コンシューマデバイスの信頼性確保に向けた取組み

～開発方法論の国際標準化に向けて～

2013 年 9 月

独立行政法人情報処理推進機構
Information-technology Promotion Agency, Japan

目次

1. はじめに

一般利用者が使用するコンシューマデバイス（Consumer Device）のディペンダビリティを確保するための開発方法論（Dependability Assurance Framework For Safety-Sensitive Consumer Devices）に関して、(独)情報処理推進機構 技術本部 ソフトウェア高信頼化センター（以下、IPA/SEC）コンシューマデバイス安全標準化 WG の場を中心に、複数の企業、団体等からの有識者に参画いただいて、国際標準化に向けた取組みを進めてきた。本書では、この国際標準化に向けた提案の要請（RFP[1]）成立に至るまでの取組みの経緯、及びその後の標準化に向けた検討内容について解説する。

1.1. 背景

コンシューマデバイスとは、自動車、サービスロボット、スマート家電、スマートハウス等の利用者が自ら使用する工業製品のことをいう。

コンシューマデバイスは、利用者の生活を更に豊かで利便性の高いものにするため、今後も様々なものが開発・提供され、自動車での自動運転やパーソナルケア用のロボット等のように高機能・インテリジェンス化しつつ、急激に普及していくと考えられる。

一方、コンシューマデバイスは、様々な一般利用者（老若男女、知識・経験・技能の違い等）により、様々な環境（異なった地域・天候・温度・場所等）で利用される。そのため、それらに対応できるような高い安全性や信頼性が求められる。しかも、自動車やパーソナルヘルスケアロボット等においては、単に危険な場合に停止するという安全性だけでなく、利用者にとって必要とされる時には何らかの形で使い続けるという可用性も含めた信頼性が必要とされる。そのため、本書では「セーフティ」ではなく「ディペンダビリティ」との用語を用いている。ちなみに、「ディペンダビリティ」とは、JIS Z8115 によると「アベイラビリティ性能及びこれに影響を与える要因，すなわち信頼性性能，保全性性能及び保全支援能力を記述するために用いられる包括的な用語」と規定されており、一言でいえば「仮に欠陥があってもへこたれない」技術ということができる。

コンシューマデバイスの製造会社は、激化する市場競争のため、ディペンダビリティを確保しつつ、より機能的に優れたコンシューマデバイスを、早期かつ安価に提供することが求められている。しかし、既存の機能安全規格は、自動車に関する国際安全規格である ISO 26262 やパーソナルケアロボットの国際安全規格 ISO 13482 などがあるものの、コンシューマデバイスの開発プロセス、開発の仕組みに関しては、ほとんど言及

1 Request For Proposal

されていない。コンシューマデバイスのディペンダビリティを確保する開発方法論を、個々のドメインではなくドメイン横断的に規定することができれば、共通的な方法論・ツールの普及等にもつながり、ディペンダブルなコンシューマデバイスの拡大・多様化を支援することが期待できる。

1.2. 本書の目的

IPA/SEC ではコンシューマデバイス安全標準化 WG を設置し、自動車、サービスロボット、スマート家電、スマートハウス等の各分野を代表する企業からの有識者により、コンシューマデバイスのドメイン横断的に適用できるような効率的な開発方法論（Dependability Assurance Framework 、以下「DAF」）に関して検討を進め、OMG[2]への提案活動を行ってきた。ちなみに国際標準化として OMG へ提案した理由は、コンシューマデバイスの開発方法論がシステムズエンジニアリングと切り離すことができないものであり、OMG にてシステムズエンジニアリングと関連する多くの国際規格が規定されているためである。

このような標準化活動は、多くの方々の協力と理解により普及し、効果を発揮することが期待できる。このため、本書ではコンシューマデバイスの開発方法論の考え方、経緯等を説明することにより、コンシューマデバイスの開発に携わる方を中心に、より多くの方の理解を得て、開発方法論がより実務的に効果のあるものとして普及することを目的としている。

1.3. 本書の構成

本書の構成は以下のようになっている。読者が、第 2 章、第 3 章での RFP の記載が理解しにくい場合には、第 4 章の解説を併せて読むことにより理解が進むと考えられる。

第 1 章　本書の導入として、コンシューマデバイス等の概説をする

第 2 章　「コンシューマデバイスの課題」に関して、OMG の RFP を引用しつつ概説する

第 3 章　OMG にて 2013 年 3 月に発行された RFP の概要を紹介する

第 4 章　第 2 章、第 3 章で記載した RFP の趣旨、規格提案の方向性等について、ここで改めて解説する

第 5 章　全体のまとめ

---

[2] Object Management Group の略、アーキテクチャやテクノロジーに関する標準を開発する非営利団体

2. コンシューマデバイスの課題

ここでは、コンシューマデバイスの特徴、課題に関して、OMG の RFP [1]に基づき解説する。

2.1. コンシューマデバイスとは

コンシューマデバイスとは、一般消費者が使用する自動車、サービスロボット、スマートハウスなどのスマートシステム、スマート家電、医療機器などの製品群のことである。

コンシューマデバイスの不具合を防ぐことは、消費者の安全のために必要不可欠になっている。これまでの産業機器とは異なり、コンシューマデバイスは幅広く、オープンで変化の激しい環境でも常に頼れることのできる機器でなければならず、それらの機器の安全性、信頼性はもとより可用性までも、必要不可欠な機能要件となっている。表 1 は、産業機械とコンシューマデバイスをいくつかの観点で比較したものである。

表 1　産業機械とコンシューマデバイスの比較

	産業機械	コンシューマデバイス
		 
生産数	少	多
ユーザ	専門家	一般消費者
要求コスト	高	低
メンテナンス	設置現場	ユーザ、 サービスステーション
環境	工場環境 （安定、管理された）	ユーザ環境 （多様）

2.2. 現状の課題

コンシューマデバイスの機能がより複雑になってきているため、その安全性、信頼性、可用性に関しての取り組みが、コンシューマデバイスの製造会社にとってますます重要

になってきている。ディペンダビリティという概念 [2]は、それらシステムの特性の集まりとして定義され、システム開発者にとって開発上の重要な位置を占めるようになってきている。例えば、自動車は安全でなければならないだけではなく、そのドライバーが運転したいときはいつでも利用可能でなければならない。つまり、自動車は安全性と可用性を同時に満たさなければならない。

コンシューマデバイスがディペンダブルであることを保証することは非常に重要であるが、それを明確に示した規格はなく、どのようにディペンダビリティを保証するかを示した基準もない。

現在、コンシューマデバイスの様々な特性（安全性、信頼性など）は IEC61508 や ISO26262 のようなこれまでの産業機械に対して存在する国際標準を基に、それぞれ個別に保証の仕方が議論されている。しかしながら、既存の国際標準は、開発プロセスや開発の仕組みについてはあまり言及していない。例えば、自動車製造会社では、頻繁なシステム開発者同士のコミュニケーションによって自動車機能を開発しているが、開発のプロセスは非常に柔軟で即興的な要素の強いガバナンスの下で開発を進めている。

2.3. 課題の解決策

昨今のコンシューマデバイスに対する開発方法論では、システムズエンジニアリングとアジャイル的な繰り返しによる開発手法をとっており、既存の国際標準で規定されたプロセス要件と現実的な開発プロセスのバランスを保っている。

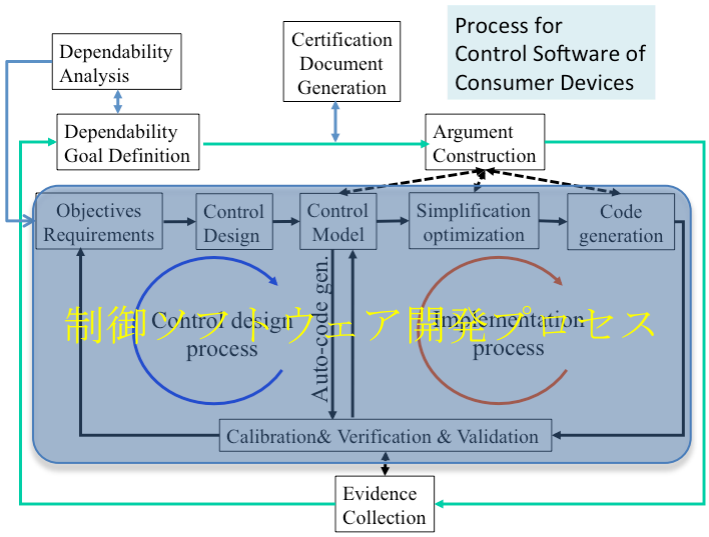

図 1 制御ソフトウェア開発のアジャイル的な繰り返しの例

したがって我々は、以上のような国際標準と実際のコンシューマデバイス開発現場の

ギャップを埋めながら、いかにコンシューマデバイスのディペンダビリティを保証するかを考えていかなければならない。その中には、オープンで、変化の激しい使用環境で問題なく動作する複雑な機能の開発をしつつ、短くなり続けている製品のライフサイクルと機能要件の変化のスピードをサポートすることも含まれている。

図 1 は、制御ソフトウェア開発において、ディペンダビリティを確保するためのアジャイル的な繰り返しプロセスの例を示している。図 1 の中心部には、制御ソフトウェア開発でのモデルベース技術などを活用したアジャイル的な開発プロセスがある。その周辺の 5 つのボックスは、ディペンダビリティを保証するためのプロセスを構成する要素であり、それぞれ以下のようなことを行う。

- Dependability Analysis：ディペンダビリティに関する要求を分析する
- Dependability Goal Definition：ディペンダビリティ保証のための目標（Goal）を設定する
- Argument Construction：ディペンダビリティを保証するための議論（Argument）を構築する
- Evidence Collection：ディペンダビリティを保証していることの証拠（Evidence）を収集する
- Certification Document Generation：ディペンダビリティを保証していることを示す文書を生成する

つまり、ディペンダビリティの保証に向けて、リスクやハザードを分析し、リスクに対処する目標（Goal）を設定し、全体的な議論（Argument）を明らかにして、それがきちんと実現された証拠（Evidence）を収集する、との流れになっている。図 1 はディペンダビリティ保証のためのソフトウェア部分に関する 1 つの事例にすぎないが、このような考え方でコンシューマデバイス全体のディペンダビリティを保証するためのプロセスを規定する必要がある。

3. 提案の要請（RFP）

OMG では、規格の提案を行う前に提案の要請（RFP）が発行される必要がある。RFP は、規格の提案を行いたい組織などが、一定の手順（3.2 を参照）を踏んだのちに RFP 案として提出し、TC（Technical Committee）の投票の結果、正式に発行される。

このような流れは、提案者だけの提案を採用するのではなく、広く提案を受け付けて、より多くの意見をマージするような形で提案を作り上げることを目的としており、OMG 会員からの提案を広く受け付ける。

本章では、OMG での規格提案の流れと、2013 年 3 月に発行されたコンシューマデバイスのディペンダビリティ保証に関する RFP の内容について説明する。

3.1. OMG と国際標準規格

OMG では数多くの国際標準規格の提案を受け付けて、数多くの国際標準規格を公開している。代表的なものとしては、UML や SysML がある。UML は Unified Modeling Language の略で、オブジェクト指向ソフトウェアのモデリング言語の標準規格である。

オブジェクト指向ソフトウェア開発では、モデリング記法として UML を使用している場合が多い。誰もが世界共通の規格を使うことで、UML で記述したソフトウェアモデルが世界中どこでも同じ意味で解釈されるため、誤解や意識の不整合を防ぐことができる。

SysML は System Modeling Language の略で、UML を基に規格化されたシステム記述言語である。UML がオブジェクト指向ソフトウェアを対象にしているのに対して、SysML はハードウェアとソフトウェアを含むシステムを記述することができる。

このように OMG では、記述言語やプロトコルなどを標準規格化して、似たような言語やプロトコルが横行し混乱することを防いでいる。コンシューマデバイスの提案も標準規格になった際は、同様の扱いになる。

3.2. 規格提案の流れ

OMG への提案は、3 ヵ月に 1 度開催されるテクニカルミーティングを中心に、各種提案などの提出、レビュー及び投票などが行われて、幾つかのハードルを超えることで最終的に標準規格発行に漕ぎ着けることになる。提案の流れは以下のようになる。

① 提案を計画している旨のホワイトペーパーをテクニカルミーティングで提出する
② ①の次のテクニカルミーティングで「情報の要請」（RFI : Request for Information）を提出して提案に関する情報を要請する（投票で通過すると、RFI は正式に発行される）

③ RFI に関するレスポンスを受け付けて、3 ヵ月後のテクニカルミーティングでレスポンスを FIX し、どのような提案をすべきかを提示する

④ ③が問題なければ⑤へ、ダメなら次回のテクニカルミーティングでホワイトペーパーを提出して、提案のスコープや手順を明確にする

⑤ 次回のテクニカルミーティングでは、「提案の要請」（RFP）のドラフトを提出する（3 ヵ月間公開）

⑥ 次回のテクニカルミーティングで RFP を提出し、レビューと投票を受ける

⑦ ⑥の投票で通過すると、規格提案を行いたい組織から「提案の意向表明」（LOI：Letter of Intent）を受け付ける（3 ヵ月間）
⑥の投票が不通過なら、次回のテクニカルミーティングで修正した RFP を再度提出する

⑧ LOI 提出締め切り後、約 5 ヵ月後に「初期提案」（Initial Submission）を提出する

⑨ ⑧の直後のテクニカルミーティングで初期提案のレビューを受ける

⑩ 次回のテクニカルミーティングで「変更提案」（Revised Submission）の中間レビューを受ける

⑪ 次回のテクニカルミーティングで正式な変更提案を提出し、レビューと投票を受ける

⑫ ⑪の投票で通過すると 3 ヵ月間公開して、最終投票を受ける。問題が無ければ正式な標準規格の初期バージョン（Ver. 1.0）として発行される

⑬ ⑪の投票で通過できなかったら、修正を行って次のテクニカルミーティングで再度提出を行い、レビューと投票を受けるが、これも通らなかったら、RFP からやり直さなければならない

下表はコンシューマデバイス標準規格の着手から Ver. 1.0 が発行されるまでのスケジュールである。着手してから 3 年を要することになる。Ver. 1.0 が発行されると、タスクフォースが立ち上がり、随時バージョンアップを行っていくことになる。

表 2　コンシューマデバイス標準規格の提案スケジュール

年月	アクティビティ
2011 年 09 月	① White Paper(提案概要)提出
2011 年 12 月	② RFI(Request for Information：情報の要請)提出
2012 年 06 月	③ RFI レスポンス FIX

2012 年 09 月	④ White Paper(提案のスコープとロードマップ)提出
2012 年 12 月	⑤ RFP(Request for Proposal：提案の要請)ドラフト提出
2013 年 03 月	⑥ RFP 提出、発行
2013 年 06 月	⑦ LOI(Letter of Intent：提案の意思表明)提出
2013 年 11 月	⑧ Initial Submission(初期提案)提出
2013 年 12 月	⑨ Initial Submission(初期提案)レビュー
2014 年 03 月	⑩ Revised Submission(変更提案)の中間レビュー
2014 年 06 月	⑪ Revised Submission(変更提案)最終提出
2014 年 09 月	⑫ 最終投票

3.3. OMG に提出した RFP の内容

実際に OMG に提出した RFP [1]の主要な箇所を抜粋して日本語化したものを以下に示す。所々に斜体字でコメントや解説を入れている。

なお、国際規格では一般的に、要求事項は Shall 又は Shall not、推奨事項は Should 又は Should not と表記することになっているが、本 RFP の日本語化に伴い意味的に伝わりにくい部分は、その旨を注記した上で表現を変えている。

3.3.1. 動機（Motivation）

コンシューマデバイスの開発能力を向上するための手段を提案することが、今回の RFP 提出の動機である。その意味において、コンシューマデバイスの開発能力を向上するためには次の 3 つの概念を導入することが必要であると考えている。これらの概念はそれぞれ独立の概念として別々に導入してもよいし、3 つ同時に導入してもよい。

最初の概念はディペンダブルなシステムを開発するプロセスに関して、である。ディペンダビリティアシュアランスは開発プロセスの中の様々な活動を考慮する必要がある[i]。例えば、コンシューマデバイスの製品導入までの時間を削減するための素早い繰り返しによる開発である。ユーザを取り巻く環境の多様性を考慮すると、コンシューマデバイスの機能を保証する唯一の手段は、できるだけ全てのユーザのユースケースをカバーするため制御ソフトウェアを繰り返し素早く検証することである。“保証ケース”は、これらの繰り返しによる開発を、とりわけ各繰り返し時に発生する検証及び妥当性確認作業を、適切に組織立てて説明する必要がある[i]。

2 番目の概念は、“Proven In Use” [3]である。コンシューマデバイスでは、全ての開発を一から開始することはほとんどない。今日の大抵のソフトウェアは既存のソフト

ウェアを基に徐々に機能を追加しながら拡張されてきている。したがって、新規に開発されたコンポーネントと既存のコンポーネントの整合性を検証することが重要となる。さらによくあることでは、M&A により異なる企業で開発されたシステムを統合する場合に、既存でしかも複雑なコードを基に、さらに新機能を追加することがある。両方の場合とも “Proven In Use”の概念がコンシューマデバイスのディペンダビリティを保証するためには重要となる。

最後の概念はシステムズエンジニアリングとモデルベース開発の導入である。これらを用いて、開発期間を短縮していくことが不可欠であり、短縮された時間を利用して、さらに多くのユースケースでコンシューマデバイスが検証されなければならない。検証が開発における最も時間を費やす作業として一般的に認識されている。実機で実施している検証作業も、コンピュータを用いたシミュレーションを用いれば、時間短縮が可能となる。ここで言うシミュレーションとは、コンシューマデバイスのハードウェア、操作者、エンドユーザ、物理的な環境をモデル化したシミュレーション環境で、コンシューマデバイスの制御ソフトウェアを実行できることを意味している。システムズエンジニアリングとモデリングの MBD（Model Based Development）の特長及び複雑な HW（ハードウェア）と SW（ソフトウェア）のシミュレーションは、OMG SysML（Systems Modeling Language）と AADL（Architecture Analysis & Design Language）, EAST-ADL, ArchiMate, Simulink で既に述べられている。したがって、今回提案するディペンダビリティ保証の枠組み（DAF ：Dependability Assurance Framework）は、既存のモデリング言語とディペンダビリティ保証の観点を統合することを目的とする[i]。

提出される DAF は、エンジニアリング手法を拡張し、アシュアランスケース、特性を記述するためのコンセプトモデル、及びプロセスモデルを記載できるような観点を追加しなければならない[i]。エンジニアリングプロセスの一部として、明示的な議論と証拠を使用できるようにすることで、保証しようとする特性についての開発者の主張を正当化できるようにすることを狙いとしている。

3.3.2. 標準化の必要性

標準化されたコンシューマデバイスに対する DAF があれば、様々なツールを相互に利用可能にすることができ、産業界に広く認知されているベストプラクティスの規範を確立し、共通の方法論を広く浸透させ、より安全なコンシューマデバイスを生産するコストを削減することができる。

ディペンダビリティアシュアランスは一般に、システムズエンジニアリングプロセスを拡張したものである。ただし、標準化された DAF では、システムズエンジニアリン

グの方法論やモデリング言語・ツールの選定を制限してはならない[i]。言い換えれば、多くの異なるシステムズエンジニアリング手法に適合する柔軟なインタフェースを提供する必要がある[i]。

3.3.3. 提案のスコープ

本 RFP が求めている内容は、コンシューマデバイスのディペンダビリティを保証する方法である。特に、以下の内容を織込んだ正当と認められるディペンダブルなコンシューマデバイスを開発できる規格を具体化することである。

① ディペンダビリティの要素を定義する一つ以上の Dependability Conceptual Model（DCM）*⇒4.1 にて解説*

② コンシューマデバイスに対するディペンダビリティアシュアランスケース（Dependability Assurance Case : DAC）を組み立てるために使用する一つ以上のテンプレート*⇒4.2 にて解説*

③ ディペンダブルなコンシューマデバイスを開発するための素早い繰り返し要素を含んだプロセスを定義する一つ以上の Dependability Process Model（DPM）*⇒4.3 にて解説*

DAC テンプレートは、図 1 に描かれた DAC の視点を使用しなければならない。また、コンシューマデバイスの開発プロセスを記述するプロセス視点を参照しなければならない。

図 2 に DPM の具体例を示す。この DPM では、組込み制御ソフトウェアが開発されディペンダビリティアシュアランスケースが同時に生成されていることを示している。

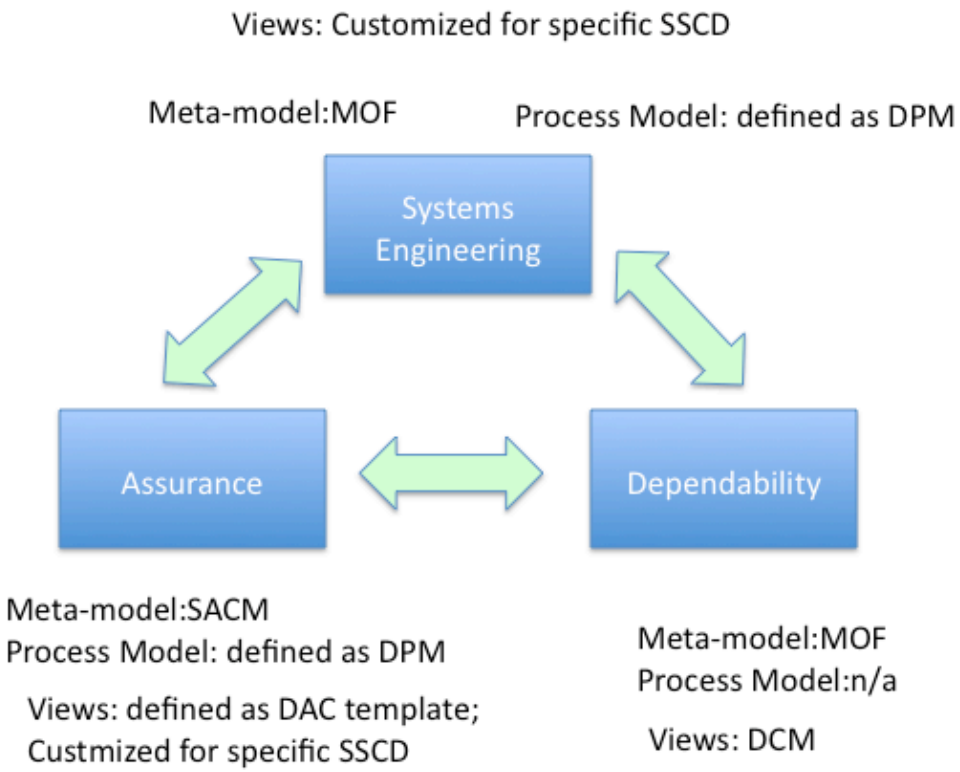

図 2　3 つの異なる観点の関係

※ 本 RFP により規格化しようとしているディペンダビリティアシュアランスフレームワーク（DAF） は、DCM、DAC と DPM から構成される。つまり DCM、DAC、DPM とそれぞれの関係を示すことで DAF を提案することができる。

3.3.4. OMG 規格との関係

① **OMG MARTE** [4]**との関係**

DAF は、MARTE 1.1 (formal/11-06-02)の非機能特性を参照してもよい。

② **OMG SACM** [5]**との関係**

SACM は、保証ケースのメタモデルを提供する。したがって、DAC は SACM (formal/2013-02-01)に準拠しなければならない。

③ **OMG MOF** [6]**との関係**

OMG MOF (formal/2011-08-07) は、保証ケース、DCM、DPM のメタモデルを提供する。

④ **OMG XMI** [7]**との関係**

OMG XMI (formal/2011-08-09) は、MOF メタモデルの変換と相互動作保証を提供する。

⑤ **OMG SysML** [8] **,SPEM** [9]**, BPMN** [10]**との関係**

OMG SysML (formal/2012-06-01), BPMN (formal/2011-01-03), SPEM (formal/2008-04-01) は、システムやプロセスのモデル化に関係する。SPEM と BPMN は、DPM をモデル化する際考慮しなければならない[i]。

⑥ OMG ODM [11]との関係

OMG ODM (formal/2009-05-01) は、概念をモデル化することに関係する。これらの仕様は、DCM の語彙をモデル化する際に考慮しなければならない[i]。

※ ここでは、OMG が既に公開している標準規格との関係を明確にしている。OMG に提案するためには、既に規格化されているものがあれば、それに準拠しなければならない。既存の規格から逸脱することは許されない。

3.3.5. OMG 外の標準との関係

① GSN [12] Community Standard

GSN とは、GSN community standard version 1.0, 2011 で定義されたアシュアランスケースに対するグラフィカル記述方法である。GSN は既に SACM と一貫性を保っている。

② ISO 26262 [13]

ISO 26262 とは、自動車における機能安全設計をカバーする国際標準で、自動車の安全ライフサイクル活動を提供する。

提案では、ISO 26262 の範囲を拡張することを目的としている[i]。既存の ISO 26262 は E/E (Electrical/Electronic)システムの課題のみ言及しているためである。例えば、エンジンストールは、自動車の安全問題と認識されるが、ISO 26262 では品質問題と位置付けられ、スコープ外となってしまう。DAF によって、ISO 26262 を拡張することで、ISO 26262 でカバーし切れない安全問題を議論できるようにする。

③ IEC 61508 [14]

IEC 61508 は、全ての産業の機能安全をカバーする国際標準である。ISO 26262 は、自動車の E/E システムに対する IEC 61508 の適用規格であり、EN 50128 は列車への適用である。

④ ISO 15026

ISO 15026 は、システムとソフトウェアの保証に関する国際標準で、アシュアランスケースとライフサイクルプロセスへの関係付けに対する一般的なフレームワークを提供する。

本 RFP に対する提案は、ISO/IEC 15026 の必須要件と一貫性を保っていなければならない[ii]。

⑤ **ISO/IEC 62741**

ISO/IEC 62741 は、ディペンダビリティ保証に対する国際標準で、ディペンダビリティアシュアランス議論に対する一般的なフレームワークを提供する。

本 RFP に対する提案は、ISO/IEC 62741 の必須要件と一貫性を保っていなければならない[ii]。

⑥ **ISO/FDIS 13482**

ISO/FDIS 13482 は、パーソナルロボットの安全要件に対する国際標準である。

本 RFP に対する提案では、ISO/FDIS 13482 で要求されている特定の安全要件を規定する必要はない。しかし、コンシューマデバイスのディペンダビリティの一部として安全を保証する一般的な議論構造（argument structure）を提供しなければならない。

⑦ **TCG standards**

Trusted Computing Group (TCG) は、IT セキュリティ標準グループで、相互運用可能な信頼されるコンピューティングプラットフォームに対する、特にハードウェアに関するオープンかつベンダー非依存のグローバル産業の標準を開発、定義、啓蒙することを目的としている。

TCG 標準は、IT セキュリティ問題のみを扱うが、本 RFP に対する提案では、ディペンダビリティ全般を扱う。

⑧ **DEOS Whitepaper** [15]

DEOS Whitepaper は、JST CREST Dependable Embedded Operating System Project により発行され、ディペンダブルな組込みシステムの開発を目的としている。

⑨ **DA Draft Guidance on Infusion Pumps**

DA Draft Guidance on Total Product Life Cycle: Infusion Pumps - Premarket Notification [510(k)] は、2013 年に発行予定である。

⑩ **ISO 14971**

ISO 14971 は、医療機器のリスク管理システムを扱う ISO 規格である。最新版は、2007 年発行である。

⑪ **FDA Quality System Regulation, 21 CFR Part 820, Design Controls**

FDA 21 CFR Part 820, は Quality System Regulation QSR outlines Current Good Manufacturing Practice CGMP regulations としても知られており、人間によって使用されることを意図した完成デバイスの設計・製造・パッケージング・ラベル付け・設置・修理に使用される方法論・制御・設備を管理する規格である。

⑫ **IEC 60601-1**

IEC 60601 は、医療電子機器の安全と効果に対する技術標準である。International

Electrotechnical Commission によって 1977 年に初版が発行され 2011 年までに定期的に更新されている。10 の付随標準と約 60 の特定標準が存在する。IEC 60601-1 は、基本安全と不可欠性能に対する医療電子機器要件を扱う。

⑬ ASTM F2761

本標準は、統合医療環境を構築する装置を統合するための一般要件、モデル及びフレームワークを提供する。特に、医療機器と他の装置を安全に統合するために必要な特性を規定する。

※ ここでは OMG 以外の規格との関係を明確にしている。ISO や IEC などでは機能安全に関する規格が多く提出されている。ISO や IEC 以外の規格も多く存在している。関係すると思われる規格やドキュメントを列挙している。

3.3.6. 必須要件

① 提案は一つ以上の DCM を定義しなければならない
② 各 DCM はディペンダビリティの要素とそれらの関係を定義しなければならない
③ 提案は DAC テンプレートを定義しなければならない
④ DAC テンプレートは SACM に準拠していなければならない
⑤ 各 DAC テンプレートは、一つ以上の DCM を参照しなければならない
⑥ 提案は DPM を定義しなければならない
⑦ 各 DAC テンプレートは一つ以上の DPM を参照しなければならない
⑧ DPM は素早い開発プロセス、繰り返し開発プロセス、差分開発プロセスを記述することができなければならない
⑨ DPM は素早いディペンダビリティアシュアランス開発プロセス、繰り返しのディペンダビリティアシュアランス開発プロセス及び差分ディペンダビリティアシュアランス開発プロセスを記述することができなければならない
⑩ 提案は DAC, DCM , DPM に関連する用語をコンシューマデバイスの内容に即して定義しなければならない
⑪ DCM と DPM は、そのモデルのインスタンスが XMI を使用して変換可能であることを保証するために、適切な MOF ベースの言語で定義されなければならない
⑫ 提出された DPM と DAC は特定のコンシューマデバイス用にカスタマイズ可能でなければならない。また、それらのコンシューマデバイスのシステムアーキテクチャに関連付けられなければならない
⑬ DAC テンプレート は、コンシューマデバイスのディペンダビリティを保証する議論構造（argument structure）を再利用可能なように定義しなければならない。

また、どのようにディペンダビリティに関する主張が達成され、理由付けられ、証明されたかを明記しなければならない

⑭ DAC テンプレートは 、“Proven In Use”による証拠を織込むことが可能でなければならない

⑮ DAC テンプレートは、固有のシステムズエンジニアリングの視点を参照するシステム固有の主張を追加することによってカスタマイズ可能でなければならない

⑯ DAC テンプレートのカスタマイズは、プロセスモデルに基づき特定のシステムズエンジニアリングの視点を参照にするカスタマイズプロセスの視点で、サポートできなければならない

※ この原文には shall が使われていて、ここに挙げたことは要求事項として必ず守らなければならない。国際規格では一般的に、要求事項は Shall（しなければならない）、Shall not（してはならない）と表記する。

3.3.7. 必須でないフィーチャ

① DCM は、コンシューマデバイスのディペンダビリティに適用可能な標準を使用してもよい

② DAF は、システムエンジニアリングプロセスに関連した要素を含んでもよい

③ DAF は、プロセスの視点、保証の視点、特性の視点間の様々な関係を定義しても良い。それにより保証ケースは、プロセス内の各活動と特性の要素に従い、そして参照して構成することができる

④ DAF は、保証の観点を提供する目的で、SysML,AADL,EAST-ADL,ArcihMate, Simulink のような既存のシステムズエンジニアリングモデル言語を拡張してもよい

⑤ DAF は、プロセスの観点を提供する目的で、既存のシステムズエンジニアリングモデル言語を拡張してもよい

⑥ DAF は、システム特性に対する概念モデルを記述する観点を提供する目的で、既存のシステムズエンジニアリングモデル言語を拡張してもよい

⑦ DAF は、保証ケースがシステムアーキテクチャとプロセスの各活動に従い、そして参照して組み立てられるように、システムズエンジニアリングの観点とプロセスの観点の関係を定義してもよい

⑧ DAF は、保証ケースが特性要素に従い、そして参照して組み立てられるように、保証の観点と特性の観点の間の関係を定義してもよい

⑨ DAC テンプレートは、DAF のシステムズエンジニアリングの要素を参照してもよ

い

⑩ DAC テンプレートは、モジュール型議論構造をサポートして、SACM の機能を追加してもよい

※ この原文では May が使われており、必須でないフィーチャを規定している。国際規定では一般的に、許可事項として May（してもよい）、May not（する必要がない）と表記する。

3.3.8. 議論される課題

提出された提案の評価中には、以下の課題が考慮される。これらは、提案された基準となる仕様の一部であってはならない [ii]。各レスポンスで、これらの課題について記述しなければならない。

- DCM 及び DAC テンプレートの表現、インスタンス化、及び実装の簡素性
- 他の OMG 規格や非 OMG 規格との互換性
- 他の規格との依存関係又は参照関係
- 議論構造の再利用性

3.3.9. 評価基準

提案は、その仕様の一貫性、実現可能性、幅広いコンシューマデバイスへ多用途性の観点で評価される。

4. 提案のフレームワーク

ここで改めて、本 RFP で求めている開発方法論（DAF）を構成する 3 つの要求（ディペンダビリティのメタモデル（DCM）、ディペンダビリティアシュアランスケース（DAC）のテンプレート、ディペンダビリティを保証するプロセス（DPM））に関して、その必要性や今後の標準化に向けた方向性等を解説する。

4.1. ディペンダビリティ規格のメタモデル（DCM）

(1) DCM の必要性

コンシューマデバイスのディペンダビリティを保証する規格を一から作成することは困難なため、自動車の機能安全規格である ISO 26262 をベースとする。しかし、ISO などの規格は一部説明を補足するために図などが使われている以外は、そのほとんどはテキストにより記述されていて構造等が分かり辛い。そこで ISO 26262 の Part1～Part3（今回の提案の範囲）をメタモデル（概念モデル）として、規格の構造を見える化することで、複雑な要素が絡み合った規格を易しく見通すことができるようにした。なお、OMG で標準化などの議論をするためにもメタモデルは必要で、今回のような規格に関しては UML のクラス図を使用して記述することが適切と判断した。

(2) DCM 提案の考え方・方向性

ISO 26262 のメタモデルをベースにして、安全性、信頼性、可用性などを汎化する形でディペンダビリティの要素などの規格の概念構成を見える化したメタモデルを DCM として提案する。

例えば、ISO 26262 の Part1 で記述されている Fault & error & failure について、メタモデルとして扱うと、次のようになる。

システムの Fault（不良）が原因となり、その結果システムが Error（誤り）という状態になる。この Error という状態は内部的なものであり、これが外部へ出てくると Failure（故障、或いは障害）という状態になるということが表現されている。安全を保つために Fault の発見、Failure のコントロールといった技術的な解決策として Safety mechanism が存在する。これはさらに上位の Safety measure という概念のサブクラスとして定義される。

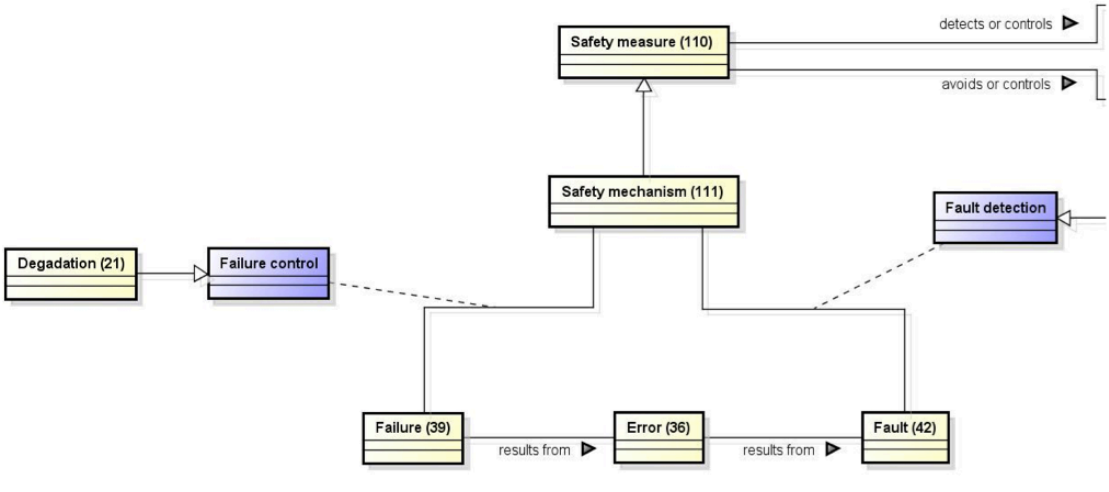

図 3　Fault & error & failure メタモデルの一部

このようにほとんどテキストで表現されている ISO 26262 をメタモデルで表現することで、どのような概念が存在し、それらがどのような関係になっているかを一目で見通すことができるようになる。

ここで、Fault & error & failure をコンシューマデバイスの標準規格のためにどうすべきかを検討する。

エラーモデルとして Fault-Error-Failure というエラー状態の遷移についてどのように定義するかが重要である。ここで、Fault、Error、Failure を結ぶ関連（results from）において矢印の方向が、遷移とは逆になっているのは、results from が因果関係を表しているからであり、遷移としては、Fault-Error-Failure の順に変化して行く。Fault-Error-Failure の定義とそれらの関係については、従来の定義を踏襲することが望まれる。

Threat（脅威）は、通常はセキュリティにおけるとして理解されるが、その意味を含むさらに大きな概念として Fault、Error、Failure との関連で定義されている。その解釈は図 4 のように定義することが可能である。

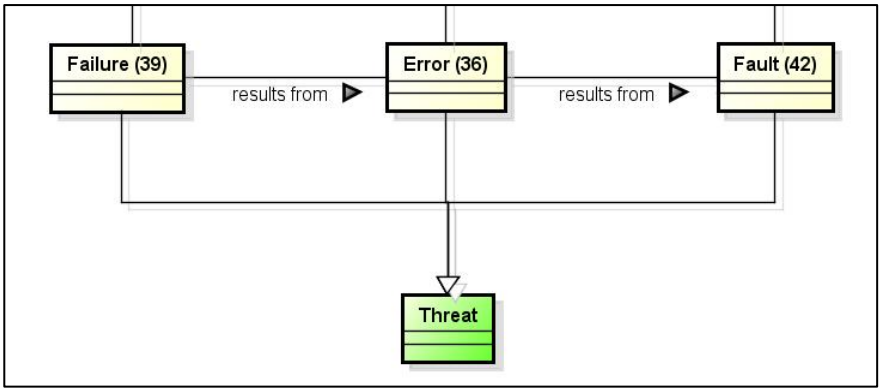

図 4　Fault、Error、Failure と脅威の関連

ここでは、ISO 26262 Part 1 において定義されている、Fault、Error、Failure に対して、その全ては、Threat の性質が継承されるサブクラスとして定義されている。

また、ISO 26262 Part 1 において Fault、Error、Failure を検出、制御する方法として Safety measure、Safety mechanism が定義されているが、これらも Dependability に汎化することが必要であると考えられる。さらにこれらを検出、コントロールする方法を詳細に定義したのが図 5 である。

図 4 では、Safety mechanism と Error の間の関係は、ISO 26262 の文章に見当たらなかったため何の関係もないとしたが、図 5 では Dependability mechanism と Error の間に本来あるべき関連を引いた。

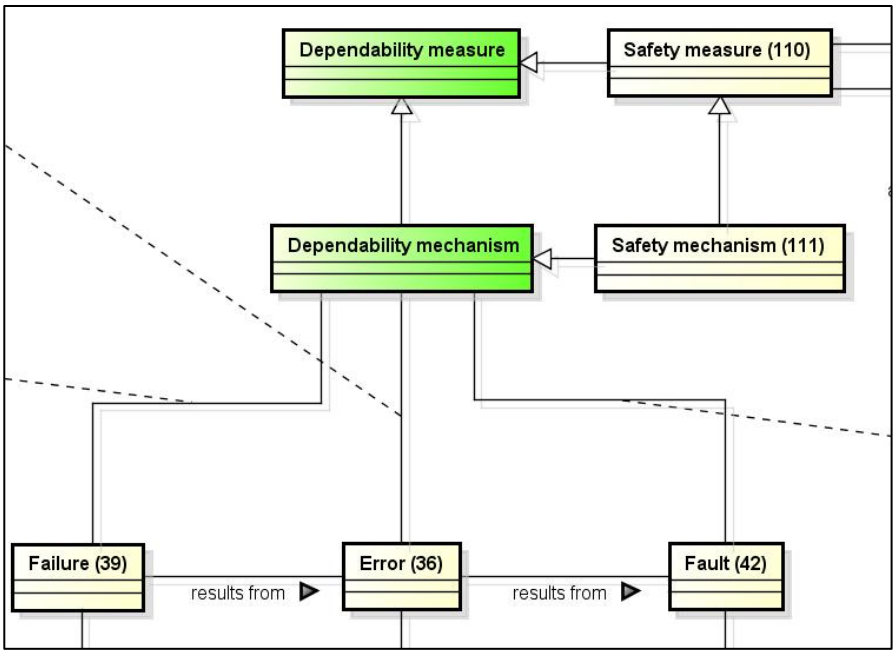

図 5　Dependability measure への拡張

これに、ディペンダビリティを達成する手段を付け加えると図 6 及び図 7 のようになる。

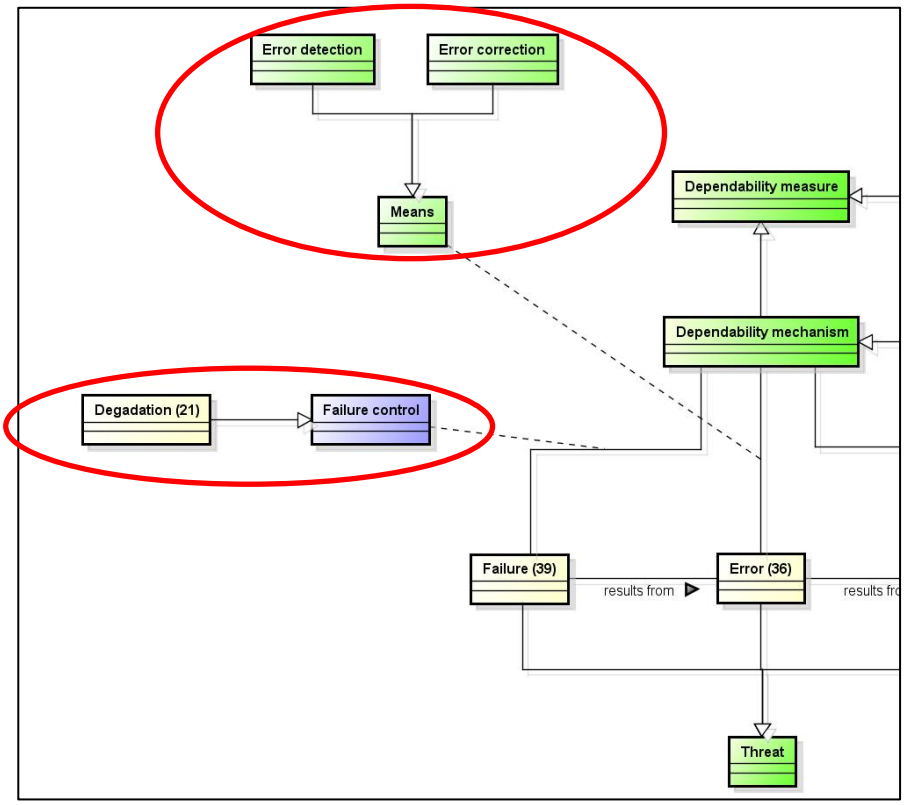

図 6　Failure と Error への対応

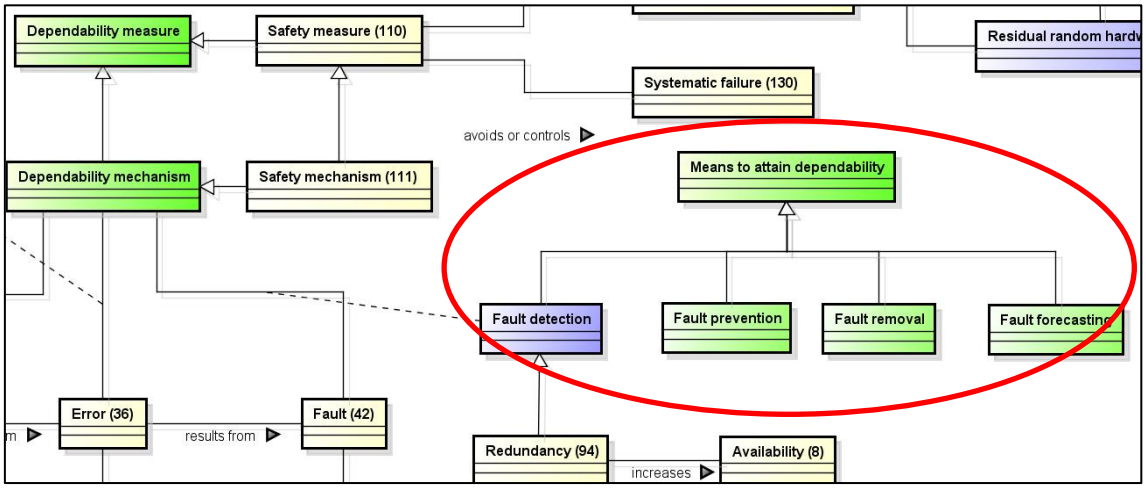

図 7　Fault への対応

ここで、Dependability mechanism と Error の間に新たに関連を引いたので、この関連にメカニズムを実現する手段として関連クラスを追加した。さらに図 7 の Fault の

関連クラスは、Means to attain dependability（ディペンダビリティを達成する方法）を汎化して、実現手段の拡張を図っている。

さらに、Dependability はセキュリティの一部を含む形で定義されている。そのため、Asset や Vulnerability などセキュリティにおいて重要な概念も導入すべきか検討する必要があると思われる。

4.2. ディペンダビリティアシュアランスケース（DAC）のテンプレート

(1) DAC のテンプレートの必要性

提案する DAC のこのテンプレートとは、様々なコンシューマデバイスに拡張して適用できるようなディペンダビリティアシュアランスケース［16］［17]の雛形のことである。DAC を最初から作るのではなく、雛形であるテンプレートをドメイン毎に拡張できるようにすることで、DAC 作成の作業効率を向上することができ、また DAC の記述のサンプルとしても有効である。

DAC は、GSN（Goal Structuring Notation）［12]により、トップゴールから合意形成を図りながらサブゴールを導出し、それらが成立していることの証拠（エビデンス）を明示する。このエビデンスが成立している限り、システムのディペンダビリティは保証される。

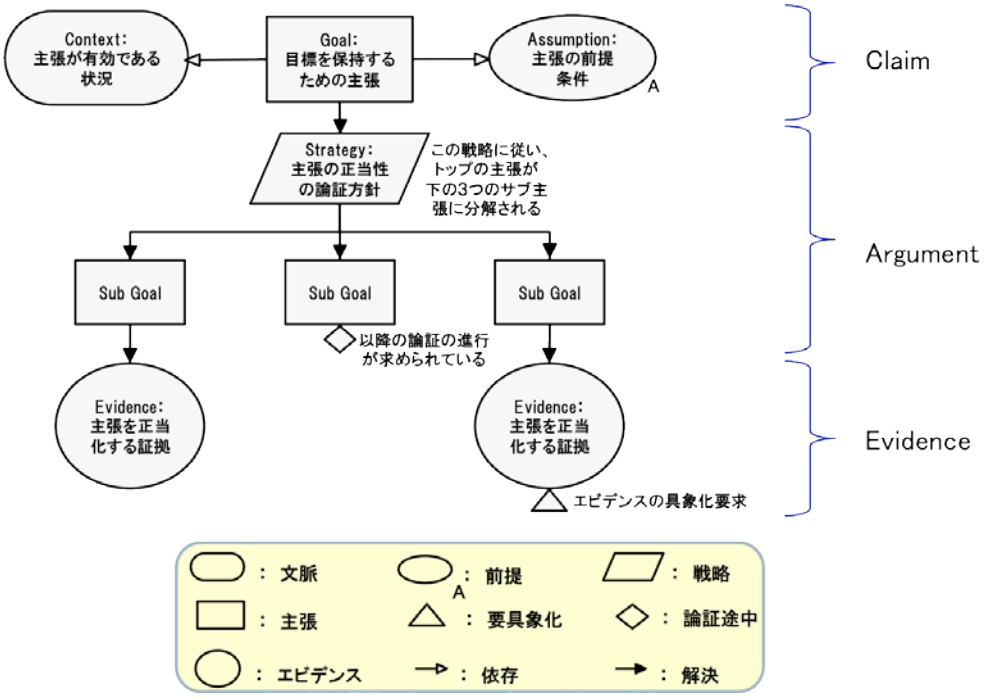

図 8　GSN の構造と要素

GSN にてアシュアランスケースを記述するためには、まず、目標（Goal）を保持す

るための主張（Claim）を明確（主張が有効である状況や主張の前提条件も考慮）にする。次に主張の正当性の論証方針（Strategy）を立てて議論（Argument）して、サブの主張（Sub Goal）に分解する。最終的なサブの主張に対して、その証拠（Evidence）を示す（図 8）。GSN ではこのようにして、目標を達成していることの議論を記述することができる。

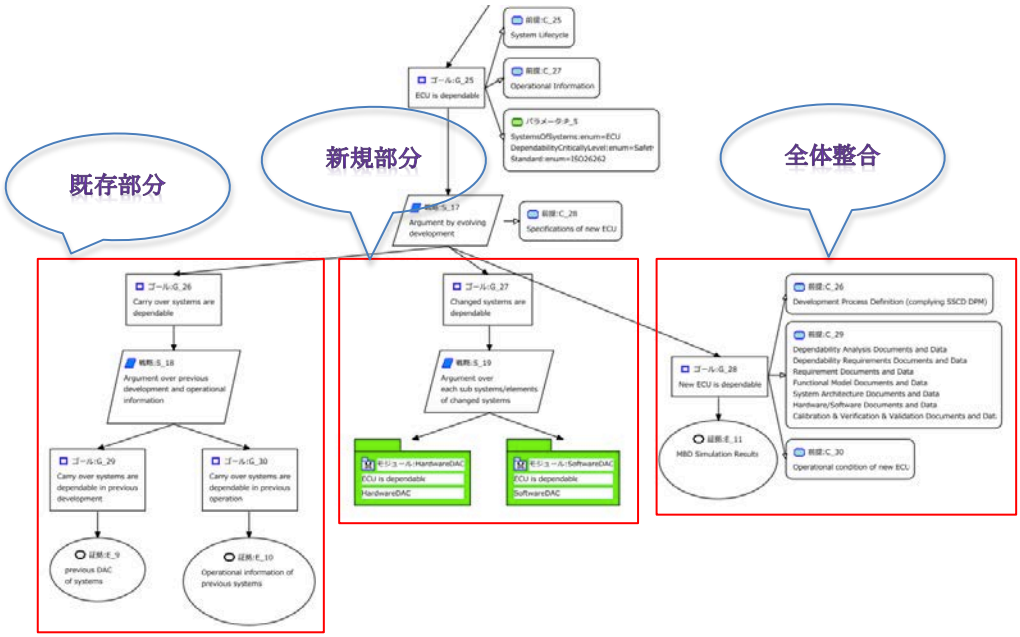

図 9　エンジンストールの DAC

(2) DAC のテンプレート提案の考え方・方向性

我々の中で最も詳細化しているエンジンストールの DAC（図 9）の事例を基にテンプレート案を作成する。この事例から得られたパターン（既存部分のディペンダビリティ、新規部分のディペンダビリティ、全体整合のディペンダビリティ）を利用して DAC のテンプレート（既存部分の保証テンプレート、新規部分の保証テンプレート、全体整合の保証テンプレート）を作成する（図 10）。作成したテンプレートは、基本となるパターンに対して、汎用化するためのプロパティを設定して、そのプロパティを変更することで、他のドメインにも対応できるようにする。

幾つかのパターンに対して、テンプレートが十分か、他のドメインで適用できるかなどの検証を行う必要がある。

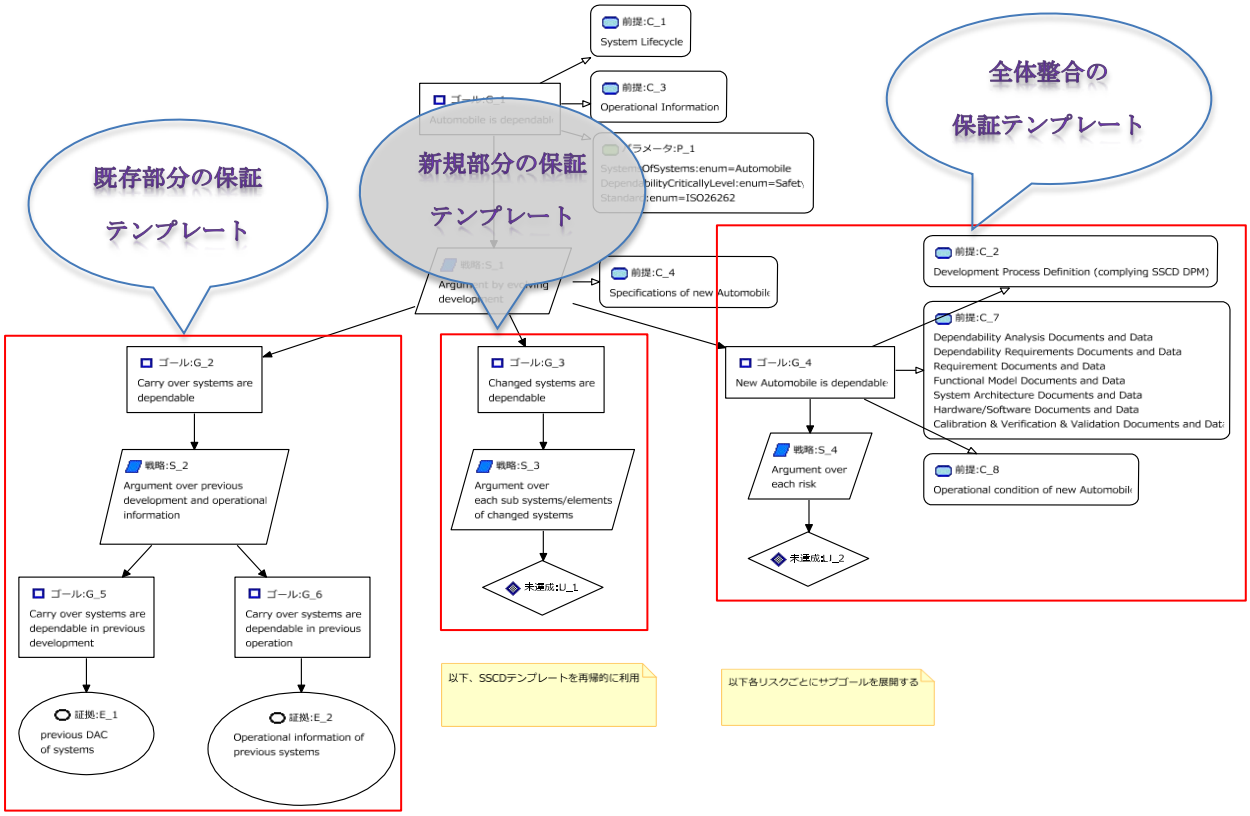

図 10　DAC のテンプレート案

4.3. ディペンダビリティを保証するプロセス（DPM）

(1) DPM の必要性

利用者層、利用環境等の多様性に対応する必要のあるコンシューマデバイスに関して、ディペンダビリティを保証するための開発プロセスの必要性については、第 2 章で説明した。この開発プロセスに関しては、機能安全等の国際規格等を考慮したシステムズエンジニアリングプロセスとの整合性や、現実の開発現場を考慮したアジャイル的な繰り返しによる開発形態との整合性がとれるものである必要がある。

(2) DPM 提案の考え方・方向性

DPM も曖昧さを排除したメタモデルとして記述するため、BPMN（Business Process Modeling Notation）を使って定義する。ちなみに、BPMN は OMG で標準化されたプロセスを記述するための表記法である。

BPMN でモデル化するプロセスとして図 11（図 1 と同じもの）であるが、これ自体は制御ソフトウェアのプロセスなので、システムズエンジニアリング [18]のプロセスに置き換える必要がある。

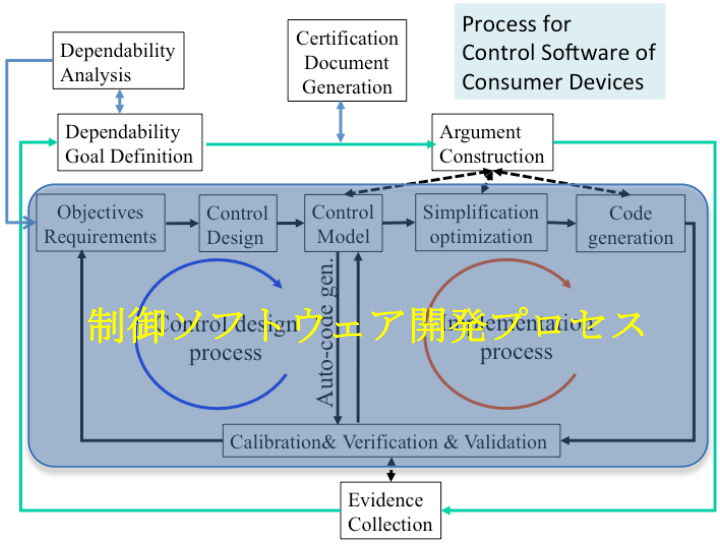

図 11　制御ソフトウェア開発のアジャイル的な繰り返しの例
に対してディペンダビリティを保証するプロセス

まず開発プロセスとして、システムズエンジニアリングのプロセスを検討する。現段階では、SAFE（Safe Automotive soFtware architEcture）の考え方 [19]を参考とした図 12 のようなプロセスをベースに考えている。システム要求を導き、それを実現するためのシステムのアーキテクチャを定義する。それを基にハードウェアとソフトウェアのサブシステムに分割し、サブシステムを開発（設計実装）する。最後にそれらを統合して、検証と妥当性確認を行う、との流れになっている。

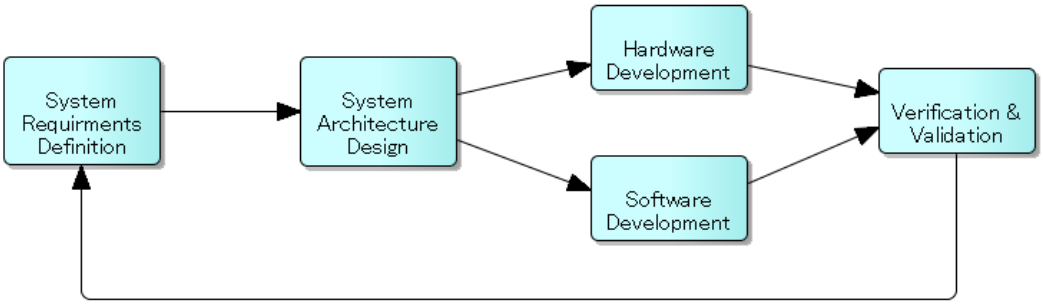

図 12　システムズエンジニアリングのプロセス

このようなシステムズエンジニアリングのプロセスに、図 11 を基にしたディペンダビリティを保証するプロセスを追加する。

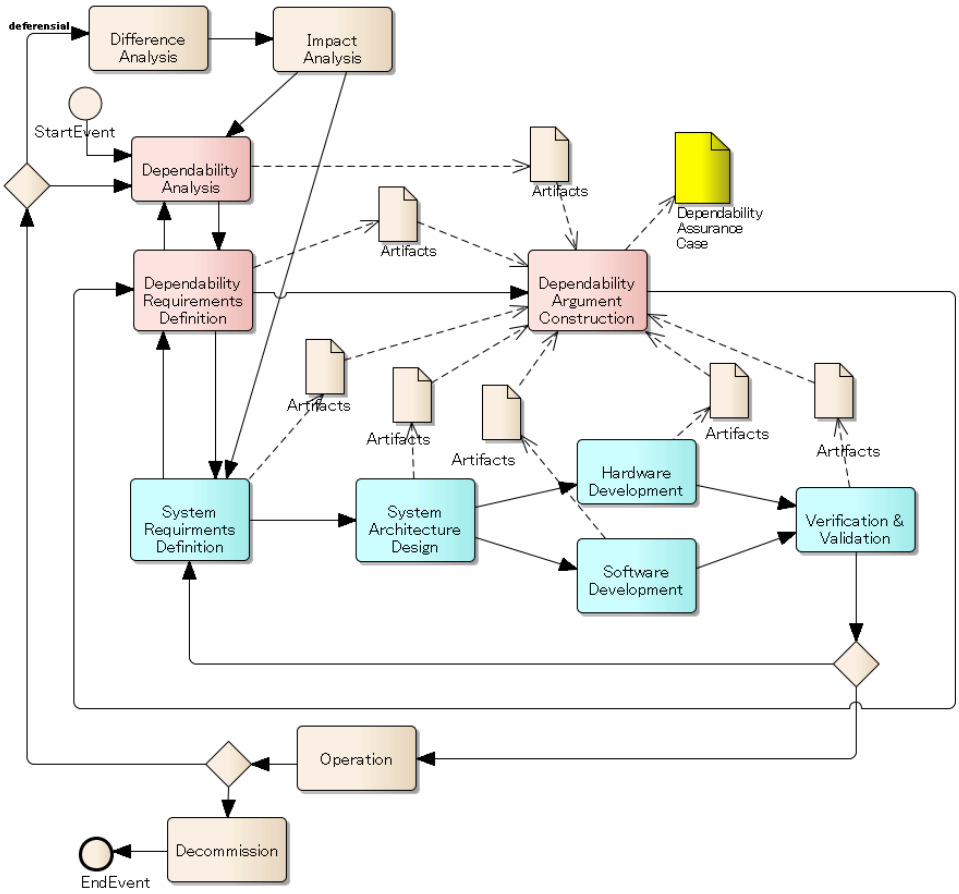

図 13　ディペンダビリティ保証プロセス案

まずリスクハザードを分析（Dependability Analysis）して、ディペンダビリティ要求（Dependability Requirements Definition）を設定し、さらにディペンダビリティを保証する議論構造（Dependability Argument Construction）を導いて、DAC（Dependability Assurance Case）を作成する。その DAC でシステムのディペンダビリティが成立していることを保証するためには、システムズエンジニアリングプロセスの各ワークフロー（アクティビティ）からシステムがディペンダブルである証拠を成果物（“Artifacts”）として取得して、ディペンダビリティが成立していることを検証する必要がある。このようにして作成したプロセスが図 13 である。図 13 では更に、システムズエンジニアリングのプロセスに、運用（Operation）や廃棄（Decommission）を含めて、コンシューマデバイスのライフサイクルを通してディペンダビリティを保証するプロセスを追加した。なお、Difference Analysis と Impact Analysis は派生開発を考慮したアクティビティであり、Difference Analysis は差分解析を行い、Impact Analysis は変更に対するインパクト分析を行う。

5. おわりに

国民が安全かつ安心して暮らせるためには、コンシューマデバイスのように一般消費者自身が使用するような製品のディペンダビリティを確保することが極めて重要である。そのためには、言語による曖昧さを排除し明確に解釈できるメタモデルで記述された概念モデル（DCM）の下で、明確な目標とその実施、実施の証拠を明確にして、ディペンダビリティを確保（DAC）するための適切なプロセス（DPM）にしたがって開発される必要がある。ここでは、それらの 3 つの要件から構成されるコンシューマデバイスのディペンダビリティを確保するための開発方法論（DAF：Dependability Assurance Framework）に関して、2013 年 3 月に発行された RFP に基づき、考え方、経緯等を解説した。

このような標準化活動は、多くの方々の協力と理解により普及し、効果を発揮することが期待できるため、読者の方には、特にコンシューマデバイスの開発に携わる方々を中心に規格化に向けた動きを注視していただくとともに、開発方法論として各開発現場への導入方法等をご検討いただければ幸いである。

なお、本書で解説したコンシューマデバイスの開発方法論（DAF）の検討は、IPA/SEC コンシューマデバイス安全標準化 WG の委員諸氏によるものであり、更に OMG のテクニカルミーティング等で有意なコメントをいただいた OMG メンバの方々も含め、ここで感謝の意を表したい。

参考文献

1. **RFP.** Dependability Assurance Framework For Safety-Sensitive Consumer Devices RFP. (オンライン) Members Only
http://www.omg.org/techprocess/meetings/schedule/Dependability_Assurance_Framework_For_Safety-Sensitive_Consumer_Devices_RFP.html.
2. **AVIZIENIS.** Basic Concepts and Taxonomy of Dependable and Secure Computing. Avizienis et al.
IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 1, NO. 1, JAN-MAR 2004.
3. **UseInProven.** [PROVEN]. 61508 Association Policy document: Proven in Use. (オンライン) http://www.61508.org/?page_id=143.
4. **MARTE.** Modeling and Analysis of Real-time Embedded Systems. (オンライン) http://www.omg.org/spec/MARTE/.
5. **SACM.** Structured Assurance Case Metamodel. (オンライン)
http://www.omg.org/spec/SACM.
6. **MOF.** Meta Object Facility Specification. (オンライン)
http://www.omg.org/spec/MOF/.
7. **XMI.** XML Metadata Interchange Specification. (オンライン)
http://www.omg.org/spec/XMI.
8. **SysML.** Systems Modeling Languag. (オンライン)
http://www.omgsysml.org/.
9. **SPEM.** Software & Systems Process Engineering Metamodel. (オンライン)
http://www.omg.org/spec/SPEM/.
10. **BPMN.** Business Process Model and Notation. (オンライン)
http://www.bpmn.org/.
11. **ODM.** Ontology Definition Metamodel. (オンライン)
http://www.omg.org/spec/ODM/.
12. **GSN.** Goal Structuring Notation community standard version 1.0, 2011 GSN contributors. (オンライン)
http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf.

13. **ISO26262.** 2011. Road vehicles – Functional safety. International Organization for Standardization, Geneva, Switzerland.
14. **IEC61508.** 2010, Functional safety of electrical/electronic/programmable electronic safety-related systems. International Electrotechnical Commission, Geneva, Switzerland.
15. **DEOS.** DEOS Whitepaper. (オンライン)
http://www.dependable-os.net/osddeos/data/DEOS-FY2011-WP-03J.pdf.
16. **山本修一郎.** ディペンダビリティケースの必要性と留意点. 名古屋大学情報連携統括本部情報戦略室. (オンライン) 2013 年 2 月 15 日.
https://acs.is.nagoya-u.ac.jp/index.php?module=User&action=DownloadFile&id=6797&file_id=16673&folder_id=4806.
17. **倉光研究チーム、松野、恩田、山本研究グループ.** D-Case ディペンダビリティ合意形成のための手法とツール. DEOS プロジェクト. (オンライン)
2013 年 5 月 1 日.
http://www.dependable-os.net/osddeos/data/DEOS-FY2013-DC-02J.pdf.
18. **西村秀和 , 鈴木尚志.** 複雑化する統合システム（SoS）の開発方法論　モデルベースシステムズエンジニアリング導入の手引き. IPA/SEC. (オンライン)
2013 年 8 月 23 日. https://www.ipa.go.jp/sec/reports/20130823.html.
19. **SAFE.** Overview presentation showing the main SAFE ideas and outcomes. http://www.safe-project.eu/SAFE-Download.html#Concepts.

協　力

平成 24 年度　消費者機械安全標準化 WG

主査	新 誠一	国立大学法人 電気通信大学
副主査	田口 研治	独立行政法人産業技術総合研究所
委員	秋山 進	株式会社デンソー
委員	大畠 明	IPA/トヨタ自動車株式会社
委員	神余 浩夫	三菱電機株式会社/公益社団法人計測自動制御学会
委員	金川 信康	株式会社日立製作所
委員	白坂 成功	慶應義塾大学大学院
委員	中川 雅通	パナソニック株式会社
委員	神徳 徹雄	独立行政法人産業技術総合研究所
委員	平鍋 健児	株式会社チェンジビジョン
委員	松野 裕	国立大学法人 名古屋大学

(平成 25 年 3 月時点)

平成 25 年度　コンシューマデバイス安全標準化 WG

主査	新　誠一	国立大学法人電気通信大学
副主査	田口　研治	独立行政法人産業技術総合研究所
委員	石﨑　直哉	公益社団法人計測自動制御学会／トヨタ自動車株式会社
委員	大畠　明	トヨタ自動車株式会社
委員	金川　信康	株式会社日立製作所
委員	相馬　大輔	株式会社シーエーブイテクノロジーズ
委員	中川　雅通	パナソニック株式会社
委員	中坊　嘉宏	独立行政法人産業技術総合研究所
委員	松野　裕	国立大学法人電気通信大学
委員	宮崎　比呂志	富士通株式会社

(平成 25 年 9 月時点)

[i]英語の文献では「should」と表記されているが、前後の文脈より強めの表現としている。
[ii]英語の文献では「should not be inconsistent with」と表記されているが、前後の文脈より強めの表現としている。