] 1.00000					1
	1/3	Reusability [system] -1.00000	1/2	Reusability [application] -1.00000					-1
			1/2	Reusability [platform] -1.00000					-1
	1/6	Cost [system] 0.50000	1/2	Development cost [system] 1.00000					1
			1/2	Training cost [system] 0.00000					0
	1/3	Modifiability [system] 0.50000	1/3	Extensibility [platform] 1.00000					1
			1/3	Updatability [platform] 0.00000	1/2	Compliant with standard [platform] -1.00000			-1
					1/2	Bug fix [platform] 1.00000			1
			1/3	Productivity [system] 0.50000	1/2	Analyticity [platform] 1.00000			1
					1/2	Tool support [system] 0.00000	1/2	Tool support [application] -1.00000	-1
							1/2	Tool support [platform] 1.00000	1

4.4. 考察

4.4.1. 有効性

提案手法の有効性を確認するため, ECU ソフトウェアに対するアーキテクチャ選定の事例を用いた実験を行った. この結果が示すように, 重み付きソフトゴールは, 車載ソフトウェア分野における NFR ソフトゴールと操作ソフトゴールの妥当な関係の分析において有効である. 1 つの事例に基づいた評価結果であるが, 他の評価に適用しても同様の結果が期待できる. 例えば, セキュリティ要求とセーフティ要求の衝突問題は, 重み付きソフトゴールを使用することで定量的に解決することが可能である[33]. また, 今回の事例は, エンタープライズアーキテクチャにおけるアプリケーションアーキテクチャへの適用事例に相当するが, ビジネスアーキテクチャ, テクノロジーアーキテクチャの評価にも適用可能である[2].

4.4.2. 先行研究との違い

本手法では, 先行研究で提案された重み付きソフトゴール[33]を形式化することができた. 先の研究では, 評価を直感的に実施していたが, 本手法では操作ソフトゴールの重みに基づいてソフトゴールの重みを表形式で計算する方法を提案している. また, 車載システム開発におけるソフトウェアのアーキテクチャ選定の事例に本手法を適用した結果についても考察している.

4.4.3. 本手法の限界

本手法では, 重み付きソフトゴールの有効性を分析しているが, その有効性は車載分野の 1 つの事例に適用した場合においてのみ確認されている. 本手法の汎用性を示すために, より多くの事例で評価していく必要がある. また, 属性付きゴールを用いた定量的な評価手法を 2.3 節で比較しているが, それらを統合することで新たな重み付きソフトゴール手法の構築が期待できる. 今後の課題として, それら手法の統合について検討する必要がある.

4.5. 結論

本手法では, NFR フレームワークに対して重み付けしたソフトゴールを導入した. 本手法の評価実験により, 操作ソフトゴールで定義した実現方式の品質を定量的に判断できることを確認することができた. また, 車載ソフトウェアのアーキテクチャ選定に対する適用事例から，本手法の車載分野への適用性の可能性を示すことができた.

今後の課題として, 提案手法の適用実験を繰り返し, NFR フレームワークを定量的に評価できるよう拡張したその他の分析方法との比較実験に取り組んでいく．GSN も定量的な属性を有したゴール拡張の候補の一つである．また，その他の重み付けゴール指向手法との統合を考慮していく必要がある.

5. 7 人の侍フレームワークを用いた標準ソフトウェア資産の評価知識

本章では，1.1.4 節で述べた課題④を解決する方法として，Martin が提唱した 7 人の侍フレームワーク[5]を用いて熟練エンジニアの知見を可視化する方法，およびその知見を用いて標準ソフトウェア資産を評価する方法を提案する．

5.1. はじめに

自動車業界では，自動運転システムのような次世代システムの開発を支えるために，車載ソフトウェアの効率的な開発手法が必要とされている．プロダクトラインは生産性を改善する一つの有効な手法であるが，実際の開発プロジェクトにおいて有効なプロダクトラインを実現するためには，適切に可変点が設計された標準ソフトウェア資産を維持することが不可欠である．しかしながら，車載ソフトウェアの大規模，複雑化にともない，ソフトウェア開発のすべての工程をエンジニアが理解することは難しくなってきており，実際の開発現場では熟練したエンジニアの知識に頼らざるを得ない状況となっている．

本章では，7 人の侍フレームワークを用いて，車載ソフトウェア開発活動に関連するメタモデルを構築し，それに基づいてプロダクトラインに有用な可変点を抽出する方法を説明する．さらに，抽出した可変点を評価基準として，プロダクトラインの運用に用いる標準ソフトウェア資産の有効性を定量的に評価する方法を提案する．

5.2 節で 7 人の侍フレームワークを用いて抽出したメタモデルについて説明した上で，5.3 節で標準ソフトウェア資産の評価手法を説明する．さらに，5.4 節で提案手法の有効性を確認するために実施した適用実験について示し，その結果に基づいて，5.5 節で本提案の有効性を考察する．最後に，5.6 節で本章の結論と今後の課題について述べる．

5.2. 車載ソフトウェア開発メタモデル

本節では，7 人の侍フレームワークを開発活動の参照モデルに位置付け，標準的な車載ソフトウェア開発活動を定義したメタモデルを説明する．このメタモデルでは，Problem (P1)，Intervention system (S2)のみ以下のように事例の特

徴を記述しているが，それ以外は一般的な車載ソフトウェア開発で再利用可能な粒度で記述している．本メタモデルの全体像を図 5-1 に示す．

- Problem (P1)：
  危険な運転が事故を引き起こすことを事例として記述
- Intervention system (S2)：
  P1 で記述した危険運転の回避に用いる機能を定義

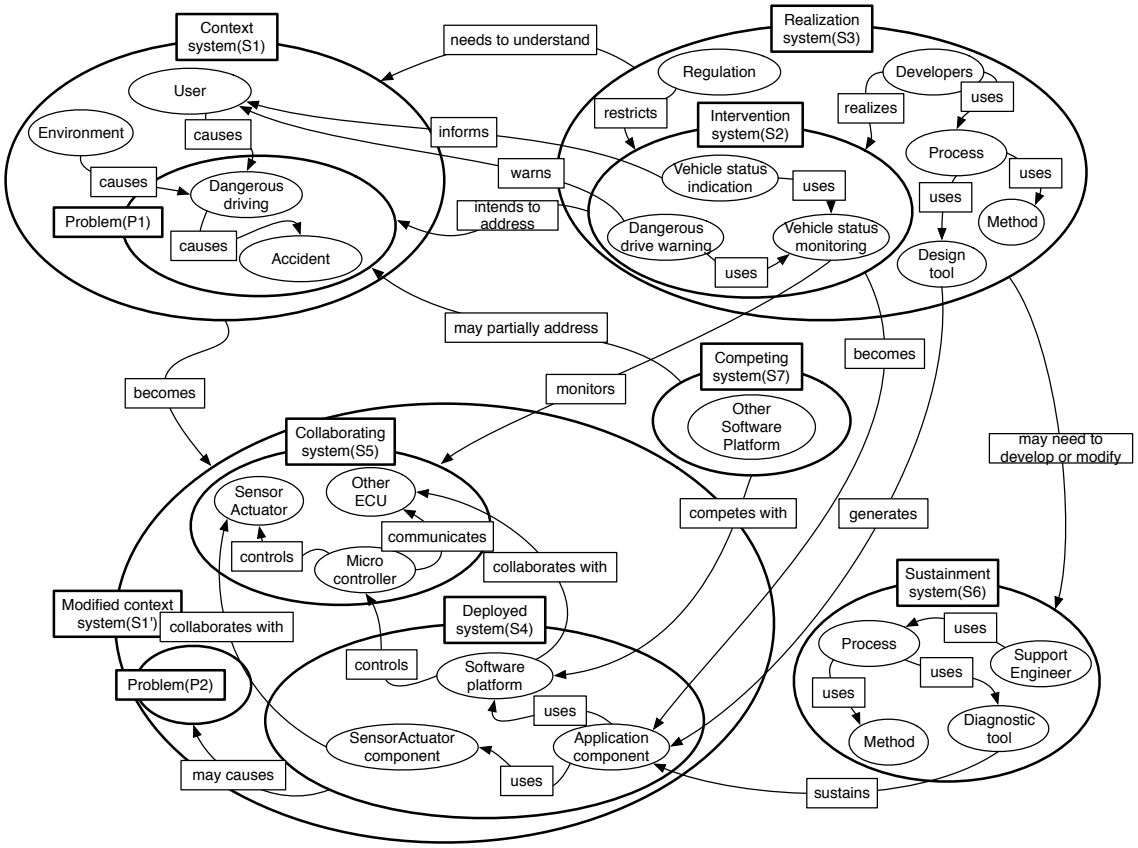


図 5-1 7 人の侍フレームワークを用いた車載ソフトウェア開発活動メタモデル

5.2.1. Context system(S1)

S1 は Problem (P1)を引き起こす原因を記述する．本事例では，車載ソフトウェアが搭載されるシステムを取り巻く環境（例：気候），およびシステムを利用するユーザを対応づけている．S1 の具体的な活用方法としては，S1 の要素をシステムに応じて具体化していくことで，プロダクトラインの可変点につながる要素を導き出すことができる．例えば，ユーザはドライバとして具体化され，さ

らにドライバが有する属性として使用する言語を連想することができる．この結果から，システムの可変点として言語の考慮が必要であることが分かる．

5.2.2. Intervention system(S2)

車載ソフトウェアに対する要求は，ユーザの利益に関連した機能要求と機能要求を実現する際に遵守すべき非機能要求に大別できる[69]. 2 種類の要求のうち，S2 では P1 を解決するために必要な機能要求を記述する．本事例では，車両状態通知，危険運転警告，車両状態監視として具体化されている．なお，機能要求を詳細化する際には，S2 と関連した要素に従って具体化を進めることができる．例えば，車両状態監視は Collaborating system (S5)を監視する関係にある．この関係を利用して，S2 の要素は，S5 の要素と関連づけてセンサ・アクチュエータ状態監視のように具体化できる. 最後に，上述した非機能要求は，後述する S3 と S5 で記述される．

5.2.3. Realization system(S3)

S3 では S2 に記述された機能要求を実現する上で考慮の必要がある開発資源と制約について定義する. 開発資源は，PMTE[70]に基づくと，開発者，プロセス，開発手法，設計支援ツールとして具体化することができる．また，制約は車載ソフトウェアが準拠すべき法規に基づいて具体化できる．さらに，非機能要求は，上述した法規のような制約と処理速度のような機能要求の属性に分類できる[69]．後者については S5 で記述する．

5.2.4. Deployed system(S4)

S4 では車載ソフトウェアの構造に関連した要素を定義する．S4 の要素は，他システムとの関係から以下のように導出できる．

- S2 に定義された機能要求を実現する「アプリケーションコンポーネント」が導出できる
- S5 に定義されたマイクロコントローラと他 ECU (Electronic Control Unit) を制御する役割を担う「ソフトウェアプラットフォーム」が導出できる
- S5 に定義されたセンサ・アクチュエータを制御する役割を担う「センサ・アクチュエータコンポーネント」が導出できる

上述した通り，他システムの要素と対応する要素で分割することで外部変化に

起因する影響を吸収できる適切なソフトウェア構造の定義が可能となる．なお，AUTOSAR では，システムを取り巻く外部環境の要素を写像する形で車載ソフトウェアの構造を定義している（図 5-2 参照）．このため，本節で説明したソフトウェア構造とも合致している．

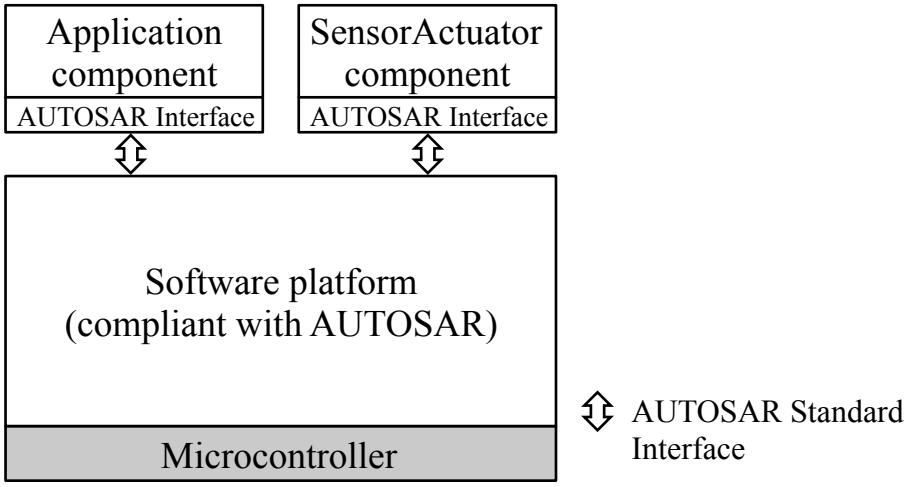


図 5-2 AUTOSAR ソフトウェア構造

5.2.5. Collaborating system(S5)

S5 では S2 を実現するために S4 によって制御される対象を記述する．具体的には，マイクロコントローラ，センサ・アクチュエータ，他 ECU で構成される．S4 はマイクロコントローラに搭載され，センサ・アクチュエータは S4 によって制御される．他 ECU は S4 の通信対象を意味する．さらに，前述した機能要求の属性に関する非機能要求は，本システムの属性として記述される．

5.2.6. Sustainment system(S6)

S6 では S4 を保守するために必要なリソースを定義する．このリソースの分類は S3 と同様に定義する．

5.2.7. Competing system(S7)

S7 では S4 に定義されたソフトウェアプラットフォームの代替を定義する．具体的には他社製ソフトウェアプラットフォームが相当する．ソフトウェアプラットフォームは，AUTOSAR に準拠して開発されており，S3 の内容に従って変

更することができる.

5.2.8. Modified context system(S1')

S1'は S4，S5，Problem(P2)で構成される．P2 は S2 を S4 として実現した際に発生した追加課題として記述される.

5.3. 標準ソフトウェア資産の評価手法

本節では，標準ソフトウェア資産の評価基準として，標準ソフトウェア資産の効果的な可変点を抽出する手法，および抽出した可変点を用いて標準ソフトウェア資産を定量的に評価する手法を説明する．なお，本節の前提として，5.2 節で定義した車載ソフトウェア開発活動メタモデルを車載ソフトウェア開発の理想形に位置付け，その構成要素の変化を最小限の影響範囲で吸収できるアーキテクチャを有した標準ソフトウェア資産が優れているものとする.

5.3.1. 評価基準の抽出

車載ソフトウェア開発活動メタモデル（図 5-1 参照）に基づいて評価基準を抽出する方法について説明する．抽出手順は以下の通りとなる.

手順 1： Deployed system (S4)の要素に関連する他システムの要素を抽出する.

手順 2： 手順 1 で抽出した要素を横軸，S4 の要素を縦軸とした表を作成する（以降，評価表と呼ぶ）.

手順 3： 評価表の縦軸，横軸の要素間に関係がある場合，該当するセルに印を付ける.

表 5-1 に上記手順に従って作成した評価表を示す.

表 5-1 評価表

Evaluation factors		Application component	Sensor Actuator component	Software platform
Intervention system (S2)	Vehicle status indication	X		
	Dangerous drive warning	X		
	Vehicle status monitoring	X		
Realization system (S3)	Design tool	X		
Collaborating system (S5)	SensorActuator		X	
	Other ECU			X
	Microcontroller			X
Sustain system (S6)	Diagnostic tool	X		
Competing system (S7)	Other software platform			X

5.3.2. 有効性の評価

5.3.1 節で作成した評価表を用いて標準ソフトウェア資産の有効性を評価する方法について説明する．本提案では有効性の評価ビューとして，重み付きソフトゴールを用いた SIG (Softgoal Interdependency Graphs)を採用する[33], [71]. SIG は，非機能要求を表現する NFR ソフトゴールをゴールツリー形式で構造化する記法であり，さらにソフトウェアアーキテクチャの候補など非機能要求の達成方式を表現する操作ソフトゴールとの関係を可視化することができる．本提案では，標準ソフトウェア資産のアーキテクチャを操作ソフトゴール，その評価指標を NFR ソフトゴールに対応づけることで，標準ソフトウェア資産の有効性を定量的に評価する．

手順 1： “Compliance for the standard software architecture”を SIG のトップゴールとして定義する．

手順 2： 評価表の列要素に従って NFR ソフトゴールを下位の NFR ソフトゴールに分割する．

手順 3： 分解した NFR ソフトゴールに対して重みを定義する．重みの合

計は 1.0 になるように定義し，分解元の NFR ソフトゴールのラベル名として SIG に記述する．

手順 4： 評価対象を操作ソフトゴールで定義し，最下位の NFR ソフトゴールと関連づける．

手順 5： 最下位の NFR ソフトゴールに相当する変化は，評価表の行方向で印が付いた列の要素のみに影響を与える．このため，評価対象のアーキテクチャが，その影響を吸収できる場合は，最下位の NFR ソフトゴールと操作ソフトゴールの関係を肯定関係(+1)とし，そうでない場合は否定関係(-1)とする．

手順 6： 以下の計算式に従って標準ソフトウェア資産の評価値 $A_w(g)$を計算する．

$$A_w(g) = \Sigma_{h \text{ in } Child(g)} P_w(h) * C_w(h) * A_w(h)$$

5.4. 標準ソフトウェア資産に対する事例評価

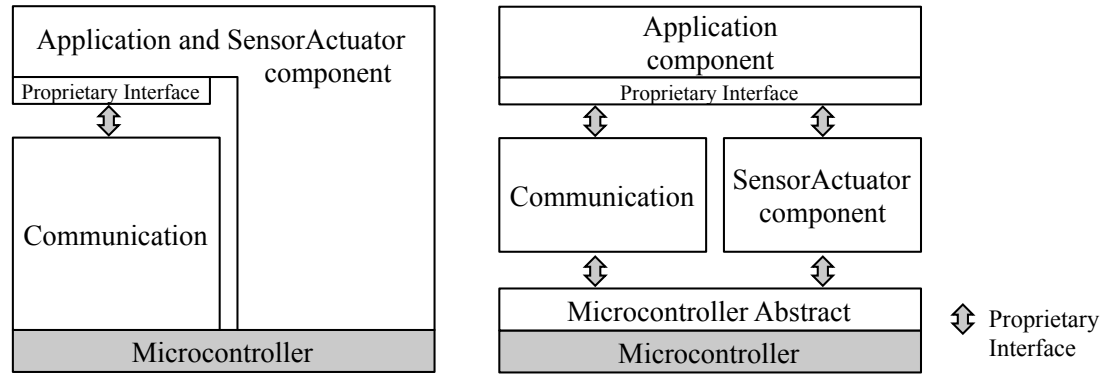
5.4.1. 有効性の確認方法

本節では，事例に基づいて提案手法の有効性を確認する方法について説明する．本事例評価では, 図 5-3 に示す 2 つのソフトウェアアーキテクチャを評価する．評価対象(a)は再利用性を十分考慮しておらず，機能要求を実現するアプリケーションコンポーネントとセンサ・アクチュエータを制御するセンサアクチュエータコンポーネントが一体化したソフトウェアアーキテクチャとなっている．唯一，他 ECU，診断ツールとの通信を担う通信コンポーネントのみ通信仕様の変更に対応できるよう分離されている．

一方，評価対象(b)は再利用性を考慮しており，マイクロコントローラ，センサアクチュエータ，通信仕様の変更にそれぞれ対応できる単位でコンポーネントを分割したアーキテクチャを採用している．

また，評価対象(a), (b)ともにコンポーネント間の接続には独自仕様のインタフェースを採用しており，自動車業界で標準規格となっている AUTOSAR 仕様には準拠していない．

なお，本実験の前提として，この標準ソフトウェア資産は，同一の機能を様々な環境の車載システムに搭載する際に利用されるものとする．



(a) Not considering reuse of software　　(b) Considering reuse of software

図 5-3 評価対象

5.4.2. 実験結果

評価対象(a), (b)を重み付きソフトゴールで評価した結果をそれぞれ図 5-4, 図 5-5 に示す．評価対象(a), (b)ともに AUTOSAR に準拠したインタフェースを採用していないことから, 設計ツール, ソフトウェアプラットフォームの変更に対して否定関係となる．評価対象(a)は，通信コンポーネントを除くコンポーネントが一体化しているため，通信コンポーネントで対応できる他 ECU，診断ツールの変更以外の変更との関係は否定関係となる．

表 5-2, 表 5-3 は上述した重み付きソフトゴールに基づき，それぞれの評価値を計算した結果である. 評価対象(a)の評価値は-0.44 であり, 評価対象(b)は 0.11 であり，標準ソフトウェア資産の有効性において，評価対象(b)は評価対象(a)に比べて優れていると言える．この結果は，本実験の前提とした，評価対象(a)：再利用性を考慮していない，評価対象(b)：再利用性を考慮している，にも整合している．

なお，本実験は 5.4.1 節で定めた前提条件に基づいており，Deployed system (S4)が搭載される Collaborating system (S5)と Competing system (S7)の重みを他よりも重要な要素として定義している．

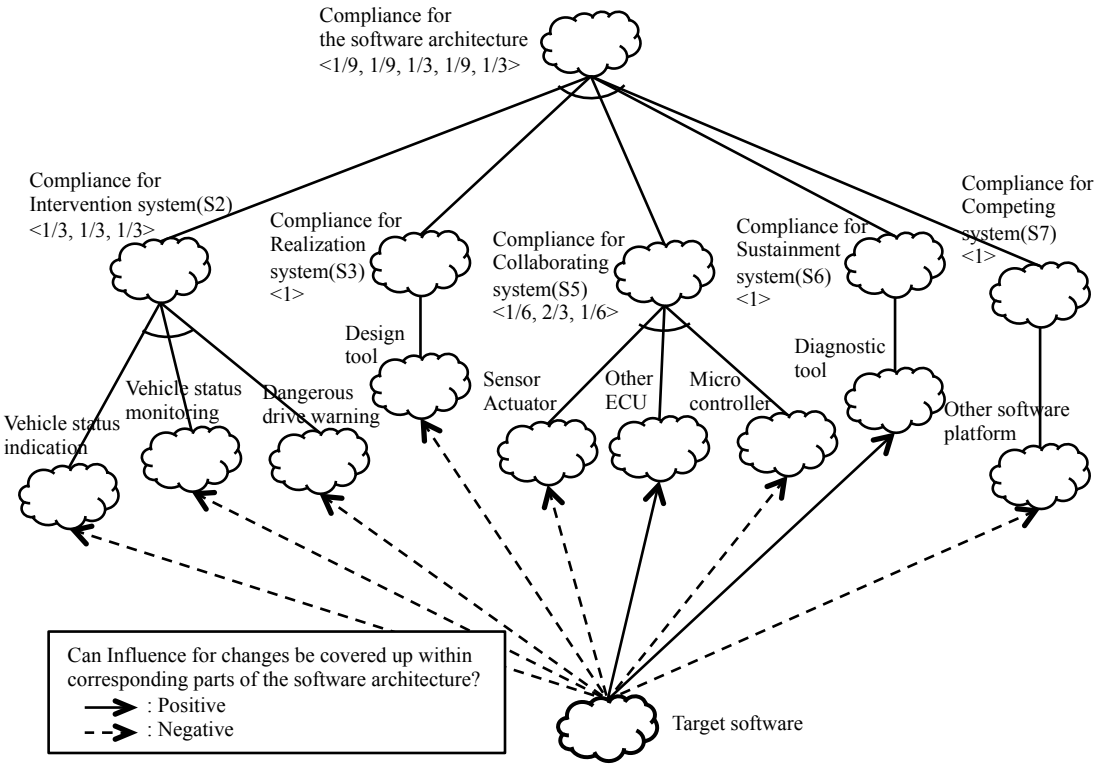


図 5-4 評価対象(a)の評価結果

表 5-2 表形式で算出した評価対象(a)の評価結果

SIG decomposition structure					Target system (a)
Compliance for the standard software architecture -0.44	1/9	Compliance for Intervention system (S2) -1.00	1/3	Vehicle status indication	-1
			1/3	Vehicle status monitoring	-1
			1/3	Dangerous drive warning	-1
	1/9	Compliance for Realization system (S3) -1.00	1	Design tool	-1
	1/3	Compliance for Collaborating system (S5) 0.00	1/6	SensorActuator	-1
			2/3	Other ECU	1
			1/6	Microcontroller	-1
	1/9	Compliance for Sustainment system (S6) 1.00	1	Diagnostic tool	1
	1/3	Compliance for Competing system (S7) -1.00	1	Other software platform	-1

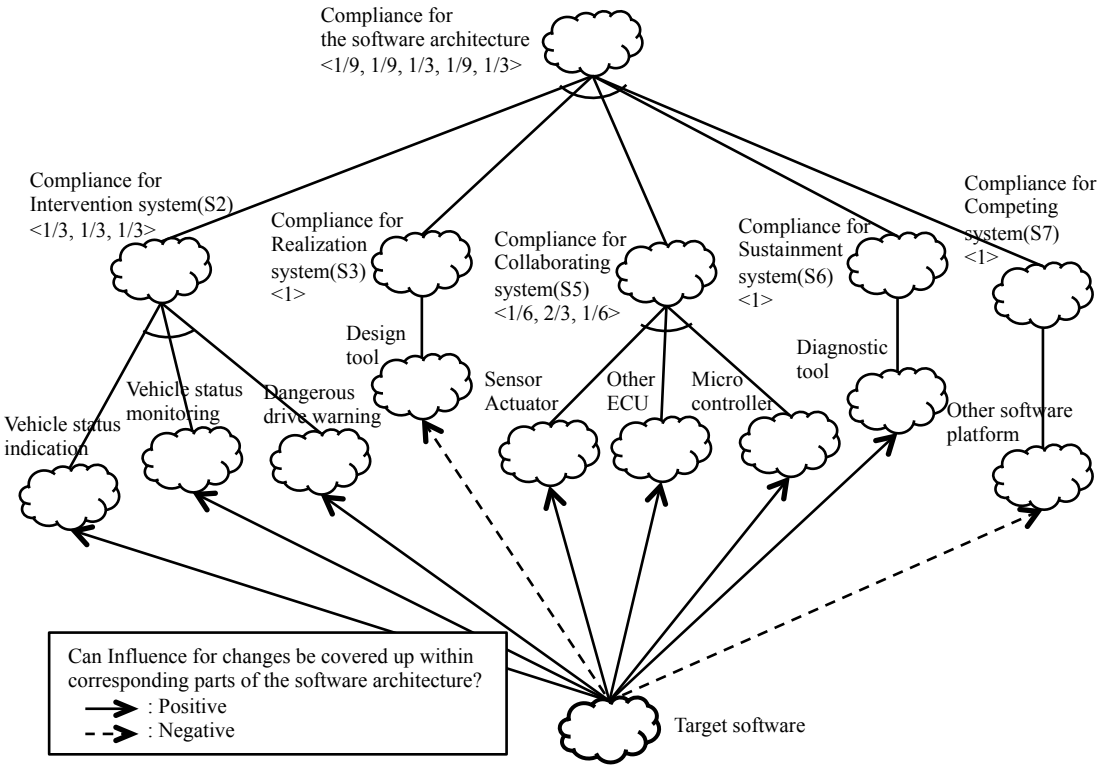


図 5-5 評価対象(b)の評価結果

表 5-3 表形式で算出した評価対象(b)の評価結果

SIG decomposition structure					Target system (b)
Compliance for the standard software architecture 0.11	1/9	Compliance for Intervention system (S2) 1.00	1/3	Vehicle status indication	1
			1/3	Vehicle status monitoring	1
			1/3	Dangerous drive warning	1
	1/9	Compliance for Realization system (S3) -1.00	1	Design tool	-1
	1/3	Compliance for Collaborating system (S5) 1.00	1/6	SensorActuator	1
			2/3	Other ECU	1
			1/6	Microcontroller	1
	1/9	Compliance for Sustainment system (S6) 1.00	1	Diagnostic tool	1
	1/3	Compliance for Competing system (S7) -1.00	1	Other software platform	-1

5.5. 考察

5.4.2 節の実験結果から，本提案手法を用いることで標準ソフトウェア資産の有効性を定量的に評価できることが確認できた．また，車載ソフトウェア開発活動メタモデルがソフトゴールに相当する評価基準を抽出する際に有効であることも確認できた．車載ソフトウェア開発活動に関連した知識は，車載ソフトウェアの大規模，複雑化に伴い，より難解になってきている．今回の実験結果から，7 人の侍フレームワークは知識の共有に関して有効であることが確認できた．このフレームワークは 7 つの要素のみから構成されており，EAST-ADL，AUTOSAR といった大規模な規格と比べて理解が容易であるため，実際の開発プロジェクトへの適用も容易であることが期待できる．

しかしながら，本提案では，実際の車載ソフトウェア開発において頻繁に変更が発生すると予想される Intervention system (S2)の変更について考慮していない．今後の課題として，S2 の詳細化を含めた様々な事例評価の必要がある．

5.6. 結論

本提案では，車載ソフトウェア開発活動に関する知識を 7 人の侍フレームワークに基づいたメタモデルとして可視化している．さらに，そのメタモデルを利用した手法の一つとして，標準ソフトウェア資産の評価手法を考案している．さらに，標準ソフトウェア資産のアーキテクチャ候補として，変化に対する影響範囲の異なる 2 種類のアーキテクチャを用意し，その比較実験の結果から本手法の有効性を確認している．比較実験においては，重み付きソフトゴールを用いることで，車載ソフトウェア資産の再利用性を定量化した評価値で示すことができ，その結果がそれぞれのアーキテクチャの特性と整合していることを確認できた．

今後の課題として，7 人の侍フレームワークに基づいて作成したメタモデルの実用性を高めていくために，より多くの事例，特に実際の車載ソフトウェア開発に適用して評価、改善に取り組む必要がある．

6. SPRME を用いたアシュアランスケース作成手法の提案

本章では，1.1.5 節で述べた課題⑤を解決する方法として，保証対象となるソフトウェアの構造，期待される品質特性，想定されるリスクと対策，およびその対策結果に基づいて統一的にアシュアランスケースを生成する手法を提案する．また，生成したアシュアランスケースの有効性を確認するために，ソフトウェアレビューを対象とした従来手法との比較実験の結果について述べる．

6.1. はじめに

車載ソフトウェア開発企業が自社の開発状況を調べたところ，開発プロジェクトが必要とするレビューアの人数に対して，適切な能力を有したレビューアが割り当てられている割合は 5 割程度であった．レビューアには，製品の特性に関する十分な知識，および製品の品質確認に用いる手法の習得が求められるが，熟練者となるためには長い育成期間が必要となる．

本提案では，上記課題に対して，熟練者の経験に基づいたレビュー結果を可視化するアシュアランスケースの作成手法を提案する．本提案手法は，レビュー対象を整理するために SPRME[31]を採用している．SPRME は，表 6-1 に定義された 5 つの要素の頭文字から命名されており，それらを用いてレビュー対象を明確に分類することができる．本章では SPRME の構成要素をイタリック体で示している．

本節以降では， 6.2 節では SPRME を用いたアシュアランスケースの作成方法を提案する．6.3 節では本提案の提案手法を適用した結果について説明する．最後に 6.4 節で本提案の結論と今後の課題について述べる．

表 6-1 SPRME 要素の説明

要素	内容
Subject	レビュー対象の構成要素とその関係
Property	*Subject* に対して要求された品質基準
Risk	HAZOP[14]のような分析手法を用いて導出された *Property* の達成を阻害する障害
Measure	*Property* を達成するためにリスクを解消する方法
Evidence	*Measure* が *Risk* を解決したことを証明できる事実

6.2. 提案手法

本節では，ソフトウェアレビューにおいて有用なアシュアランスケースの作成手法を提案する．提案手法が採用する SPRME は，GSN のような議論モデルの記法を参照して定義されており，製品品質の妥当性をステークホルダに説明する際に有効である．このため，本手法はソフトウェア製品の妥当性を議論するソフトウェアレビューにおいても有用であることが期待される．

SPRME は図 6-1 に示すように GSN の要素と関連づけることができる．以下にメタモデルの関係に従った SPRME モデルの実体から GSN で記述されたアシュアランスケースへの変換手順を示す．

手順 1 GSN の最上位のゴールに"*Subject* satisfies *Property*"を定義し，"All nodes satisfy *Property*"と"All relationships satisfy *Property*"の二つのサブゴールに分解する．

手順 2 手順 1 で分解されたそれぞれのサブゴールを，ノード，あるいは関係の実体数分だけ分解する．それぞれの分解されたサブゴールは，"Instance of nodes or relationships satisfy *Property*"のように定義する．それらのサブゴールは，"Instance of *Property*"に基づいて分解され，それぞれの分解されたサブゴールは，"Instance of node or relationship satisfies an Instance of *Property*"として定義される．

手順 3 手順 2 で分解されたサブゴールは Risk に基づいて分解され，それぞれの分解されたサブゴールは，"Instance of node or relationship addresses an instance of *Risk*"として定義される．それらのサブゴールには，GSN の前提ノードとして"*Measure* for an instance of *Risk*"，証拠ノードとして *Evidence* が関連付けられる．

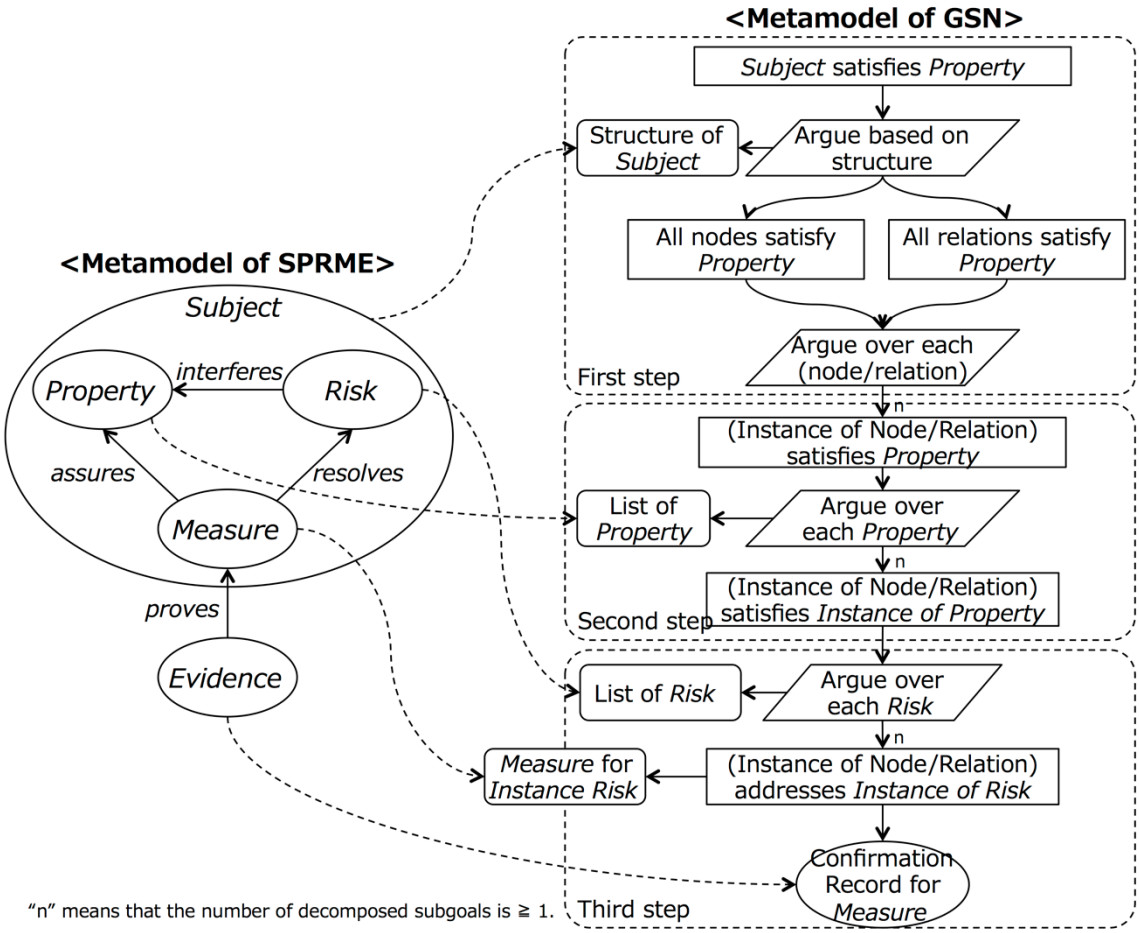


図 6-1 SPRME と GSN の関係のメタモデル

さらに，ソフトウェアレビューにおける有用なアシュアランスケースを作成するために，本提案手法では SPRME 要素に表 6-2 の内容を対応づけている．

表 6-2 SPRME 要素に対応づけた内容

要素	内容
Subject	ソフトウェアの構造
Property	ソフトウェアに対するレビューの合格基準
Risk	熟練したレビューアの経験に基づいた確認項目
Measure	Subject と Risk の組み合わせから導出できる問題の解決策
Evidence	解決策が実装されていることを確認した記録

なお，以下では，この変換手順を用いる上での前提条件として，*Subject* が *Property* を持つことを，1）構成要素とその相互関係で分解する場合，2）*Property* を下位の *Property* に分解する場合，3）これらのいずれでもない場合に分けて説明する．

まず，*Subject* が *Property* を持つことを，構成要素とその相互関係に分解しない場合で，かつ，*Property* を下位の *Property* に分解しない場合，*Subject* が *Property* を持つ *Risk* を列挙して，*Subject* が各 *Risk* に対処できていることを証拠によって確認することができる．この場合，手順 1 と手順 2 を省略して手順 3 に基づいて GSN を作成する．このとき、「Argue over each Risk」の下位にある主張を「*Subject* addresses Instance of *Risk*」として分解することができる．

次に，*Subject* が *Property* を持つことを，構成要素とその相互関係に分解しない場合であって，かつ，*Property* を下位の *Property* に分解する場合，手順 1 を省略して，手順 2 と手順 3 に基づいて GSN を作成する．この場合，「Argue over each *Risk*」の下位の主張を「*Subject* satisfies Instance of *Property*」として分解することができる．

また，*Subject* を分解する場合で，*Property* を分解しない場合は、手順 2 を省略して，手順 1 に次いで手順 3 を実施することにより，GSN を分解できる．

最後に，構成要素とその相互作用で分解する場合には、手順 1 から，手順 2，手順 3 を用いて GSN を分解することができる．

以上で述べたように，図 6-1 のモデルは、1）*Subject* が構成要素とその相互作用に分解する場合，2）*Property* を分解する場合だけでなく，それぞれを分解しない場合にも適用できる．

6.3. 実験

図 6-2 に *Subject* に相当する評価事例のソフトウェア構造を示す．この事例が示す Model Checker は，Model file からチェック対象となる Block のみを抽出し，それらに対してエラーチェックを行った上で，その結果をコンソールに出力する機能を有している．

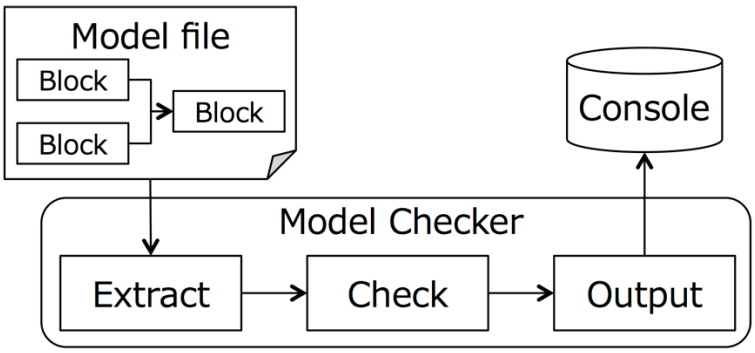


図 6-2 *Subject* に相当するソフトウェア構造

また，この事例において *Property* は以下の内容を採用している．

・ Completeness of input patterns for the model checker.

さらに *Risk* には，ソフトウェア構造の関係に関する以下の熟練者の知識のみを採用し，それ以外の知識については省略している．

・ There is no data.
・ Data is out of range.

本事例は, 実際のソフトウェア製品の一部であり, 図 6-2 に示した Check が, Extract の出力が空の場合に動作不良を引き起こす欠陥を含んでいる（このケースは Model file に Extract の対象となる Block が存在しない場合に相当する）. 表 6-3 に従来のレビューと本提案を用いたレビューの比較結果を示す．本事例において，従来のレビューのレビューアは，3 年以上の開発経験を有していた．さらに，レビュー対象の入力となる Block と Model file のパターン，および Console の入力となるレビュー対象の出力のパターンを十分に確認していた．しかしながら, 熟練者の知識に相当する確認項目に従って, レビュー対象の内部要素の関係を全て確認していなかったため，ID.5 に関係する欠陥を検出することができなかった．また，従来のレビュー記録では，図 6-3 に示すようにレビューアが指摘した欠陥は図中の下段の表のように記録されているが，レビューアが内容を確認して問題ないと判断した内容に関しては図中の上段の表のように一般論のレベルでしか記録されていない．このため，マネージャが製品品質を

判断する際の情報としては，製品がどの範囲で確認されているか把握することが困難な状況であった．一方，提案手法では，SPRME に基づいて統一的に生成されたアシュアランスケースがレビュー記録として提示されるため，レビューアの確認範囲を具体的に把握した上で品質を判断することが可能となる．図 6-4 に従来のレビューの仕方では検出できなかった問題を検出したアシュアランスケースの抜粋を示す．図中の G9 が従来のレビューの仕方で見落とされた項目に相当している．

表 6-3 従来手法と提案手法の欠陥検出能力に関する比較結果

ID	確認対象	確認項目	従来	本提案
1.	Model file と Extract の関係	There is no data	検出	検出
2.		Data is out of range	検出	検出
3.	Block と Extract の関係	There is no data	検出	検出
4.		Data is out of range	検出	検出
5.	Extract と Check の関係	There is no data	**未検出**	検出
6.		Data is out of range	検出	検出
7.	Check と Output の関係	There is no data	検出	検出
8.		Data is out of range	検出	検出
9.	Output と Console の関係	There is no data	検出	検出
10.		Data is out of range	検出	検出

ID.	Review point	Evidence	Reviewer
1.	Features defined in the requirement document are tested?	The following review result details	CONFIRMED

<Review result details>

ID.	Problems	Status	Reviewer
1.	Testing environment shall be equal to User environment.	RESOLVED	CONFIRMED
2.	The case of “out of data” for M3-O2 shall be confirmed, other relations shall be reconfirmed.	RESOLVED	CONFIRMED

図 6-3 従来のレビュー記録抜粋

6.3.1. 制約事項

提案手法の有効性は，一つの小規模なソフトウェアに対してのみ確認されている．このため，大規模なソフトウェアに対して適用し，その有効性を評価する必要がある．また，本章の実験では，ソフトウェアレビューにおける確認項目として，レビュー対象の構造毎に確認できる基本的な 2 項目のみを採用している．このため，より複雑な確認項目を多数採用した事例を用いてその有効性を評価する必要がある．

6.4. 結論

本章では，SPRME を用いて統一的にアシュアランスケースを作成する手法を提案した．また，本手法で作成したアシュアランスケースの有効性を確認するために，ソフトウェアレビューの事例を対象とした比較実験を行った．本実験の結果から，提案手法で作成したアシュアランスケースは，レビューアの能力に依存した従来のレビュー方法よりも高い欠陥検出能力を有することが確認できた．

上記結果から，SPRME を用いた統一的なアシュアランスケース作成法は，セーフティ，セキュリティといった領域以外のアシュアランスケースの作成においても有用であることが確認できた．

今後の課題としては，本提案手法の実用性を評価するために，より大きな規模の事例に適用してその有効性を確認する必要がある．例えば，大規模なアシュアランスケースをレビューアが理解するためには，多くの時間を要することが予想されるため，その可読性の改良が期待される．また，熟練者の知識を資産化して再利用できる仕組みの考案も期待される．さらに，開発現場において本提案手法を容易に実践するための支援環境の開発に取り組んでいく必要がある．

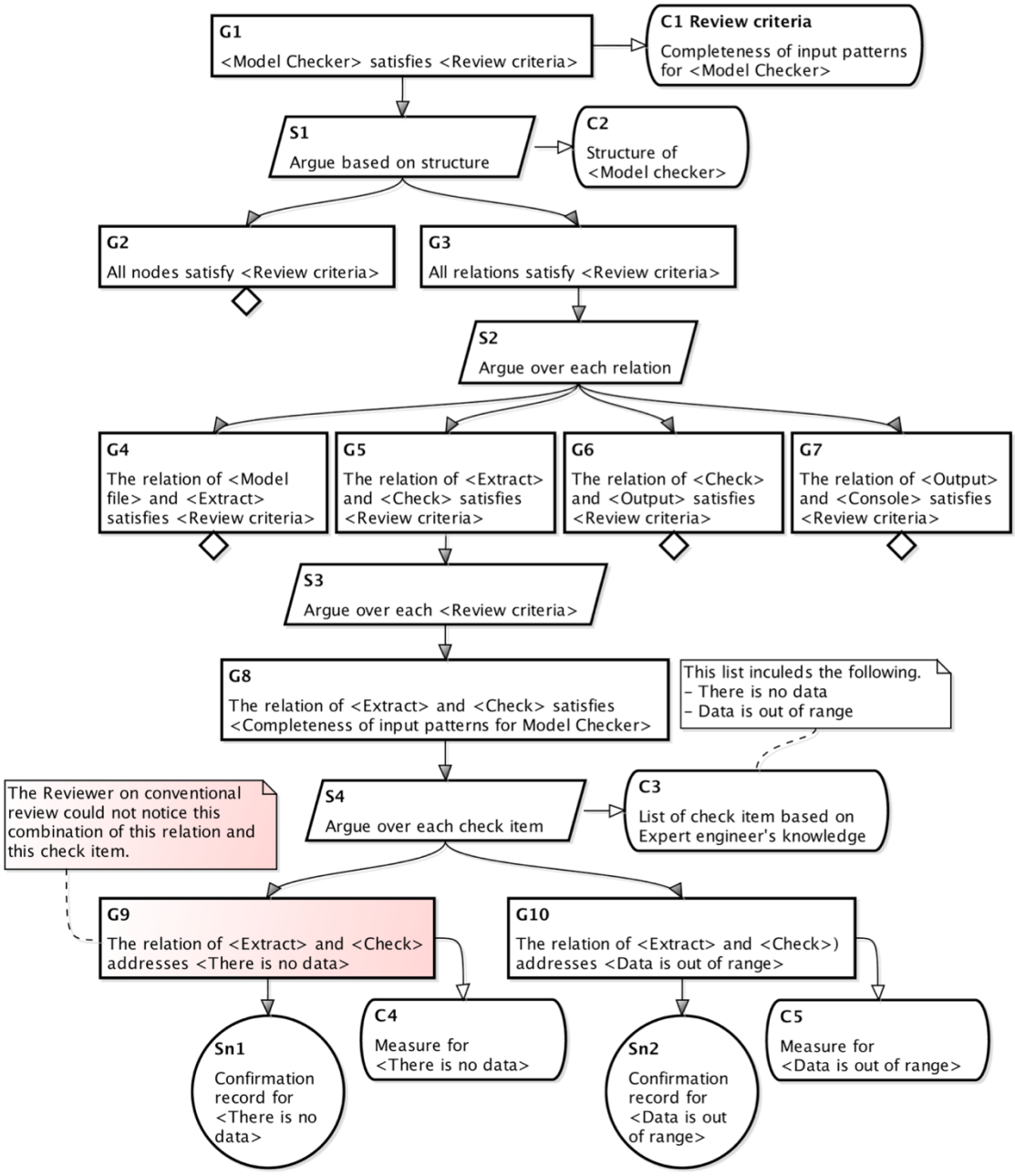


図 6-4 本手法を用いて作成したアシュアランスケース抜粋

7. 結論

7.1. 本研究のまとめ

本研究が対象とする車載システム開発分野は，従来，運用開始後の振る舞いが変化しない前提でセーフティ要求を中心に品質保証する特性を有していた．一方，自動走行などを想定した今後の車載システム開発では，運用開始後も常に進化することに価値がある人工知能のようなシステムとの連携が一般的になるため，セーフティ要求に加えてセキュリティ要求にも対応する必要がある．

上記の背景に対して，本研究では，セーフティ要求，セキュリティ要求といった相反する可能性のある要求群で構成されたディペンダビリティ要求の保証技術に関する研究を実施した．運用開始後の進化を前提としたディペンダビリティ保証フレームワークとしては，The Open Group が提唱する O-DA (Open Dependability through Assuredness)が存在しており，その運用には想定されるシステムのリスクを網羅的に確認し，その結果を関係者と正しく合意形成した記録が不可欠となる．アシュアランスケースは，関係者との議論の前提を明文化した上で，議論内容を構造化して記録する文書であり，自然言語だけでなく，GSN (Goal Structuring Notation)などのモデリング言語を用いて記述することが可能である．しかしながら，従来のアシュアランスケースに関する研究では，1.1 節で示した不足が存在していた．本研究では，これらの不足に対応するため，D-Case を用いた安全部分析結果の説明手法の提案（不足①），非機能要求の定量的評価手法の提案（不足③），7 人の侍フレームワークを用いた標準ソフトウェア資産の評価（不足④），品質特性に基づくアシュアランスケース作成手法の提案（不足⑤），を実施した．本研究において得られた主な結果を次節以降で述べる．

7.1.1. D-Case を用いた安全分析結果の説明手法の提案

3 章では，車載システム開発において従来から用いられているセーフティ分析手法と，D-Case と呼ばれるアシュアランスケースの記述法との統合手法を提案した．セーフティ分析に関する従来の研究では，個々の「分析技術」の応用やその組合せ方法について考慮しているが，第三者に対する客観的な「説明技術」との組合せまでは考慮していない．

本手法では，HAZOP，FTA を用いた安全分析の結果を D-Case を介して論理

的に統合する手法を考案した．提案手法では，システムの安全性を主張した最上位のゴールを，HAZOP，FTA を用いたリスク分析結果に基づいて，最小単位になるまで下位のゴールに分割する．さらに，最下位のゴールにはそのゴールの合格基準として FTA を用いて抽出したリスク原因の対策を紐付け，その基準を達成できる証拠を関連付けることで D-Case を完成させる．

さらに，エンジニア 10 名を対象として，HAZOP, FTA を個別に用いた従来形式の分析結果と上述した提案手法を用いて作成した D-Case の比較実験を行った．本実験では，それぞれの分析結果に対して同一の質問を行い，正答率に関しては本手法が 90％前後，従来形式が 50％程度であり，回答時間に関しては本手法が約 4 分，従来形式が約 9 分という結果であった．この結果から，本手法を用いて作成した D-Case が従来形式の分析結果より優れることを確認できた．なお，実際の車載ソフトウェア開発現場では，製品品質を保証する際に確認内容の網羅性の担保が重要となる．D-Case は，ゴールの分割根拠を明確に示すことができる記法であり，開発現場のマネージャが上記視点で製品品質を判断する際の有効なビューとなる可能性が高い．今後の課題として，本手法の現場導入に取り組み，その実績に基づいて本手法の有効性を評価していきたい．

7.1.2. 非機能要求の定量評価手法の提案

4 章では，異なる特性を有した非機能要求の衝突問題を解決する方法として，非機能要求の満足度を定量的に評価する手法を提案した．非機能要求を満足できる最適なアーキテクチャを導き出すためには，衝突する非機能要求に対してトレードオフの判断が必要となる．しかしながら，その判断理由を定量化して客観的に説明できる手段が用意されていなかった．NFR フレームワークをはじめとする従来の研究では，非機能の分解は考慮しているが，子ノード間の重み関係を考慮していない．

本手法では，NFR フレームワークを拡張して，分解に関する重み付けをソフトゴールの属性として持たせる記法を考案し，子ノードの間のトレードオフ関係を評価できるようにした．これにより，熟練者の有する非機能要求に関する知識の体系化と重み付けによる設計方針の満足度の定量化が可能となった．また，上記内容をテーブル形式で表現し，定量化の計算を容易にするための変換則を考案した．IoT 機器との接続が一般的となる今後の車載システム開発では，従来重視されてきたセーフティ要求に加えて，セキュリティ要求の考慮も不可欠とな

る．本手法は，相反するセーフティ要求，セキュリティ要求のトレードオフ関係を定量的に評価できる有用な手法になると予想される．
今後の課題として，本手法を実際の車載システム開発に適用して，その有効性を評価する．

7.1.3. 7 人の侍フレームワークを用いた標準ソフトウェア資産の評価知識

5 章では，ゴール指向分析手法におけるゴールの分解根拠に利用可能な参照モデルとして，7 人の侍フレームワークを用いた車載ソフトウェア開発活動メタモデルを定義した。さらに，このメタモデルを利用して，プロダクトラインを適切に運用する際に重要となる標準ソフトウェア資産のアーキテクチャ評価手法を提案した．アーキテクチャ評価に用いる NFR フレームワークの SIG を作成するためには，プロダクトラインで扱う対象に応じて可変要素を想定し，その内容を構造化したソフトゴールとして定義する必要がある．しかしながら，開発経験の浅いエンジニアが開発活動全体を把握することは困難であり，その状況下で可変要素を抽出して構造的にソフトゴールを定義することは難しい．本手法では，熟練者が有する開発活動に関する知識を 7 人の侍フレームワークに基づいて整理した車載ソフトウェア開発活動メタモデルの構造に従い，ソフトゴールを段階的に分解する手法を提案した．さらに提案手法の妥当性を確認するため，特性の異なる下記 2 つのアーキテクチャの評価実験を行った．

- 他社製品，市販開発環境と互換性のない自社独自のアーキテクチャ
- 自動車業界標準である AUTOSAR を部分的に採用したアーキテクチャ

提案手法を用いて評価した結果は，上記 2 つのアーキテクチャの特性と整合しており，車載ソフトウェア開発活動メタモデル，および提案した評価手法の妥当性を確認することができた．

7.1.4. 品質特性に基づくアシュアランスケース作成法の提案

6 章では，ディペンダビリティを構成する様々な品質特性の要求に対するアシュアランスケースを生成する手法として，SPRME を用いた統一的なアシュアランスケース作成法を考案し，さらにその有効性を確認した．SPRME は，アシュアランスケースの保証対象を 5 つの観点 (Subject：保証対象の構造，Property：

保証対象に期待される特性，Risk：特性の達成を阻害するリスク，Measure：リスクを解消する対策，Evidence：対策が備わっている証拠）で整理できるメタモデルを提供している．本手法では，このメタモデルとアシュアランスケースの構成要素の対応関係を定義することで，アシュアランスケースを統一的に生成する変換則を明らかにした．また，本手法を用いて作成したアシュアランスケースの有効性を確認するために，ソフトウェアレビューを対象とした従来手法との比較実験を行った．本実験の事例とした従来のレビュー手法は，レビューアの能力に依存して実施しており，網羅性の観点で抜け漏れが発生していた．一方，提案手法では，確認すべき事項と対象の組み合わせがゴールツリー形式で提示されるため，網羅性の観点で従来手法よりも高い欠陥検出能力を有していることが確認できた．さらに，レビュー記録についても，従来のレビュー記録の形式には欠陥に関する指摘は記録できるが，レビューアが問題ないと判断した範囲に関する記録を残すことができない．このため，マネージャが最終的な品質を判断する際に，レビューアの確認した範囲を踏まえて判断することが難しい状況となっていた．一方，提案手法では，確認した範囲がアシュアランスケースとして漏れなく提示されるため，マネージャがレビューアの確認範囲を踏まえて品質を判断することが可能となる．

上述の結果から、SPRME を用いて作成したアシュアランスケースは、セーフティ、セキュリティを保証する際だけでなく、その他の品質特性を保証する際においても有用であることが確認できた。今後の課題として，複雑なシステムへの適用実験を繰り返し，本手法の実用面における課題抽出とその解決に取り組んでいく．

7.2. 今後の課題

上記で述べた通り，本研究では，衝突の可能性がある要求で構成されたディペンダビリティ要求を保証するために，それぞれの要求に対するアシュアランスケースの統一的な作成法，および要求間の衝突を定量的な根拠に基づいて解消する手法を考案した．本研究では，車載システム分野におけるセキュリティケース作成法の提案までは実施できなかったが，ArchiMate と組み合わせたセキュリティケース作成法[3]の適用実験を終えており，その結果が Communications in Computer and Information Science (CCIS) - Springer に採録される見通しである．

本研究では，図 7-1 に示すようにアシュアランスケースの作成に軸足を置いて研究活動を推進してきた．その際，全ての研究において，保証対象がモデル記述等の制約された言語で記述され，その内容が適切な粒度と正しい関係で構成されていることが，高品質なアシュアランスケースを作成する上で不可欠であった．今後の課題として，車載システム開発の入力となる要求記述やその要求を実現するシステムアーキテクチャ定義に用いるモデル記述言語の有効性評価を進め，その利用方法の考案，および自動化による作成効率の向上や人為ミス混入の解消を狙った支援環境の開発に取り組んでいきたい．

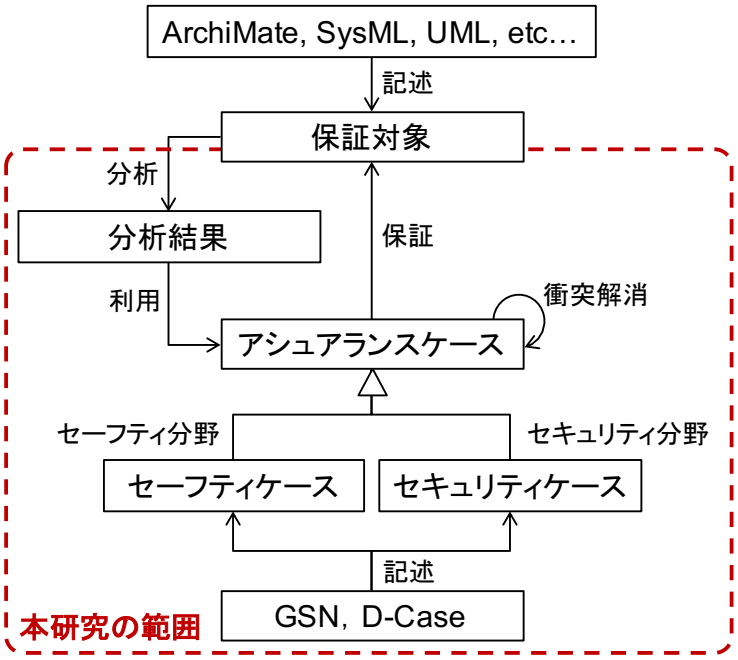


図 7-1 本研究の対象範囲

謝辞

本研究をまとめるにあたり，種々の御指導，御鞭撻，御支援を賜りました名古屋大学大学院情報学研究科情報システム学専攻教授 山本修一郎博士，同教授 高田広章博士，同准教授 森崎修司博士に心から感謝の意を表します．

また，本研究に取り組む機会を与えてくださった株式会社デンソークリエイト顧問 林修吉氏，同社専務取締役 辻村健治郎氏，同社常務取締役 藤井友康氏，同社取締役 加藤宏幸氏，同社イオタ推進部 木下稔章部長，同社事業推進部 宮川英明部長，同社技術リソーセス部 広瀬智部長にお礼申し上げます．

本研究を進めるにあたり，株式会社デンソークリエイト技術リソーセス部 山田ひかり氏，同社技術リソーセス部 林香織氏には，多くの議論を交わす中で素晴らしい示唆を与えて頂きました．お二人の多大な御支援に深く感謝致します．また，様々な面でサポートして頂きました日本アイ・ビー・エム シニアプロジェクトマネージャ 山本佳和氏，株式会社チェンジビジョン 代表取締役 平鍋健児氏，同社取締役 小阪暢之氏，同社執行役員 岩永寿来氏，同社 高井利憲氏，同社 藤元謙次氏，および名古屋大学大学院情報学研究科情報システム学専攻山本研究室の諸氏に心から感謝致します．

最後に，本研究に挑戦することを後押しし，様々な面でいつも支えになってくれた素晴らしい妻 俊子に心から感謝の意を表します．

参考文献

[1] The Open Group, Dependability through Assuredness (O-DA) Framework. 2013, pp.1–69.

[2] A. Josey, TOGAF® Version 9.1-A Pocket Guide. Van Haren Publishing, 2011.

[3] S. Yamamoto and N. Kobayashi, “Mobile Security Assurance through ArchiMate,” *IT CoNvergence PRActice (INPRA)*, vol.4, no. 3, pp.pp. 1–8, Sep. 2016.

[4] Autosar Technical Overview. AUTOSAR. [Online]. Available: http://www.autosar.org/about/technical-overview/.

[5] J. N. Martin, “3.1.2 The Seven Samurai of Systems Engineering: Dealing with the Complexity of 7 Interrelated Systems,” vol.14, no. 1, pp.459–470, Jun. 2004.

[6] S. Yamamoto and S. Morisaki, “A case study on architecture quality assurance service using O-DA,” Conference on ENTERprise Information Systems 2016, Sep. 2016.

[7] N. Kobayashi, “The evaluation of O-DA Template application ,” *SIG-KSN 20*, 2017.

[8] C. C. Howell, S. Guerra, S. L. Pfleeger, and V. Stavridou-Coleman, “Workshop on assurance cases: best practices, possible obstacles, and future opportunities 1 July 2004, Florence, Italy,” 2004 International Conference on Dependable Systems and Networks, pp.841–841, 2004.

[9] Y. Matsuno, V. Patu, and S. Yamamoto, “A Survey on Structured Documents for Assurance Cases,” *KBSE2012-20*, Jul. 2012.

[10] T. Kelly and R. Weaver, “The Goal Structuring Notation - A Safety Argument Notation,” Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, 2004.

[11] K. Attwood, P. Chinneck, M. Clarke, G. Cleland, M. Coates, T. Cockram, and P. Williams, “GSN Community Standard Version 1,” 2011.

[12] M. Tokoro, Open Systems Dependability, Dependability Engineering for Ever-Changing Systems. CRC Press, 2012.

[13] Y. Matsuno, J. Nakazawa, and M. Takeyama, “Towards a language for communication among stakeholders,” *(PRDC)*, 2010.

[14] J. Dunjó, V. Fthenakis, J. A. Vílchez, and J. Arnaldos, “Hazard and operability (HAZOP) analysis. A literature review,” Journal of hazardous materials, 2010.

[15] R. de Queiroz Souza and A. J. Álvares, “FMEA and FTA analysis for application of the reliability-centered maintenance methodology: case study on hydraulic turbines,” ABCM Symposium Series in Mechatronics, 2008.

[16] T. Kelly and R. Weaver, “The goal structuring notation–a safety argument notation,” Proceedings of the dependable systems and networks 2004 workshop on assurance cases, 2004.

[17] M. Tokoro, Open Systems Dependability. CRC Press, 2015.

[18] K. Allenby and T. Kelly, “Deriving safety requirements using scenarios,” Fifth IEEE International Symposium on Requirements Engineering, pp.228–235, 2001.

[19] P. Fenelon and B. Hebbron, “Applying HAZOP to software engineering models,” Risk Management And Critical Protective Systems: Proceedings of SARSS, 1994.

[20] F. Ding, S. Yamamoto, and N. Abrahim, “The Method of D-Case Development Using HAZOP Analysis on UML Models,” in Knowledge-Based Software Engineering, vol.466, no. 54, Springer, Cham, Cham, 2014, pp.617–629.

[21] Y. Matsuno and K. Taguchi, “Parameterised Argument Structure for GSN Patterns,” 2011 11th International Conference on Quality Software (QSIC), pp.96–101, 2011.

[22] Y. Matsuno and S. Yamamoto, “A Framework for Dependability Consensus Building and In-Operation Assurance,” Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, pp.1–17, Mar. 2013.

[23] S. Yamamoto, “A Knowledge Integration Approach of Safety-critical Software Development and Operation based on the Method

Architecture," Procedia - Procedia Computer Science, vol.35, pp.1718–1727, 2014.

[24] I. Habli, I. Ibarra, R. S. Rivett, and T. Kelly, "Model-Based Assurance for Justifying Automotive Functional Safety," Proc. 2010 SAE World Congress, vol.1, 2010.

[25] R. Hawkins and T. Kelly, "A software safety argument pattern catalogue," The University of York, 2013.

[26] T. Kelly and J. A. McDermid, "Safety Case Construction and Reuse Using Patterns," in Safe Comp 97, no. 5, Springer London, London, 1997, pp.55–69.

[27] T. Kelly, "Arguing safety: a systematic approach to managing safety cases," University of York, 1999.

[28] R. Bloomfield and P. Bishop, "Safety and Assurance Cases: Past, Present and Possible Future – an Adelard Perspective," in Making Systems Safer, no. 4, C. Dale and T. Anderson, eds. Springer London, London, 2010, pp.51–67.

[29] R. Palin and I. Habli, "Assurance of automotive safety–A safety case approach," *International Conference on Computer Safety, Reliability, and Security*, 2010.

[30] V. Patu and S. Yamamoto, "How to develop Security Case by combining real life security experiences (evidence) with D-Case," Procedia Computer Science, vol.22, pp.954–959, 2013.

[31] S. Yamamoto, S. Morisaki, and N. Atsumi, "A unified approach on assurance case development method based on models ," SIG-KSN, 2015.

[32] N. Subramanian and J. Zalewski, "Quantitative Assessment of Safety and Security of System Architectures for Cyberphysical Systems Using the NFR Approach," vol.PP, no. 99, pp.1–13, 2014.

[33] S. Yamamoto, "An Approach for Evaluating Softgoals Using Weight," The Asian Conference on Availability, Reliability, and Security, AsiaARES, vol.9357, pp.203–212, 2015.

[34] H. Kaiya, H. Horai, and M. Saeki, “AGORA: Attributed goal-oriented requirements analysis method,” Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on, pp.13–22, 2002.

[35] A. Kokune, M. Mizuno, K. Kadoya, and S. Yamamoto, “FBCM: Strategy modeling method for the validation of software requirements,” vol.80, no. 3, pp.314–327, 2007.

[36] S. Saito and S. Yamamoto, “The Incremental Goal Evolution Process Methodology.,” vol.proc. of workshops and doctoral consortium, pp.254–261, 2006.

[37] V. R. Basili and D. M. Weiss, “A Methodology for Collecting Valid Software Engineering Data,” vol.10, no. 6, pp.728–738, 1984.

[38] C. Alberts and A. Dorofee, “OCTAVE-The Operationally Critical Threat, Asset, and Vulnerability Evaluation,” 2001.

[39] B. Len, C. Paul, and K. Rick, “Software architecture in practice,” 2003.

[40] EAST-ADL Association, EAST-ADL Domain Model Specification Version V2.1.12. 2013, pp.1–244.

[41] S. Fürst, J. Mössinger, S. Bunzel, and T. Weber, “AUTOSAR–A Worldwide Standard is on the Road,” *14th International VDI Congress Electronic Systems for Vehicles*, 2009.

[42] P. Cuenot, P. Frey, R. Johansson, and H. Lönn, “Developing automotive products using the east-adl2, an autosar compliant architecture description language,” *Embedded Real-Time Software Conference*, 2008.

[43] The AUTOSAR development partnership, Layered Software Architecture Release 4.2.2. 2015, pp.1–165.

[44] The AUTOSAR development partnership, AUTOSAR Software Component Template Release 4.2.2. 2016.

[45] The AUTOSAR development partnership, AUTOSAR System Template Release 4.2.2. 2015, pp.1–1437.

[46] The AUTOSAR development partnership, AUTOSAR Specification of ECU Configuration Release 4.2.2. 2015, pp.1–267.

[47] The AUTOSAR development partnership, AUTOSAR Specification of ECU Resource Template Release 4.2.2. 2015, pp.1–55.

[48] D. Chen, R. Johansson, H. Lönn, and H. Blom, “Integrated safety and architecture modeling for automotive embedded systems*,” 2011.

[49] P. Cuenot, D. J. Chen, S. Gerard, and H. Lönn, “Managing complexity of automotive electronics using the East-ADL,” *Engineering Complex Computer Systems*, pp.353–358, 2007.

[50] R. Flores, C. Krueger, and P. Clements, “Mega-scale product line engineering at General Motors,” Proceedings of the 16th International Software Product Line Conference, New York, New York, USA, pp.259–268, 2012.

[51] A. Leitner, R. Mader, C. Kreiner, and C. Steger, “A development methodology for variant-rich automotive software architectures,” 2011.

[52] K. Schmid and I. John, “A customizable approach to full lifecycle variability management,” 2004.

[53] S. Thiel, S. Ferber, T. Fischer, and A. Hein, “A case study in applying a product line approach for car periphery supervision systems,” SIG-KSN-017-04, pp.43–55, 2001.

[54] M. Voelter and I. Groher, “Product line implementation using aspect-oriented and model-driven software development,” *Software Product Line Conference*, pp.233–242, 2007.

[55] Y. Matsuno and S. Yamamoto, “An evaluation of argument patterns to reduce pitfalls of applying assurance case,” Assurance Cases for Software-Intensive Systems (ASSURE), 2013 1st International Workshop on, pp.12–17, 2013.

[56] N. Kobayashi and S. Yamamoto, “The Effectiveness of D-Case Application Knowledge on a Safety Process,” *Procedia Computer Science*, vol.60, pp.908–917, 2015.

[57] K. Attwood and T. Kelly, “Controlled expression for assurance case development,” Proceedings of the 23rd Safety-Critical Systems Symposium on Engineering Systems for Safety, 2015.

[58] K. Attwood, P. Conmy, and T. Kelly, “The Use of Controlled Vocabularies and Structured Expressions in the Assurance of CPS,” *ADA USER*, 2014.

[59] R. Hawkins, I. Habli, and T. Kelly, “The Need for a Weaving Model in Assurance Case Automation,” *www-users.cs.york.ac.uk*, 2015.

[60] R. Hawkins, I. Habli, D. Kolovos, R. Paige, and T. Kelly, “Weaving an Assurance Case from Design: A Model-Based Approach,” 2015 IEEE 16th International Symposium on High Assurance Systems Engineering (HASE), pp.110–117, Dec. 2014.

[61] R. Palin, D. Ward, I. Habli, and R. Rivett, “ISO 26262 safety cases: Compliance and assurance,” 6th IET International Conference on System Safety 2011, pp.1–6, 2011.

[62] 独立行政法人 情報処理推進機構, セーフティ設計・セキュリティ設計に関する 実態調査結果. 独立行政法人情報処理推進機構, 2015, pp.1–73.

[63] D. J. Pumfrey, “The principled design of computer system safety analyses.,” University of York, 1999.

[64] International Electrotechnical Commission, IEC 61882: Hazard and operability studies (HAZOP studies)-Application guide. International Electrotechnical Commission, 2001.

[65] S. Yamamoto, S. Morisaki, N. Atsumi, J. Kondo, and H. Oobayashi, “An experimental evaluation of GSN review ,” SIG-KSN, vol.18, 2016.

[66] N. Leveson, System safety and computers. Addison-Wesley, 1995.

[67] T. Kelly, “A six-step method for the development of goal structures,” 1997.

[68] H. Heinecke, K. P. Schnelle, H. Fennel, and J. Bortolazzi, “AUTomotive Open System ARchitecture-an industry-wide initiative to manage the complexity of emerging automotive E/E-architectures,” *Convergence*, 2004.

[69] M. Glinz, “On Non-Functional Requirements,” 2007 IEEE International Conference on Requirements Engineering, pp.21–26, Jul. 2007.

[70] J. A. Estefan, “Survey of model-based systems engineering (MBSE) methodologies,” *Incose MBSE Focus Group*, 2007.

[71] L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, Non-functional requirements in software engineering. 2012.

本論文に関する原著論文

学術誌論文

[72] 小林展英, 森崎修司, 山本修一郎, “D-Case を用いた安全分析結果の説明手法の提案.,” 情報処理学会論文誌, 58(2), 521-530., 2017（第3章）

[73] Kobayashi, N., Morisaki, S., Atsumi, N., & Yamamoto, S., “Quantitative Non Functional Requirements evaluation using softgoal weight.,” Journal of Internet Services and Information Security (JISIS), 6(1), 37-46., 2016（第4章）

査読付き国際会議論文

[74] Kobayashi, N., & Yamamoto, S., “The Effectiveness of D-Case Application Knowledge on a Safety Process.,” Procedia Computer Science, 60, 908-917., 2015（第3章）

[75] Kobayashi, N., Yamada, H., Utsunomiya, H., Morisaki, S., & Yamamoto, S., “The Evaluation Knowledge of Standard Software Asset using The Seven Samurai Framework.,” Procedia Computer Science, 96, 782-790., 2016（第5章）

[76] Kobayashi, N., “Assurance case development method using SPRME for software reviews.,” The 35th International Conference on Conceptual Modeling (ER2016), 2016（第6章）

査読付き国内会議論文

[77] 小林展英., “D-Caseを用いたレビューを見える化する方法の導入事例.,” 12th Workshop on Critical Software System, 2015（第3章）

国内研究会

[78] 小林展英., “安全分析プロセスにおけるD-Caseの有効性.,” SIG-KSN 16, 2015（第3章）