人工知能学会
第16回知識流通ネットワーク研究会
SIG-KSN-016-06

システム安全検証知識の体系化について

山本修一郎

名古屋大学
愛知県名古屋市千種区不老町

On A Systematic Approach for the System Safety Verification and Validation Knowledge

Shuichiro Yamamoto

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

概要

システムの安全性を検証するために必要と考えられる知識を整理すると，基礎知識，計画知識，分析設計知識，検証確認知識に分類できる．本稿では現代システムのための安全検証知識を体系化する試みを紹介する．

In this paper, the knowledge for system safety verification and validation will be categorized by the basic knowledge, the plan knowledge, the analysis and design knowledge, and the verification and validation knowledge. This paper also describes a systematic approach for constructing the knowledge for the modern safety verification and validation.

1. はじめに

安全なシステムを構築するための手法をまとめた書籍として，セーフウェア[1]がある．しかし，同書が 1990 年代に執筆されたため，2011 年に策定された ISO26262 における機能安全の概念や，エンタープライズ系情報システムの障害事例について言及されてはいない．このような現代システムの安全対策知識を知らなければプロジェクトリスクに総合的に対応できず，プロジェクトを成功に導くことはできない．このため，本稿では，現代システムを安全に開発・運用するために必要となるシステム安全検証知識を体系的に構築する試みを紹介する．また，この知識体系を習得するための研修コースの構成事例についても述べる．

2. システム安全検証知識の構成

システムの安全性を検証するために必要と考えられる知識を整理すると，基礎知識，計画知識，分析設計知識，検証確認知識に分類できる．安全なシステムを実現するためには，システム安全の必要性を理解し障害に対処することの重要性を理解することが大切である．したがって，システム安全計画を立案し，システムの安全分析と安全設計を実施する必要がある．このため，システム安全に関する基礎知識の理解が重要になる．また，システム安全をライフサイクルプロセスの観点からとらえる必要がある．さらに，開発したシステムの安全性を検証確認する必要がある．以下では，これら 10 項目の知識の内容について概説する．

2.1 現代システムのリスクと障害の根本原因

システム安全検証では，まず，基礎知識として，現代システムのリスクと障害の根本原因の整理法を学ぶ必要がある．現代システムのリスクには，大規模複雑化する IT システムのリスクと，システム開発技術のリスクの 2 つがある[2]．大規模複雑な IT システムでは，要素システムが異なる組織に

よって独立に所有され管理されるために，相互連携作用による影響を事前に予測できないという「システム連携リスク」や，人間，組織，社会，政治的要因を技術的要因に加えて総合的にとらえる「社会技術的システムリスク」を知る必要がある．これらのリスクを識別できなければ，対処できないからである．

システム開発技術リスクでは，急速な技術変化や予測不可能な技術の出現という「技術変化リスク」，ユーザー，発注者，開発者，保守者などのステークホルダ間で顕在化する意図や要求の矛盾に起因する「ディペンダビリティリスク」や，独立なシステムはどこにもなく，すべてのシステムが一体となってエコシステム（ECOSYSTEM）を構成することに起因する「相互依存性リスク」とそれらへの対策の必要性を理解する必要がある．

障害の根本原因では，最近のネット地図騒動，クラウドサーバ事件，高速鉄道停止事件，ネット通販騒動など，社会問題化したシステム障害事例の根本原因と再発防止策の事例を理解することで，発生事象，根本原因，再発防止策を系統的に整理して共有するための基礎知識を学習できる．これにより，問題事象と原因，再発防止策を関連付けて共有する事例ベースの作成知識を習得できる．

2.2　システム安全用語の基礎知識

システム安全用語の基礎知識では，ディペンダビリティ[3]と,その属性，脅威，対処手段についての用語と，システム安全の基本概念を学習する必要がある．ディペンダビリティ属性には，可用性（Availability），信頼性（Reliability），安全性（Safety），機密性（Confidentiality），一貫性（Integrity），保守性（Maintainability）がある．

ディペンダビリティの脅威として，フォールト（Faults），エラー（Errors），故障（Failures）がある．フォールトが原因，エラーが状態，故障が結果である．しかし，実際のプロジェクト組織などでは，これらが区別されずに用いられることも多いので，用語の意味を明確に定義する必要がある．

ディペンダビリティの脅威への対抗手段として，障害回避（Fault Prevention），障害耐性（Fault Tolerance），障害除去（Fault Removal），障害予測（Fault Forecasting）がある．

システム安全の基本概念では，安全を考えた設計，システム全体の安全性，システム安全とトレードオフ，擬陽性と擬陰性，ソフトウェアとシステム安全，障害対策などの概念を理解する必要がある．たとえば，ソフトウェアの実行がシステムのハザードに寄与しないときシステムは安全である[1]．また，システムの安全な状態を危険状態だと誤判定することを擬陽性という．これに対して，システムが危険な状態であるのに安全状態だと誤判定することを擬陰性という．イプシロンロケットの最初の発射では，システムが擬陽性だと判断したために，発射されなかった．このように，擬陽性などの概念を理解しておくことでシステム安全を考慮したシステム設計や検証が可能になる．

2.3　システム安全原則

システム安全原則の知識では，事故原因の階層モデル，安全文化，ヒューマンファクター，安全原則の具体的な事例について習得する．たとえば，事故の因果モデルでは，①安全文化の欠如が組織管理の不備をもたらす，② 不適切な設計管理と不適切な運用管理が作業品質の低下を導く．③顕在化プロセス（故障，不安全行為）が発生する，④結果として人の健康財産への損害が発生するという4段階を提示している[4].

組織構成員の安全に対する一般的な態度と取り組みのことを，組織における「安全文化」という．事故を引き起こす安全文化の欠如例，自信過剰と自己満足，安全に低い優先順位を割り当てるとどうなるか，安全性のトレードオフなどについての知識を習得する必要がある．

ヒューマンファクターでは，過誤強制情況（Error Forcing Context）について理解しておくことが重要である．個人的，環境的，社会的要因を背景として過誤強制情況が形成されると，不安全行為を誘発することが多い．すなわち，時間的な余裕がなく，あいまいな情報しか与えられない条件下では，どんな人でもほぼ確実に誤判断するのである．ヒューマンエラーを除去するためには，過誤強制情況を識別して除去する必要がある．

2.4　システム安全プログラムの策定

システム安全プログラムは，組織的にシステムの安全性についての計画の立案と，実行管理ならびに，実施結果を評価することである．

組織的な安全プログラムの構築知識として，①米国の航空安全報告システム（ASRS, Aviation Safety Reporting System），②JAXA の安全プログラム，③O-DA の高保証性アーキテクチャ開発手法[5]（オープングループが 2013 年 8 月に標準化したディペンダビリティに関する技術標準），④IT サービス継続性管理（たとえば，ITIL）などの事例について理解する．これらの事例を横断的に理解することによって，プロジェクトにおけるシステム

安全プログラムの構築法についての視野を広げることができる．

なお, O-DA は筆者らも参加して日本が中心となって提案してオープングループで採択された標準規格である．

2.5 ハザード分析技法

ハザード分析の技法知識には，ハザード分析プロセスと，ハザード分析モデルの知識がある．代表的なハザード分析モデルには，①FTA（Fault Tree Analysis），②ETA（Event Tree Analysis），③FMEA（Failure Modes and Effects Analysis），④HAZOP（Hazard and Operability Analysis），⑤ STPA（System Theoretic Process Analysis）がある．これらの技法はハードウェアやプラントを対象に提案されてきたが，ソフトウェアシステムに対しても適用されてきている[6][7]. STPAはシステム理論による事故因果モデルSTAMP（Systems-Theoretic Accident Modeling and Processes）に基づく新しいハザード分析手法である[8]．STPAでは，FTAなどが目指した「障害原因を除去する」ことから，「システム動作についての安全性制約を強制する」ことに，焦点を転換した点に特徴がある．

これらの多様な分析技法知識の相互関係や使い分け，組合せ方についても理解する必要がある．

2.6 システム安全要求分析

システム安全要求分析知識には，①要求仕様作成，②ソフトウェア安全要求分析，③要求仕様の完全性基準がある．システムが動作する状況下で，ソフトウェアの安全な挙動を規定するために要求仕様が完全である必要がある．ソフトウェア安全プロセスにおけるソフトウェア安全要求分析の位置づけを表 1 に示す[9]．この表から分かるように，ソフトウェアが安全であるというためには，システムハザードが発生しないようにソフトウェア安全要求が定義されていることが必要である．したがって，ハザード分析の知識がなければ，ソフトウェアの安全性を保証できないのである．このように，ハザード分析と安全要求の知識は現代システム開発にとって，不可欠の知識になっている．

2.7 システム安全分析・設計検証

システム安全分析・設計検証知識として，独立検証確認についての標準技法と適用事例がある[10][11][12].

国際標準では，検証（Verification）とは，システムが正しく作られていることに対する証拠を提示することである．これに対して，妥当性確認（Validation）とは，作成されたシステムが正しいこと対する証拠を提示することである. 国内では，検証と妥当性確認が混同されることが多いので注意する必要がある．そもそもこのような混乱が生じる原因は，安全技術に関するきちんとした知識がエンジニアに提供されていないことにある．

また，独立性については，①重要なプロジェクトに対して，全社的な観点から投資する財政独立性，②重要性アセスメントに従って，標準作業構造に基づいてスコープとタスクを定義する技術独立性，③Office of Safety and Mission Assurance（OSMA）による機能的な管理をする管理独立性がある．とくに，技術独立性が理解できないと，安全性の検証・確認ができないことを注意しておく．この理由は，独自技術で最高の安全性を達成したと主張しても，その技術が第三者によって客観的に評価できなければ，その安全性が達成されたかどうか不明だからである．標準的な技術によって第三者が評価できる安全性が求められている．ISO26262[13]で機能安全についての安全管理が標準化されていることの理由もここにある．第三者による検証確認については文献[14]に説明がある．

表 1　ソフトウェア安全プロセス

プロセス	説明
ソフトウェアハザード分析	システムハザードに基づいて，ハードウェアとソフトウェアのインタフェースについてのハザードまで追跡することによりソフトウェアハザードを識別する
ソフトウェア安全要求分析	ソフトウェアハザードをもたらさないように，ソフトウェアの挙動についての安全要求を明らかにする
一貫性確認	ソフトウェア要求仕様とソフトウェアの安全要求との一貫性を確認する
完全性確認	人間とシステムのインタフェース仕様を含めたソフトウェア要求仕様の完全性を確認する

2.8 システム安全設計法

システム安全設計法の知識には，安全設計プロセスと，安全設計技法がある．安全設計プロセスでは，①過去の障害事例に学ぶためのプロセス，②設計対象に対するハザード分析に基づくプロセス，③設計変更プロセスがある．

事例に学ぶためには，①一般的な安全設計原理に基づく基準，②再発防止実施規定，③ソフトウェア設計特性，④設計チェックリストとして整理しておく必要がある．ハザード分析では，①安全制約条件とハザードに基づいて，設計基準，要件，試験要件，対人インタフェース要件を作成する方法と，②安全要件と制約条件をコードまで追跡することにより，ハザードを制御する方法がある．

識別されていないハザードが顕在化すると，設計変更が必要になる．したがって設計変更プロセスでは，①条件変更と判断の妥当性審査，②設計判断の記録，③インシデント管理を考慮する必要がある．審査するためには設計判断が記録されていなくてはならない．したがって設計判断の記録では，①前提条件，②判断基準，③設計判断理由を明記しておく必要がある．ハザードの顕在化に伴うインシデント管理では，設計判断根拠の妥当性分析と，障害に学ぶための教訓材料の記録が必要である．

2.9 システム安全保証技法

システム安全設計・保証技法の知識として，安全性ケースの基本概念と標準化動向がある．安全ケースを構成する基本要素には，安全性についての主張とその主張が成立することを前提条件と証拠に基づいて説明するための論理的な構造がある．安全性ケースは，上述した機能安全規格でも推奨されており，欧米を中心として自動車分野や医療分野で，最近注目されている．安全性ケースを国内で導入する場合の最大の障壁は，前提と証拠に基づいて主張を論理的に説明するという基礎的な訓練をほとんどの要員が受けていないことである．わが国では論理的な文章の訓練が決定的に不足しているので，安全性ケースの構成要素を理解するだけではなく，主張を明晰に論じる訓練が必要である[15]. 安全性ケースの作成で混乱が生じる理由は，安全性ケースの前提となる開発対象システムやそのコンテクストが明確に理解できていないためである．したがって安全性ケースの作成の準備段階として，システムコンテクストの分析についても教育する必要がある．システムコンテクストには上述したシステム安全原則や，機能安全規格などの法制度も含まれる．

安全性ケースに関連する標準規格には，ISO/IEC 15026[16]， ISO26262[13]，オープングループによる O-DA[5]などがある．

2.10 システム安全保証プロセス

システム安全保証プロセス知識には，①説明責任，②システム安全組織，③システム安全文書，④安全情報システムの知識がある．

システム安全保証における説明責任を遂行するためには，①システムの安全対策を命令・決定する権利を持つ組織の構成員が，②安全対策の結果を計測・評価することが必要である．

システム安全組織を確立するためには，システム安全部門を設置することにより，安全管理活動を実施する．安全保証プロセスは，①安全要求とハザードの識別，②安全設計，③安全コンポーネントの分析，④安全審査報告，⑤システム安全の最終報告から構成される．

システム安全文書として，システム安全組織，システム安全計画，システム安全評価基準，システム安全データ，ハザード分析，などを記述したシステム安全プログラムを策定する．

安全プログラムを円滑に実施するためには，ハザードとその対策を文書化し，追跡するための組織横断的な安全情報システムを用意する必要がある．代表的な安全情報システムとして ASRS[17]がある. 安全情報システムの留意点としては, すべてのハザードを記録すること，決定の内容と，その理由，問題の優先順位が重要になる．

3. 知識の習得方法

上述したシステム安全検証知識を習得する場合，個人ごとの座学だけでは不十分である．学習した知識をグループ形式で実践に近い形式で討論することにより，疑問点が明確になるだけでなく，他の参加者から新たな気づきを得ることができる．したがって，知識項目ごとに数時間のグループ討論と発表による意見交換を実施する必要がある．しかし、一方で、体系化された知識を習得するだけの十分な時間的余裕がない場合もある。したがって、安全検証知識の概要を簡潔に理解するための、簡易な習得方法を提供することも必要であろう。このため、研修の実施例では、①安全検証知識の概要を簡潔に習得できる基礎研修、②安全検証知識を講義をグループ討論でしっかり習得する応用研修について、筆者による経験事例を紹介しよう。

4. 研修の実施例

本稿で紹介したシステムの安全検証知識体系に基づいて，ビジネスコミュニケーション社と協力

して教材を開発した．具体的な研修の構成を表 2 に示す．この知識構成に従って、①講義形式の基礎研修と，②講義とグループ演習からなる応用研修を実施した．

4.1 基礎研修事例

基礎研修では，表 2 の基礎知識と計画知識を各 3 時間で 2 回に分けて講義を実施する．

この教材に基づく基礎研修を大手通信サービス企業の 18 名の担当者に対して実施した．

講義内容に対するアンケートでは，講義内容が参考になったと全員が回答した．内訳は「非常に参考になった」が 8 名，「参考になった」が 10 名であった．具体的な参加者の意見を，①知識の体系化，②事例，③業務との関係に分類して紹介すると以下のようになる．

【知識の体系化】
・様々な角度から安全を確認する手段があることが勉強になった．
・システム安全の評価基準が種々あることを知り興味がわいた．
・システム安全の考え方がよく理解できた．今まで考えていなかった知識をたくさん知った．
・システム機能の継続性についてこれまで体系的に学んだことがなかった．
・通常の業務では知る機会のない知識が多く，参考になった．

【事例】
・分かりやすい事例と体系的な考え方で，日常の混乱した頭を整理する機会になった．
・事例が多く，イメージがつかみやすかった．
・多くの事例を取り上げて，その中での課題・解決の考え方・過程の構成法がよく理解できた．

【業務との関係】
・サービスの信頼性・作業の安全性を確認する業務に就いていたので，参考になった．
・実際の業務でシステムを運用しているので，今の運用での不足点が明確になった．
・講義で学んだように，例外や想定外の事象を予見しておくプロセスを充実させていきたい．
・運用業務に就いていないので，参考になった．

とくに，「運用業務に就いていないので参考になった」という意見を補足すると，次のようになる．サービスの運用で，欠陥がどのような仕組みで発生するか，それに対してどのように対処するかという知識が，運用の担当者にはよく理解できる．これに対して，運用を担当していない方でも，今回の安全知識を学ぶことで，サービス運用の安全性とリスク識別・分析・対策の知識が参考になったということであろう．

4.2 応用研修事例

応用研修では，表 2 のテーマごとに，4 時間の研修を 10 回実施した．研修の内容は，①講義，②グループ演習，③発表，④講評である．このため，テーマごとにグループ演習用の課題事例を用意した．

講師による講義（90 分）では，テーマについて基本概念を講述する．グループ演習（90 分）では，学習した講義内容に基づいて，グループごとに課題事例について討論する．発表（45 分）では，グループ演習の結果をグループごとに発表して意見を交換する．講師による講評（15 分）では，発表に対して講師から総合評価を実施する．

この教材に基づいて，大手通信サービス企業に対する研修を実施することにより，教材内容の妥当性を確認している．

表 2　システム安全検証知識の構成

分類	知識	概要
基礎	現代システムのリスクと障害の根本原因	社会基盤システムの危険性，システム障害事例
	システム安全用語の基礎知識	欠陥，エラー，障害，故障
計画	システム安全原則	安全文化，安全ビジョンの定義手法
	システム安全プログラムの策定	組織的な安全プログラムの構築事例
分析設計	ハザード分析技法	ハザード分析プロセスとハザード分析モデル（FTA，FMEA，HAZOP，STPA）
	システム安全要求分析	ディペンダビリティ要求，安全要求仕様の構成
	システム安全分析・設計検証	独立検証確認，IV&V 事例
検証確認	システム安全設計法	安全設計の構成要素，安全設計プロセス
	システム安全保証技法	安全性ケースの作成と確認，安全ケースを用いたシステム安全評価
	システム安全保証プロセス	安全保証プロセスの構築と監視

5. おわりに

本稿では，システム安全性について，基本概念，安全要求分析，安全設計，安全性検証・確認と安全性ケースによる保証，安全管理などの知識を体系的に整理して紹介した．これらの知識は従来のソフトウェア工学では，非機能要求とアーキテクチャの観点から安全性に触れる程度であった．ソフトウェア工学の教科書でディペンダビリティについて紹介しているのは，Sommerville[18]だけである．しかし本稿で紹介したように体系的に整理しているわけではない．システムの安全性について教えられていなければ，ソフトウェア開発者が安全なシステムの設計や安全性の検証を実行できないのは当然である．したがって安全なシステムを開発する上で，システム安全検証知識の体系的な教育が必要である．知らなかったから障害が発生してもやむを得ないという態度では社会的な説明責任を果たすことはできない．システム安全検証知識を体系的に訓練すべきである．

またシステム安全性についての書籍として，セーフウェアがよく知られている．しかし，エンタープライズ系大規模情報システム障害については触れていない．また，1995 年の著作でありその後のディペンダビリティ技術の進展と社会情勢の変化を考慮すると，本稿で示したようなシステムの安全性に関する検証確認知識の記述が必要である．また，我が国でのシステム障害事例を体系的な観点から学ぶことが、担当者によるシステムリスク対策能力の向上に寄与すると考える．

参考文献

[1] Nancy Leveson: *Safeware - System Safety and Computers*, Addison-Wesley, 1995, 松原友夫監訳, 「セーフウェア」, 翔泳社, 2009.

[2] Ian Sommerville et al.: “Large-Scale Complex IT Systems”, Communications of The ACM, pp.71-77, Vol. 55, No. 7, 2012.

[3] AVIZZIENIS et al.: “BASIC CONCEPTS AND TAXONOMY OF DEPENDABLE AND SECURE COMPUTING”, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 1, NO. 1, JANUARY-MARCH 2004.

[4] 古田一雄，長崎晋也: 「安全学入門」, 日科技連, 2007.

[5] Open Group: *Dependability through Assuredness™ (O-DA) Framework*, Open Group Standard Real-Time and Embedded Systems, 2013.

[6] Felix Redmill, Morris Chudleigh, James Catmur: *System Safety: HAZOP and Software HAZOP*, John Wiley & Sons, 1999.

[7] Troubitsyna, E.: “Elicitation and Specification of Safety Requirements”, Systems, 2008. ICONS 08. Third International Conference on, pp. 202-207, 2008.

[8] Leveson, N. and Thomas, J.: *An STPA Primer*; http://sunnyday.mit.edu/STPA-Primer-v0.pdf, 2013, アクセス日: 2014年7月3日

[9] 山本修一郎: “要求の完全性”, 連載要求工学,第107回, 月刊ビジネスコミュニケーション, http://www.bcm.co.jp/site/youkyu/youkyu107.html, アクセス日: 2014年7月3日.

[10] JAXA編: 「IV&Vガイドブック【虎の巻】」, JAXA, 2013.3.

[11] IEEE: *1012-2004 - IEEE Standard for Software Verification and Validation*, IEEE, 2005.

[12] NASA: IV&V, http://www.nasa.gov/centers/ivv/about/index.html, アクセス日: 2014年7月3日.

[13] ISO: *ISO26262 Functional Safety*, ISO, 2011.

[14] 山本修一郎: “緊急:今，なぜ第三者検証が必要か”, 連載要求工学, 第70回, 月刊ビジネスコミュニケーション, http://www.bcm.co.jp/site/youkyu/youkyu70.html, アクセス日: 2014年7月3日.

[15] 山本修一郎: “保証ケース作成上の落とし穴”, 連載要求工学, 第94回, http://www.bcm.co.jp/site/youkyu/youkyu94.html, アクセス日: 2014年7月3日.

[16] ISO/IEC: *ISO/IEC 15026-2:2011, Systems and Software engineering - Systems and Software assurance - Part2: Assurance case*, ISO, 2011.

[17] NASA: ASRS(Aviation Safety Reporting System), http://asrs.arc.nasa.gov/index.html, アクセス日: 2014 年 7 月 3 日.

[18] Ian Sommerville: *Software Engineering*, Ninth Edition, Addison-Wesley, 2011