オープンシステム・ディペンダビリティのための形式アシュランスケース・フレームワーク（FFO）

奥野 康二[※1]　木下 修司[※1]　木下 佳樹[※1,※2]　武山 誠[※1,※2]　中原 早生[※1,※2]

オープンシステムのディペンダビリティを主張する形式アシュランスケースのためのフレームワークを構築し，オープンシステム・ディペンダビリティの要件を再検討すると共に，車載システム及び防災システムにおける事例研究を行った．

Formal assurance case Framework for Open systems dependability (FFO)

Koji Okuno[※1], Shuji Kinoshita[※1], Yoshiki Kinoshita[※1,※2], Makoto Takeyama[※1,※2] and Hayao Nakahara[※1,※2]

A framework for formal assurance cases that claim dependability of open systems is developed. Requirements for open systems dependability are reexamined and case studies in automotive systems and disaster management systems are given.

1　はじめに

本稿では，IPAによる「ソフトウェア工学分野の先導的研究支援事業（RISE）」の委託研究「オープンシステム・ディペンダビリティのための形式アシュランスケース・フレームワーク」の研究成果[1]を紹介する．

この研究では，システムがオープンシステム・ディペンダビリティ（Open systems dependability，以下OSD）を達成していることを議論する形式アシュランスケースを書くためのフレームワークFFO（Formal assurance case Framework for Open systems dependability）を作成した．また，その作成過程で得られた知見をもとに，システムがオープンシステム・ディペンダビリティを達成するためのシステムライフサイクルプロセスへの要件を考察し，国際標準案IEC 62853 Open systems dependability[7]策定の技術的根拠を与えた．

本論文の構成は以下の通りである．第2節では，アシュランスケースをはじめとする研究背景を述べる．第3節では，フレームワークFFO自体の研究成果を述べる．第4節と第5節は事例研究の紹介で，それぞれFFOを車載システム，防災システムに適用した成果を述べる．第6節で総括する．

2　背景

2.1　アシュランスケース

アシュランスケース（assurance case）は，具体的なシステムの安心・安全に関する議論（アシュランス議論）の記録文書である．アシュランスケースはシステムの利害関係者間の合意事項の記録，契約文書や認証における提出文書，あるいは事故調査委員会の資料として，近年急速に注目され始めた．

アシュランスケースの利用は，プラントや軍事技術など，高度な安全性が要求される，いわゆるsafety critical systemに関する安全性議論に用いられることから始まった．現在では車載，鉄道，航空，医療システムの認証に関してアシュランスケースの提出が要請される国際標準があるなど，その需要は増加傾向にある．

膨大なアシュランスケースの構造的な理解を助けるため，図式を用いた構造化アシュランスケースの記法が幾つか提案され，それぞれ普及が図られている（GSN[9]，CAE[2]，D-Case[10][11]など）．アシュランスケースの国際標準ISO/IEC 15026-2[17]が発行されている．

我が国でも2010年代からD-Caseなど，アシュランスケース

※1　神奈川大学プログラミング科学研究所　　※2　神奈川大学理学部情報科学科

に関連する研究が盛んになり，独立行政法人情報処理推進機構（IPA）や一般社団法人ディペンダビリティ技術推進協会（DEOS協会）において，普及活動が展開されている．

2.2 形式アシュランスケース

形式アシュランスケースとは，整合性検査を自動化するため形式言語によって記述されたアシュランスケースである．筆者らは，関数型プログラミング言語Agdaによる記法，D-Case in Agda [10]（図1）を研究開発している．Agdaは構成的型理論に基づく関数型プログラミング言語であり，高い論理的表現力を持つ．

D-Case in Agdaにより，アシュランスケース記述における整合性の問題が解決された．アシュランスケースは，システムの安心・安全に関する議論の記録文書であるため，対象システムが大きくなり，その議論が重ねられることで，膨大な文書となる．そのような文書の細部にまでわたって整合性を取るのは困難だが，一方で，ほんの少しの不整合も，システムの大きなトラブルに直結し得る．

GSNなどの図式による現行のアシュランスケース記述では，アシュランス議論が何に基づくのか，が明示されないことが問題である．D-Case in Agdaでは，それを形式言語Agdaで定義された概念体系（オントロジー）として明示した上で議論を形式記述することにより，機械的な整合性検査をAgda処理系によって行う．これにより，アシュランスケースの本質的な部分の検討に人間の注意を集中することを可能にした．

2.3 OSD

オープンシステムは，システムの境界が定義できず，その機能が時間と共に変化するようなシステムである．オープンシステム・ディペンダビリティ（OSD）とは，オープンシステムが目的，目標，環境及び性能の変化に適応し，説明責任を絶え間なく達成して，想定されるサービスを，要求されたときに要求通り提供する能力のことである [10]．OSDでは，ある一時点のシステムの機能や状態がディペンダブルであるからと言って，システムがディペンダブルであるとは言えない点に注目する．そのためには，システムライフサイクルの観点から長期的にディペンダビリティを達成させなければならない．その際，単にシステムが障害や環境変化に対応できるということだけでなく，説明責任や合意形成などのhuman factorをも考慮することが必要である．

SoS（System of Systems）やIoT（Internet of Things）に関するディペンダビリティ達成のためにはOSD達成が必要である．また，OSDはシステムのレジリエンス（resilience）[5] とも関連する．防災計画やセキュリティ対策では想定外の災害や攻撃への対処が重要であるが，OSD達成のためには，どこまでを想定するのかを明確にすることが求められる．

```
module 安全要求仕様
  (PB : 前提条件[基本機能]-type)
  (PS : 前提条件[安全機能を含む]-type PB)
  (C : 追加の支援情報-type)
  (D : システム設計[基本機能]-type PB C)
  where
  record 安全分析-対象ハザードの発生原因分析[FTA]-項目-type : Set where
    field
      対象安全目標ID : ID-type 安全目標
      安全分析-対象ハザードの発生原因分析[FTA] : FT図-type
  安全分析-対象ハザードの発生原因分析[FTA]-type : Set
  安全分析-対象ハザードの発生原因分析[FTA]-type =
    List (安全分析-対象ハザードの発生原因分析[FTA]-項目-type)
  module 技術安全要求仕様 (SA : 安全分析-対象ハザードの発生原因分析[FTA]-type) where
    record 技術安全要求仕様-項目-type : Set where
      field
        システムブロック名 : システムブロック名-type
        基本事象 : 基本事象-type
        安全要求 : 安全要求-type
        ASIL : ASIL-type
        時間間隔 : Time-type
```

図1　Agdaによる形式アシュランスケースの記述例

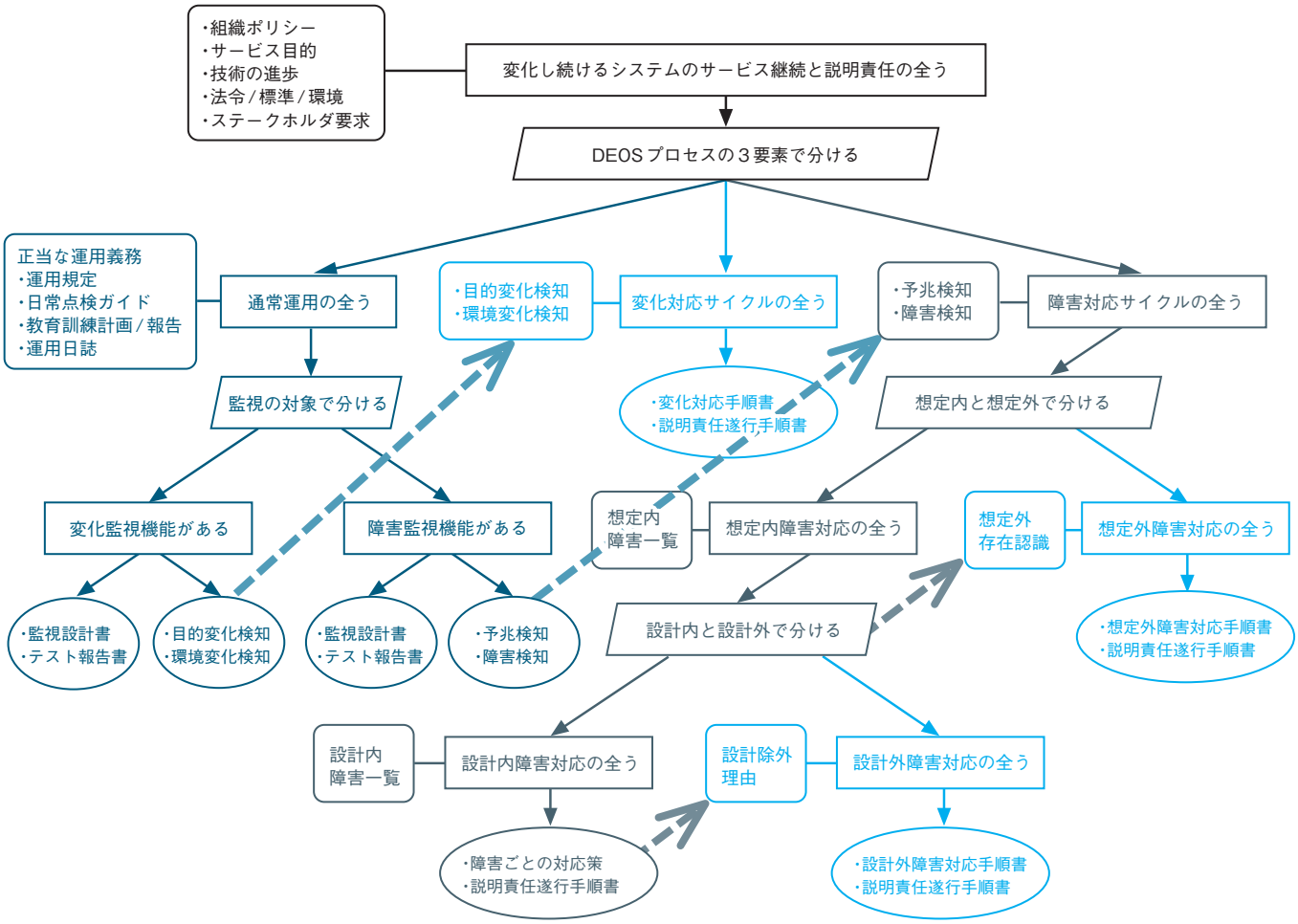

図2　DEOS基本構造のGSNによる表現

3　FFO

3.1　DEOS基本構造

FFOは，OSD達成を議論する形式アシュランスケースのフレームワークである．これは，D-Case in AgdaをOSD達成の議論向けに詳細化したものである．OSD達成に必要な議論のモデルを，先行研究の成果であるDEOS基本構造[10] Chapter 3. 4. 1に従い作成した．

図2はDEOS基本構造をGSN記法によるアシュランスケースで示したものである．議論の最上位ゴールは，「変化し続けるシステムのサービス継続と説明責任の全う」である．DEOS基本構造に基づき，これを以下の3つのサブゴールに分けた．

- **通常運用**：変化監視と障害監視が正しく機能する
- **変化対応**：システムの目的や環境の変化が検知されたときに正しく対応する
- **障害対応**：システム障害に対して，正しく応じる

また，形式アシュランスケースのAgda言語による実装として，以下のような工夫を考案した．

- 形式アシュランスケースの中で用いる語彙を定義する部分（Context）と，形式アシュランスケースが提示する主張を議論する部分（Argument）を，別のモジュールとして明確に分離した．D-Case in Agdaではこのような明確な分離がなされていなかった．

- 証拠の吟味，手順の遂行などのように，形式アシュランスケースの議論部分で繰り返して使われる議論の類型を幾つか見出す，Agdaの関数として実装した．例えば，証拠の吟味は，変化の検知，変化対応への移行，障害検知，障害対応への移行などで議論される．これらの議論では，監査結果，開発証憑，運用証憑の3つが必ず用いられる．この共通部分を1カ所にまとめて機械的に具体化することにより，形式アシュランスケース記述，理解，保守が容易になることが期待される．

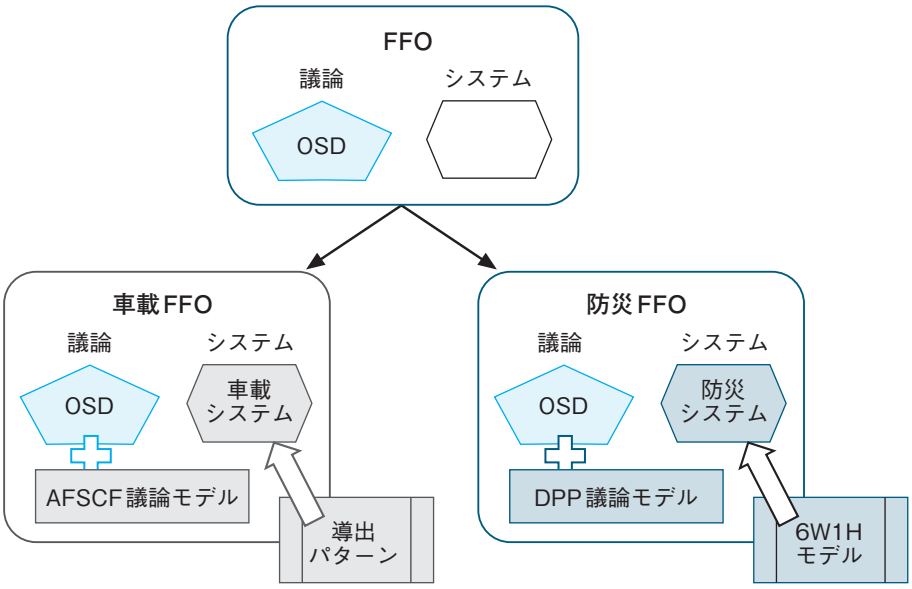

図3　FFOとその事例研究の関係

3.2　FFOを利用したアシュランスケース記述

FFOは，オープンシステム一般に適用可能なフレームワークである．FFOを具体的なシステムの議論に適用するには，以下の手順を踏む．

1. FFOの適用対象となるシステムを明確に定義する（記述する）．
2. FFOが提供するOSD一般の議論モデル（DEOS基本構造）に合わせて，技術領域個別に議論モデルを構築する．
3. 1.2.及びFFOに従ってアシュランスケースを作成する．

1.や2.は必ずしも自明な作業ではない．そこで本研究では，車載システム（第4節）と防災システム（第5節）を対象とした事例研究において，このようなシステム定義の枠組みや議論モデルの構築も研究対象とした．

図3は本研究の各成果の関連を示したものである．まず，FFOはシステムの技術領域を限定しない，オープンシステム一般に適用可能なフレームワークである．また，2つの具体的な技術領域において，FFOに基づくディペンダビリティ議論を考察した．

車載システムのOSD達成に必要な議論を検討し，議論モデルとしてAFSCF議論モデルを考案した．また，システム定義の枠組みとして，導出パターンを考案した．

防災システムのOSD達成に必要な議論を検討し，議論モデルとしてDPP議論モデルを考案した．また，システム定義の枠組みとして，6W1Hモデルを考案した．

4　事例：車載システムへの応用

車載システムの機能安全を規定するISO26262 [3] は，システムライフサイクルのテクニカルプロセスのうち開発設計プロセスに対する要求を主に定める．しかしこれだけでは，機能安全を超えた，開発設計時に作り込むことができないような安全要求を十分に盛り込むことが困難である．

機能安全に加えて，OSDを要求することにより，開発設計時に想定できなかった安全要求を，運用，保守，合意プロセス，説明責任遂行などに対して盛り込むことができる．これらを達成することにより，機能安全の，より現代的な実現が可能になる．そこでFFOを車載システムの機能安全に具体化したAFSCF（Automotive Functional Safety Case Framework）議論モデルを考案した．また，車載システムの定義を明確にするため，導出パターンを考案した．

4.1　導出パターン

導出パターンとは，下位の仕様から上位の仕様を導く導出のパターンである．これを利用することにより，仕様書の段階的な詳細化が容易になり，開発者の負担が軽減される．

システム開発における仕様書は一般に，基本設計書と詳細設計書のように，上位の仕様から下位の仕様へと段階的に詳細化される．その際，それら仕様書間に要請される性質には，「下位の仕様が満たされれば，上位の仕様が満たされる」ことがある．例えば，詳細設計書を記述する場合，その仕様がすべて適切に実装されたときに，基本設計書の仕様が満たされなければいけない．

図4にその例を示す．左側では機能Aが機能A1/A2/A3に適切に分割され，詳細設計書が記述されたのに対して，右側では機能Bを詳細化した際に，本来記述すべき機能B3の詳細設計書が欠落している．そのため，詳細設計書B1/B2の仕様が適切に実装されたとしても，機能Bは満たされない．

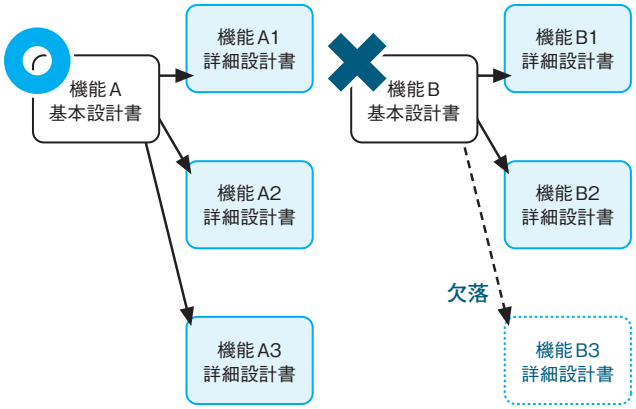

図4　下位仕様と上位仕様との関係

そこで「下位の仕様が満たされれば，上位の仕様が満たされる」ことの議論を明示する．それによって，このような仕様の不備を事前に発見することが可能になる．

これを示したのが図5である．左の例では，機能Aの基本設計書，機能A1，A2，A3の詳細設計書を記述したのちに，「機能A1，A2，A3の詳細設計書の仕様が満たされれば，機能Aの基本設計書の仕様が満たされる」ことが示されている．つまり，欠落がないことの検討の根拠を提供している．一方，右の例では，機能Bの基本設計書，機能B1と機能B2の詳細設計書を記述したのちに，「機能B1とB2の詳細設計書の仕様が満たされれば，機能Bの基本設計書の仕様が満たされる」ことが示されていない．つまり，B1，B2以外にB3の詳細設計書が必要であることが判明する．

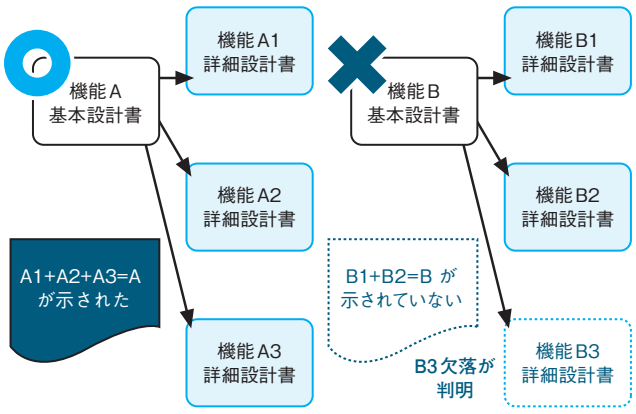

図5　仕様の不備を事前に発見

一般に，上位仕様と下位仕様の対応を示すには，それらの項目間の対応を追跡する必要がある．更に，妥当性の議論のためには，単に対応させるだけでなく，対応することの理由まで明示することが必要である．しかし，この作業は必ずしも自明ではない（どのような記述であれば「対応している」と言えるのかは高度な判断を要する）上に，コスト面の負担が大きい．また，度重なる仕様修正が起こると，そのコストは増大する．

そこで，「下位の仕様が満たされれば，上位の仕様が満たされる」ことが示された仕様書のテンプレートを用意しておく．そのようなテンプレートがあれば，開発者はそのテンプレートに沿って具体的な仕様を埋めていくだけで，適切な上位の仕様書と下位の仕様書が完成する．このように，対応関係だけでなくその妥当性の議論まで記すためのテンプレートが導出パターンである．

4.2　AFSCF議論モデル

車載システム開発がISO 26262に準拠するためには，安全性を示す議論を安全ケース（安全性のアシュランスケース）として提出することが要求される．それらを各社が個別に実施することは容易ではない．そのため，（社）JASPARによって，ISO 26262に準拠した開発を行うための機能安全テンプレート[4]が提供されている．

そこで，IEC 62853 国際標準案のアシュランスケースモデル[7] Annex BとISO 26262に準拠した安全ケースのレイヤーモデル[6]を統合した，AFSCF（Automotive Functional Safety Case Framework）議論モデルを考案し，Agdaで記述した．

ISO 26262は安全ケースの提出を要求しているが，その詳細は規定されていない．AFSCF議論モデルは，安全ケース執筆の指針を提供する．

図6はAFSCF議論モデルをGSNで示したものである．トップゴールは「OSD-機能安全の達成」（システムの正常でない振る舞いによるハザードに起因する不適当なリスクは回避される）である．機能安全の達成は次の6つのサブゴールに分割できると考えられる．6つのうち，1，2は機能安全の観点からの分割であり，3-6はそれらに加え，OSD達成のために必要な要件をDEOS基本構造に従って追加したものである．

1. 「機能安全の達成」（システムの正常でない振る舞いによるハザードに起因する不適当なリスクは回避される）
2. 「環境」（適切な環境のもとで機能安全を達成している）
3. 「合意形成」（システム，システムの目的，目標，環境，性能，ライフサイクル，及びこれらの変化に関する共通理解と明示的合意を確立し，維持する）

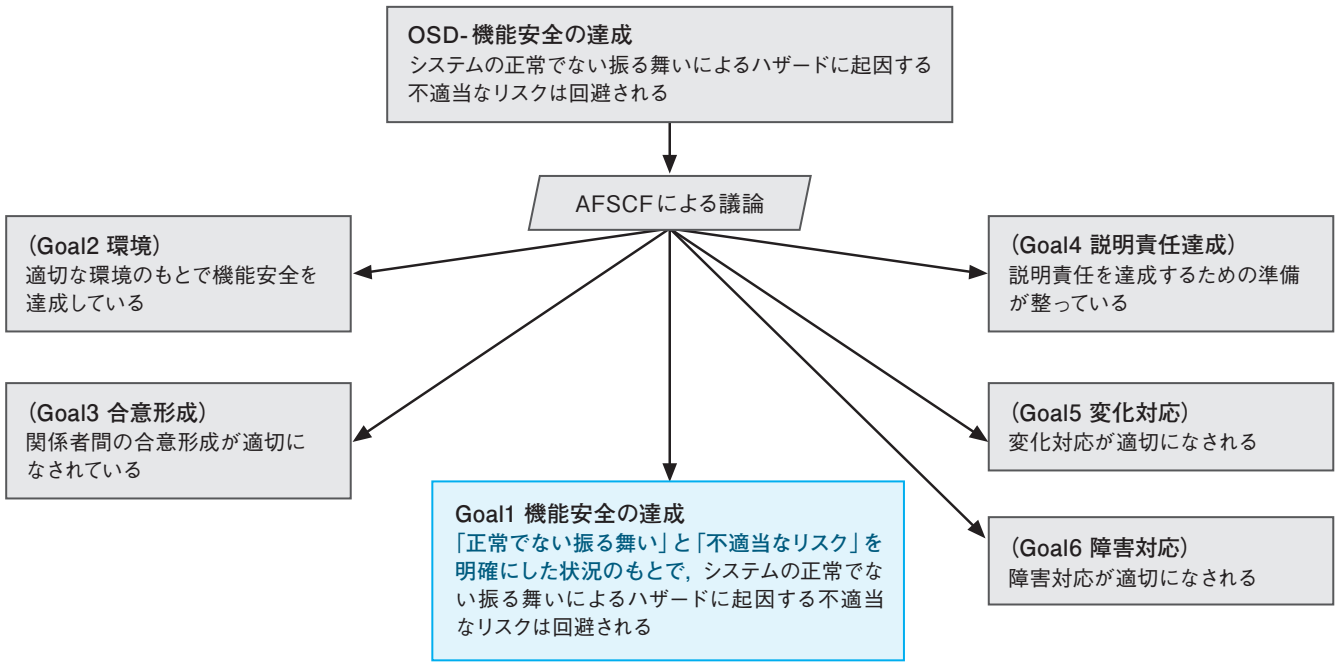

図6　AFSCF議論モデル

4. 「説明責任達成」(システムに関する合意事項の不履行がステークホルダや一般社会に及ぼす影響を同定し，合意事項の遂行を改善して，システムに関する確信と信用を保ち，潜在的な被害に対する補償を確実にする)
5. 「変化対応」(要求，環境，目標及び目的が変化しても，システムの「目的にかなった(fit-for-purpose)」状態を維持する)
6. 「障害対応」(障害に際してもサービス中断と損害を最小にとどめ，その状況のもとで最も適切なやり方で，可能な限りサービス提供を続ける)

更に，このうち1.「機能安全の達成」について，ISO 26262に基づく開発を4つのフェーズに分け，そのフェーズごとに議論することと定めた．[14]にその詳細がある．

5　事例：防災システムへの応用

もう一つのFFOの適用事例として，防災システムを選んだ．ここで言う防災システムとは，地方自治体の防災業務のことである．地方自治体に警察・消防・公共インフラ企業などが協力して実施する防災関係業務は「地域防災計画」という行政文書によって規定される．これは，1961年に施行された災害対策基本法[13]に基づき，地方自治体が作成する文書である．

平塚市との共同研究により，平塚市地域防災計画[12]を対象に防災FFOを作成し，防災システムのオープンシステム・ディペンダビリティを主張する形式アシュランスケースの記述を試みた．その際，システム記述の枠組みとして6W1Hモデル[16]を，議論の枠組みとしてDPP議論モデルを考案した．

5.1　6W1Hモデル

6W1Hモデルとは，6つのW(Who, What, Whom, Where, When, Why)を持つアクションが構成するツリーである．6W1Hの1HとはHowのことである．一般に，「ある1つの作業の記述」は，更に細かい複数の作業の記述によって表現できる．この関係のことをHowと呼ぶ．これが，システムのアシュランス議論に必要なシステムの明確な記述を提供する．

防災業務をシステムライフサイクルとして定式化することにより，様々な局面でやるべき業務を明確にすることができると考えられる．システムライフサイクルについては，国際規格ISO/IEC/IEEE 15288[8]があるので，これに基づいた防災業務の定義を試みた．

そのためにはまず，システム自体が明確に記述されていなければならない．しかし，それは地域防災計画をもとに自明に構築できるものではないことが明らかになったため，記述の枠組みとして6W1Hモデルを考案した．

図7は，6W1Hモデルを利用したシステムの記述例である．

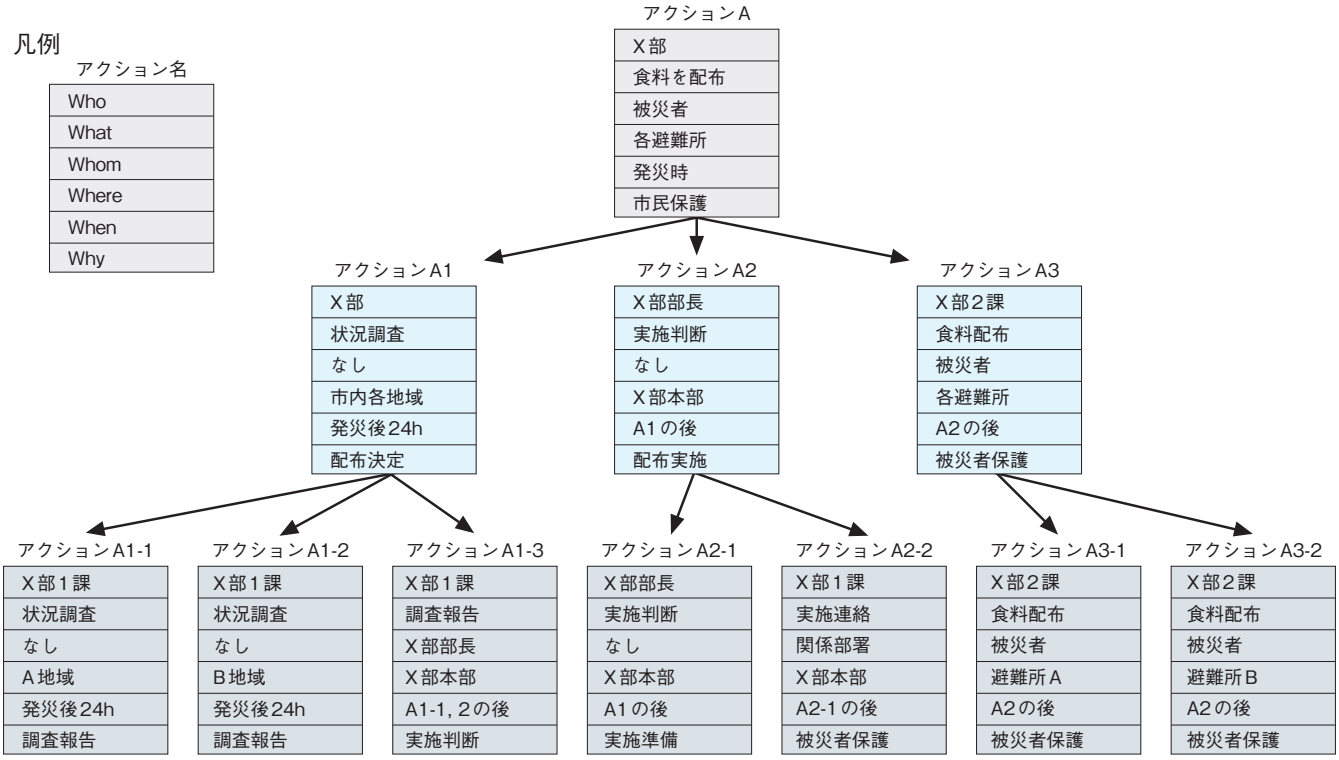

図7　6W1Hモデルの記述例

発災時の食料供給業務を例に取り，Who, What, Whom, Where, When, Whyの6つのラベルを1つのアクションに付け，それを段階的に詳細化した．例えばアクションA「X部は市民保護のため，発災時に，被災者に対して，各避難所で食料を配布する」は，以下の3つのアクションによって詳細化される．

- **A1**：X部は食料配布決定のため，発災後24時間以内に市内各地域の状況を調査する
- **A2**：X部部長は，食料配布実施のため，アクションA1の後にX部本部において実施を判断する
- **A3**：X部2課は，被災者保護のため，アクションA2の後に各避難所において被災者に食料を配布する

これらのアクションは，アシュランス議論に必要な単位まで更に詳細化することができる．地域防災計画に記述が不足しがちな主語（Who）が明示されるとして，平塚市の防災担当者から肯定的評価を得た．発災時の給水業務を事例とした記述実験では，6W1Hモデルの49個のアクションを構成した．そのうち，主語を明確にしなければならなかったアクションは19個にのぼった．主語を明確にすることにより，地域防災計画をシステムとして定義可能にし，それについてのアシュランス議論が可能になった．

5.2　DPP議論モデル

防災システムのオープンシステム・ディペンダビリティに関する議論モデルの一例がDPP議論モデルである．「DPP」とは，Decision（決定），Preparation（準備），Provision（実施）の頭文字を取ったものである．多くの業務は以下の3つに分類されるという考えに基づく．

- **決定（Decision）** ある業務を実施するために必要な情報収集と開始の判断及び，実施した業務の確認と継続・終了の判断
- **準備（Preparation）** ある業務を実施するために必要な人的，物的リソースの準備，実施計画の作成
- **実施（Provision）** 準備された業務の実施

図8はDPP議論モデルを，発災時の給水業務に対して適用した例である．このような決定，準備，実施による議論の分割は，給水だけでなく，物資配給全般や応援要員派遣などの様々な防災業務に適用可能である．

6W1HモデルおよびDPP議論モデルに基づき記述した給水業務のアシュランスケースは，DEOS基本構造に基づく防災業務全体のアシュランスケースの一部として位置付けられる．すなわち，発災時の給水業務とは，DEOS基本構造における障害対応のうち，設計内障害対応の一つに当たる[1]．

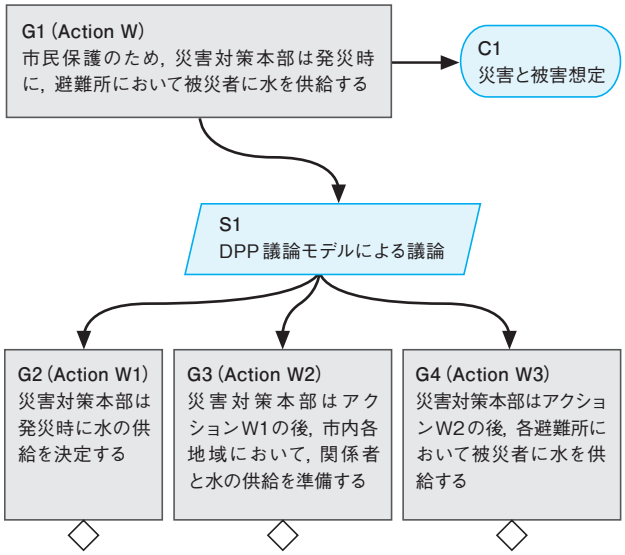

図8　DPP議論モデルの給水業務への適用

6　考察

本研究は，以下の2つの内容をアシュランスケース開発フレームワークとして明示した．

- 形式アシュランスケースの様式とその処理技術
- オープンシステム・ディペンダビリティの概念体系

フレームワークの試作により，オープンシステム・ディペンダビリティの要件について，より洗練された理解を得ることができた．それらは，策定中の国際標準案[7]に反映されている．車載システム及び防災システムにおける事例研究によって更にこの理解を深めることができた．

しかし，ソフトウェア開発現場へのFFO導入手法，ほかの技術領域への適用手法などの構築は，今後の課題である．本研究においては，車載システムにおいてAFSCF議論モデルを，防災システムにおいてDPP議論モデルを構築した．現場の技術者によって，これらのモデルを具体的なシステムに適用し，その工数を計測するなど，更なる実証評価が必要である．

導出パターンが用意する仕様書のテンプレートが適切であるとはどういうことかの明示も，今後の課題として残されている．

本研究の後，プロセスを組み合わせたライフサイクルモデルへの要件を規定するなどの後継活動の必要がある．このような要件は，社会における一定のコンセンサスを得て初めて利用可能なものとなっていく．そのためにもオープンシステム・ディペンダビリティ要件の国際標準化は重要であると考える．

謝辞

本研究は，IPA RISE事業の一テーマとして行われた．AFSCF議論モデルと導出パターンの研究にあたって（株）デンソーのご協力をいただいた．また，6W1HモデルとDPP議論モデルの研究にあたって平塚市防災危機管理部災害対策課のご協力をいただいた．ここに記して深甚なる感謝の意を表する．

【参考文献】

[1] 2014年度ソフトウェア工学分野の先導的研究支援事業「オープンシステム・ディペンダビリティのための形式アシュランスケース・フレームワーク」成果報告書，神奈川大学．URL: http://www.ipa.go.jp/sec/rise/#03-1, 2017-08-05閲覧.
[2] Adelard, Safety Case Structuring: Claims Arguments and Evidence. URL: https://www.adelard.com/asce/choosing-asce/cae.html, 2017-08-05閲覧.
[3] ISO 26262 Road Vehicles — Functional Safety. ISO Standard (2011)
[4] （社）JASPAR, 機能安全テンプレート，2013. URL: https://www.jaspar.jp/disclosureDocument/#anchor-8, 2017-08-05閲覧.
[5] J.-C. Laprie, From Dependability to Resilience, The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2008.
[6] I. Habli, et. al., A Layered Model for Structuring Automotive Safety Arguments, 10th European Dependable Computing Conference (EDCC 2014), Newcastle upon Tyne, UK, May 2014.
[7] IEC 62853 Open systems dependability (unpublished Committee Draft for Vote).
[8] ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes
[9] Origin Consulting Limited, GSN Community Standard, Version 1, York, 2011.
[10] M. Tokoro (ed.), Open systems dependability 2nd edition, CRC Press, 2015.
[11] 所眞理雄編著，DEOS —変化し続けるシステムのためのディペンダビリティ工学，近代科学社，2014.
[12] 平塚市地域防災計画，平塚市防災会議，2015. URL: http://www.city.hiratsuka.kanagawa.jp/bosai/page-c_01661.html, 2017-07-20閲覧.
[13] 災害対策基本法．URL: http://law.e-gov.go.jp/htmldata/S36/S36HO223.html, 2017-08-05閲覧.
[14] 中原早生，木下佳樹，自動車機能安全・OSDアシュランスケースの為のAFSCF議論モデル，Science Journal of Kanagawa University Vol.27, 2016.
[15] 木下佳樹，DEOS関連国際標準の動向，第2回DEOS協会オープンシンポジウム，2015. URL: http://deos.or.jp/link/obj/pdf/hyoujunka_bukai.pdf, 2017-08-05閲覧.
[16] Shuji Kinoshita and Yoshiki Kinoshita, The 6W1H Model as a Basis for Systems Assurance Argument, 4th International Workshop on Assurance Cases for Software-intensive Systems, LNCS, vol. 9923, pp.63-74, Springer, 2016.
[17] ISO/IEC 15026-2:2011 Systems and software engineering — Systems and software assurance — Part 2: Assurance case