人工知能学会
第13回知識流通ネットワーク研究会
SIG-KSN-013-07

メソッド・アーキテクチャ手法に基づく 重要安全ソフトウェア開発運用手法の提案

山本 修一郎

名古屋大学 情報連携統括本部　情報戦略室
愛知県名古屋市千種区不老町

A Proposal on Software Knowledge Integration for Safety Critical Software Development and Operation based on Method Architecture

Shuichiro YAMAMOTO

Nagoya University
Furo-cho, Chikusa-ku, Nagoya Aichi Japan

概要

安全性を確認するための開発運用手法を現場に導入するためには実践的なソフトウェア開発運用知識体系群との統合が必要である．このため，ソフトウェアとその開発運用プロセスの安全性を保証するための安全性ケース手法と，複数の知識体系とを，多様なプラクティスを統一的に記述できるメソッド・アーキテクチャ手法に基づいて統合する手法を具体化することにより，ソフトウェア開発運用の全工程を通じて安全性を保証する方法を提案する．

Abstract

It is necessary to integrate practical software development and operation body of knowledge to deploy development and operation methods for assuring safety. In this paper, an approach based on the method architecture is proposed to develop an integrated method for describing various software related bodies of knowledge and the safety case for assured software life cycle processes.

1 はじめに

社会的に大きな影響を与えるITシステムで用いられるソフトウェアの安全性を保証するためには，開発運用対象としてのソフトウェアの安全性を確認するだけでなく，ソフトウェア開発運用プロセスの安全性を含めて保証できる開発運用手法が必要である．

また，安全性を確認するための開発運用手法が実際のソフトウェア開発運用プロジェクトで活用されるためには，開発運用プロジェクトの現場で活用されているソフトウェア開発運用知識体系群との統合が必要である．

このため，ソフトウェアとその開発運用プロセスの安全性を保証するための安全性ケース手法[1]-[3]と，複数の知識体系とを，多様なプラクティスを統一的に記述できるメソッド・アーキテクチャ手法[4]に基づいて統合する手法を具体化することにより，ソフトウェア開発運用の全工程を通じて安全性を保証する方法を提案する．

2 研究の背景

航空宇宙分野, 医療機器分野, 自動車分野などで，システムの安全性を保証する方法として，安全性ケース（Safety case）への期待が高まっている．

筆者らは，安全性ケース手法を国内におけるソフトウェア開発と運用の現場導入を容易化するために，2012年9月にディペンダビリティケース実証評価研究会(http://www.dcase.jp)を設立して普及活動を進めている．また国際的には，国際標準化組織The Open Group（TOG）に対して，エンタープライズ・アーキテクチャのフレームワークTOGAF（The Open Group Architecture Framework）[5]に関する安全性の高いアーキテクチャ開発手法AADM（Assured Architecture Development Method）[6]の提案活動に参画し，2013年7月にOpen Dependability through Assuredness（O-DA）として日本発のTOG標準として採択された．しかし，O-DA標準も策定されたばかりであり，まだ十分に安全性ケース手法の現場導入が進んでいるとは言えない．

この普及・標準化活動経験から，国内外において安全性ケース手法の利用が進展しない主な理由として次の諸点が明らかになった．すなわち，ソフトウェア開発者，運用者およびIT企業経営者，ユーザ企業経営者などとの間で安全性ケース手法に対する認

識が十分ではないために，安全性ケース手法の導入目的を明確に定められないこと，開発運用現場において活用されているSWEBOK(SoftWare Engineering Body Of Knowledge)[7]，PMBOK(Project Management Body Of Knowledge)[8]，CMMI[9], BABOK(Business Analysis Body Of Knowledge)[10]，REBOK (Requirements Engineering Body Of Knowledge)[11], ITIL(IT Infrastructure Library)[12][13], SQuARE[14]などのソフトウェア開発運用知識体系と安全性ケースの結合が不十分であること，したがって開発運用プロセスへの具体的な導入法が確立されていないことから安全性ケース手法を効果的に適用する組織能力が整備されていないことなどである．

欧米では，重要安全システムに対する安全性ケースの適用が義務付けられ，またO-DA標準が認められたように，発注者，開発者，運用者，監視当局などのステークホルダ間での合意形成手法として安全性ケースの有用性や必要性が認識されている．しかし，O-DAはTOGAFと融合された段階であり，O-DA標準の現場適用はまだこれからである．またTOGAF以外のソフトウェア開発運用知識体系と安全性ケースの結合が不十分である．また，開発運用プロセスとソフトウェア成果物の安全性ケースとの関係が不明確であることや，安全性ケースとFTA(Fault Tree Analysis)，FMEA(Failure Mode Effect Analysis)，HAZOP(Hazard and Operability)などの従来手法との融合方法の具体化などの導入上の課題がある．

このため，開発運用知識の構成要素と安全性ケース作成知識との関係や，システム開発運用プロセスの安全性を分析・評価するプラクティスには限界があった．

海外では，ソフトウェア開発プラクティスを理論的に再構成する研究として，メソッド・アーキテクチャに基づく，SEMAT (Software Engineering Method and Theory) [15]が提案され，本研究の分担者らによって日本支部も設立されている．しかし，現在のSEMATでは，安全性ケースをはじめ運用知識やプロジェクトマネジメント知識を扱っていない．

メソッド・アーキテクチャという観点を取り入れることによって，上述のような複合的な知識体系を有効活用できるように安全性ケース手法の適用性を向上する必要がある．

本研究の代表者や分担者は，我が国において，安全性ケース手法やSEMATを普及させ，その効果をソフトウェア開発の現場にもたらすべく活動を進めている．一部の企業で安全性ケース手法やSEMATの導入に向けた取り組みが見られるものの，我が国におけるこれらの手法の融合とその普及は，まだ進んでいない．

本研究では，SEMATのメソッド・アーキテクチャの概念に基づいて，多様な知識体系を安全性ケース手法と結合することにより，ソフトウェア開発運用ライフサイクルの各段階において安全性を保証する包括的な方法を提案する．また，安全性ケース手法におけるパターン化やエディタなどの要素技術，導入研修活動の実績，知識工学手法，開発プラクティスの再利用などに関する研究代表者および分担者のこれまでの成果に基づいて，運用工程も含むソフトウェア開発運用ライフサイクル全般に亘って，メソッド・アーキテクチャに基づいてソフトウェアの安全性を効果的に確認できる方法を提案するとともに，その有用性を事例研究によって実証する．

本研究は，メソッド・アーキテクチャの概念に基づいて，①多様な知識体系を安全性ケース手法と統合するとともに，②複数の知識を適切に組み合わせてソフトウェアライフサイクルの全段階において活用する方法を提案することにより，安全性の高いソフトウェアの開発運用を可能とする広範で高度な研究である．

3 関連研究

以下では，メソッド・アーキテクチャと安全性ケース手法ならびにソフトウェア知識体系を説明する．

3.1 メソッド・アーキテクチャ[4]

メソッド・アーキテクチャでは，プラクティスを組み合わせてメソッドを定義することができる．プラクティスは,基本的なプラクティスであるエッセンス・カーネル (Essence Kernel)とエッセンス言語(Essence Language)で記述する．これによって，上位のメソッドを作成するために，プラクティスを他の複数のプラクティスと安全に合成できる．メソッド・アーキテクチャの基本概念は次のとおりである．

【メソッド】複数のプラクティスからメソッドは構成される．メソッドは開発者の日常的な活動を支援するために何が期待されるかだけでなく何が実行されたかを含む計画と結果からなる動的な記述である．

【プラクティス】指定された目的を実行するための反復できる方法がプラクティスである．プラクティスは仕事のある側面を解決するための系統的で検証可能な手順を提供する．プラクティスは複数のメソッドの構成要素である．

【エッセンス・カーネル】エッセンス・カーネルはソフトウェア開発手法の本質的な要素である．エッセンス・カーネルを用いて他の分野のカーネルを定義することができる．

【エッセンス言語】エッセンス言語は，メソッド，プラクティス，カーネルを定義するためのドメイン専用言語である．

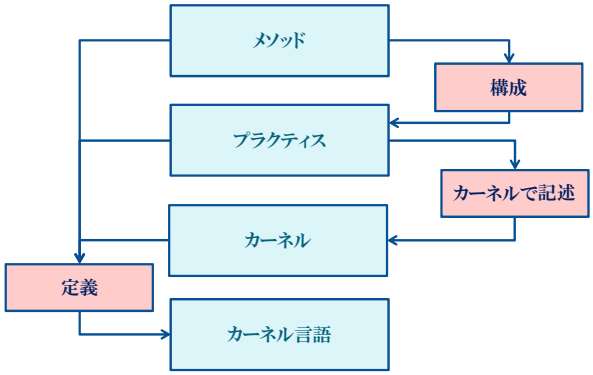

図 1　メソッド・アーキテクチャ

3.2 安全性ケース

安全性ケースの構成要素は，主張（claim），説明（strategy），前提(コンテクスト，context)，証拠

（evidence），未展開である[1][2]．構成要素間の関係は，主張，説明，証拠を関連付ける矢線と，主張と説明をコンテクストと関連付ける白抜きの矢線の２種類である．GSN(Goal Structuring Notation)による安全性ケースの記述例を図 2 に示す．
【主張】システムが達成すべき性質を示す矩形．下位主張や戦略に分解される
【説明】主張の達成を導くために必要となる論証を示す平行四辺形．下位主張や下位説明に分解される．
【コンテクスト】主張や説明が必要となる理由としての外部情報を示す楕円．
【証拠】主張や説明が達成できることを示す証拠
【未展開】具体化していない主張や説明を示すひし形

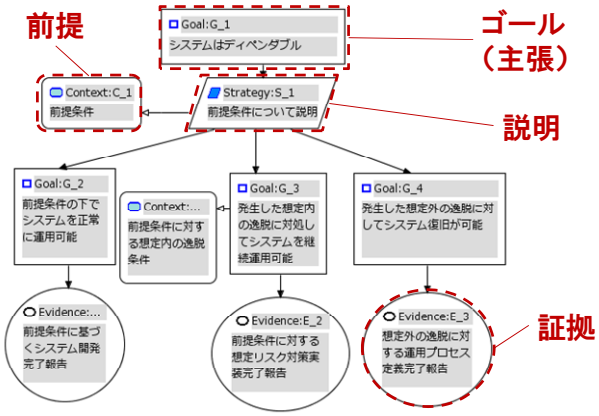

図 2 安全性ケースの例

筆者らは TOGAF における O-DA 規格の策定を推進している[6]．O-DA の高保証アーキテクチャ開発手法（Assured Architecture Development Method, AADM）では，①アーキテクチャリポジトリによる証拠文書と安全性ケースの管理手法，②主張間の優先順位の合意形成手法，③安全性ケースのスコープ定義手法，④主張の定量的評価尺度の定義手法，⑤安全性ケース作成能力の評価手法，⑥安全性ケースレビュ手法，⑦安全性ケース統合手法，⑧運用ならびに開発プロセスに対する安全性ケース作成手法，⑨安全性ケースと障害分析，リスク管理手法との統合手法，⑩安全性ケースの主張とシステム要求の追跡管理手法を確立する必要があるとされている．
これらの課題を解決するため，筆者らは安全性ケースのパターン化ならびに開発運用プロセスに対する安全性ケースの作成手法の研究を進めている[18]-[40].
以下では，3.3 から 3.10 でソフトウェア知識体系の概要を述べ，3.11 で共通性についての比較結果と課題を明らかにする．

3.3 SWEBOK[7]

ソフトウェアエンジニアリング基礎知識体系 SWEBOK (Guide to the Software Engineering Body of Knowledge (SWEBOK V3)では，知識領域(Knowledge Area) ごとに，①概要(Introduction)，②知識領域の構成要素（トピックス），③トピックス参考文献対照表，④推奨参考文献，⑤関連文献一覧を記述している．
SWEBOK では知識領域をトピックスに，2～3 階層で階層的に分解している．
この階層分解は特定のアプリケーション分野，ビジネス利用，開発手法などに依存しないように構成されている．ただし，SWEBOK では，必要な知識は参考文献を直接参照することを推奨しているので知識を詳細に記述してはいない．
SWEBOK の知識領域は，①ソフトウェア要求，②ソフトウェア設計，③ソフトウェア構築，④ソフトウェアテスト，⑤ソフトウェア保守，⑥ソフトウェア構成管理，⑦ソフトウェア開発管理，⑧ソフトウェア開発プロセス，⑨ソフトウェア開発ツールと方法，⑩ソフトウェア品質である．

3.4 PMBOK[8]

PMBOK のプロセスには知識領域とプロセスグループという２つの側面がある．
PMBOK の知識領域には，①プロジェクト統合管理，②プロジェクトスコープ管理，③プロジェクト時間管理，④プロジェクト経費管理，⑤プロジェクト品質管理，⑥プロジェクト人材管理，⑦プロジェクトコミュニケーション管理，⑧プロジェクトリスク管理，⑨プロジェクト調達管理がある．プロセスグループには，①開始，②計画，③実行，④監視制御，⑤終了がある．

3.5 CMMI[9]

CMMI (Capability Maturity Model Integration) モデルは組織が開発プロセスを改善するための実践知識を体系化している．CMMI-Dev.V1.3 では，プロセス領域について，目的，概要，関連プロセス領域，具体的ゴール，実践的活動，成果物の事例，一般的なゴール，一般的活動を記述している．

3.6 BABOK[10]

BABOK 2.0 では，ビジネス分析知識を①知識領域，②作業，③技法の３階層によって体系的に記述している．知識領域については，①知識領域の定義，②関連する作業，③関連する技法，④知識領域への入力，と⑤知識領域からの出力を記述している．
作業については，①作業の目的，②作業内容の記述，③関連するステークホルダ，④作業への入力，⑤作業からの出力，⑥作業の要素作業，⑦作業で活用される技法を記述している．

3.7 REBOK[11]

要求工学知識体系 REBOK(Requirements Engineering Body Of Knowledge) はユーザとベンダが協調した要求開発を実践するための知識体系であり，以下の特徴がある．
(1) ベンダ，ユーザを問わない共通の知識体系
(2) 要求アナリストに加えて，エンドユーザ，経営者など，要求開発に関与するステークホルダが，必要に応じて習得すべき範囲と水準を整理した知識体系
(3) ビジネス要求，システム要求，ソフトウェア要求の 3 つのスコープに応じた知識体系

(4) エンタープライズシステム, 組込みシステムの要求開発に共通する技術の知識体系．ただし，ドメイン固有の知識は個別に定義

3.8 ITIL[12][13]

ITIL(IT Infrastructure Library)では，目的に適した信頼性の高い安定したサービスを顧客に提供し，事業から信頼できるプロバイダとみなされるように，サービスマネジメントにおける実践的な知識を提供している．ITIL にはサービス・ライフサイクル全体を網羅するプロセスベースのフレームワークであり，戦略，設計，移行，運用，継続的改善という 5 プロセスからなる．

3.9 TOGAF[5][6]

TOGAF V9 は 7 部から構成されている．第 1 部では，主要概念の説明と用語が定義されている．第 2 部は，エンタープライズ・アーキテクチャを開発するための段階的手法であるアーキテクチャ開発手法 ADM が説明される. ADM の説明では, 概要, 目的, 手順，入力，出力を記述している．

第 3 部では, ADM を適用するためのガイドラインと技法を説明している．

第 4 部では，アーキテクチャコンテンツ・フレームワークを説明している．アーキテクチャコンテンツ・フレームワークには，メタモデル，再利用のためのアーキテクチャ･ビルディング･ブロック, ADM の各工程の生産物が含まれる．第 5 部では，エンタープライズ・アーキテクチャ活動の生産物を格納する分類体系であるエンタープライズ・コンティニュームとツールを説明する．エンタープライズ・コンティニュームは，エンタープライズ・アーキテクチャで生産される全情報を体系的に分類して関係付けて格納できる仕組みのことだと考えればいい．第 6 部では TOGAF 参照モデルがファウンデーション・アーキテクチャと統合情報基盤参照モデルとして説明される．エンタープライズ・コンティニュームと参照モデルによって，ビジネス・ケイパビリティからエンタープライズ・アーキテクチャの実践状況を獲得して，ビジネスの現状をビジネス・ビジョンに対して提示できるようになる．

最後に, 第 7 部では, EA 活動の運営に必要な組織, プロセス，スキル，ロール，責任などが，アーキテクチャ・ケイパビリティ・フレームワークとして説明される．

3.10 SQuaRE[14]

SQuaRE (Software Product Quality Requirements and Evaluation)は, ソフトウェア製品品質要求評価のための新たなソフトウェア品質要求の標準である．SQuaRE では，スコープ，適合性，規範的参考文献，用語定義，ソフトウェア品質要求フレームワーク，品質要求の要求を記述している．

ソフトウェア品質要求フレームワークでは，目的，ソフトウェアとシステム，関係者と関係者の要求，ソフトウェアの性質，ソフトウェア品質測定モデル，ソフトウェア品質要求，システム要求の構成，品質要求ライフサイクルモデルを記述している．

品質要求の要求では，一般要求，ステークホルダ要求，ソフトウェア要求を記述している．

3.11 知識体系の比較[16]

ソフトウェア知識体系を対象として共通的な記述項目を分析した結果を表 1 にまとめる．この結果から知識体系間で共通する部分があること，したがって適切に統合する必要があることが分かる．このため，本稿で指摘しているように，メソッド・アーキテクチャによって，横断的に知識を記述し適切かつ安全に結合する手法が必要である．

4 研究課題

安全性ケース手法の開発運用現場への導入容易性を向上することによって，重要安全ソフトウェアの開発と運用の効率化・高品質化を支援する手法を提案するとともに，支援ツールを開発する．ソフトウェアの開発から運用まで含めたライフサイクル全般に対する安全性ケース手法を実現するために，以下の三つの研究課題を掲げる．

4.1 安全性ケース手法に基づくソフトウェアの開発運用プロセスの高安全化

具体的なソフトウェアの開発運用事例に基づいて, 安全性ケース手法の適用上の課題を明らかにするとともに，効果的な解決策を提示する必要がある．

たとえば，O-DAのAADM(Assured Architecture Development Method)を用いて，事業ビジョンからはじめて情報システムが扱うデータとアプリケーション機能ならびに使用されるIT技術の構成，システムの実装および運用シナリオに対して安全性ケースを記述して重要安全性を分析し保証できることを実証する．また，BABOKやITILなど，TOGAF以外の知識体系に対しても, O-DAのような安全性拡張についての手法を提案する．

安全性を保証するためにはシステムリスクを可能な限り緩和できるような安全対策要求がソフトウェアで実装されていることを確認する必要がある．そこで，①システム要求やシステム設計に対してリスクを列挙するとともに，安全性ケースの完全性をレビュする手法や，②システム障害に対するオントロジーや事例ベースを用意することにより，安全性要求・設計に抜け漏れがないことを確認する方法を提案する．

4.2 メソッド・アーキテクチャによる知識体系の再構築

上述した安全性ケース手法と現場で活用されている実践的な知識体系の高安全化手法を，知識体系横断的に融合することで，開発運用現場へ円滑に展開する方法を提示する．

具体的には，ソフトウェア工学知識体系，要求工学知識体系，プロジェクトマネジメント知識体系，運用知識体系などの知識をメソッド・アーキテクチャ手法に基づいて系統的に融合できる安全性の高いソフトウェアの妥当性確認・保証方法を提案する．とくに安全性ケース手法では証拠に基づいて，安全

性についての意図をソフトウェアやその開発運用プロセスが保証できることを客観的に説明して確認することができる．したがって本手法の研究では，①異なる知識体系に共通する客観的な証拠に基づいて横断的に統合するとともに，②事例研究によってその有効性を実証する．

4.3 支援ツールの開発

本研究で提案するメソッド・アーキテクチャに基づく重要安全ソフトウェア開発運用手法を効果的に支援するツールの設計と試作を行う．

安全性ケースの作成を支援するエディタ[41]などのツールにおいては，適用事例に基づく安全性ケースパターンや障害パターンをリポジトリで蓄積管理し，再利用を可能とする．また安全性の確信度や議論の完全性の分析を支援する機能の実現を目指す．また，異なる知識体系の融合や安全性ケースの再利用で発生する用語や知識の矛盾問題を解消するため，オントロジーを開発する．

これらのリポジトリやオントロジーには，安全性ケース手法によって高安全化された各種の知識体系を適用するためのガイドや教材も含んでおり，メソッド・アーキテクチャ手法によるソフトウェア開発運用手法の教育および普及に利用する．

5 考察

5.1 実用性と学術性

本研究は，メソッド・アーキテクチャ手法に基づいてソフトウェア開発運用の現場で活用されている複数の知識体系を安全性ケース手法に適切に融合できる方法を提案する実用性の高い実践的研究である．複合的なソフトウェア開発運用知識体系と安全性ケースの横断的な統合化理論を提案する点に学術的な特徴がある．

5.2 提案手法の展開プロセス

また実際のソフトウェア開発運用の事例研究に基づいて有用な知見を蓄積・再利用することにより，セミナーや講習会を通じて，提案手法の公開と普及を推進することで，社会基盤ソフトウェアの高安全化に貢献する．

5.3 ソフトウェア知識体系の高安全化

国内外の重要安全ソフトウェアの開発運用現場において，安全原則と安全目標に基づくソフトウェア自体の高安全化だけでなく，メソッド・アーキテクチャの概念に基づいてソフトウェア開発運用知識を系統的に再構築することにより，ソフトウェア開発・運用プロセスの高安全化にも貢献する．

6 まとめと今後の課題

本稿では，重要安全ソフトウェアを実践的に開発運用するために，メソッド・アーキテクチャと安全性ケースを用いて，現場で活用されているソフトウェア知識体系を高安全化するとともに適切に統合する取組みについて紹介した．ただし，調査対象とした知識体系の事例は 8 件であり，一般化するためには，他の知識体系についても評価する必要がある．たとえばテストについては安全性ケース（保証ケース）を用いてテストの十分性を確認する手法を構築している[17]ので，メソッド・アーキテクチャによって今回検討した知識体系と統合できる可能性がある．

本稿で提案した手法をまとめると，図 3 のようになる．今後，4 節で提示した課題について研究するとともに，異なる知識体系の技法知識を①どのようにして統合できるか，②安全性ケースを用いてどのように高安全化できるかについても検討していく予定である．

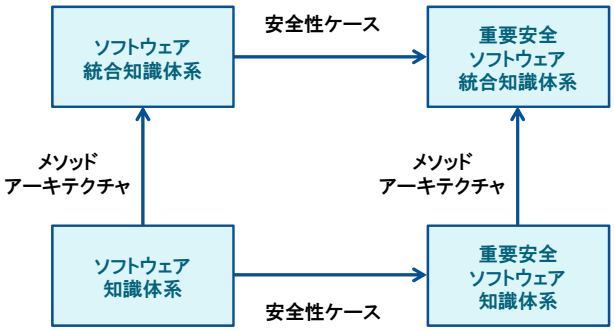

図 3 重要安全ソフトウェア知識体系の統合

参考文献

[1] Kelly, T. P, A Six-Step Method for the Development of Goal Structures, York Software Engineering, 1997
[2] T. Kelly. “Arguing Safety, a Systematic Approach to Managing Safety Cases”. PhD Thesis, Department of Computer Science, University of York, 1998
[3] Robin Bloomfield and Peter Bishop, Safety and Assurance Cases: Past, Present and Possible Future – an Adelard Perspective, 2010
[4] OMG, Essence – Kernel and Language for Software Engineering Methods, ad/2013-02-01, http://www.omg.org/spec/Essence/1.0, 2013
[5] THE Open GROUP, TOGAF V.9 A Pocket Guide, 2008
[6] The Open Group, The Open Group Real-Time & Embedded Systems Forum, Dependability through Assuredness™ Standard (O-DA), 2013
[7] Guide to the Software Engineering Body of Knowledge (SWEBOK V3), http://www.computer.org/portal/web/swebok/home
[8] PMBOK ガイド, http://www.pmi.org/
[9]CMMI, CMU/SEI-2010-TR-033, 2010
[10] IIBA 日本支部，ビジネスアナリシス知識体系ガイド, 2009
[11] 情報サービス産業協会 REBOK 企画 WG 編， 要求工学知識体系 REBOK(Requirements Engineering Body Of Knowledge)，近代科学社，2011
[12] ITIL, itSMF japan http://www.itsmf-japan.org/itil
[13] iTSMF, ITIL V3 Foundation Handbook, 2009
[14] Jorgen Boegh, A New Standard for Quality Requirements, IEEE Software, pp.20-27, January/ February, 2008.
[15] Ivar Jacobson, Pan Wei Ng, Paul E. McMahon, Ian Spence, and Svante Lidman, “The Essence of Software Engineering – Applying the SEMAT Kernel”, Addison-Wesley Pearson Education, 2013
[16] 山本修一郎,ソフトウェア知識体系のメタモデルについての一考察，人工知能学会 第12回知識流通ネットワーク研究会, SIG-KSN-012-02,

http://www4.atpages.jp/sigksn/conf12/SIG-KSN-012-02.pdf
[17]山本修一郎, 保証ケース手法に基づくテスト十分性に関する合意形成手法の提案, 人工知能学会 第12回知識流通ネットワーク研究会, SIG-KSN-013-02, http://www4.atpages.jp/sigksn/conf13/SIG-KSN-013-02.pdf
[18] 山本修一郎, 松野裕, ディペンダビリティケース作成法に関する一考察, KBSE研究会, IEICE-112, vol. IEICE-SS-164, No. IEICE-KBSE-165, pp.61-66, 2012
[19] 松野 裕, 高井利憲, ヴァイセ パテゥー, 山本修一郎, アシュアランスケース構築法の提案, KBSE 研究会, 2012
[20] 松野裕, 山本修一郎,ユースケース分析に基づくディペンダビリティケース作成法の提案, KBSE研究会, IEICE-112, vol. IEICE-KBSE-419, KBSE2012-61, pp.19-24
[21] 高間翔太, 松野裕, 山本修一郎, スーパーコンピュータ運用手順に対するディペンダビリティの確認手法の提案, 信学技報, vol. 112, no. 165, KBSE2012-18, pp. 37-42 2012
[22] 高間翔太, 松野裕, 山本修一郎, ディペンダビリティ・コンテクストの推定手法の提案, KBSE 研究会, 信学技報, vol. 112, no. 314, KBSE2012-42, pp. 25-30, 2012
[23] 徳野達也, 松野裕, 山本修一郎, エンタープライズアーキテクチャ開発プロセスに対するディペンダビリティケース作成法の提案, 信学技報, vol. 112, no. 165, KBSE2012-36, pp. 145-150 2012
[24] 徳野達也, 松野裕, 山本修一郎, TOGAF NEXT に対する ADM プロセステンプレートの提案, KBSE 研究会, 信学技報, vol. 112, no. 314, KBSE2012-55, pp. 103-108, 2012
[25] 山本修一郎, 松野裕, ディペンダビリティケース分解パターンについての考察, KBSE 研究会, 信学技報, vol. 112, no. 496, KBSE2012-80, pp. 67-72, 2013
[26] 山本修一郎, 松野裕, ディペンダビリティケースへの責任属性導入法の検討, KBSE 研究会, 信学技報, vol. 112, no. 314, KBSE2012-52, pp. 85-90, 2012
[27] 松野裕, ヴァイセ バトゥ, 山本修一郎, アシュアランスケースへの構造化文書の適用に関する調査, 信学技報, vol. 112, no. 165, KBSE2012-20, pp. 49-54, 2012
[28] 猿渡卓也, 松野裕, 星野隆, 山本修一郎, Modular GSN の定式化, KBSE研究会, 信学技報 vol.112, No.165, pp.151-156, 2012
[29] Shuichiro Yamamoto, Yutaka Matsuno, d* framework: Inter-Dependency Model for Dependability, DSN 2012
[30] Shuichiro Yamamoto, Yutaka Matsuno, A review method based on a matrix interpretation of GSN, JCKBSE2012
[31] Yutaka Matsuno, Shuichiro Yamamoto, Consensus Building and In-operation Assurance For Service Dependability?, CD-ARES 2012, LNCS 7465, pp.639-653.
[32] Shuichiro Yamamoto, Tomoko Kaneko, Hidehiko Tanaka, A Proposal on Security Case based on Common Criteria, Asia ARES 2013
[33] Shuichiro Yamamoto, Yutaka Matsuno , An Evaluation of Argument Patterns to Reduce Pitfalls of Applying Assurance Case, 1st International Workshop on Assurance Cases for Software-intensive Systems (Assure 2013)
[34] Yutaka Matsuno , Shuichiro Yamamoto, An Implementation of GSN Community Standard, 1st International Workshop on Assurance Cases for Software-intensive Systems (Assure 2013)
[35] 松野裕, 高井利憲, 山本修一郎, D-Case 入門, ～ディペンダビリティ・ケースを書いてみよう！～, ダイテックホールディング, 2012, ISBN 978-4-86293-079-8
[36] 松野裕, 山本修一郎, 実践 D-Case,～ディペンダビリティ・ケースを活用しよう！～, ダイテックホールディング, 2013, ISBN 978-4-86293-091-0
[37]山本 修一郎, 主張と証拠, 978-4-86293-095-8, ダイテックオンデマンド出版, 2013
[38]山本 修一郎, 保証ケース作成上の落とし穴 : 要求工学, (2013/3/17) - Kindle
[39]山本 修一郎, 保証ケース議論分解パターン : 要求工学, (2013/3/17) - Kindle
[40]山本 修一郎, システムとソフトウェアの保証ケースの動向 : 要求工学, (2013/3/18) - Kindle
[41] D-Case エディタ, http://www.dependable-os.net/tech/D-CaseEditor/

表 1　ソフトウェア知識体系の共通性分析

	SWEBOK	PMBOK	CMMI	BABOK	REBOK	ITIL	TOGAF	SQuaRE
SWEBOK	ソフト開発	開発プロジェクト	開発管理	要求工学	要求工学	設計・移行	ソリューション実装監督	ソフト品質
PMBOK		プロジェクト管理	開発管理	要求管理	リスク管理	運用管理	EA 管理	品質改善プロセス
CMMI			開発，調達，サービス	プロセス改善	プロセス改善	サービス成熟度	EA 成熟度	品質改善プロセス
BABOK				ビジネス分析	ビジネス要求	ビジネス要求	ビジネスアーキテクチャ	品質保証
REBOK					要求工学	運用要求	要求管理	品質要求
ITIL						サービス	運用監視	サービス品質
TOGAF							EA	EA 品質
SQuaRE								ソフト品質