独立行政法人情報処理推進機構 委託

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

平成 28 年 2 月

神奈川大学

本報告書は独立行政法人情報処理推進機構　技術本部　ソフトウェア高信頼化センターが実施した「2014 年度ソフトウェア工学分野の先導的研究支援事業」の公募による採択を受け神奈川大学理学部（研究責任者　木下佳樹）が実施した研究の成果をとりまとめたものである.

目次

研究成果概要

1. 背景

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などとして，近年急速に注目され始めた．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が，現在では車載，鉄道，航空，医療システムの認証に関する以下のような国際標準においてアシュランスケースの提出が要請されているなど，その需要は増加傾向にある．

- ISO26262 Road vehicles - Functional safety
- IEC62425 Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling
- DO-178C/ED-12C Software Considerations in Airborne Systems Equipment Certification
- FDA 510(k) 注入ポンプ（infusion pump）市販前届出

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いくつか提案され，それぞれ普及が図られている（GSN[14], CAE[1], D-Case[16][17]など）．さらに，整合性検査を進めるため，形式言語による記法が，形式化アシュランスケースと呼ばれて研究されている．アシュランスケース自身に関する国際標準化も進められており，以下のような標準が出版されている．

- IEC/ISO 15026-2:2011 Systems and software engineering -- Systems and software assurance -- Part 2: Assurance case
- IEC 62741:2015 Demonstration of dependability requirements - The dependability case
- Object Management Group (OMG), Structured Assurance Case Metamodel (SACM), Version 1.1, 2013
- IEC 80001-2-9 Application of risk management for IT-networks incorporating medical devices -- Part 2-9: Application guidance -- Guidance for use of security assurance cases to demonstrate confidence in IEC/TR 80001-2-2 security capabilities（策定中）

我が国でも 2010 年代にはいって D-Case や D-Case in Agda など，アシュランスケースに関連する研究がはじまり，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. 実施内容

2.1. 研究の全体像

本研究では，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 Formal assurance case Framework for Open systems dependability (FFO) を作成した．

また，こ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 IEC 62853 Open systems dependability 策定の技術的根拠を与えた．

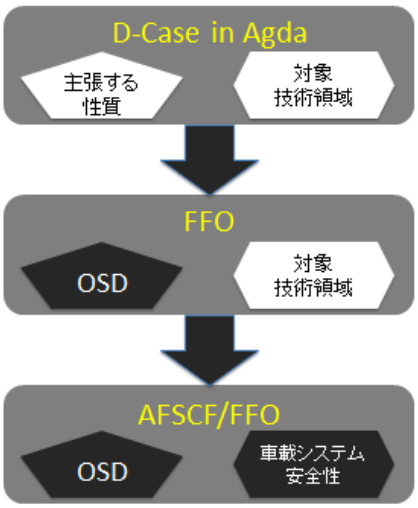

図-1　研究の全体像

先行研究（「利用者施行ディペンダビリティの研究」JST CREST 制度による研究）で開発した D-Case in Agda は

(i) 形式アシュランスケースが主張する，システムの性質

(ii) 形式アシュランスケースが対象とする，システムの技術領域

の二つのパラメータをもっていた．このうち，パラメーター(i)をオープンシステム・ディペンダビリティ (OSD) に具体化したものが FFO である．D-Case in Agda の単なる具体化にとどまらず，以下のような工夫を考案し，FFO に反映させた．

(iii) 形式アシュランスケースの中で用いる語彙（オントロジー）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．

(iv) 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いくつか見出し，Agda の関数として実装した．例えば証拠の吟味，手順の遂行などの類型を提供した．

技術領域によらず，オープンシステム・ディペンダビリティ一般を取り扱う「一般 FFO」をまず開発した．このレベルでは，ソフトウェアライフサイクルが通常運用の他に，合意形成，障害対応，変化対応，説明責任の四つのプロセスビューを持つことが要件である．FFO

は，これらの要件達成の本質に関するアシュランスケースを，抽象度の高いレベルで構築可能にした．

次に，具体的な技術領域として車載システムおよび地域防災計画を選び，一般 FFO にそれぞれの領域知識を具備させたフレームワークを試作した．本研究の目的は FFO を試作し，その有効性を評価することであるが，有効性評価のためにシステムの現場から意見をきくためには，(ii)も具体化したフレームワークをさらに試作して提示する必要があった．そこで，FFO を何らかの技術領域に具体化したバージョンを作り，産業界からの協力を得て，具体化バージョンに基づいた形式アシュランスケースを試作してもらって有効性を評価しようと計画した．

自動車部品メーカーと地方自治体（平塚市）から協力を得ることができた．これに伴い，FFO を車載システムに具体化した AFSCF 議論モデルと，地域防災システムに具体化した防災 FFO を考案した．しかし形式アシュランスケースは先端技術であり，対象技術領域の専門家に形式アシュランスケースを書いてもらうことは困難であった．一方で本研究チームは形式アシュランスケース記述のための技術は持ち合わせているものの，具体的な技術領域の知識を持ち合わせない．そのため，対象技術領域における利用に基づく有効性評価は困難であった．

FFO が依拠するオープンシステム・ディペンダビリティのライフサイクルを明確に記述した．また，本研究で得られたオープンシステム・ディペンダビリティの要件を IEC 62853 の制定活動に反映させることができた．オーストラリアや英国の標準化活動と協力して，上記の二つのライフサイクル概念を比較対照し，産業界における既存のライフサイクル概念と調和をとりながら，オープンシステム・ディペンダビリティのライフサイクル概念を確立し，FFO が依拠する概念を提供した．Face to face の研究討論の場として国際標準 WG（IEC TC56 WG4, ISO/IEC SC7 WG7）の plenary meeting や interim meeting の場を利用した．これらの WG はライフサイクル概念について，産業界，官界，学界にまたがるメンバーによる議論の場を提供しているためである．

本研究は，形式アシュランスケースというアシュランスケースの形態（とその形態に基づく処理技術）と，オープンシステム・ディペンダビリティの概念体系というアシュランスケースが保証すべき内容の二つ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，また，その過程で得られたオープンシステム・ディペンダビリティ達成のための要件を国際規格として具現化した．

3. 研究成果

以下の 4 つの研究目標をたてた．

研究目標1.　オープンシステム・ディペンダビリティ一般のための FFO の開発
研究目標2.　特定の技術領域における FFO の開発
研究目標3.　事例研究による有効性評価
研究目標4.　FFO が依拠するシステムライフサイクル概念の確立

3.1. 研究目標 1 オープンシステム・ディペンダビリティー般のための FFO の開発

FFO の機能のうち，対象領域によらず共通して必要となるものを，プログラミング言語 Agda によって実装する．図 2 の DEOS 基本構造（[16]Chapter 3.4.1）に従い作成した．

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，これを三つのサブゴール

- 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時に正しく対応する
- 障害対応：システム障害に対して，正しく応じする

に分けた．

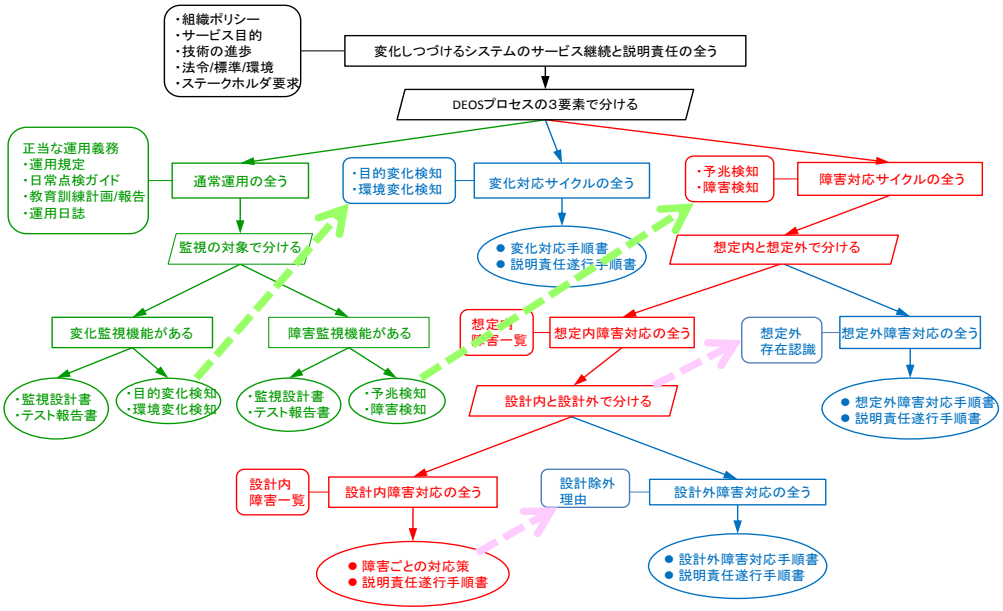

図-2　DEOS 基本構造

3.1.1. 一般オントロジーの形式記述

各サブゴールに対して，ゴール達成の議論に使う語彙（オントロジー）を決める文脈を表すモジュール C-通常運用，C-変化対応，C-障害対応を設けた．これらのモジュールは単に語彙を提供するだけでなく，それらの「型情報」も同時に定めている．型情報は，語彙の使い方をさだめるものである．Agda は構成的型理論に基づく言語なので，型情報を定めることにより，いわゆるデータ型の情報にとどまらず，語彙の意味を規定する公理をも記すことができた．

3.1.2. 一般議論フレームワークの形式記述

各種証憑から各サブゴールを導き出す議論の構造を，通常運用ケース，変化対応ケース，障害対応ケース の三つの Agda プログラムとして表現した．

証憑や手順書の存在を証拠として用いる議論のパターンを関数 証拠の吟味 と 手順の遂行 として実装して，議論の再利用を図った．

3.1.3. 一般 FFO の標準体系における位置づけの明確化

FFO が依拠するシステムライフサイクルとそれを対象とするアシュランスケースの構造に国際標準体系（IEC 62853 および ISO/IEC 15288 及び周辺の標準）の中に位置づけた．

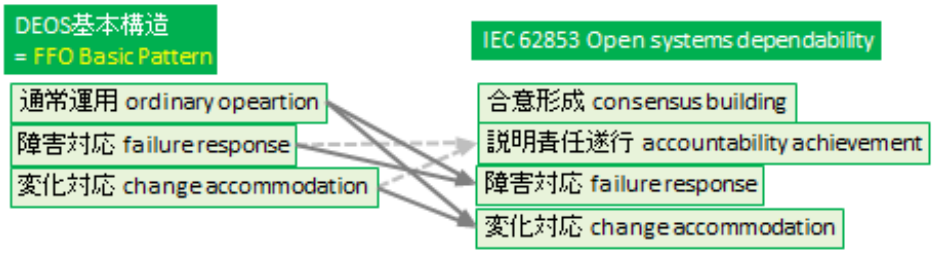

図-3　DEOS 基本構造と IEC62853 要件との対応

3.2.　研究目標 2 特定の技術領域における FFO の開発

特定の技術領域として，車載システムと防災システムをとりあげ，それぞれの技術領域に FFO を具体化したものを作成した．

3.2.1. 車載システム

車載システムの機能安全を規定する ISO26262 とオープンシステム・ディペンダビリティを規定する IEC62853 を合わせて要求することにより，機能安全の，より現代的な実現が可能になる．そこで我々はオープンシステム・ディペンダビリティのための FFO を車載システムの機能安全に具体化した AFSCF 議論モデルを考案した．

(1) 車載システム向けのオントロジーと議論フレームワークを形式記述

ISO26262 に従った開発を行うための機能安全テンプレート[8]が，（社）JASPAR によって提供されている．このうち，「機能安全技術テンプレート システム開発編」，および「機能安全技術テンプレート ソフトウェア開発編」の語彙定義を分析し，プログラミング言語 Agda で記述した．また，IEC62853 のアシュランスケースモデル（[11], Annex B）と ISO26262 準拠機能安全ケースのレイヤーモデル[10]を融合させて AFSCF 議論モデル（Automotive Functional Safety Case Framework）を考案し，Agda で記述した．

(2) AFSCF 議論モデルの標準体系における位置づけ

AFSCF 議論モデルは，ディペンダビリティ議論に関して IEC 62853 に準拠している．機能安全議論に関しては，ISO26262 に準拠している．以上が国際標準に関する位置づけである．

ISO 26262 はアシュランス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．それに伴い，JASPAR 機能安全テンプレートもアシュランスケースに関する詳細を規定して

いない．AFSCF 議論モデルに基づいて，JASPAR 機能安全テンプレートへの補遺を作成することができる．

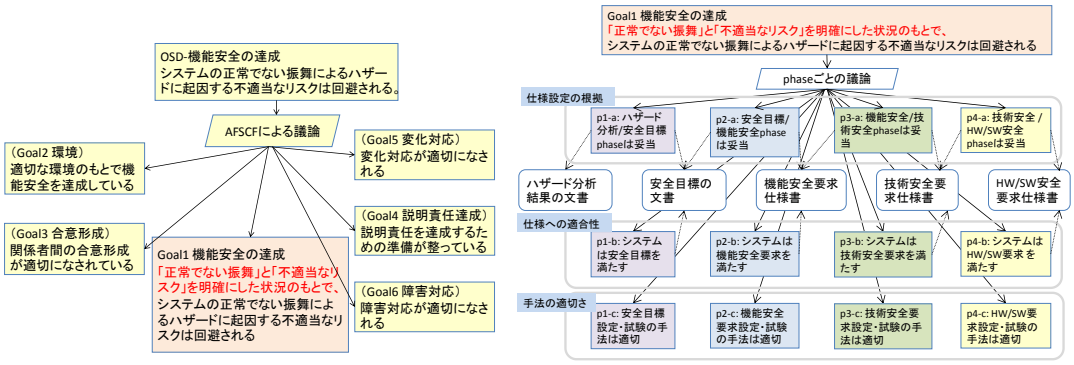

図-4　OSD 機能安全の達成の 6 つのサブゴール、機能安全の達成の phase ごとの議論

3.2.2. 防災システム

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防災業務にあたる，地方自治体の防災業務を研究の対象とした．地方自治体が行う防災関係業務は，「地域防災計画」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[19]に基づき地方自治体が作成している文書である．地域防災計画をもとに防災 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．

(3) 防災オントロジーの形式記述

地域防災計画が定める防災業務というオープンシステムのシステムライフサイクルを ISO/IEC/IEEE15288 に基づいて定義しようと試みた．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．

ISO/IEC/IEEE15288 では，システムライフサイクルにおける作業の集まりを「プロセス」と呼ぶ．30 のプロセスが定義されており，これらのプロセスを組み合わせて 1 つのシステムライフサイクルが構築される．プロセス間の時系列は ISO/IEC/IEEE 15288 ではなくシステムライフサイクル model によって定められる．

1 つのプロセスは複数の「アクティビティ」によって構成され，1 つのアクティビティは複数の「タスク」により構成される．その構造とは別に，1 つのプロセスには複数の「アウトカム」が定められている．アウトカムとは，あるプロセスを正しく実装（具体化）した際に実現する状態や成果のことである．

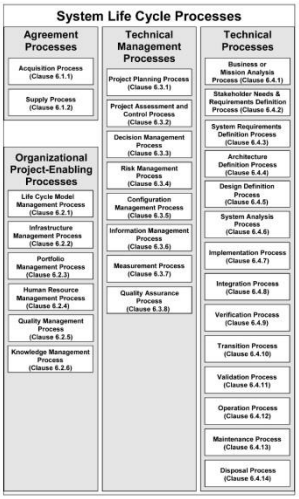

図-5　ISO/IEC/IEEE15288 が定める 30 のプロセス

ISO/IEC/IEEE 15288 は，タスクがどのようなものかを定めているが，実際にタスクをどのように記述していけばよいかは明らかでない．アクティビティ，プロセスについても同様である．そこで，プロセス・アクティビティ・タスクにどのような事項を記述すればよいかの枠組みを作る必要が生じた．その枠組みが 6W1H モデルである．

6W1H モデルとは，大まかに言えば，ISO/IEC/IEEE15288 に基づいてプロセス，アクティビティ，タスクを記述する際に 6 つのラベルを付ける手法のことである．6 つのラベルとは Who，What，Whom，Where，When，Why である．一般に，「ある 1 つの作業の記述」は，さらに細かい複数の作業の記述によって表現できる．ISO/IEC/IEEE15288 においても，1 つプロセスが複数のアクティビティで表現され，1 つのアクティビティが複数のタスクで表現されている．この，「1 つの作業と複数の作業との対応」を How と呼ぶ．

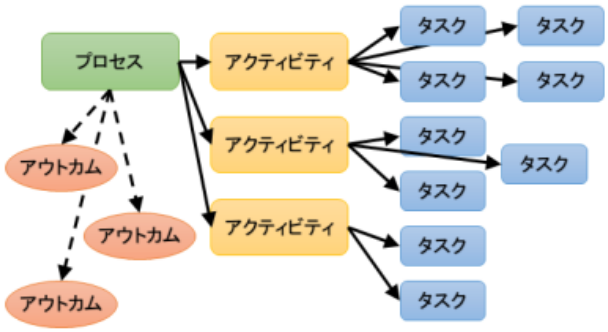

図-6　ISO/IEC/IEEE15288 におけるプロセスの構成

図-6 は，6W1H モデルを利用して ISO/IEC/IEEE15288 のプロセスを記述した例である．Who，What，Whom，Where，When，Why の 6 つのラベルを各プロセス，アクティビティ，タスクに付けた．これらのラベル付けについては，それがどのような作業であるのか容易に理解できるラベルにする必要がある．これらの定め方の指針を表-1 にまとめた．

表-1　6W1H モデルにおけるラベル付けの指針

ラベル名	記述の指針	例
Who	作業を行う個人名，役割名，組織名	X 部，X 部 Y 課，X 部 Y 課 Z 担当，X 部部長
What	作業．特に「何」を「どうする」のかに注意	水道管の被害状況を調査，給水車を運転
Whom	作業を行う対象がある場合に，対象となる個人名，役割名，組織名．ない場合「なし」	被災者，X 部 1 課，なし
Where	作業を行う場所	避難所 A，各避難所，X 部本部
When	作業を行う日時もしくは，作業の前後関係	発災後 24 時間以内，3 月 12 日，作業 A 後
Why	作業を行う目的，なぜその作業の成果物が必要か	被災者生命保護のため，給水準備のため

図-7 は，6W1H モデルの形式アシュランスケース記述言語 Agda による記述である．

```
module 6W1H where

open import Data.List

record 6W1H-Parameters : Set1 where
  field
    Who   : Set
    Whom  : Set
    What  : Set
    Where : Set
    When  : Set
    Why   : Set

module 6W1H-Model-Definition(6W1H-param : 6W1H-Parameters) where
  open 6W1H-Parameters 6W1H-param

  record 6W1H-Model : Set where
    inductive
    field
      who    : Who
      whom   : Whom
      what   : What
      where_ : Where
      when   : When
      why    : Why
      how    : List 6W1H-Model
```

図-7　6W1H モデルの Agda コード

(4) 防災議論フレームの形式記述

後述する事例研究に 6W1H モデルを適用して，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を与えた一例が DPP 議論モデルである．「DPP」とは，Decision（決定），Preparation（準備），Provision（実施）の頭文字をとったものである．多くの業務は表-2 の 3 つに分類されるという考えに基づく．

表-2　DPP 議論モデルの 3 分類とその意味

業務の分類	意味
決定(Decision)	ある業務を実施するために必要な情報収集と情報に基づいた開始・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

(5) 事例研究の結果を用いた成果の改善

6W1H モデルと DPP 議論モデルは，後述する神奈川県平塚市との共同研究を通して生まれたもので，それ自体が事例研究において随時議論し，改善した成果である．研究の順序および他の成果物との関係を図-8 に示す．

まず，地域防災計画に基づく防災オントロジー記述のために 6W1H モデルを考案した．次に，それを適用してプロセスを記述した結果，DPP 議論モデルを発見した．また，それらの研究にあたっては，一般 FFO，平塚市地域防災計画ほか関連文書，国際標準 ISO/IEC/IEEE15288 を参照した．

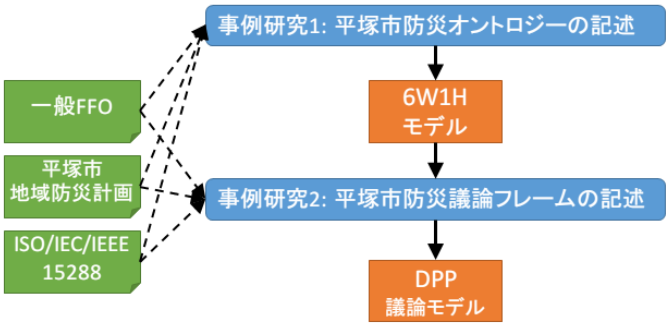

図-8　研究成果と事例研究との関係

(6) 防災 FFO の標準体系における位置づけ

一般 FFO と防災 FFO との比較を下図に示す．前述の一般 FFO に 6W1H モデルと DPP 議論モデルを加え，一般 FFO が持つフレームワークの一部を防災向けに修正したのが，防災 FFO である．

なお，6W1H モデルや DPP 議論モデルは，防災という技術領域のみならず，ISO/IEC/IEEE15288 に依拠したシステムライフサイクルを定めるための普遍的な枠組である．

図-9　一般 FFO と防災 FFO との比較

3.3.　研究目標 3 事例研究による有効性評価

3.2 で選んだ対象記述領域における事例研究への協力者を探索し，3.2 で作成したその対象技術領域に具体化した FFO の有効性評価を試みた．

3.3.1. 車載システム

(1) 導出パターン

導出パターンは，上位の仕様から下位の仕様を導出するパターンである．これを利用することで，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ていく．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-10 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．

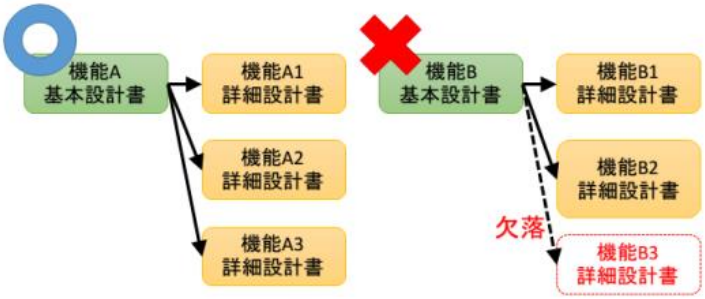

図-10　下位仕様と上位仕様との関係の例

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」ことをあらかじめ示しておくことが重要になる．それによって，上のような問題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-11 である．右の例で，機能 B1 と機能 B2 の詳細設計書が満たされても，機能 B が満たされるとは限らない，とすれば，何か別の機能が必要であることが判明する．B1，B2，B3 の三つから B が導かれるような B3 が必要である．

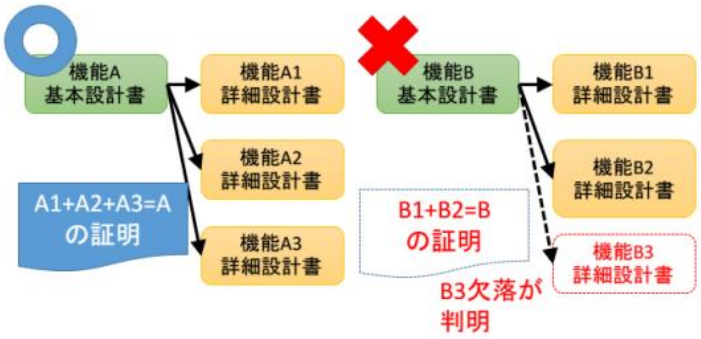

図-11　下位仕様と上位仕様との関係を事前に確認することで不備を発見

しかし，どうすればこの事実が示されるのかは開発者にとって自明ではない上に，ひとつひとつの下位仕様について上位仕様との関係を検討するのはコスト面の負担が大きいためである．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ことが考えられる．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．このようなテンプレートが導出パターンである．

3.3.2. 防災システム

神奈川県平塚市の協力を得ることができ，同市の地域防災計画に規定されている防災業務を対象として防災 FFO を適用したアシュランスケース記述実験を行った．

アシュランスケース記述を試みたが，防災業務のシステムライフサイクルは，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．これには大きく 2 つの理由がある．まず，地域防災計画が長年の加筆修正に伴って複雑かつ膨大になり，容易には理解が困難な文書となっていること．地域防災計画に記述すべき事項は，防災基本計画等によって規定されているが，どのような様式で記述すべきかの規定はないためである．もうひとつの理由は，防災計画がシステムライフサイクルの考えとは独立に策定されていることである．防災業務がシステムであるという考えは，世間一般に認知されたものではない．そのため，地域防災計画の執筆者が，防災業務とはシステムであるとは考えておらず，システムライフサイクルについても考慮にないことは当然である．

防災業務のシステムライフサイクルを構築するために，防災 FFO の 6W1H モデルを用いることとした．防災計画の全体を対象とすることは困難なので，応急対策時の給水業務に対象を絞った．これについて ISO/IEC/IEEE15288 に定める Tailored Process として「給水プロセス」を表-3 のように定義した．

表-3　6W1H モデルによる給水プロセスの定義

ID	Who	What	Whom	When	Where	Why
P	市	飲料水等を供給	被災者等	発災時	避難所等	市民の生命・身体の保護
A1	市	給水の実施を判断	×	発災後	災害対策本部設置場所	必要とする市民への給水
T1-1	総務部被害調査班、特別調査班	水道の被害状況を調査	×	発災後	×	給水開始/終了の決定
T1-2	総務部被害調査班、特別調査班	水道の被害状況を報告	総合対策部総合調整班	T1-1の後	×	給水開始/終了の決定
T1-3	土木復旧部	交通の状況を調査	×	発災後	×	給水開始/終了の決定
T1-4	土木復旧部	交通の状況を報告	総合対策部総合調整班	T1-3の後	×	給水開始/終了の決定
T1-5	給水部	給水体制の進行状況等を調査	×	給水開始後	×	給水継続/終了の決定
T1-6	給水部	給水体制の進行状況等を報告	総合対策部総合調整班	T1-5の後	×	給水継続/終了の決定
T1-7	災害対策本部	給水の実施を判断	×	T1-2,4,6の後	災害対策本部設置場所	必要とする市民への給水
A2	市	給水業務を準備	被災者等	A1の後	市内各地	必要とする市民への給水
T2-1	総合対策部広報班	汲み置きを連絡	自主防災組織	A1の後	×	飲料水の確保
T2-2	自主防災組織	汲み置きを呼びかけ	被災者等	T2-1の後	×	飲料水の確保
T2-3	県企業庁平塚水道営業所	貯水量を確認	×	A1の後	平塚配水池	飲料水の確保
T2-4	県企業庁平塚水道営業所	貯水量を連絡	災害対策本部	T2-3の後	×	飲料水の確保
T2-5	協定締結事業者	飲料水の状況を確認	×	A1の後	事業所	飲料水の確保
T2-6	協定締結事業者	飲料水の状況を連絡	災害対策本部	T2-5の後	×	飲料水の確保
T2-7	給水部	非常用貯水タンクの状況を確認	×	A1の後	非常用貯水タンク所在地	飲料水の確保
T2-8	給水部	非常用貯水タンクの状況を連絡	災害対策本部	T2-7の後	×	飲料水の確保
T2-9	消防部	火災の状況を確認	×	A1の後	災害対策本部	飲料水の確保
T2-10	給水部	臨時給水栓の設置を協議	消防部	T2-9の後	×	飲料水の確保
T2-11	給水部、避難部	臨時給水栓を設置	×	T2-10の後	消火栓所在地	飲料水の確保
T2-12	避難部	ろ水機を移動	×	A1の後	保管場所から利用場所へ	飲料水の確保
T2-13	避難部	耐震性プールの水をろ過	×	T2-12の後	ろ水機利用場所	飲料水の確保
T2-14	県企業庁平塚水道営業所	配水管を復旧	×	A1の後	配水管故障個所	飲料水の確保
T2-15	県企業庁平塚水道営業所	応急給水栓を設置	×	T2-14の後	避難所等	飲料水の確保
T2-16	給水部	給水の計画を決定	×	A1の後	災害対策本部設置場所	必要とする市民への給水
A3	市	給水業務を実施	被災者等	A2の後	避難所等	必要とする市民への給水
T3-1	給水部	給水車を移動	×	A2の後	元の場所から平塚配水池へ	飲料水の供給
T3-2	給水部	飲料水を移送	×	T3-1の後	平塚配水池から給水車へ	飲料水の供給
T3-3	給水部	給水車を移動	×	T3-2の後	平塚配水池から目的地へ	飲料水の供給
T3-4	給水部	給水車を移動	×	A2の後	元の場所から各事業所へ	飲料水の供給
T3-5	給水部	飲料水を移送	×	T3-4の後	事業所	飲料水の供給
T3-6	給水部	給水車を移動	×	T3-5の後	事業所から目的地へ	飲料水の供給
T3-7	給水部	給水車を移動	×	A2の後	元の場所から非常用貯水タンク所在地へ	飲料水の供給
T3-8	給水部	飲料水を移送	×	T3-7の後	非常用貯水タンク所在地	飲料水の供給
T3-9	給水部	給水車を移動	×	T3-8の後	非常用貯水タンク所在地から目的地へ	飲料水の供給
T3-10	市民、事業所	水道水を汲み置き	×	A2の後	自宅、各事業所	飲料水の確保
T3-11	避難所運営委員会等	飲料水を供給	被災者等	T3-3,6,9の後	目的地	市民の生命・身体の保護
T3-12	避難所運営委員会等	飲料水を供給	被災者等	A2の後	消火栓所在地	市民の生命・身体の保護
T3-13	避難所運営委員会等	耐震性プールの水を供給	被災者等	A2の後	ろ水機利用場所	市民の生命・身体の保護
T3-14	県企業庁平塚水道営業所	飲料水を供給	被災者等	A2の後	応急給水栓所在地	市民の生命・身体の保護

※IDの頭文字P=プロセス、A=アクティビティ、T=タスク

給水プロセスを定義する過程で DPP 議論モデルを作成することとなった．給水プロセスのアシュランスケースを，DPP 議論モデルに従って記述した（図-12）．赤枠で示したのが，DPP 議論モデルに従ってアシュランスケースのゴールを 3 つのサブゴールに分割した部分である．そして，防災 FFO の形式アシュランスケース・フレームワークを適用して，平塚市の防災業務全体のアシュランスケースを記述した（図-13）．青枠で示したのが，給水プロセスのアシュランスケース全体にあたるサブモジュールである．

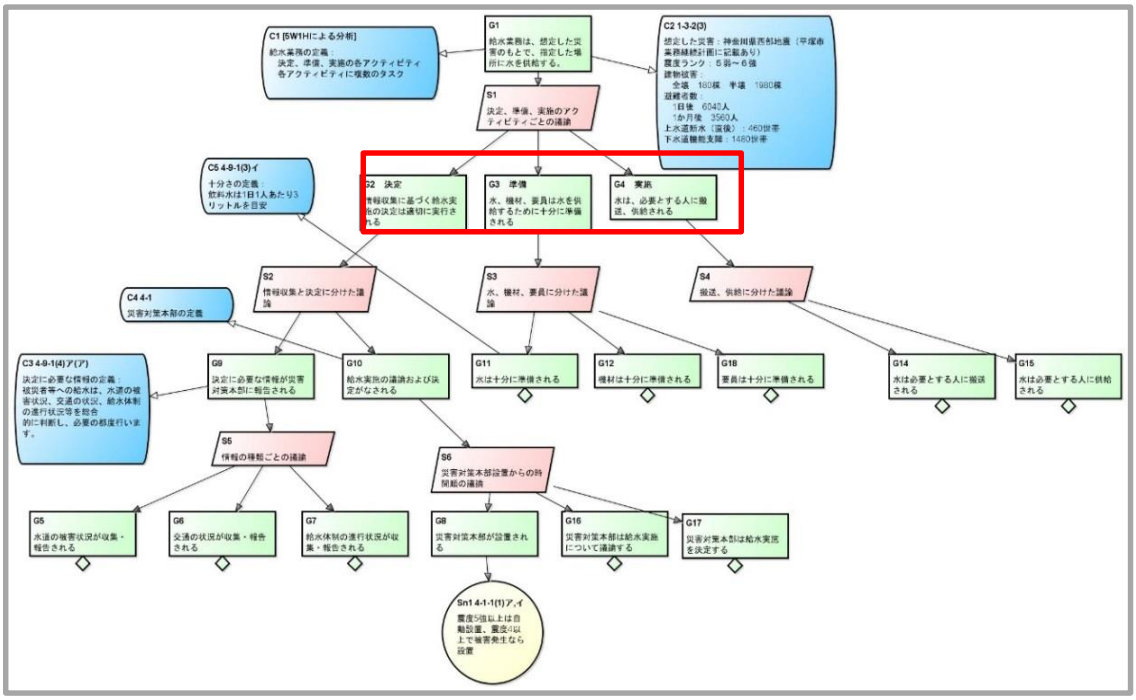

図-12　給水業務のアシュランスケース

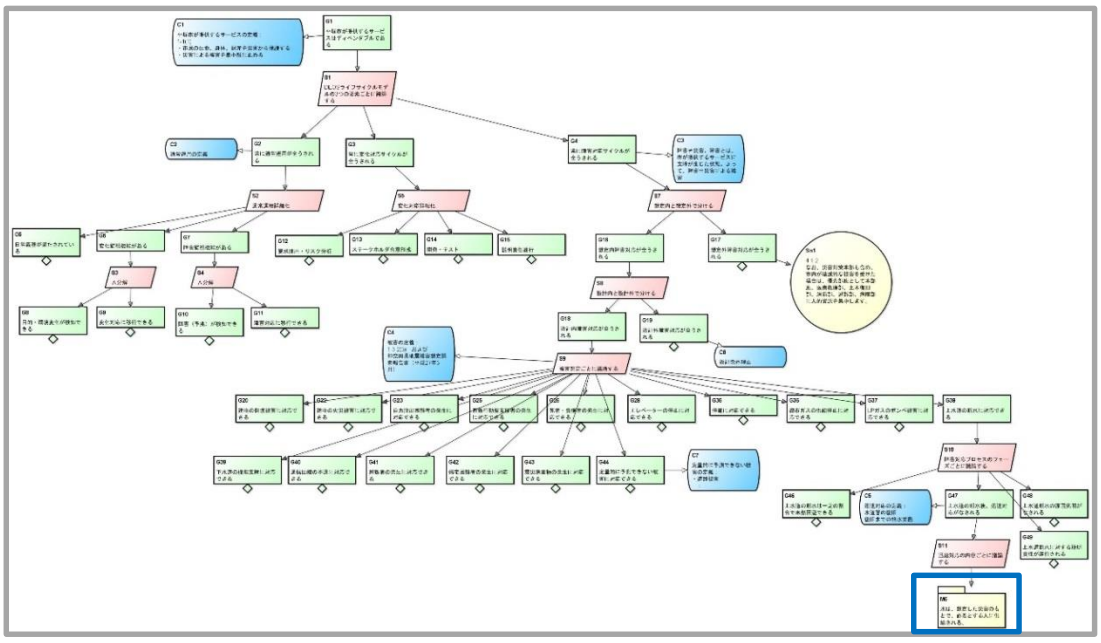

図-13　防災業務全体のアシュランスケース

このアシュランスケースでは，平塚市の防災業務全体のアシュランスケースのうち，障害対応の部分を想定される被害ごとに分割した．これにより，発災時に給水業務が実施される理由は，上水道の断水という被害（障害）に対応するためであるということが明らかになっている．

3.4. 研究目標 4 FFO が依拠するシステムライフサイクル概念の確立

システムライフサイクルに関連する既存の国際標準における用語定義を比較対照した上で，これらに矛盾しない形でオープンシステム・ディペンダビリティを達成するライフサイクルモデルとして提供する．

3.4.1. ライフサイクルの記述

DEOS プロセスのグラフを解釈して，意図する遷移系をペトリネットを用いて導き出す方法を考案した．ライフサイクルモデルの記述では，プロセスを節にし，節の間の辺によってプロセスの遷移を表現するグラフがよく用いられる．先行研究で提案された DEOS プロセスの記述もそのようなものである．しかし，このグラフを解釈して，意図する遷移系をどのように導き出すのかは自明ではない．我々は，ペトリネットを用いた解釈を提出した．

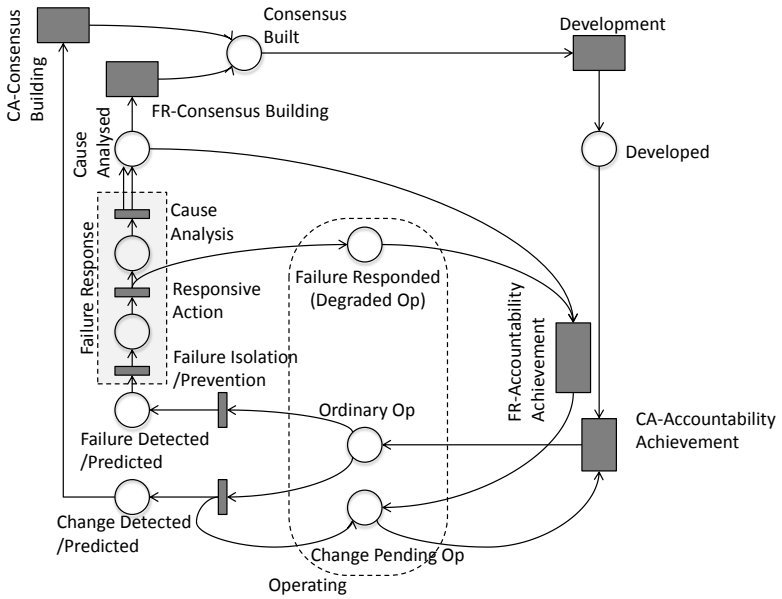

図-14　ペトリネットによる DEOS プロセスの表現

3.4.2. 国際標準との比較対照

ペトリネットによる定式化に加え，DEOS プロセスの各ステージと，ISO 国際標準体系においてシステムライフサイクルプロセスに関する最上位標準と位置付けられている ISO/IEC/IEEE 15288 が定義するシステムライフサイクルプロセスとの関連を明示したライフサイクルモデルを DEOS ライフサイクルモデルとして提出した．

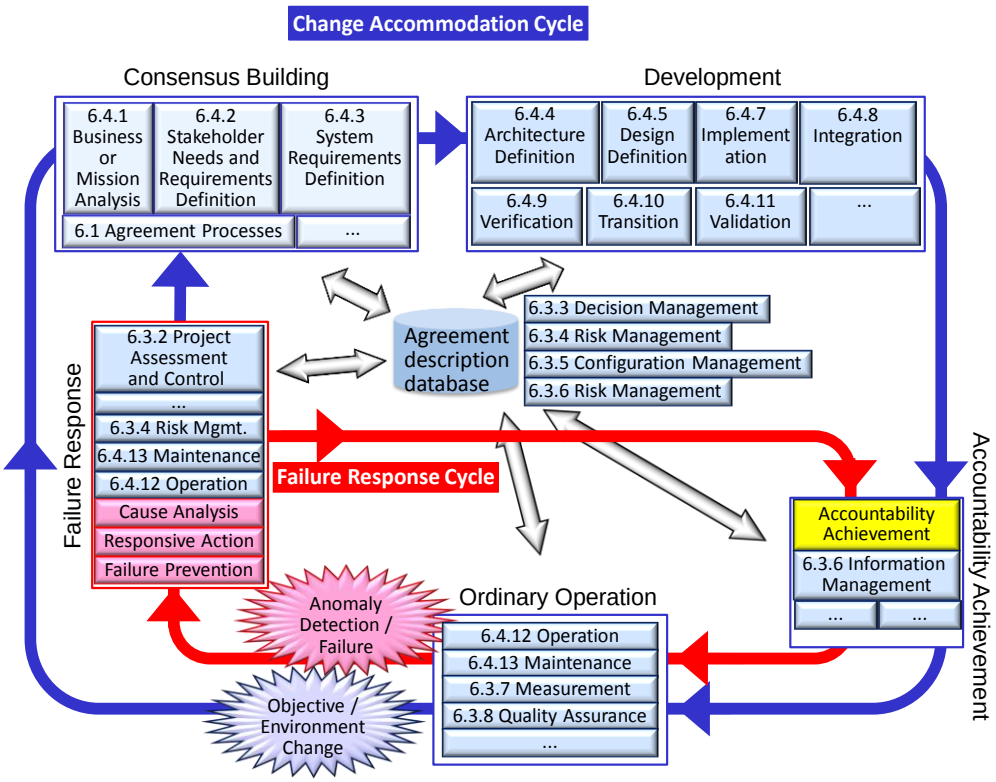

図-15　DEOS ライフサイクルモデル表現

4.　考察

本研究は，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ことを主張するアシュランスケースを書くためのツールを試作することが当初の目的であった．以下を当然のこととして（実施計画の想定仮説に明示せずに）想定して，本研究が開始された．

- アシュランスケースの書き方は，明らかである．さらに，
- オープンシステム・ディペンダビリティを要求するとはどういうことなのかが，すでに明らかになっている．

しかし，研究を開始すると，意外にも

- アシュランスケースの書き方についての課題が多数浮かび上がった．また，
- オープンシステム・ディペンダビリティがどういう属性なのか，明らかでない点が多々あった．

この結果，ツール試作と，オープンシステム・ディペンダビリティおよびアシュランスケースについてのより洗練された理解を得る作業が，相互に他を引き起こしつつ研究が進められた．研究による効果を考えるとき，アシュランスケースに関する知識が十分に広まっていない現状では，研究開始当初に目的としていたツールの存在そのものよりも，ツール試作を通して得られたオープンシステム・ディペンダビリティおよびアシュランスケースについての，より洗練された要件や概念が得られ，それが国際標準に反映されていることのほうが，産業を含めた社会に貢献する度合いが大きいと考えられる．

本研究により，策定中の国際標準 IEC 62853 Open systems dependability の技術的根拠が確立した．この標準は，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にシステムライフサイクルプロセスが満たすべき要件を規定している．

本研究の後，さらに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はじめて利用可能なものとなっていく．そのためにも国際標準化は重要であるが，本研究およびその先行研究により，オープンシステム・ディペンダビリティに関する研究成果は，必要があれば直ちに，国際標準として策定する活動を開始できる体制ができている．

1 研究の背景および目的

1.1 背景

1.1.1 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などとして，近年急速に注目され始めた．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が，現在では車載，鉄道，航空，医療システムの認証に関する以下のような国際標準においてアシュランスケースの提出が要請されているなど，その需要は増加傾向にある．

- 自動車の機能安全(ISO26262 Road vehicles - Functional safety)
- 鉄道信号の安全性(IEC62425 Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling)
- 航空の安全性(DO-178C/ED-12C Software Considerations in Airborne Systems Equipment Certification)
- 医療電子機器（FDA 510(k) 注入ポンプ（infusion pump）市販前届出）

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いくつか提案され，それぞれ普及が図られている（GSN[14]，CAE[1]，D-Case[16][17]など）．さらに，整合性検査を進めるため，形式言語による記法が，形式化アシュランスケースと呼ばれて研究されている（2.1.2 参照）．アシュランスケース自身に関する国際標準化も進められており，以下のような標準が出版されている．

- IEC/ISO 15026-2:2011 Systems and software engineering -- Systems and software assurance -- Part 2: Assurance case
- IEC 62741:2015 Demonstration of dependability requirements - The dependability case
- Object Management Group (OMG), Structured Assurance Case Metamodel (SACM), Version 1.1, 2013
- IEC 80001-2-9 Application of risk management for IT-networks incorporating medical devices -- Part 2-9: Application guidance -- Guidance for use of security assurance cases to demonstrate confidence in IEC/TR 80001-2-2 security capabilities（策定中）

アシュランスケースは，安全性に関する安全ケース（safety case）として始まった．2000 年代にはいって，セキュリティや信頼性（reliability, dependability）などのシステムの他の属性に関する議論の文書も取り扱われるようになり，属性に関して一般的なアシュランスケースの名が広まった．

我が国でも 2010 年代にはいって D-Case や D-Case in Agda など，アシュランスケースに関連する研究がはじまり（2.1.2 参照），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．

1.1.2 オープンシステム・ディペンダビリティ

システムの目的，目標，環境およ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，要求された時に要求どおり提供する能力はオープンシステム・ディペンダビリティ（OSD）とよばれており，我が国の研究プロジェクトの成果である（2.1.2 参照）．この概念はシステムのレジリエンス(resilience)[9]と共通点がある．また，説明責任や合意形成などの human factor への考慮や，変更管理（change management），リスク管理（risk management）などとも関連させながら，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ようとするものである．

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 達成が必要条件だとされている．また，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．

1.2 研究課題

アシュランスケースに関する研究課題のうち，本プロジェクトで取り扱うのは，

- 論理的整合性の担保
- オントロジーの明示
- 評価基準の明示

の三つである．

アシュランスケースは一般に大部な文書となるが，一方で，些細な論理的瑕疵が文書全体を無意味なものにする可能性がある．アシュランスケースにとって，論理的な整合性の担保だけでは十分ではないが，これは少なくとも必要である．しかし，膨大な記述の詳細にわたって厳密な検査を行うのは容易ではない．機械的検査の方法が求められている．

アシュランスケースの用語とその定義の集まり，つまりオントロジーを明示することは，アシュランスケースの「様式」を指定することにあたる．様式を用意することによって，チームによって執筆されるアシュランスケースの記述が統一され，執筆が容易になるだけでなく，アシュランスケースを読む側にとっても，理解を進めやすくなる．とくにアシュランスケースが認証に用いられる場合には，必要記載事項がだれにでも明らかになるように示すことが求められる．標準化された骨組みのもとで，予め用意された汎用的な部品を具体化したり組み合わせたりする穴埋め作業によってアシュランスケースを構築できるのが望ましい．

さらに，アシュランスケースを評価する判断基準が求められる．与えられたソフトウェアの安心・安全が達成されている，ということを議論するやり方は，一般にはいくつもあり得る．その中でどの議論展開が適切であるか，を評価して決めなければならない．そのときの判断基準が求められる．

1.3 研究の意義

本研究の対象であるアシュランスケースは，複雑で大規模なソフトウェアの全体像を掌握するための技術である．近年，医療，自動車，原子力，交通などの高信頼ソフトウェアが要求される分野でアシュランスケースを認証に取り入れる動きが出ており，国際標準化も

進んでいる．複雑で大規模なソフトウェアの全体像把握は，1970 年代に叫ばれたソフトウェア危機以来，一貫してソフトウェア工学の中心的課題である．したがって，簡単に完全な解決が得られるとは思われないものの，アシュランスケースはこの方向に着実な一歩を進める技術であり，本研究はソフトウェア工学の中心的課題解決に寄与するものである．

次に，本研究は形式アシュランスケース(Formal Assurance Case, FAC)のアプローチをとっている．このアプローチにより，大部なアシュランスケースの整合性を計算機によって検査することが可能になり，システムの検査官の注意を内容的な検査に集中させ，その結果，システムの安全安心の質が向上する．また，形式アシュランスケースは，proof assistant の技術をアシュランスケース構築に適用することを可能にするため，整合的でないようなアシュランスケースは当初から構築されず，整合的なものだけが構築される，いわゆる CbyC (correctness by construction)をアシュランスケースに対して実現する．

本研究では，国際標準化の会議におけるシステムライフサイクルの概念に関する議論が，研究討論の一部として利用される．本研究はシステムライフサイクルの概念に依拠するものである．システムライフサイクルのような抽象的な概念は，国際標準に規定されない限り，産業界において社会的に存在しないも同然である．この意味で，標準化はソフトウェアの高品質化に必須の新たなサービスの社会的枠組と価値を創出し，社会に認知させるために不可欠な過程である．標準化のこのような位置づけは情報産業（第三次産業，サービス産業）に特徴的なものであり，第二次産業（工業）における標準化の位置づけとは異なる．本研究は標準化活動の新しいニーズに応える情報技術研究活動の先端的事例を提供する．

2 実施内容

2.1 研究アプローチ

2.1.1 研究の全体像

本研究では，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 Formal assurance case Framework for Open systems dependability (FFO) を作成した．

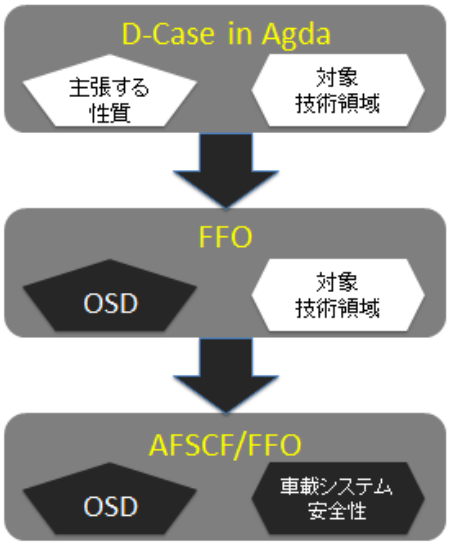

図 2-1　研究の全体像

先行研究（2.1.2 参照）で開発した D-Case in Agda は

(i) 形式アシュランスケースが主張する，システムの性質

(ii) 形式アシュランスケースが対象とする，システムの技術領域

の二つのパラメータをもっていた．このうち，パラメーター(i)をオープンシステム・ディペンダビリティ（OSD）に具体化したものが FFO である．D-Case in Agda の単なる具体化にとどまらず，以下のような工夫を考案し，FFO に反映させた．

(iii) 形式アシュランスケースの中で用いる語彙（オントロジー）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．

(iv) 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いくつか見出し，Agda の関数として実装した．例えば証拠の吟味，手順の遂行などの類型を提供した．

技術領域によらず，オープンシステム・ディペンダビリティ一般を取り扱う「一般 FFO」をまず開発した．このレベルでは，ソフトウェアライフサイクルが通常運用の他に，合意形成，障害対応，変化対応，説明責任の四つのプロセスビューを持つことが要件である．FFO

は，これらの要件達成の本質に関するアシュランスケースを，抽象度の高いレベルで構築可能にした．

次に，具体的な技術領域として車載システムおよび地域防災計画を選び，一般 FFO にそれぞれの領域知識を具備させたフレームワークを試作した．本研究の目的は FFO を試作し，その有効性を評価することであるが，有効性評価のためにシステムの現場から意見をきくためには，(ii)も具体化したフレームワークをさらに試作して提示する必要があった．そこで，FFO を何らかの技術領域に具体化したバージョンを作り，産業界からの協力を得て，具体化バージョンに基づいた形式アシュランスケースを試作してもらって有効性を評価しようと計画した．

自動車部品メーカーと地方自治体（平塚市）から協力を得ることができた．これに伴い，FFO を車載システムに具体化した AFSCF 議論モデルと，地域防災システムに具体化した防災 FFO を考案した．しかし形式アシュランスケースは先端技術であり，対象技術領域の専門家に形式アシュランスケースを書いてもらうことは困難であった．一方で本研究チームは形式アシュランスケース記述のための技術は持ち合わせているものの，具体的な技術領域の知識を持ち合わせない．そのため，対象技術領域における利用に基づく有効性評価は困難であった．

FFO が依拠するオープンシステム・ディペンダビリティのライフサイクルを明確に記述した．また，本研究で得られたオープンシステム・ディペンダビリティの要件を IEC 62853 の制定活動に反映させることができた．オーストラリアや英国の標準化活動と協力して，上記の二つのライフサイクル概念を比較対照し，産業界における既存のライフサイクル概念と調和をとりながら，オープンシステム・ディペンダビリティのライフサイクル概念を確立し，FFO が依拠する概念を提供した．Face to face の研究討論の場として国際標準 WG（IEC TC56 WG4, ISO/IEC SC7 WG7）の plenary meeting や interim meeting の場を利用した．これらの WG はライフサイクル概念について，産業界，官界，学界にまたがるメンバーによる議論の場を提供しているためである．

本研究は，形式アシュランスケースというアシュランスケースの形態（とその形態に基づく処理技術）と，オープンシステム・ディペンダビリティの概念体系というアシュランスケースが保証すべき内容の二つ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，また，その過程で得られたオープンシステム・ディペンダビリティ達成のための要件を国際規格として具現化した．

2.1.2 関連するこれまでの研究について

科学技術振興機構 CREST 制度による DEOS プロジェクトにおいて，形式アシュランスケース(Formal Assurance Case, FAC)の研究が行われた．アシュランスケースは元来，日本語や英語などの自然言語によってシステムの安全・安心に関する議論を記述したものであった．これに対して，形式アシュランスケースは形式言語によって議論を記述したものである．

DEOS プロジェクトでは形式アシュランスケース技術に基づいて，アシュランスケース支援システム D-Case in Agda が開発された．D-Case in Agda は プログラミング言語 Agda によってアシュランスケースを記述し，CbyC (Correctness by Construction)，整合性検

査，プレースホルダによる記述の段階的詳細化，抽象化，モジュール化，グループ共同作業など，D-Case に対する自動処理を支援するシステムであった．

形式アシュランスケースは，上記の三つの問題のうち「論理的な整合性の担保」のための方法を与える．形式アシュランスケースに対しては，プログラミング言語処理技術を用いて，整合性検査自動化の支援が比較が可能である．

「オントロジーの明示」,「評価の判断基準」の問題について，学界や標準において解決策が検討されているが，論理的な整合性の担保と結びついた解決策には至っていなかった．

2.1.3 研究目標

(1) 想定する仮説

以下に，本研究で想定する仮説を列挙する．

仮説1 オープンシステム・ディペンダビリティは，複雑かつ大規模なソフトウェアのディペンダビリティを議論する上で欠かせない性質である．

仮説2 アシュランスケースの概念体系を与える用語と意味，つまりオントロジーを明示することによって，アシュランスケース構築や評価の品質が向上する．対象のオントロジーが明示されない議論は，曖昧なものになりがちである．

仮説3 アシュランスケースのオントロジーと議論を明確に分離するとその品質が飛躍的に向上する．

仮説4 アシュランスケースを形式言語で記述し，計算機処理を可能とすることによって，アシュランスケースの整合性，厳格さ，一括処理の利便を大幅に向上させることができる．

仮説5 アシュランスケース・フレームワークの有効性を，それを用いた記述実験によって評価することができる．しかも，実験の数が多ければ多い程，それらを総合した有効性の評価の信憑性は高くなる．

仮説6 アシュランスケース・フレームワークが参照するライフサイクルなどの抽象的概念は，国際標準に規定されていない限り，社会的には存在しないも同然である．従って，アシュランスケース・フレームワーク自体も，それが依拠する概念が国際標準に規定されない限り，普及が困難である．

仮説7 アシュランスケースの計算機処理により，アシュランスケースのライフサイクルにかかるトータルコストが，現状の半分以下に削減される．

(2) 到達目標の設定

想定する仮説に基づいて以下の到達目標を設定した．

到達目標 1 FFO 提供

オープンシステム・ディペンダビリティ達成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（FFO（Formal assurance case Framework for Open systems dependability））の提供．FFO は，ソフトウェアの開発フレームワークと同様に，アシュランスケース記述の部品と骨組みを用意し，それらを組み合わせることによってアシュランスケースを構築するフレームワークである．想定する仮説 1 によれば，このよ

うなアシュランスケースの構築は，複雑かつ大規模なソフトウェアのディペンダビリティを議論するために避けて通ることができない過程である．

到達目標 2 オントロジー明示

アシュランスケース構築および評価に用いるためのオントロジーを明示する記述機構，さらにオントロジーと議論の分離を促進する記述機構の提供．想定する仮説 2 と仮説 3 によれば，このような記述機構により，アシュランスケース自体の品質が向上するばかりでなく，その評価過程の品質も向上する．

到達目標 3 形式記述機構の提供

アシュランスケースの形式記述を可能にする機構の提供．想定する仮説 4 によれば，その結果，アシュランスケース一括処理の自動化（機械化）が可能になり，アシュランスケースの整合性，厳格さが向上する．

到達目標 4 記述実験

少なくとも一つの技術領域について，FFO に基づくアシュランスケース記述実験がなされて，その有効性の評価が得られること．想定する仮説 5 によれば，記述実験による評価は数多くなされればなされる程，評価の信憑性が高くなる．

到達目標 5 国際標準との整合

FFO の内容と国際標準との整合．想定する仮説 6 によれば，整合性を達成することは FFO の普及のための必要条件（前提条件）である．

到達目標 6 トータルコスト半減

アシュランスケースの構築，保守をはじめとするライフサイクルのコストを現状の半分以下に削減する．想定する仮説 7 によれば，この目標達成のためには，アシュランスケースの計算機処理を可能にすれば十分である．

(3) 研究目標の設定

本研究では上記(2)到達目標の設定を目指すため、以下の研究目標を設定した．

研究目標1.　オープンシステム・ディペンダビリティ一般のための FFO の開発

研究目標2.　特定の技術領域における FFO の開発

研究目標3.　事例研究による有効性評価

研究目標4.　FFO が依拠するシステムライフサイクル概念の確立

各研究目標の概略と，到達目標との関係を以下に説明する．

1) 研究目標 1 「オープンシステム・ディペンダビリティ一般のための FFO の開発」について

この目標は，システムのオープンシステム・ディペンダビリティ達成を主張する形式アシュランスケースを記すために必要な語彙の定義と議論の構造を，プログラミング言語 Agda のモジュールとして実装したフレームワーク FFO（Formal assurance case Framework for Open systems dependability）を試作することである．どのような技術領域のシステムに対しても，オープンシステム・ディペンダビリティ達成を主張するために最低限必要な語彙定義と議論構造だけを FFO は提供するものとする．

研究目標 1 が達成されれば，到達目標 1，到達目標 2，到達目標 3，到達目標 6 が達成される．なぜならば，FFO が実装されるので，到達目標 1 が達成される．また，そこではオープンシステム・ディペンダビリティ一般についてオントロジーが明示されるので，到達目標 2 が達成され，実装が言語 Agda によって行われるので到達目標 3 が達成され，さらに，到達目標 3 と仮説 7 により到達目標 6 が達成されるからである．

2) 研究目標 2 「特定の技術領域における FFO の開発」について

この研究目標では，車載ソフトウェア，スマートグリッド，インターネット等の技術領域から一つを選び，[研究目標 1 オープンシステム・ディペンダビリティ一般のための FFO の開発] の達成において実装した一般的フレームワークを，その技術領域に具体化して，その領域独特の事情を反映したフレームワークを実装することとする．また，[研究目標 3 事例研究による有効性評価] の結果を受けて，この実装を改善する．実装の過程では，[到達目標 2 オントロジー明示] が達成されるよう考慮することとする．

本研究目標 2 は，技術領域を特定せずオープンシステム・ディペンダビリティ一般に設定した研究目標 1 の結果が，少なくともいくつかの具体的な技術領域では有効であることを示すことにより，研究目標 1 の達成の意義を与えるために設定された．

3) 研究目標 3 「事例研究による有効性評価」について

実働のソフトウェアライフサイクルに FFO を適用してアシュランスケースの記述実験を行う．このことによって FFO の有効性を評価し，必要があれば FFO を改善する．

研究目標 3 の達成により，到達目標 4 が達成される．

4) 研究目標 4 「FFO が依拠するシステムライフサイクル概念の確立」について

FFO はシステムのディペンダビリティに関するフレームワークであるが，産業界ではディペンダビリティライフサイクル(IEC 60300-1)とシステムライフサイクル(ISO/IEC 15288)が独立に規定されているのが現状である．これら既存のライフサイクル概念を比較対照した上で，双方に矛盾しない形でオープンシステム・ディペンダビリティのライフサイクルを確立し，FFO が依拠する概念として提供した．

研究目標 4 の達成により，到達目標 5 が達成される．

2.2 研究の活動実績・経緯

研究の手順と進捗実績表を以下に示す．

(1) 研究の手順

表 2-1　進捗実績表の進捗状況表を基に研究の手順を説明する．

オープンシステム・ディペンダビリティ一般を取り扱う FFO をまず開発する（研究目標 1-作業項目(ア)(イ)(エ)）．このレベルでは，ソフトウェアライフサイクルが通常運用の他に，合意形成，障害対応，変化対応，説明責任の四つのプロセスビューを持つことなどが要件である．FFO は，これらの要件達成の本質に関するアシュランスケースを，抽象度の高いレベルで構築可能にする．

次に，特定の技術領域を選び，FFO にその領域知識を具備させ，抽象レベルの高い主張を具体的証憑で支えるアシュランスケースの構築を可能とする（研究目標 2-(ア)（イ)(エ)）．

この抽象レベルでの FFO の開発や有効性評価には実働のソフトウェアライフサイクル関係者による協力が必要である．初めの一般 FFO の開発と並行して，事例研究の協力先を探索し，共同研究契約を締結する（研究目標 3-(ア)(イ)）とともに記述実験，事例研究を実施する（研究目標 3-(ウ)(エ)）．記述実験の結果を検討して開発した FFO の有効性を評価し，妥当でない点を改善する（研究目標 1-(ウ)，研究目標 2-(ウ)，研究目標 3-(オ)）．

FFO の前提となるべきオープンシステム・ディペンダビリティのライフサイクルは未だ確立したものではないため，研究討論の場として国際標準 WG（IEC TC56 WG4, ISO/IEC SC7 WG7）の plenary meeting や interim meeting の場を利用し，この概念を確立して明確に記述する（研究目標 4-(ア)(イ)）．

本研究では，形式アシュランスケースというアシュランスケースの形態（とその形態に基づく処理技術）と，オープンシステム・ディペンダビリティの概念体系というアシュランスケースがアシュアすべき内容の二つをアシュランスケース開発フレームワークとして実用化し，オープンソースとして公開する．

表 2-1　進捗実績表

作業項目	H27 年度																		H28 年度																	
	6 月		7 月		8 月		9 月		10 月		11 月		12 月		1 月		2 月		3 月		4 月		5 月		6 月		7 月		8 月		9 月		10 月		11 月		12 月		1 月		2 月	
目標１　一般 FF0 の開発 (ア)：一般オントロジーを形式記述する		■	■	■	■	■	■		■	■	■	■	■	■	■	■	■																					
(イ)：一般議論フレームワークを形式記述する					■	■			■	■	■	■	■	■	■	■	■																					
(ウ)：研究目標３の結果を用いて改善する																																				■	■	■	
(エ)：標準体系における位置づけを明確にする				■	■	■	■					■					■	■	■	■																			
目標２　特定の技術領域における FF0 の開発 (ア)：特定技術領域向けオントロジー形式記述													■	■	■	■	■	■	■	■	■	■	■															
(イ)：特定技術領域向け議論フレーム形式記述																							■	■	■	■	■	■	■	■	■	■	■	■	■	■			
(ウ)：研究目標３の結果を用いて改善する																																				■	■	■	
(エ)：標準体系における位置づけを明確にする															■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
目標３　事例研究による有効性評価 (ア)：アシュランスケース記述事例研究相手先探索		■	■	■	■	■	■	■	■	■	■	■	■	■																							
(イ)：事例研究協力者との契約									■	■	■	■	■	■																							
(ウ)：アシュランスケース記述対象の技術領域調査											■	■	■	■	■	■	■	■	■	■					■	■	■	■										
(エ)：アシュランスケース記述実験																													■	■	■	■	■	■	■	■	■	■	■	■	
(オ)：(エ)の過程および結果に基く FF0 の有効性評価																																				■	■	■	
目標４　システムライフサイクル概念の確立 (ア)：ライフサイクルの記述	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	
(イ)：国際標準との比較対象																	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
最終報告書																																				■	■	■	■

(2) 内部・外部打合せの実施状況と，学会及び研究討論参加状況

① 内部打合せ実施状況

毎週 1 回を基本に合計 62 回実施した．研究方針を審議して定め，あるいは軌道修正し，研究活動を進めた．研究員間の意思疎通を図った．

② 外部打合せ実施状況

自動車部品メーカー，平塚市とそれぞれ共同研究契約を結び，折々で事例研究のための情報交換をした．前者との共同研究を進める中で領域知識についての知見を頂戴した．後者とは 22 回打ち合わせを実施して，地域防災計画およびその周辺領域についても一定の知見を得た．

③ 学会及び研究討論参加状況

- システムアシュランス研究会
  - ✧ 内容：形式アシュランスケースに関する研究討論の場を設けるために開設し運営(研究全般)
  - ✧ 期日/場所：2015-07/KU ポートスクエア，2014-12/同左，2014-07/同左
  - ✓ 口頭発表：木下佳樹「IEC 62853 オープンシステムズディペンダビリティの最新動向」
  - ✓ 口頭発表：武山誠「DEOS ライフサイクルモデルについて」
  - ✓ 口頭発表：木下佳樹「IEC TC56 Dependability 活動報告」
  - ✓ 口頭発表：木下佳樹「DEOS 標準化動向」
- WOSD (Workshop on Open Systems Dependability, (IEEE ISSRE Workshop))
  - ✧ 内容：OSD に関する研究討論の場，開設し運営(研究全般)
  - ✧ 期日/場所：2015-11/Washington DC,US，2014-11/Naples,Italy
  - ✓ 口頭発表：Yoshiki Kinoshita, “Open systems dependability standardization activity in IEC TC56”
- DEOS シンポジウム
  - ✧ 内容：国内での OSD に関する研究討論の場(研究全般)
  - ✧ 期日/場所：2015-06/慶應大学日吉キャンパス，2014-06/同左
  - ✓ 口頭発表：木下佳樹「DEOS 関連国際標準の動向（IEC62853:Open Systems Dependability)」
  - ✓ 口頭発表：武山誠「ディペンダビリティ技術の国際標準化動向と DEOS」
- Workshop on Logical Analysis of Descriptions and their Representations (NII Shonan Seminar)
  - ✧ 内容：アシュランスケース，システム仕様，産業標準などの記述に対する論理的解析手法の適用に関する国際ワークショップ(研究全般)
  - ✧ 期日/場所：2015-01/湘南国際村
  - ✓ 口頭発表：Yoshiki Kinoshita “Formal assurance case”

- ✓ 口頭発表：Makoto Takeyama “Formal Assurance Case in Agda (FACIA)”
- ASSURE (International Workshop on Assurance Cases for Software-intensive Systems)
  - ✧ 内容：アシュランスケース研究全般に亘る研究討論
  - ✧ 期日/場所：2015-09/Delft,the Netherlands
  - ✓ 口頭発表：Shuji Kinoshita, “Towards Assurance Arguments of Local Disaster Management Plans”
  - ✓ パネル：Yoshiki Kinoshita, “The Role of Argumentation in Certification and Safety Risk Management”
- SAFECOMP (International Conference on Computer Safety, Reliability & Security)
  - ✧ 内容：安全性・信頼性・セキュリティ研究全般に亘る研究討論
  - ✧ 期日/場所：2015-09/Delft,the Netherlands
- Adelard LLP meeting (Robin Bloomfield 等)
  - ✧ 内容：研究全般に亘る研究討論
  - ✧ 期日/場所：2015-09/London,UK
- AIM XXI (Agda Implementors’ Meeting XXI)
  - ✧ 内容：Agda 言語の応用・実装に関する研究討論
  - ✧ 期日/場所：2015-06/Goteborg,Sweden, 2014-10/Tallinn,Estonia
- AAA (International Workshop on Argument for Agreement and Assurance)
  - ✧ 内容：合意形成・アシュランス議論の研究全般に亘る研究討論
  - ✧ 期日/場所：2015-11/慶応大学日吉キャンパス
  - ✓ ポスター展示とデモ：Shuji Kinoshita, “Formal Assurance Case in Agda (FACIA)”
- DSW (ディペンダブルシステムワークショップ)
  - ✧ 内容：ディペンダビリティ―研究全般に亘る研究討論
  - ✧ 期日/場所：2015-12/熱海, 2014-12/熱海
  - ✓ ポスター発表：中原早生「AFSCF 自動車機能安全議論について」
  - ✓ ポスター発表：木下修司「地域防災計画のアシュランス議論に向けて」
  - ✓ ポスター発表：木下修司「平塚市地域防災計画の整合性検査」
- IEC TC56 Plenary
  - ✧ 内容：目標４に関する研究討論と国際標準制定
  - ✧ 期日/場所：2015-10/Glasgow,UK, 2014-10/Praha,Czech
- ISO/IEC SC7 WG7
  - ✧ 内容：目標４に関する研究討論と国際標準制定
  - ✧ 期日/場所：2015-11/Hoboken,US, 2015-05/Rio de Janeiro,Brasil
- TC56 WG4 meeting
  - ✧ 内容：目標４に関する研究討論と国際標準制定の国内会議
  - ✧ 期日/場所：2014-08, 2014-10, 2014-12, 2015-02, 2015-06, 2015-08, 2015-11, 2016-01 /日本規格協会

2.3 研究実施体制

(1) 実施体制

実施体制を以下に示す.

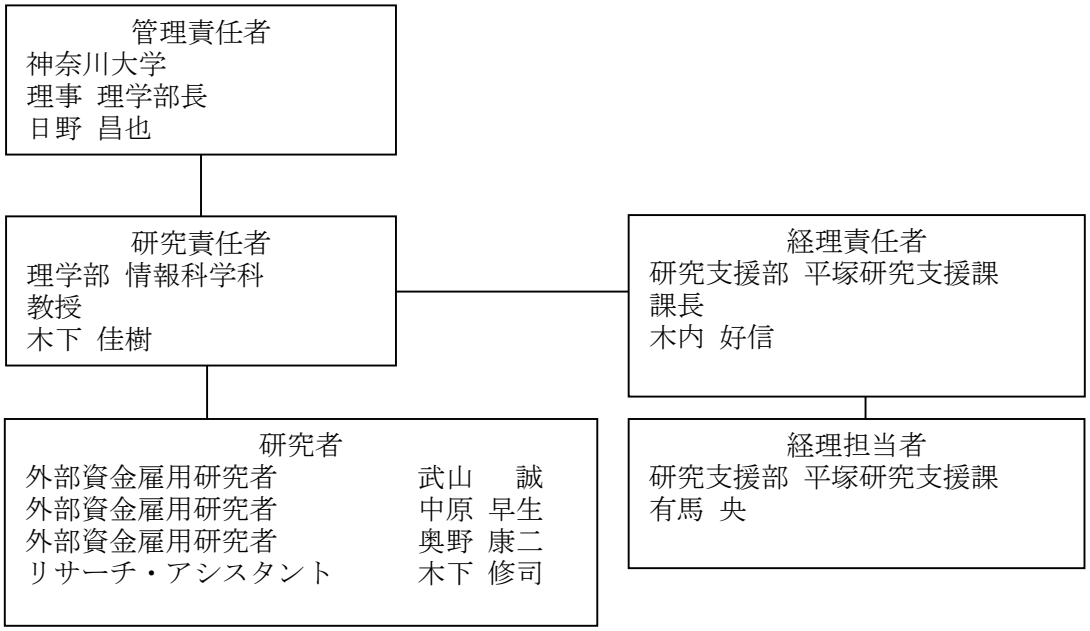

図 2-2 実施体制図

(2) 研究者プロフィール

研究責任者，各研究者のプロフィールの役割等を以下に示す.

① 研究責任者プロフィール

(ふりがな) 氏　名	きのした　よしき 木下 佳樹
生年月日	1956（昭和31）年 8月14日
所属機関	神奈川大学
所属（部署名）	理学部 情報科学科
役　職	教授
住　所	〒259-1293 神奈川県平塚市土屋2946
TEL	0463-59-4111（代）
E-mail	yoshiki@kanagawa-u.ac.jp

【学歴（大学卒業以降）】 1981.3 東京大学理学部情報科学科卒業 1983.4-1984.3 同上　研究生 1984.4 東京大学大学院理学系研究科情報科学専攻 1989.3 同修了（理学博士）	【職歴】 1981.4-1983.3 テキサスインスツルメンツ 1989.4-2001.3 通産省工技院電子技術総合研究所 2001.4-2013.3（独）産業技術総合研究所 2013.4より現職

【研究実績】

1991-1998「新ソフトウェア構造化モデル」研究分担者（通商産業省産業技術開発制度）

2002-2007「検証における記述量爆発問題の構造変換による解決」研究代表者（JST CREST）

2004-2010 産業技術総合研究所システム検証研究センター長

2008-2013「利用者指向ディペンダビリティの研究」研究代表者（JST CREST）

【主な論文・著書】

● Yoshiki Kinoshita and Makoto Takeyama. Assurance case as a proof in a theory: towards formulation of rebuttals. In C. Dale and T. Anderson, eds., *Assuring the Safety of Systems*, ISBN: 978-1481018647, pp. 205-230, 2013.

● Yoshiki Kinoshita and Anthony John Power. Data refinement and algebraic structure. *Acta Informatica*, 36(9/10), 693-719, 2000.

● Yoshiki Kinoshita, Anthony John Power and Makoto Takeyama. Sketches. *Journal of Pure and Applied Algebra*, 143, 275-291, 1999.

● Yoshiki Kinoshita. A bicategorical analysis of E-categories. *Mathematica Japonica*, 47(1), 157-169, 1998. MR 99b:18010.

● Yoshiki Kinoshita and Anthony John Power. Lax naturality through enrichment. *Journal of Pure and Applied Algebra*, 112(1): 53-72, 1995.

② 研究者略歴と役割

1) 武山 誠 (Ph.D (Computer Science))

- 職名：神奈川大学プログラミング科学研究所プロジェクト研究員，理学部情報科学科非常勤講師
- 専門分野：型理論，ディペンダビリティ
- 略歴：エディンバラ大学計算機科学科博士課程終了，クイーンズ大学計算科学科，シャルマース工科大学計算機科学科，(独) 産業技術総合研究所システム検証研究センターを経て 2013 年より現職
- 担当：研究目標 1 および 4 を担当

2) 中原 早生

- 職名：神奈川大学プログラミング科学研究所プロジェクト研究員，理学部情報科学科非常勤講師
- 専門分野：プログラム理論
- 略歴：京都大学大学院理学研究科数理解析専攻博士課程中途退学，京都大学数理解析研究所助手，東京大学理学部情報科学科助手，広島大学総合科学部助教授，産業技術総合研究所招聘研究員を経て 2013 年より現職.
- 担当：研究目標 2 および 3 のうち，車載システムを担当

3) 奥野 康二

- 職名：神奈川大学プログラミング科学研究所プロジェクト研究員
- 専門分野：組込みシステム
- 略歴：横浜国立大学電気工学科卒業，(株) 精工舎，矢崎総業 (株)，(独) 産業技術総合研究所を経て 2014 年より現職
- 担当：研究目標 2 および 3 のうち，車載システムを担当

4) 木下 修司

- 職名：神奈川大学プログラミング科学研究所リサーチ・アシスタント (理学研究科情報科学専攻博士課程在学中)
- 専門分野：ディペンダビリティ
- 略歴：東京大学文学部卒業，日本コントロールシステム (株) を経て，奈良先端科学技術大学院大学情報科学研究科博士前期課程修了，2014 年より現職.
- 担当：研究目標 2 および 3 のうち，地域防災システムを担当

3 研究成果

3.1 研究目標 1「オープンシステム・ディペンダビリティ一般のための FFO の開発」

3.1.1 当初の想定

(1) 研究内容

システムのオープンシステム・ディペンダビリティ達成を主張する形式アシュランスケースを記すために必要な語彙の定義と議論の構造を，プログラミング言語 Agda のモジュールとして実装したフレームワーク FFO(Formal assurance case Framework for Open systems dependability) を試作する．多様な技術領域のシステムに対してオープンシステム・ディペンダビリティ達成を主張することができる．どのような技術領域のシステムに対しても，オープンシステム・ディペンダビリティ達成を主張するために最低限必要な語彙定義と議論構造だけを FFO は提供するものとする．

(2) 想定課題と対応策：FFO の規模の大きさと複雑さ，FFO の保守

オープンシステム・ディペンダビリティ達成には，技術面だけでなく，経営面，社会倫理面など，多様な側面からの考慮が必要であり，また領域知識などは随時更新される必要がある．FFO で構築するアシュランスケースはもちろん，FFO 自体も大規模・複雑になることが予想される．本研究で記述に用いる言語 Agda は汎用プログラミング言語として先進の言語機構を備えているため，プログラミングに関するソフトウェア工学の最新の技術をアシュランスケースにそのまま適用することができる．

3.1.2 研究プロセスと成果

(1) 研究プロセス

システムのオープンシステム・ディペンダビリティ達成を主張する形式アシュランスケースのフレームワークをプログラミング Agda のライブラリとして実装する．この項目を次の 3 つの小項目に分けて進める．

以下の 3 つの小項目のうち，①②は並行してすすめ，これらが終了してから，その内容を国際標準に反映させることによって③をすすめる．

① 一般オントロジーを形式記述する

FFO では，形式アシュランスケースをオントロジー定義部分(Context)と主張の議論の部分(Argument)に分けたが (2.1.1(iii))，このうち Context 部分を Agda 言語によって実装する．

② 一般議論フレームワークを形式記述する

FFO の Argument 部分を Agda 言語によって実装する．

③ 一般 FFO の標準体系における位置づけを明確にする

FFO 試作のために用いた（形式化する以前の非形式的な）アシュランスケース・フレームワークの内容をもとに，オープンシステム・ディペンダビリティの国際標準案 IEC 62853 Open Systems Dependability の草稿を執筆し，内容を反映させる．

(2) 具体的な研究成果の内容

フレームワークが用いる議論の進め方は DEOS 基本構造（先行研究[16]Chapter 3.4.1. の研究成果）に従い，2.1.1(iii)(iv)に記した工夫を加えて FFO を作成した．

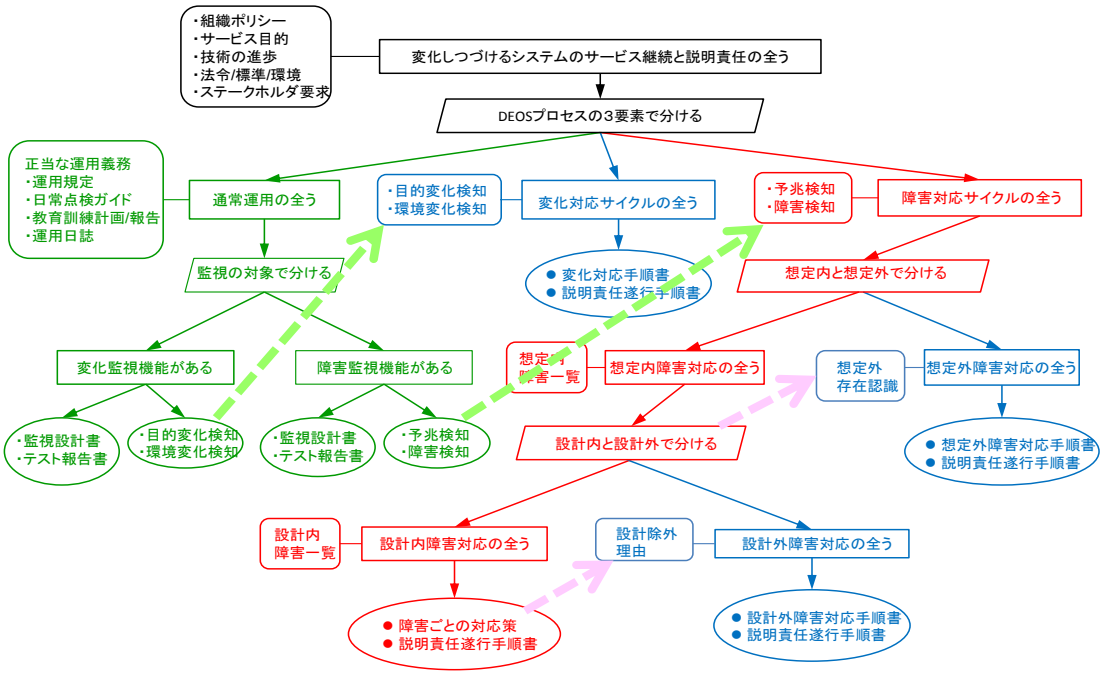

図 3.1-1 DEOS 基本構造（[16]Chapter 3.4.1）

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．これはオープンシステム・ディペンダビリティの定義

ability to accommodate changes in purpose, objectives, environment and performance and to achieve accountability continually, so as to provide expected services as and when required

に基づくものである．

DEOS 基本構造に基づき，FFO は，最上位ゴールをさらに三つのサブゴール

- 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時に正しく対応する
- 障害対応：システム障害に対して，正しく応じる

に分けた.

① 一般オントロジーの形式記述

各サブゴールに対して，ゴール達成の議論に使う語彙（オントロジー）を決める文脈を表すモジュール C-通常運用，C-変化対応，C-障害対応を設け，Context.agda にまとめた（④を参照）．これらのモジュールは単に語彙を提供するだけでなく，それらの「型情報」も同時に定めている．型情報は，語彙の使い方をさだめるものである．Agda は構成的型理論に基づく言語なので，型情報を定めることにより，いわゆるデータ型の情報にとどまらず，語彙の意味を規定する公理をも記すことができた.

② 一般議論フレームワークの形式記述

各種証憑から各サブゴールを導き出す議論の構造を，通常運用ケース，変化対応ケース，障害対応ケース の三つの Agda プログラムとして表現し，Argument.agda にまとめた.

FFO では，DEOS 基本構造を単に表現するだけではなく，システムライフサイクルの時間パラメータを導入する，変化対応サイクルや障害対応サイクル内のステージを詳細化する，などの補足を加えたうえ，証憑や手順書の存在を証拠として用いる議論のパターンを関数 証拠の吟味 と 手順の遂行 として実装して，議論の再利用を図った.

③ 一般 FFO の標準体系における位置づけの明確化

FFO が依拠するシステムライフサイクルとそれを対象とするアシュランスケースの構造に国際標準体系（IEC 62853 および ISO/IEC 15288 及び周辺の標準）の中に位置づけた．FFO（DEOS 基本構造）が定める三つのサブゴールと，IEC62853 草稿[11]が定める四つのプロセスビューへの要求事項との関係は明確であり，FFO に基づいて作られるアシュランスケースは，IEC62853 草稿[11]への適合性主張のために要請されるディペンダビリティケースとして用いることができる.

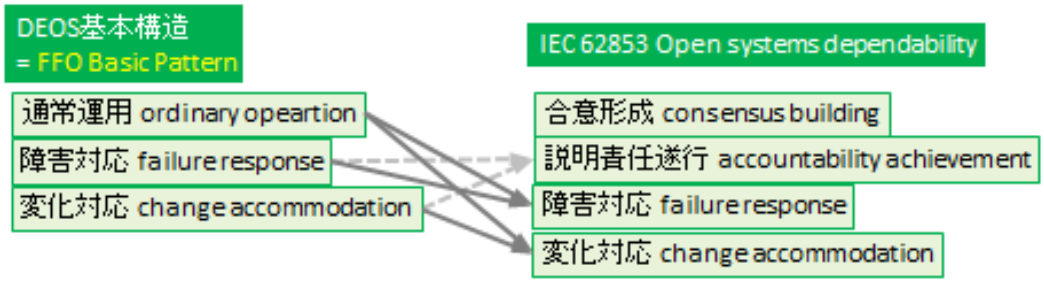

図 3.1-2 DEOS 基本構造と IEC 62853 要件との対応

④ 一般 FFO が提供する語彙定義（Contexts.agda）と議論構造（Argument.agda）

以下に一般 FFO が提供する語彙定義（Contexts.agda）と議論構造（Argument.agda）の Agda コードを示す.

```
module Contexts where
open import Notation
open import Data.Product
open import Relation.Unary  as R1
open import Relation.Binary as R2
module C-Global where
  postulate
    Time : Set
    next : Time → Time
    prev : Time → Time
  常に : ∀ {l}(P : Time → Set l) → Set _
  常に P = (t : Time) → P t
  postulate
    ⟨_⟩機能設計書-型 ⟨_⟩機能開発履歴-型 ⟨_⟩機能テスト報告書-型 ⟨_⟩機能運用証憑-型
        : Set  → Set
  record ⟨_⟩機能開発証憑-型(A : Set) : Set where
    field
      設計書        : ⟨ A ⟩機能設計書-型
      開発履歴      : ⟨ A ⟩機能開発履歴-型
      テスト報告書 : ⟨ A ⟩機能テスト報告書-型
  ⟨_⟩機能証憑監査結果-型 : Set → Set
  ⟨ A ⟩機能証憑監査結果-型 = ⟨ A ⟩機能開発証憑-型 → ⟨ A ⟩機能運用証憑-型 → A
  証憑の吟味 : {A : Set} →
    ⟨ A ⟩機能証憑監査結果-型 → ⟨ A ⟩機能開発証憑-型 → ⟨ A ⟩機能運用証憑-型 → A
  証憑の吟味 f p q = f p q
  module C-手順遂行
           {目的 : Goal}
           {手順書-型 : Set}
           {説明責任遂行手順書-型 : Set}
           {遂行される : 手順書-型 × 説明責任遂行手順書-型 → Goal} where
    手順-型 = 手順書-型 × 説明責任遂行手順書-型
    適切である : 手順-型 → Goal
    適切である x = x が 遂行される → 目的
    record 適切な手順が遂行される : Goal where
      field
        手順書    : 手順書-型
        説明責任遂行手順書 : 説明責任遂行手順書-型
      手順 = (手順書 , 説明責任遂行手順書)
      field
        適切証憑 : 手順 は 適切である
        遂行証憑 : 手順 が 遂行される
    手順の遂行 : 適切な手順が遂行される → 目的
    手順の遂行 p = 適切証憑 遂行証憑
      where open 適切な手順が遂行される p
open C-Global

module C-トップレベル where
  module atTime(t : Time) where
```

```
    postulate
      組織ポリシー-型  サービス目的-型 ステークホルダ要求-型
        環境-型 法令/標準-型 技術の進歩-型 : Set
    record システム外要因-型 : Set where
      field
        組織ポリシー        : 組織ポリシー-型
        サービス目的        : サービス目的-型
        ステークホルダ要求 : ステークホルダ要求-型
        環境                : 環境-型
        法令/標準           : 法令/標準-型
        技術の進歩          : 技術の進歩-型
    postulate
      通常運用の全う 変化対応サイクルの全う 障害対応サイクルの全う : Goal
      システム外要因 : システム外要因-型
  open atTime public

  変化しつづけるシステムのサービス継続と説明責任の全う
    = 常に (通常運用の全う
            且つ 変化対応サイクルの全う
            且つ 障害対応サイクルの全う)
  DEOS プロセスの 3 要素で分ける :
    常に 通常運用の全う →
    常に 変化対応サイクルの全う →
    常に 障害対応サイクルの全う →
    変化しつづけるシステムのサービス継続と説明責任の全う
  DEOS プロセスの 3 要素で分ける p q r t = (p t , q t , r t)

module C-システム where
  data  変化対応ステージ : Set where
    要求抽出・リスク分析 ステークホルダ合意形成 開発・テスト 説明責任遂行
                                          : 変化対応ステージ
  data  障害対応ステージ : Set where
     未然回避 迅速対応 原因究明 説明責任遂行 : 障害対応ステージ
  data  活動種別 : Set where
    通常運用  : 活動種別
    変化対応  : 変化対応ステージ → 活動種別
    障害対応  : 障害対応ステージ → 活動種別
  module atTime(t : Time) where
    postulate
      システム状態-型   : Set
      サービスレベル-型 : Set
    record 監視対象-型 : Set where
      field
        システム状態   : システム状態-型
        サービスレベル : サービスレベル-型
    postulate
      監視対象 : 監視対象-型
    postulate
```

```
      活動中 : 活動種別 → システム状態-型 → Proposition
    open 監視対象-型 監視対象
    変化対応開始 = 活動中 (変化対応 要求抽出・リスク分析) システム状態
    障害対応開始 = (活動中 (障害対応 未然回避) 又は
                    活動中 (障害対応 迅速対応)) システム状態
    障害対応中 = Σ[ x ∈ 障害対応ステージ ] 活動中 (障害対応 x) システム状態
    変化対応中 = Σ[ x ∈ 変化対応ステージ ] 活動中 (変化対応 x) システム状態
    ステージ = proj₁
  open atTime public
module C-通常運用(t : Time) where
  module C-日常義務 where
    open C-システム.atTime t
    postulate
      運用規定-型  日常点検ガイド-型 教育訓練計画-型
        運用日誌-型 日常点検報告-型 教育訓練報告-型 : Set
    record 日常義務-型 : Set where
      field
        運用規定       : 運用規定-型
        日常点検ガイド : 日常点検ガイド-型
        教育訓練計画   : 教育訓練計画-型
    postulate
      日常義務     : 日常義務-型
    record 果たされている(x : 日常義務-型) : Proposition where
      field
        運用日誌     : 運用日誌-型
        日常点検報告 : 日常点検報告-型
        教育訓練報告 : 教育訓練報告-型
  module C-変化監視 where
    open C-トップレベル.atTime
    open C-システム.atTime
    postulate
      目的・環境変化あり :
        システム外要因-型 (prev t) → システム外要因-型 t → Proposition
    目的・環境変化発生 : Proposition
    目的・環境変化発生 = 目的・環境変化あり (システム外要因(prev t))
                                            (システム外要因 t)
    目的・環境変化検知できる : Goal
    目的・環境変化検知できる = R2.Decidable 目的・環境変化あり
    目的・環境変化検知 = 目的・環境変化検知できる
    変化対応に移行できる : Goal
    変化対応に移行できる = 目的・環境変化発生 → C-システム.変化対応開始 (next
t)
    変化対応移行 = 変化対応に移行できる
    変化監視機能がある : Goal
    変化監視機能がある = 目的・環境変化検知できる かつ 変化対応に移行できる
  module C-障害監視 where
    open C-システム.atTime t
    postulate
```

```
      In_Operation_Range : Pred₀ サービスレベル-型
      障害予兆あり       : Pred₀ システム状態-型
    障害（予兆）あり : Pred₀ 監視対象-型
    障害（予兆）あり s = let open 監視対象-型 in
      (s の サービスレベル は In_Operation_Range でない)
      または (s の システム状態 は 障害予兆あり)
    障害（予兆）検知できる : Goal
    障害（予兆）検知できる = R1.Decidable 障害（予兆）あり
    障害（予兆）検知 = 障害（予兆）検知できる
    障害（予兆）発生 : Proposition
    障害（予兆）発生 = 障害（予兆）あり 監視対象
    障害対応に移行できる : Goal
    障害対応に移行できる = 障害（予兆）発生 → C-システム.障害対応開始 (next t)
    障害対応移行 = 障害対応に移行できる
    障害監視機能がある : Goal
    障害監視機能がある = 障害（予兆）検知できる かつ 障害対応に移行できる
  open C-日常義務; open C-変化監視; open C-障害監視
  open C-トップレベル.atTime t
  postulate
    通常運用詳細化 : 日常義務 が 果たされている
                   → 変化監視機能がある
                   → 障害監視機能がある
                   → 通常運用の全う
module C-変化対応(t : Time) where
  open C-システム using (変化対応ステージ)
  open C-システム.atTime t
  open C-トップレベル.atTime t
  postulate
    ステージゴール : 変化対応ステージ → Goal
  変化対応詳細 = (p : 変化対応中) → ステージゴール (ステージ p)
  postulate
    変化対応詳細化 : 変化対応詳細 → 変化対応サイクルの全う
  postulate
    変化対応手順書-型 変化対応説明責任遂行手順書-型 : Set
    遂行される : 変化対応手順書-型 × 変化対応説明責任遂行手順書-型 → Goal
  open C-手順遂行
        {目的       = ∀ stage → ステージゴール stage}
        {手順書-型  = 変化対応手順書-型}
        {説明責任遂行手順書-型 = 変化対応説明責任遂行手順書-型}
        {遂行される = 遂行される}
      public

module C-障害対応(t : Time) where
  module C-想定外 where
    postulate
      想定外障害-型 : Set
      _への想定外対応 : 想定外障害-型 → Goal
    想定外障害対応の全う : Goal
```

```
  想定外障害対応の全う = ∀ 障害 → 障害 への想定外対応
  postulate
    想定外障害対応手順書-型 想定外障害対応説明責任遂行手順書-型 : Set
    遂行される : 想定外障害対応手順書-型
                 × 想定外障害対応説明責任遂行手順書-型
                 → Goal
  open C-手順遂行
         {目的        = 想定外障害対応の全う}
         {手順書-型   = 想定外障害対応手順書-型}
         {説明責任遂行手順書-型 = 想定外障害対応説明責任遂行手順書-型}
         {遂行される = 遂行される}
       public
module C-設計外 where
  postulate
    設計外障害-型 : Set
    除外理由  : 設計外障害-型 → Proposition
    _への設計外対応 : 設計外障害-型 → Goal
  設計外障害対応の全う : Goal
  設計外障害対応の全う = ∀ 障害 → 障害 への設計外対応
  postulate
    設計外障害対応手順書-型 設計外障害対応説明責任遂行手順書-型 : Set
    遂行される : 設計外障害対応手順書-型
                 × 設計外障害対応説明責任遂行手順書-型
                 → Goal
  open C-手順遂行
         {目的        = 設計外障害対応の全う}
         {手順書-型   = 設計外障害対応手順書-型}
         {説明責任遂行手順書-型 = 設計外障害対応説明責任遂行手順書-型}
         {遂行される = 遂行される}
       public
module C-設計内 where
  postulate
    設計内障害-型 : Set
    _への設計内対応 : 設計内障害-型 → Goal
  設計内障害対応の全う : Goal
  設計内障害対応の全う = ∀ 障害 → 障害 への設計内対応
  postulate
    設計内障害ごとの対応策-型 設計内障害対応説明責任遂行手順書-型 : Set
    遂行される : 設計内障害ごとの対応策-型
                 × 設計内障害対応説明責任遂行手順書-型
                 → Goal
  open C-手順遂行
         {目的        = 設計内障害対応の全う}
         {手順書-型   = 設計内障害ごとの対応策-型}
         {説明責任遂行手順書-型 = 設計内障害対応説明責任遂行手順書-型}
         {遂行される = 遂行される}
       public
module C-想定内 where
```

```
    open C-設計外
    open C-設計内
    data 想定内障害-型 : Set where
      設計内 : 設計内障害-型 → 想定内障害-型
      設計外 : 設計外障害-型 → 想定内障害-型
    _への想定内対応 : 想定内障害-型 → Goal
    設計内 障害 への想定内対応 = 障害 への設計内対応
    設計外 障害 への想定内対応 = 障害 への設計外対応
    想定内障害対応の全う = ∀ 障害 → 障害 への想定内対応
    設計内と設計外で分ける : 設計内障害対応の全う →
                             設計外障害対応の全う →
                             想定内障害対応の全う
    設計内と設計外で分ける p q (設計内 x) = p x
    設計内と設計外で分ける p q (設計外 x) = q x
  open C-想定外
  open C-想定内
  data 障害-型 : Set where
    想定内 : 想定内障害-型 → 障害-型
    想定外 : 想定外障害-型 → 障害-型
  _への対応 : 障害-型 → Goal
  想定内 障害 への対応 = 障害 への想定内対応
  想定外 障害 への対応 = 障害 への想定外対応
  障害対応の全う = ∀ 障害 → 障害 への対応
  想定内と想定外で分ける : 想定内障害対応の全う →
                           想定外障害対応の全う →
                           障害対応の全う
  想定内と想定外で分ける p q (想定内 x) = p x
  想定内と想定外で分ける p q (想定外 x) = q x

  open C-システム.atTime t
  open C-トップレベル.atTime t
  postulate
    対応中障害 : 障害対応中 → 障害-型
  障害対応詳細 = (p : 障害対応中) → (p の 対応中障害) への対応
  postulate
    障害対応詳細化 : 障害対応詳細 → 障害対応サイクルの全う

module Argument where
open import Notation
open import Contexts
open import Evidence
open C-Global

main-case =
  let open C-トップレベル in
  変化しつづけるシステムのサービス継続と説明責任の全う
  by DEOS プロセスの 3 要素で分ける
     § (常に 通常運用の全う          by 通常運用ケース)
```

```
    § (常に 変化対応サイクルの全う by 変化対応ケース)
    § (常に 障害対応サイクルの全う by 障害対応ケース)
where
open C-トップレベル
通常運用ケース : ∀(t : Time) → 通常運用の全う t
通常運用ケース t =
  let open C-通常運用 t ; open E-通常運用 t in
  通常運用詳細化
  § (let open C-日常義務 ; open E-日常義務 in
    日常義務 が 果たされている
    by record { 運用日誌     = 運用日誌 Ref
              ; 日常点検報告 = 日常点検報告 Ref
              ; 教育訓練報告 = 教育訓練報告 Ref })
  § (let open C-変化監視 in
    変化監視機能がある
    by ∧分解
       § (目的・環境変化検知できる by 変化検知ケース)
       § (変化対応に移行できる     by 変化対応への移行ケース))
  § (let open C-障害監視 in
    障害監視機能がある
    by ∧分解
       § (障害（予兆）検知できる   by 障害検知ケース)
       § (障害対応に移行できる     by 障害対応への移行ケース))
  where
  変化検知ケース =
    let open C-通常運用.C-変化監視 t ; open E-通常運用.E-変化監視 t in
    目的・環境変化検知できる
    by 証憑の吟味
       § (⟨ 目的・環境変化検知 ⟩機能証憑監査結果-型
         by 目的・環境変化検知-証憑監査結果 Ref)
       § (⟨ 目的・環境変化検知 ⟩機能開発証憑-型
         by record { 設計書       = 目的・環境変化検知-設計書 Ref
                   ; 開発履歴     = 目的・環境変化検知-開発履歴 Ref
                   ; テスト報告書 = 目的・環境変化検知-テスト報告書 Ref
                   })
       § (⟨ 目的・環境変化検知 ⟩機能運用証憑-型
         by 目的・環境変化検知-運用証憑 Ref)
  変化対応への移行ケース =
    let open C-通常運用.C-変化監視 t ; open E-通常運用.E-変化監視 t in
    変化対応に移行できる
    by 証憑の吟味
       § (⟨ 変化対応移行 ⟩機能証憑監査結果-型
         by 変化対応移行-証憑監査結果 Ref)
       § (⟨ 変化対応移行 ⟩機能開発証憑-型
         by record { 設計書       = 変化対応移行-設計書 Ref
                   ; 開発履歴     = 変化対応移行-開発履歴 Ref
                   ; テスト報告書 = 変化対応移行-テスト報告書 Ref
                   })
```

```
      § (⟨ 変化対応移行 ⟩機能運用証憑-型
        by 変化対応移行-運用証憑 Ref)
  障害検知ケース =
    let open C-通常運用.C-障害監視 t ; open E-通常運用.E-障害監視 t in
    障害（予兆）検知できる
    by 証憑の吟味
      § (⟨ 障害（予兆）検知 ⟩機能証憑監査結果-型
        by 障害（予兆）検知-証憑監査結果 Ref)
      § (⟨ 障害（予兆）検知 ⟩機能開発証憑-型
        by record { 設計書       = 障害（予兆）検知-設計書 Ref
                  ; 開発履歴     = 障害（予兆）検知-開発履歴 Ref
                  ; テスト報告書 = 障害（予兆）検知-テスト報告書 Ref
                  })
      § (⟨ 障害（予兆）検知 ⟩機能運用証憑-型
        by 障害（予兆）検知-運用証憑 Ref)
  障害対応への移行ケース =
    let open C-通常運用.C-障害監視 t ; open E-通常運用.E-障害監視 t in
    障害対応に移行できる
    by 証憑の吟味
      § (⟨ 障害対応移行 ⟩機能証憑監査結果-型
        by 障害対応移行-証憑監査結果 Ref)
      § (⟨ 障害対応移行 ⟩機能開発証憑-型
        by record { 設計書       = 障害対応移行-設計書 Ref
                  ; 開発履歴     = 障害対応移行-開発履歴 Ref
                  ; テスト報告書 = 障害対応移行-テスト報告書 Ref
                  })
      § (⟨ 障害対応移行 ⟩機能運用証憑-型
        by 障害対応移行-運用証憑 Ref)

変化対応ケース : ∀(t : Time) → 変化対応サイクルの全う t
変化対応ケース t =
  let open C-変化対応 t; open C-システム.atTime t; open E-変化対応 t in
  変化対応詳細化
  § (((p : 変化対応中) → ステージゴール (ステージ p))
    by λ p →
      ステージゴール (ステージ p)
      by 一般化
        § ((∀ stg → ステージゴール stg)
          by 手順の遂行
            § (適切な手順が遂行される
              by record { 手順書 = 変化対応手順書 Ref
                        ; 説明責任遂行手順書 = 説明責任遂行手順書 Ref
                        ; 適切証憑 = 適切証憑 Ref
                        ; 遂行証憑 = 遂行証憑 Ref }))))

障害対応ケース : ∀(t : Time) → 障害対応サイクルの全う t
障害対応ケース t =
  let open C-障害対応 t; open C-システム.atTime t; open E-障害対応 t in
```

```
障害対応詳細化
§ (((p : 障害対応中) → (p の 対応中障害) への対応)
   by λ p →
     (p の 対応中障害) への対応
     by 一般化{x = (p の 対応中障害)}
        § ((∀ x → x への対応)
           by 想定内と想定外で分ける
              § (let open C-想定内 in
                 想定内障害対応の全う
                 by 設計内と設計外で分ける
                    § (C-設計内.設計内障害対応の全う by 設計内ケース)
                    § (C-設計外.設計外障害対応の全う by 設計外ケース))
              § (C-想定外.想定外障害対応の全う by 想定外ケース)))
where
設計内ケース =
  let open C-障害対応.C-設計内 t; open E-障害対応.E-設計内 t in
  設計内障害対応の全う
  by 手順の遂行
     § (適切な手順が遂行される
        by record { 手順書    = 設計内障害ごとの対応策 Ref
                  ; 説明責任遂行手順書 = 説明責任遂行手順書 Ref
                  ; 適切証憑 = 適切証憑 Ref
                  ; 遂行証憑 = 遂行証憑 Ref })
設計外ケース =
  let open C-障害対応.C-設計外 t; open E-障害対応.E-設計外 t in
  設計外障害対応の全う
  by 手順の遂行
     § (適切な手順が遂行される
        by record { 手順書    = 設計外障害対応手順書 Ref
                  ; 説明責任遂行手順書 = 説明責任遂行手順書 Ref
                  ; 適切証憑 = 適切証憑 Ref
                  ; 遂行証憑 = 遂行証憑 Ref })
想定外ケース =
  let open C-障害対応.C-想定外 t; open E-障害対応.E-想定外 t in
  想定外障害対応の全う
  by 手順の遂行
     § (適切な手順が遂行される
        by record { 手順書    = 想定外障害対応手順書 Ref
                  ; 説明責任遂行手順書 = 説明責任遂行手順書 Ref
                  ; 適切証憑 = 適切証憑 Ref
                  ; 遂行証憑 = 遂行証憑 Ref })
```

3.1.3 発生した課題および今後の展望

(1) 発生した課題

① パラメータ受け渡し方式の選択

プログラミング言語 Agda の式のパラメータの受け渡しをどのように設計するべきかが課題となった．Agda の式にパラメータを設定する方法は，少なくとも以下の三つがある．

i. λ-抽象化を施して，関数とする．

ii. その式を囲うモジュールにλ-抽象化を施して，モジュール全体をパラメータ月モジュールとする．

iii. パラメータとしたい変数を postulate 宣言によって定数として導入する．

いずれの方式にも一長一短がある．その特質を以下に論じる．

方式 i.は，論理的厳密さが完全な形で得られるのが長所である．しかし，パラメータの数が多くなると関数適用の際にいちいち多数の実引数を並べなければならない．また，パラメータを具体化するときに，実引数を構成して与えなければならず，アシュランスケースのような，高レベルの記述においては，その構成が煩雑になりがちである．

方式 ii.は，方式 i.と全く同じ論理的厳密さが得られる上，関数適用の際にいちいち多数の実引数を並べる必要がなくなる．モジュールに実引数を与えて具体化すれば，その実引数にモジュール内の関数に一斉に適用されるからである．しかし，実引数の構成をしなければならないのは方式 1.と同じである．

方式 iii.では方式 i.，ii.にみられる煩雑さをさけることができるが，postulate 宣言は矛盾を導入しうるので，論理的厳密さに欠ける．また，postulate 宣言で導入した定数同士の関係を制約することができないのも欠点であろう．

FFO では，上記の得失を勘案しながら，場合に応じて三つの方式を使い分けた．

② オープンシステム・ディペンダビリティの定義のゆれ

FFO は，システムのオープンシステム・ディペンダビリティ達成を主張するアシュランスケース記述の枠組を形式的に記述したものである．その語彙定義（オントロジー）と議論構造は，2014 年に刊行された先行研究の成果[17]に記された DEOS 基本構造に基づいている．一方，オープンシステム・ディペンダビリティに関する国際標準 IEC62853（制定中）の草稿は，DEOS 基本構造から派生してはいるものの，2014 以後の IEC TC56 委員会における討論の結果，語彙定義，議論構造とも，DEOS 基本構造とは異なる発展をとげている．その結果，FFO に基づいて書かれたアシュランスケースが IEC62853 に適合していること議論するためには，DEOS 基本構造から IEC62853 への解釈が必要である．

③ ペトリネットモデルの出現と反映

本研究項目終了後，[研究目標 4 FFO が依拠するシステムライフサイクル概念の確立］のためにライフサイクルモデルのペトリネットによる数理モデルを構築した．その結果，ペトリネットモデルを反映させた FFO を構築するテーマが発生した．しかし，このテーマに取り組むリソース（マンパワーと時間）が本研究の範囲では得られなかった．

(2) 今後の展望

課題［(1)①パラメータ受け渡し方式の選択］に対しては，本研究では文脈ごとに個別に解決を図ったが，一般的な解決法の模索は，オントロジー定義の技法の重要な研究テーマになると考えられる．

課題［(1)②オープンシステム・ディペンダビリティの定義のゆれ］は，必要なマンパワーを投入して FF0 を最新の国際標準にあわせるようにすれば解決される．そのためには，特に技術的困難はないが，FF0 の研究グループと標準制定グループが密接に連携する体制を保つことが重要で，そのような人的環境を保つという，社会的な課題を解決していかなければならない．

課題［(1)③ペトリネットモデルの出現と反映］の解決のためには，必要なマンパワーの確保に加えて，FF0 に「時相」つまり時間のパラメータを導入する必要がある．時相論理を用いて導入するのがよいと考えられるが，それによって，FF0 の記述がどのように，またどの程度複雑になるのかはわからない．

3.2 研究目標 2「特定の技術領域における FFO の開発」

3.2.1 当初の想定

(1) 研究内容

車載ソフトウェア, スマートグリッド, インターネット等の技術領域から一つを選び, [研究目標 1 オープンシステム・ディペンダビリティ一般のための FFO の開発] の達成において実装した一般的フレームワークを, その技術領域に具体化して, その領域独特の事情を反映したフレームワークを実装する. また, [研究目標 3 事例研究による有効性評価] の結果を受けて, この実装を改善する.

(2) 想定課題と対応策

本研究目標達成のための作業の内容は, 研究目標 3 における有効性評価作業と密接に関連する (プログラミングとデバグの関係に似ている) ので, 3.3(2)に記載した課題と関連させながら進めていかなければならない.

3.2.2 研究プロセスと成果

(1) 研究プロセス

① 特定技術領域向けのオントロジーを形式記述する

特定の技術領域として, 車載システムと防災システムをとりあげ, それぞれの技術領域において, システムのオープンシステム・ディペンダビリティ達成を主張するアシュランスケースのための語彙定義 (オントロジー) を, Agda 言語によって記述する.

② 特定技術領域向けの議論フレームを形式記述する

特定の技術領域として, 車載システムと防災システムをとりあげ, それぞれの技術領域において, システムのオープンシステム・ディペンダビリティ達成を主張するアシュランスケースのための議論構造 (議論フレームワーク) を, Agda 言語によって記述する.

③ 研究目標 3「事例研究による有効性評価」の結果を用いて①②の成果を改善する

作業項目①②の結果は, Agda によるプログラムである. まずその第 1 版を完成し, その後研究目標 3 の結果を反映させてプログラムを改善する.

④ 特定技術領域向け FFO の標準体系における位置づけを明確にする

オープンシステム・ディペンダビリティ一般に関する国際標準体系および業界標準体系における FFO の位置づけを明確にする.

(2) 具体的な研究成果の内容

特定技術領域として，車載システムと防災システムを対象にして研究を進めた．各対象について，以降に記す。

① 車載システム

車載システムの機能安全を達成するための要求事項を規定する ISO26262[7]は，機能安全を

「電気・電子システムの正常でないふるまいにより引き起こされるハザードが原因となる，不合理なリスクの不在」

と定義している．要求事項の中には，アシュランスケース(安全ケースと呼ばれている)も含まれている．

ISO26262 は開発過程に主眼を置くため，システムライフサイクルが ISO26262 に適合しても，運用，保守過程や合意形成過程が不十分であるために機能安全を欠く場合があり得る．また，想定外の原因による障害発生についても ISO26262[7]は十分な配慮をしていない．そのため，障害対応や変化対応に重点を置くオープンシステム・ディペンダビリティの要件は，ISO26262[7]と相補的に機能すると考えられ，両者の要件を合わせて要求することにより，機能安全の，より現代的な実現が可能になる．そこで我々はオープンシステム・ディペンダビリティのための FFO を車載システムの機能安全に具体化した AFSCF (Automobile Functional Safety Case Framework) 議論モデルを考案した．

以下に車載システムを対象にした研究成果について記す．

1) 車載システム向けのオントロジーを形式記述

ISO26262 に従った開発を行うための機能安全テンプレート[8]が，(社) JASPAR によって提供され，システムレベル開発およびソフトウェア/ハードウェア開発のための文書が規定されている．このうち，システムレベル開発を対象とする「機能安全技術テンプレート システム開発編」，およびソフトウェア開発を対象とする「機能安全技術テンプレート ソフトウェア開発編」の語彙定義を分析し，プログラミング言語 Agda で記述した．

2) 車載システム向けの議論フレームワークを形式記述

(ア) AFSCF 議論モデル

IEC 62853 のアシュランスケースモデル ([11], Annex B) と ISO26262 準拠機能安全ケースのレイヤーモデル[10]を融合させて AFSCF 議論モデル (Automotive Functional Safety Case Framework) を考案した．

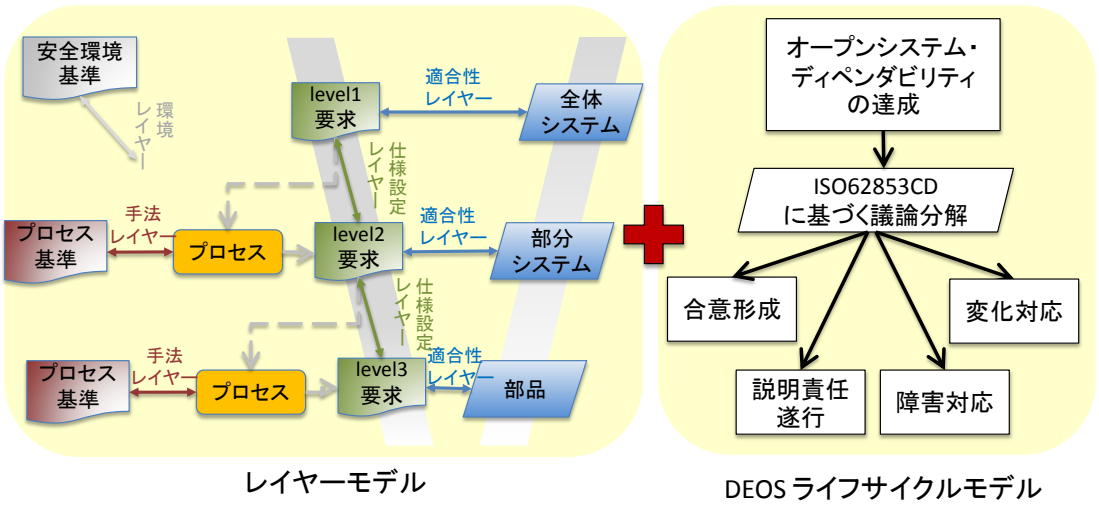

図 3.2-1 AFSCF 議論モデル＝レイヤーモデル＋DEOS ライフサイクルモデル

まず，機能安全の達成に関する議論を次のように定めた．開発プロセスに対して「機能安全の達成」というゴールの下に

- 仕様設定の根拠
- 仕様への適合性
- 手法の適切さ

の 3 つに対応する サブゴールを設定し，開発プロセス全体の管理運営に関して

- 環境

に対するサブゴールを設定した．

各サブゴールについて何を主張すべきかについて，機能安全目標から技術安全要求仕様を設定する工程(機能安全/技術安全 phase と呼ぶ）の場合を基に説明する．

- 「仕様設定の根拠」に対しては，技術安全要求仕様が満たされれば，機能安全要求仕様も満たされること，すなわち技術安全要求仕様設定の妥当性を示す．
- 「仕様への適合性」に対しては，システムが要求される仕様を満たしている事を，テスト結果，およびテスト手法の適切さにより示す．
- 「手法の適切さ」に対しては，開発に用いた，手法，プロセス，ツール，レビューが適切であることを示す．
- 「環境」に対しては，開発が品質管理基準，安全文化に照らして適切であることを示す．

一方，OSD の 4 つの要素

- 関係者間の合意形成
- 説明責任
- 障害対応の適切さ
- 変化対応の適切さ

の議論は，FFO の枠組みの中で実現できる．

以上の考察から，トップレベルのゴールを「OSD-機能安全の達成」とした「AFSCS 議論モデル」を提案した．

AFSCS 議論モデルについて説明する.

- トップレベルゴールは「OSD-機能安全の達成」(システムの正常でない振舞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.)
- 「OSD-機能安全の達成」は次の 6 つのサブゴールから達成される.
  1. 「機能安全の達成」(システムの正常でない振舞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」(関係者間の合意形成が適切に成されている)
  4. 「説明責任達成」(説明責任が達成するための準備が整っていいる)
  5. 「変化対応」(変化対応が適切になされている)
  6. 「障害対応」(障害対応が適切になされている)

ここまでを GSN で図示したものが図 3.2-2 である.

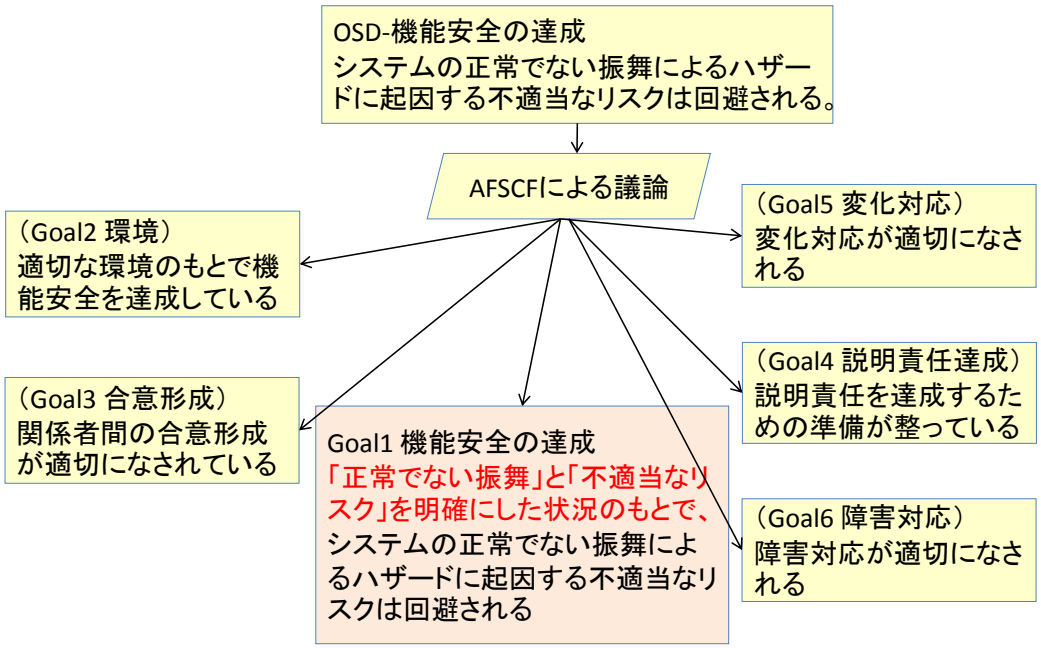

図 3.2-2 OSD-機能安全の達成の 6 つのサブゴール

以下では,「機能安全の達成」ゴール以下の構造について説明する.

(イ) 機能安全の達成

IS026262 に基づく開発を次の 4 つの phase に分けて, phase ごとに議論する. 我々が定めた 4 つの phase は

- ハザード分析/安全目標 phase
- 安全目標/機能安全 phase
- 機能安全/技術安全 phase

- 技術安全/HW/SW 安全 phase

である.

次に各 phase に対しては「仕様設定の根拠」,「仕様への整合性」,「手法の適切さ」に対応するサブゴールを定めた.

- p1. ハザード分析/安全目標 phase に対して
  - a. 仕様設定の根拠: 「ハザード分析/安全目標 phase は妥当」
  - b. 仕様への整合性: 「システムは安全目標を満たす」
  - c. 手法の適切さ: 「安全目標設定・試験の手法は適切」
- p2. 安全目標/機能安全 phase に対して
  - a. 仕様設定の根拠: 「安全目標/機能安全 phase は妥当」
  - b. 仕様への整合性: 「システムは機能安全要求を満たす」
  - c. 手法の適切さ: 「機能安全要求設定・試験の手法は適切」
- p3. 機能安全/技術安全 phase に対して
  - a. 仕様設定の根拠: 「機能安全/技術安全 phase は妥当」
  - b. 仕様への整合性: 「システムは技術安全要求を満たす」
  - c. 手法の適切さ: 「技術安全要求設定・試験の手法は適切」
- p4. 技術安全/HW/SW 安全 phase に対して
  - a. 仕様設定の根拠: 「技術安全/HW/SW 安全 phase は妥当」
  - b. 仕様への整合性: 「システムは HW/SW 安全要求を満たす」
  - c. 手法の適切さ: 「HW/SW 安全要求設定・試験の手法は適切」

の 12 個である.

「機能安全の達成」ゴールは, 以上の 12 個のサブゴールから達成される.

「機能安全の達成」ゴールに対する議論を GSN で図示する.

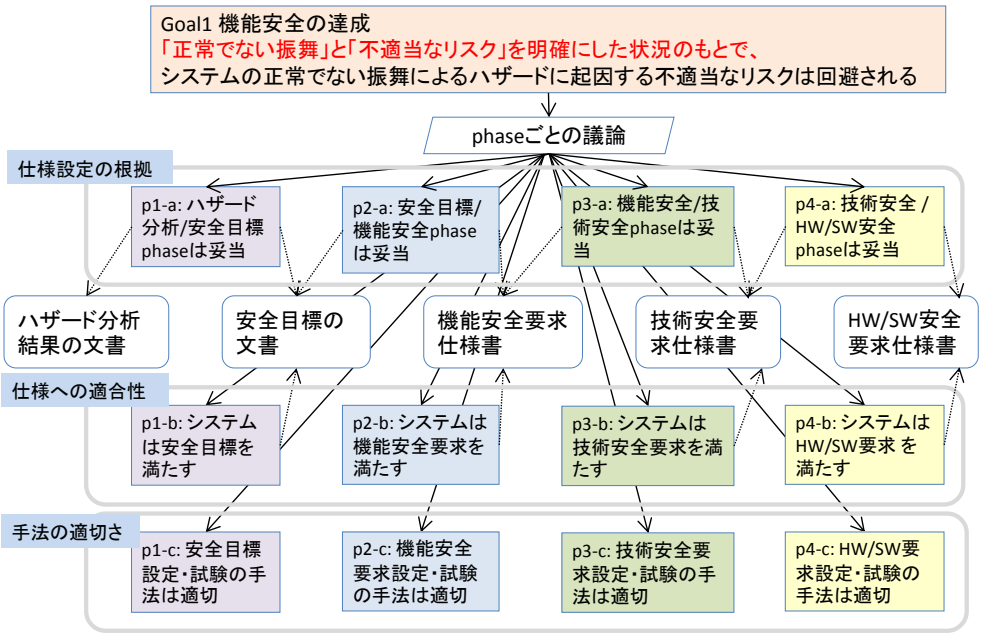

図 3.2-3 機能安全の達成の phase ごとの議論

以上の考察をふまえ、図 3.1-2 および図 3.2-3 で与えた AFSCF 議論モデルの Agda による形式記述を行なった.

3) 研究目標 3 の結果を用いた成果の改善

3.3.2(2)①において「仕様設定の根拠」のアシュランスケースを記述するにあたっては「導出パターン」を用いることが有効であることを示した. その結果をふまえ,「仕様設定の根拠」のアシュランスケース記述については,「導出パターン」の使用を前提とした議論の構造を提案した.

導出パターンを考慮に入れ, 各 phase の「仕様設定の根拠」に対する議論の構造を以下のように定めた.

- 「ハザード分析/安全目標 phase は妥当」(図 3.2-4 の青色強調部)に対して次のサブゴールを定める.

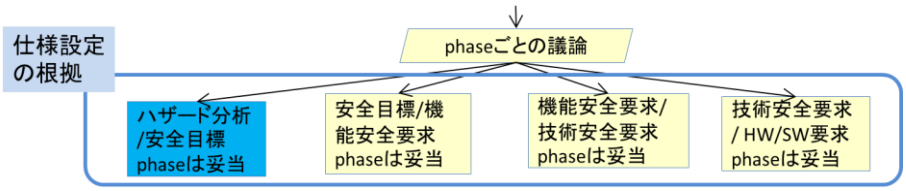

図 3.2-4 ハザード分析/安全目標 phase は妥当

- システムが安全目標を満たせば, 正常でない振舞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.

以上を GSN で図示したものが図 3.2-5 である.

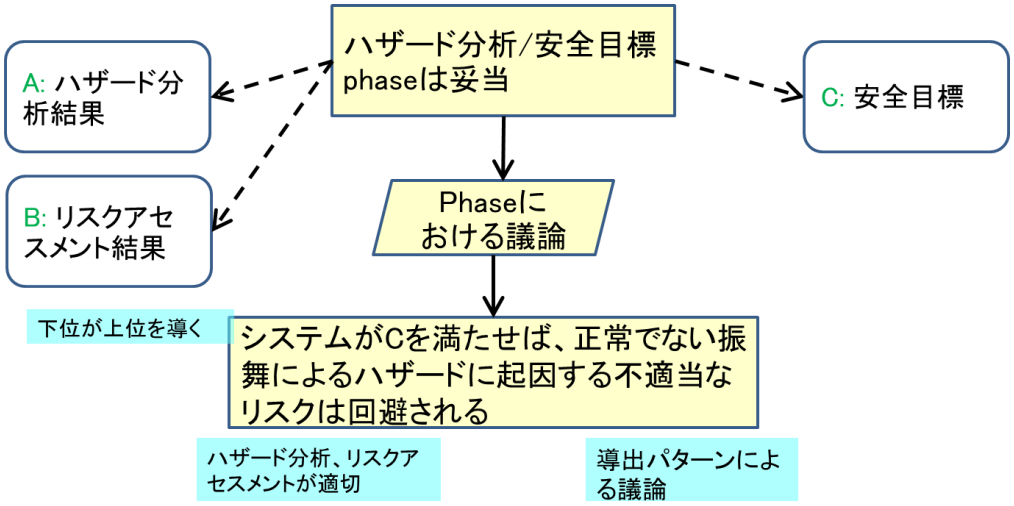

図 3.2-5 ハザード分析/安全目標 phase における「仕様設定の根拠」議論

- 「安全目標/機能安全 phase は妥当」(図 3.2-6 の青色強調部)に対して次のサブゴールを定める.

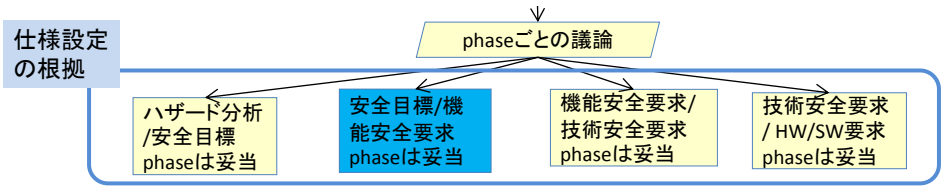

図 3.2-6 安全目標/機能安全 phase は妥当

- 機能安全要求仕様の達成は安全目標の達成を導く
- 各安全目標に対する機能安全レベルの FTA によれば機能安全要求仕様書の設定は妥当
- 各安全目標に対する機能安全レベルの FTA の実行は妥当
- 安全目標から機能安全要求仕様を導いた導出パターンは妥当

以上を GSN で図示したものが図 3.2-7 である.

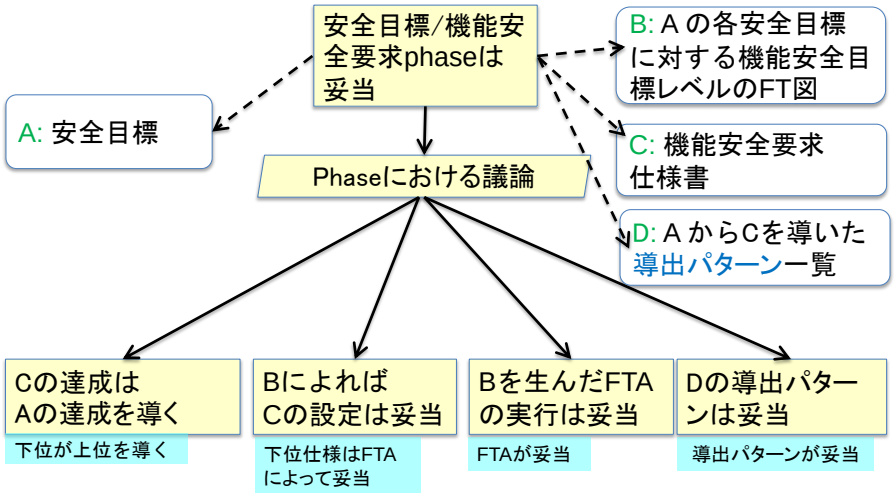

図 3.2-7 安全目標/機能安全 phase における「仕様設定の根拠」議論

- 「機能安全/技術安全 phase は妥当」(図 3.2-8 の青色強調部)に対して次のサブゴールを定める.

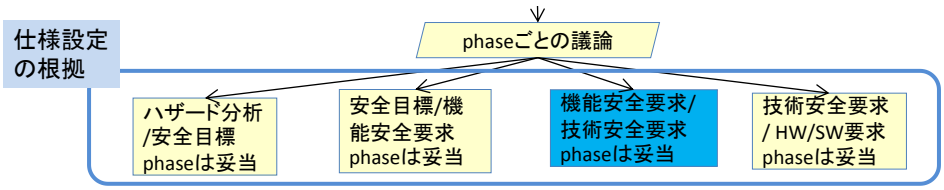

図 3.2-8 機能安全/技術安全 phase は妥当

- 技術安全要求仕様の達成は機能安全仕様の達成を導く
- 各安全目標に対する技術安全レベルの FTA によれば技術安全要求仕様書の設定は妥当
- 各安全目標に対する技術安全レベルの FTA の実行は妥当
- 機能安全要求仕様から技術安全要求仕様を導いた導出パターンは妥当

以上を GSN で図示したものが図 3.2-9 である.

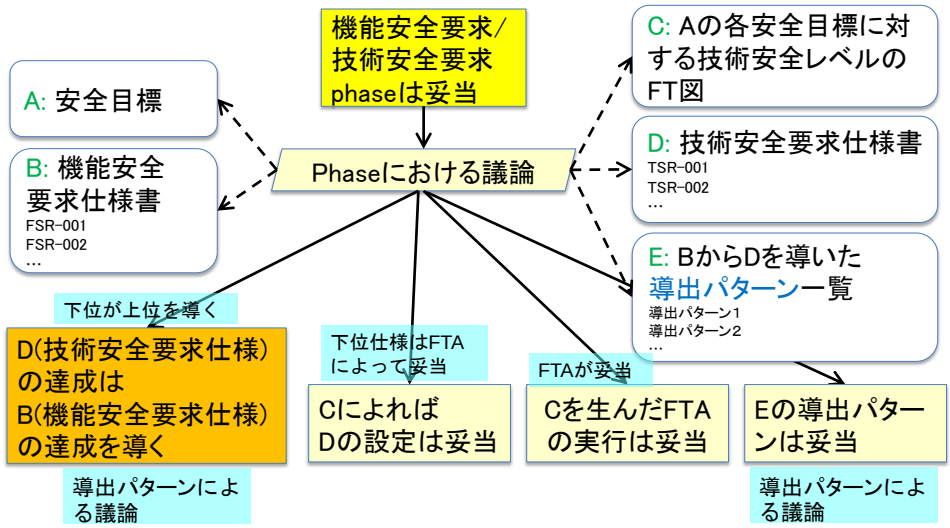

図 3.2-9 機能安全/技術安全 phase における「仕様設定の根拠」議論

- 「技術安全/HW/SW 安全 phase は妥当」(図 3.2-10 の青色強調部)に対して次のサブゴールを定める.

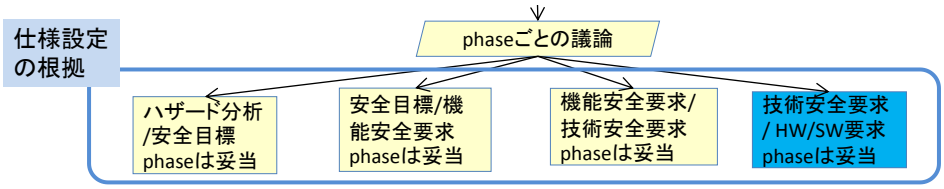

図 3.2-10 技術安全/HW/SW 安全 phase は妥当

- HW/SW 安全要求仕様の達成は技術安全仕様の達成を導く
- 各安全目標に対する HW/SW 安全レベルの FTA によれば HW/SW 安全要求仕様書の設定は妥当
- 各安全目標に対する HW/SW 安全レベルの FTA の実行は妥当
- 技術安全要求仕様から HW/SW 安全要求仕様を導いた導出パターンは妥当

以上を GSN で図示したものが図 3.2-11 である.

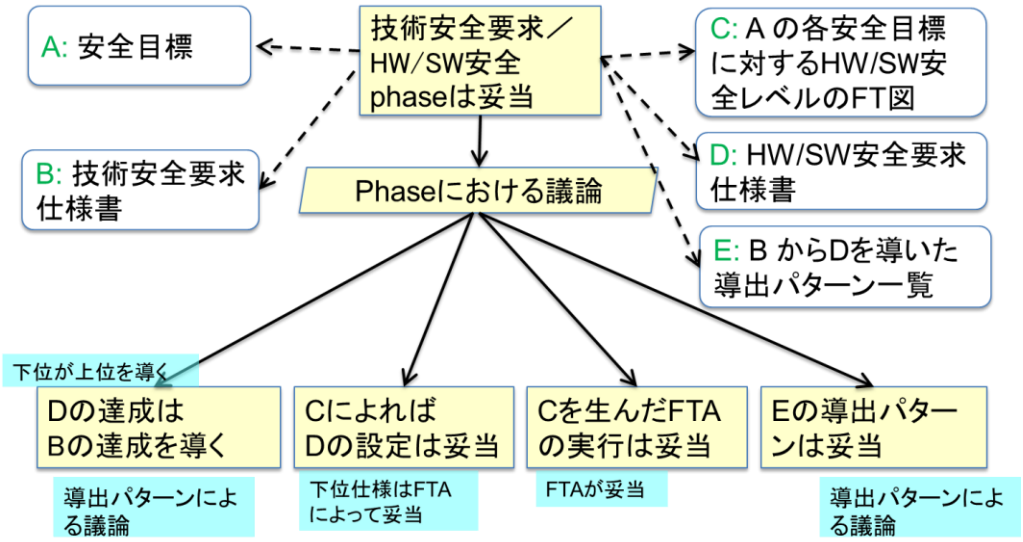

図 3.2-11 技術安全/HW-SW 安全 phase における「仕様設定の根拠」議論

4) AFSCF 議論モデルの標準体系における位置づけ

AFSCF 議論モデルは，ディペンダビリティ議論に関して IEC 62853 に準拠している．機能安全議論に関しては，ISO26262 に準拠している．以上が国際標準に関する位置づけである．

ISO 26262 はアシュランス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．それに伴い，JASPAR 機能安全テンプレートもアシュランスケースに関する詳細を規定していない．AFSCF 議論モデルに基づいて，JASPAR 機能安全テンプレートへの補遺を作成することができる．

② 防災システム

防災システムの中心をなすのは，市町村や都道府県といった地方自治体である．地方自治体の指揮のもと，警察・消防・公共インフラ企業などが協力して防災業務にあたる．そのため，特に地方自治体の防災業務を研究の対象とした．

地方自治体が行う防災関係業務は，「地域防災計画」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[19]に基づき各都道府県および各市町村が作成している文書である．災害対策基本法第 2 条において，災害とは「暴風，竜巻，豪雨，豪雪，洪水，崖崩れ，土石流，高潮，地震，津波，噴火，地滑りその他の異常な自然現象又は大規模な火事若しくは爆発その他その及ぼす被害の程度においてこれらに類する政令で定める原因により生ずる被害」と定義されている．各地方自治体では，これらの災害を想定し，平常時の対策，発災した際の応急対策やその後の復旧・復興対策を記述している．

本研究では，地域防災計画をもとに形式アシュランスケースの記述を試みた．以下に車載システムを対象にした研究成果について記す．

1) 防災オントロジーの形式記述

一般にアシュランスケースを記述に際しては，対象となるシステムやシステムライフサイクルを定義する必要がある（これがオントロジーの一部となる）．その定義には国際標準 ISO/IEC/IEEE15288（システムライフサイクルプロセス）[12]等を用いるのが一般的である．そこで，防災システムにおいても，地域防災計画が定める防災業務というオープンシステムのシステムライフサイクルを ISO/IEC/IEEE15288 に基づいて定義しようと試みた．しかし，研究を進めるにつれ，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．

ここで，ISO/IEC/IEEE15288 について簡単に説明する．この標準では，システムライフサイクルにおける作業の集まりを「プロセス」と呼ぶ．30 のプロセスが定義されており（**図** 3.2-12），これらのプロセスを組み合わせて 1 つのシステムライフサイクルが構築される．プロセス間の時系列は ISO/IEC/IEEE 15288 ではなくシステムライフサイクル model によって定められる．

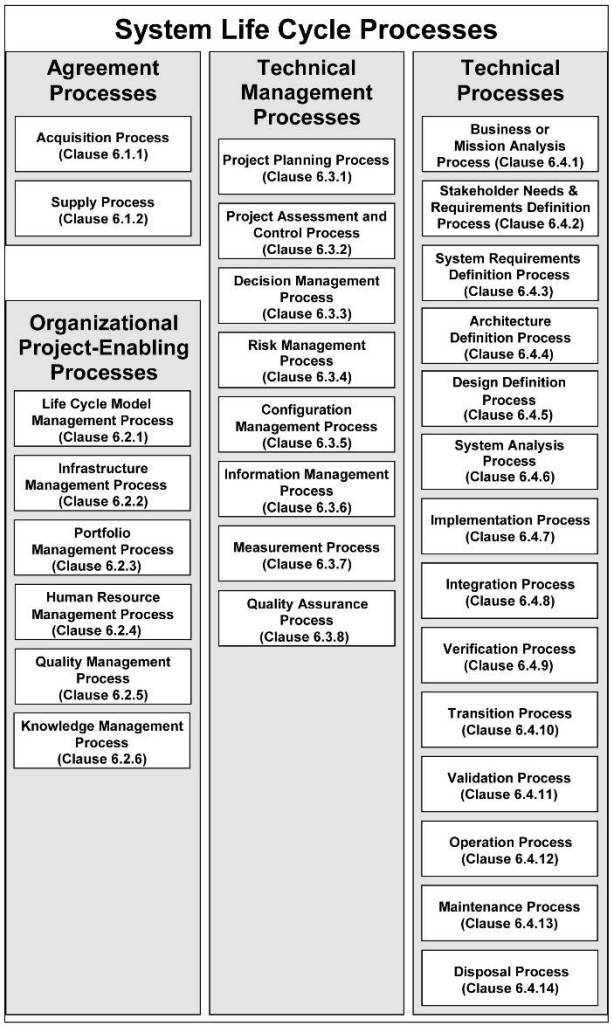

図 3.2-12　ISO/IEC/IEEE15288 が定める 30 のプロセス（[12]より引用）

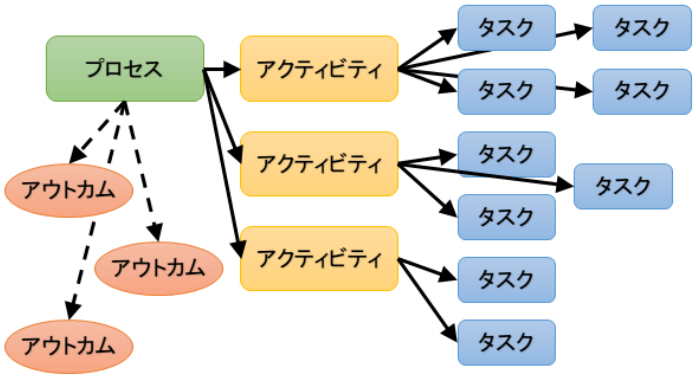

図 3.2-13　ISO/IEC/IEEE15288 におけるプロセスの構成

また，図 3.2-13 に示すように，1 つのプロセスは複数の「アクティビティ」によって構成され，1 つのアクティビティは複数の「タスク」により構成される．その構造とは別に，1 つのプロセスには複数の「アウトカム」が定められている．アウトカムとは，あるプロセスを正しく実装（具体化）した際に実現する状態や成果のことである．

このように，システムライフサイクルのプロセスについて様々な事項を定めた ISO/IEC/IEEE15288 であるが，プロセスの各アクティビティ，各タスクをどのように記述すればよいかは，必ずしも明らかではない．例えばタスクは，標準の用語定義の項では以下のように定められている．

> ” required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process” （プロセスの 1 つ以上のアウトカムの達成に貢献することを意図して，要求，推薦もしくは許可された動作）

これは，タスクがどのようなものかを定めているが，実際にタスクをどのように記述していけばよいかは明らかでない．アクティビティ，プロセスについても同様である．そこで，プロセス・アクティビティ・タスクにどのような事項を記述すればよいかの枠組みを作る必要が生じた．その枠組みが 6W1H モデルである．

6W1H モデルとは，大まかに言えば，ISO/IEC/IEEE15288 に基づいてプロセス，アクティビティ，タスクを記述する際に 6 つのラベルを付ける手法のことである．6 つのラベルとは Who，What，Whom，Where，When，Why である．これは，ジャーナリズムの世界でよく知られる 5W1H（Who，What，Where，When，Why，How）の 5W に Whom を加えたものである．

6W1H モデルの 1H とは，How のことである．一般に，「ある 1 つの作業の記述」は，さらに細かい複数の作業の記述によって表現できる．ISO/IEC/IEEE15288 においても，1 つプロセスが複数のアクティビティで表現され，1 つのアクティビティが複数のタスクで表現されている．この，「1 つの作業と複数の作業との対応」を How と呼ぶ．

図　3.2-14 は，6W1H モデルを利用して ISO/IEC/IEEE15288 のプロセスを記述した例である．Who，What，Whom，Where，When，Why の 6 つのラベルを各プロセス，アクティビティ，タスクに付けた．これらのラベル付けについては，それがどのような作業であるのか容易に理解できるラベルにする必要がある．これらの定め方の指針を表 3.2-1 にまとめた．

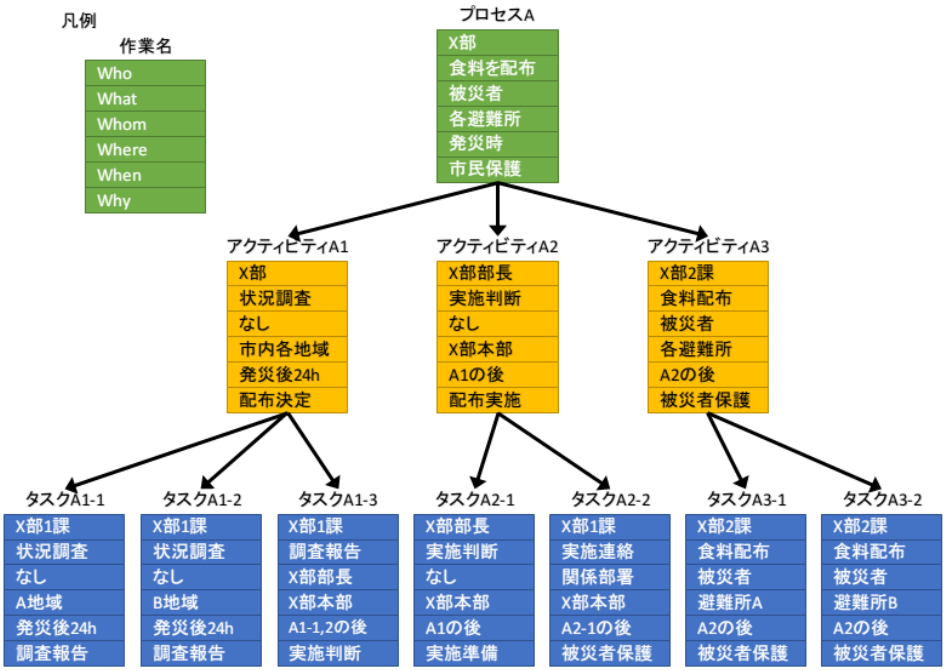

図 3.2-14　6W1H モデルを利用したプロセス記述

表 3.2-1 6W1H モデルにおけるラベル付けの指針

ラベル名	記述の指針	例
Who	作業を行う個人名，役割名，組織名	X 部，X 部 Y 課，X 部 Y 課 Z 担当，X 部部長
What	作業．特に「何」を「どうする」のかに注意	水道管の被害状況を調査する，給水車を運転する
Whom	作業を行う対象がある場合に，対象となる個人名，役割名，組織名．ない場合「なし」	被災者，X 部 1 課，なし
Where	作業を行う場所	避難所 A，各避難所，X 部本部
When	作業を行う日時もしくは，作業の前後関係	発災後 24 時間以内，3 月 12 日，作業 A の後
Why	作業を行う目的，なぜその作業における成果物が必要か	被災者の生命保護のため，給水準備のため

このようなラベル付けを行うことで，ラベルがない場合に比べてプロセスの定義が容易になると考えられる．まず，What は作業を定めるために必須の項目である．Who を定めることで，誰に権限と責任を与える必要があるか明確になる．Whom を定めることで，ある作業の対象にもれがないか確認が可能になる．Where を定めることで，作業の場所が離れている場合に，どのように情報の連絡や物資の移送を行うべきか明確になる．When を定めることで，作業どうしの時系列や依存関係が明確になる．Why を定めることで，ある作業における成果物が何か，成果物はそれで十分か，その成果物を利用する次の作業は何か，検討が容易になる．すなわち，6W1H モデルは，プロセスをアクティビティ，タスクへと分解していく際の議論の材料を提供する．

6W1H モデルを数学的に定義したものが以下の定義 3.2-1 である．また，その定義に基づいて，形式アシュランスケース記述言語 Agda を利用して記述したものが，図 3.2-15 である．

定義（6W1H モデル）
6W1H モデルは，以下のパラメータに基づいて定められる．
作業主体の集合 Who
作業の集合 What
作業客体の集合 Whom
作業場所の集合 Where
作業時刻の集合 When
作業理由の集合 Why
集合 Who, What, Whom, Where, When, Why の直積を 6Ws と呼ぶことにする．
6W1H モデル M は，各ノードが 6Ws の要素をデータに持つ木である．

定義 3.2-1 6W1H モデル

```
module 6W1H where

open import Data.List

record 6W1H-Parameters : Set1 where
  field
    Who   : Set
    Whom  : Set
    What  : Set
    Where : Set
    When  : Set
    Why   : Set

module 6W1H-Model-Definition(6W1H-param : 6W1H-Parameters) where
  open 6W1H-Parameters 6W1H-param

  record 6W1H-Model : Set where
    inductive
    field
      who    : Who
      whom   : Whom
      what   : What
      where_ : Where
      when   : When
      why    : Why
      how    : List 6W1H-Model
```

図 3.2-15 6W1H モデルの Agda コード

6W1H モデルを実際の防災システムに適用した事例は，3.3.2 に記述した．

2) 防災議論フレームの形式記述

6W1H モデルを利用して対象システムを記述することにより，防災システムのディペンダビリティについての厳密な議論が可能となる．一般に，大規模なシステムの性質を議論する場合，アシュランスケースも大規模かつ複雑になる．複雑なアシュランスケースの見通しをよくするためには，議論のパターン（以下，議論モデル）を作成しておき，システムの構造が類似している部分の議論に対しては，それを適用することが好ましい．
後述する事例研究に 6W1H モデルを適用して，このような議論モデルを明らかにしようとした．そこで生まれたのが DPP 議論モデルである．「DPP」とは，Decision（決定），Preparation（準備），Provision（実施）の頭文字をとったものである．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．

表 3.2-2 DPP 議論モデルの 3 分類とその意味

業務の分類	意味
決定(Decision)	ある業務を実施するために必要な情報収集および，情報に基づいた開始・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

事例研究においては，防災業務のうち特に発災後の給水業務を取り上げ，DPP 議論モデルによるアシュランスケース記述を試みた．

3) 事例研究の結果を用いた成果の改善

6W1H モデルと DPP 議論モデルは，後述する神奈川県平塚市との共同研究を通して生まれたもので，それ自体が事例研究において随時議論し，改善した成果である．研究の順序および他の成果物との関係を図 3.2-16 に示す．前述の 2 研究「防災オントロジーの形式記述」と「防災議論フレームの形式記述」それぞれが，事例研究とは不可分である．

まず，地域防災計画に基づく防災オントロジー記述のために 6W1H モデルを考案した．次に，それを適用してプロセスを記述した結果，DPP 議論モデルを発見した．また，それらの研究にあたっては，一般 FFO，平塚市地域防災計画ほか関連文書，国際標準 ISO/IEC/IEEE15288 を参照した．

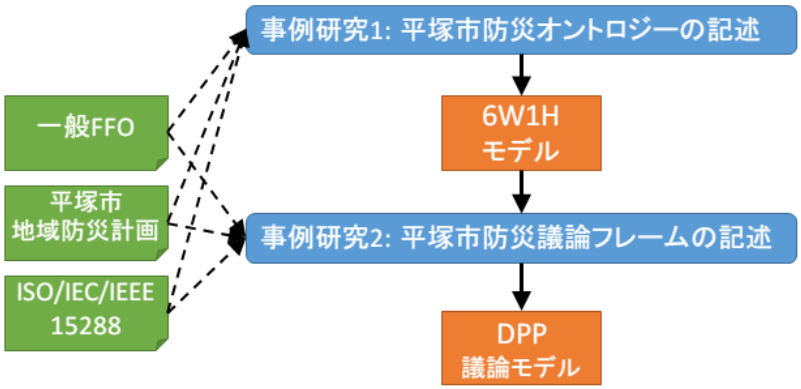

図 3.2-16　研究成果と事例研究との関係

4) 防災 FFO の標準体系における位置づけ

一般 FFO と防災 FFO との比較を図　3.2-17 に示す．前述の一般 FFO に 6W1H モデルと DPP 議論モデルを加え，一般 FFO が持つフレームワークの一部を防災向けに修正（オープンシステム一般向けの語彙を防災向けの語彙に変更）したのが，防災 FFO である．

なお，6W1H モデルや DPP 議論モデルは，防災という技術領域のみならず，ISO/IEC/IEEE15288 に依拠したシステムライフサイクルを定めるための普遍的な枠組であり，防災以外の技術領域にも適用可能である．これは，4.1 節で詳述する策定中の国際標準 IEC62853 Open systems dependability におけるオープンシステムのシステムライフサイクルが，ISO/IEC/IEEE15288 が定めるシステムライフサイクルプロセス群を利用して定義される見込みのためである．

図 3.2-17 一般 FFO と防災 FFO との比較

3.2.3 発生した課題および今後の展望

(1) 発生した課題

① システムライフサイクルプロセスの同定

FFO を適用するためには，地域防災計画をシステムライフサイクルプロセスの枠組みにはめ込む必要があった．この作業は自明なものではなく，アクティビティやタスクをどのように定めればよいのかが難しい問題であった．これに対する我々の解決策が 6W1H モデルである．タスクを定めるときの指針をこのように決定したおかげで，[研究目標 3 事例研究による有効性評価] において設定した D-P-P 議論モデルの定義を円滑に進めることができた．

② 対象システムの範囲

とくに防災システムにおいて，対象システムの範囲の決定が大きな課題となった．防災のために必要な作業が地域防災計画にすべて網羅されていると考えるのは非現実的で，実際には，市の通常業務の中に，防災のために必要な事項がいろいろな重みで含まれている．そこで，システムを市の業務全体と考えて防災業務をそのシステムの障害対応と考えるのか，システムを防災業務のみに限定するのかを選択する必要があった．この選択は，防災の専門的知見なしには難しく，自治体の防災担当者からの協力なしには決定しがたいものであった．

市町村，都道府県，国とのインターフェイスについても同様の範囲決定の問題があった．このような範囲決定のあいまいさは，まさにオープンシステムの特徴であり，オープンシステム・ディペンダビリティの有用性の根拠を提供すると考えられる．

③ 理論マップ

車載システムにおいて，安全目標，機能安全要求仕様，技術安全要求仕様などなどの phase（ライフサイクルプロセスの一般用語ではステージ）が設けられているが，本研究項目では，それらの間の関係を取り扱う必要ができた．システムが下位仕様を満たせば，必ず上位仕様を満たさなければならない．例えば，技術安全要求仕様を満たせば必ず機能安全要求仕様を満たすよう，仕様を構築することが求められる．このことをどのように理解するべきか，たとえばこのような主張を定理証明器で証明するには，形式理論をどのように配置するべきなのかは，自明ではない．普遍代数で議論される形式理論のあいだのマップの概念が適用できる可能性がある．

④ 事例観察と機密保護

本課題には，企業や自治体の現場における事例を観察することが不可欠であったが，知財保護やセキュリティ管理のために，事例観察のために必要な情報を得ることが簡単ではなかった．これは研究開始前からある程度予想していたことではあったが，企業の内部情報のみならず，同業組合の共同作業成果についても厳格な管理がなされているのは予想外であった．

(2) 今後の展望

課題［(1)①システムライフサイクルプロセスの同定］に対する我々の解決策は 6W1H モデルである．防災システムの事例では，国際規格が用意したプロセスのみでシステムライフサイクルプロセスを同定することは困難であったため，6W1H モデルによって新たなプロセスを作成した．このようなケースは，対象システムの範囲を広げることで必然的に増加すると考えられる．多様なシステムの信頼性向上のために，6W1H モデルのようなシステムライフサイクルプロセス同定手法の研究は，継続した研究が必要な分野であると考えられる．

課題［(1)②対象システムの範囲］に対する一般的な解決策があるとは思われないが，限定した状況，例えば技術領域ごとにで通用する解決策を見出すことは意義深い研究であると考えられる．このような課題は，解決策が正しいか正しくないかが問われるのではなく，関係者間の合意（賛成）が得られるかどうかが問われる種類のものである．ここでもオープンシステム・ディペンダビリティの考えが通用すると考えられる．

課題［(1)③理論マップ］は，モデル理論あるいは普遍代数（Lawvere theory など）の問題を引き起こすものであり，今回の実用的な文脈にあった理論を展開する研究は，大規模システムの信頼性向上のために本質的な寄与をなすものと考えられる．

課題［(1)④事例観察と機密保護］は，技術的な課題ではなく，本研究目標のようなフィールドワークによる研究一般に通じる研究手法の問題である．文化人類学や社会学，看護学など，人間社会を対象にするフィールドワークをおこなう分野の研究手法から学ぶことができるかもしれない．

3.3 研究目標 3「事例研究による有効性評価」

3.3.1 当初の想定

(1) 研究内容

実働のソフトウェアライフサイクルに FFO を適用してアシュランスケースの記述実験を行う．このことによって FFO の妥当性を確認し，必要があれば FFO を改善する．

(2) 想定課題と対応策

FFO の有効性評価の手段として重要であるが，実働のソフトウェアライフサイクル関係者による協力が不可欠である．協力者獲得のために，産総研システム検証研究センターや DEOS プロジェクトにおける経験とコンタクトを利用する．

3.3.2 研究プロセスと成果

(1) 研究プロセス

① アシュランスケース記述事例研究相手先探索
［研究目標 2　特定の技術領域における FFO の開発］において選んだ技術領域の事例研究フィールドを提供する協力者を探す．

② 事例研究協力者との契約
協力を得る作業における，関係者の同定，守秘義務，知財の扱い等を明確に定めた契約を，①の協力者と締結する．作業の費用のやり取りはなしとする．

③ アシュランスケース記述対象の技術領域調査
対象とする技術領域の内容を調査し，アシュランスケース記述に必用な知識を記述担当者が得る．

④ アシュランスケース記述実験
③の技術領域における具体的なシステムライフサイクルを一つ選び，それに関するアシュランスケースを研究目標 2 「特定の技術領域における FFO の開発」の成果を用いて記述する実験を行う．

⑤ ④の過程および結果を用いた FFO の有効性評価
実験において研究目標 2 「特定の技術領域における FFO の開発」の成果の問題点および改善の可能性を考察し，その有効性を評価する．

(2) 具体的な研究成果の内容

研究目標 2 で対象とした特定技術領域である車載システムと防災システムについて，それぞれの FFO の妥当性を確認した．

① 車載システム

本研究項目の研究成果は「導出パターン」の概念とその車載システムにおける利用例である．

1) アシュランスケース記述事例研究相手先探索および契約

ISO26262［7］(自動車の機能安全標準) に従った製品開発の経験を持つ自動車部品メーカーと自動車機能安全に関する共同研究契約を締結した.

2) アシュランスケース記述対象の技術領域調査

ISO26262 に従った開発を行うために (社) JASPAR によって提供されている機能安全テンプレート[8]を用いた「電動パワーステアリングシステム」の模擬開発の設計仕様書を基に事例研究を行った. ISO26262 に対応した自動車部品開発についての疑問点については, 共同研究先に尋ねることにより, アシュランスケース記述に必用な知識を得た.

3) アシュランスケース記述実験

AFSCF 議論モデルの中で「仕様策定の根拠」の議論が一番重要である. (ISO26262 で要求されている成果物だけでは議論ができないため). そこで「電動パワーステアリングシステム」の模擬開発についてのアシュランスケースを作成するにあたって, 仕様策定の根拠の議論を重点的に扱うこととし, 機能安全仕様から技術安全仕様を導く工程(機能安全/技術安全 phase)をもとに記述実験を行った.

機能安全に関する要求仕様は「危険を予知して回避する」という形のものが多く, いくつかの型に分類する事ができる. 各型について,「仕様設定の根拠」についてのパラメータ化した議論の雛形を作成しておく. 個々の要求仕様の導出時には, 上位の要求仕様, 上位レベルのシステム構成, 下位レベルのシステム構成, FTA の結果等から, 適切な議論の雛形と実パラメータを選択し, 雛形に実パラメータを代入することにより, 下位の要求仕様を導出することができる.

この事により, 負担を大きくする事なく, アシュランスケースの記述が可能となる. この雛形を用いた手法を「導出パターン」として提案した. 次節で導出パターンについて詳しく述べる.

4) 導出パターンについて

導出パターンは, 上位の仕様から下位の仕様を導出するパターンのことである (以下に定義する). これを利用することで, 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ていく. 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 3.3-1 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たため, 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.

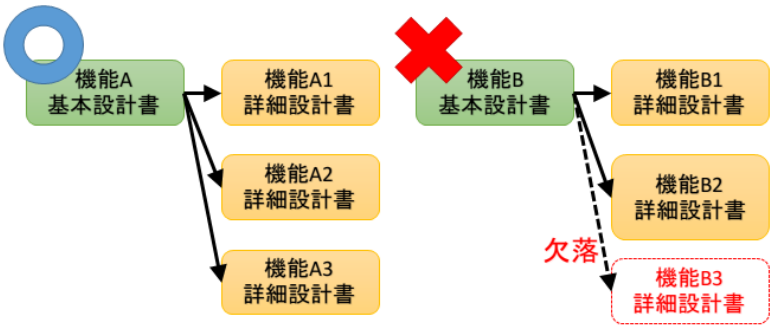

図 3.3-1 下位仕様と上位仕様との関係の例

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」ことをあらかじめ示しておくことが重要になる．それによって，図 3.3-1 のような問題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 3.3-2 である．右の例では，機能 B1 と機能 B2 の詳細設計書が満たされれば，機能 B が満たされるかを確認することで，別の機能 B3 が必要であることが判明する．

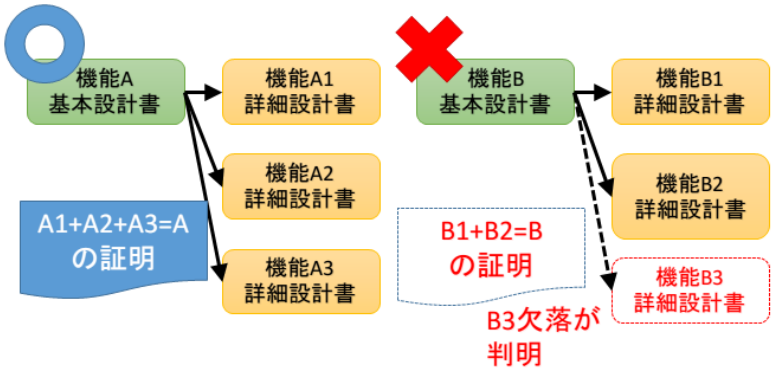

図 3.3-2 下位仕様と上位仕様との関係を事前に確認することで，不備を発見

しかし，どうすればこの事実が示されるのかは開発者にとって自明ではない上に，ひとつひとつの下位仕様について上位仕様との関係を検討するのはコスト面の負担が大きいためである．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．

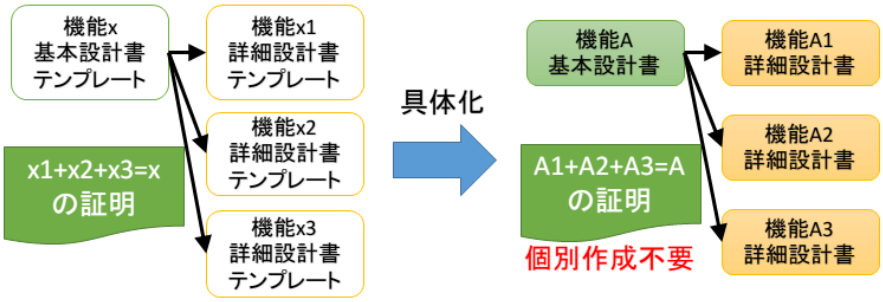

図 3.3-3 導出パターンによる仕様作成

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ことが考えられる．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．このようなテンプレートが導出パターンである（図 3.3-3）．

以下では，本研究で提案する「導出パターン」について例を交えて詳しく説明する．

(ア) 導出パターン

導出パターンは

a) 上位要求仕様のパターン

b) 下位要求仕様のパターン

c) 導出パターンの適用条件

の三要素からなる．機能安全/技術安全 phase においては

- a) は，パラメータ化された機能安全要求仕様である
- b) は，パラメータ化された技術安全要求仕様の列である
- c) は，代入によって得られた機能安全要求仕様から技術安全要求仕様の導出が妥当となるため(技術安全要求仕様を達成すると機能安全要求仕様が達成される)の実パラメータの条件である．

(イ) 例

「電動パワーステアリングシステム」模擬開発の導出パターンの適用による要求仕様の導出が妥当であること，すなわち導出パターンの適用条件を満たす実パラメータの代入で得られた，機能安全要求仕様から技術安全要求仕様の導出が妥当となることの根拠(導出パターンの妥当性と呼ぶ)は，別に提示する事が必要である．

機能安全/技術安全 phase における例によって導出パターンの説明を行なう．

機能安全要求仕様が

- FSR-001 「機能レベルブロック<アシストトルク生成>の故障<出力系異常でセルフステアに至る故障>を判別できる情報を出力する」

技術安全要求仕様が

- TSR-002　「システムレベルブロック<電流検出>の故障<電流検出が正しくない>を判別できる情報を出力する」
- TSR-015　「システムブロック<アシストトルク生成>の故障<PWM 信号演算の異常>判別できる情報を出力する」

使用する導出パターン「例１」は次のように与えられているとする.

a)　機能安全要求仕様のパターン
「機能ブロック 〈FB〉 の故障 〈FF〉 を判別できる情報を出力する」

b)　技術安全要求仕様のパターン
「システムレベルブロック<SB1>の故障<SF1>を判別できる情報を出力する」, … ,
「システムレベルブロック<SBn>の故障<SFn>を判別できる情報を出力する」

c)　導出パターン適用条件
機能ブロック 〈FB〉 の機能レベルの故障 〈FF〉 を判別できる情報としては, システムレベルでは,「システムレベルブロック 〈SB1〉 の故障 〈SF1〉 を判別できる情報」, … ,「システムレベルブロック 〈SBn〉 の故障<SFn〉 を判別できる情報」の n 個の情報で十分である.

ここで<FB>, <FF>, <SB1>, ... , <SBn>, <SF1>, ... , <SFn> はパラメータである.

(ウ) 機能安全要求仕様から技術安全要求仕様の導出

FSR-001 から TSR-002, TSR-015 の

- a) 機能安全要求仕様のパターンと機能安全要求仕様 FSR-01 を対応づける
  〈FB〉 ← 〈アシストトルク生成〉
  〈FF〉 ← 〈出力系異常でセルフステアに至る故障〉
  とすることで, a) から FSR-01 が得られる.
- 導出パターンの適用条件を抽出
  n=2 とし,
  〈SB1〉 ← 〈電流検出〉
  〈SB2〉 ← 〈アシストトルク生成〉
  〈SF1〉 ← 〈電流検出が正しくない〉
  〈SF2〉 ← 〈PWM 信号演算の異常〉
  とすることで c) 導出パターンの適用条件は次のようになる.
  「機能ブロック 〈アシストトルク生成〉 の機能レベルの故障 〈出力系異常でセルフステアに至る故障〉を判別できる情報」は, システムレベルでは,「システムレベルブロック 〈電流検出〉 の故障 〈電流検出が正しくない〉 を判別できる情報」,「システムレベルブロック 〈アシストトルク生成〉 の故障<PWM 信号演算の異常〉を判別できる情報」の 2 個の情報で十分である.
- c) 導出パターンの適用条件を確認する
  システム構成図, FTA の結果などから, 得られた適用条件が正しいことを確認する.
- 技術安全要求仕様を導出する

適用条件が確認されたパラメータの代入を b) 技術安全要求仕様のパターンに適用して，技術安全要求仕様 TSR-002，TSR-015 を得る．

導出パターン「例 1」の妥当性が示されていれば，機能安全要求仕様 FSR-001 に対応する技術安全要求仕様は TSR-002，TSR-015 としてよい．逆に，技術安全要求仕様 TSR-002，TSR-015 を達成するシステムは，機能安全要求仕様 FSR-001 を達成すると主張できる．

仕様導出時の，導出パターンの適用情況を GSN 風に記述すると次のようになる．

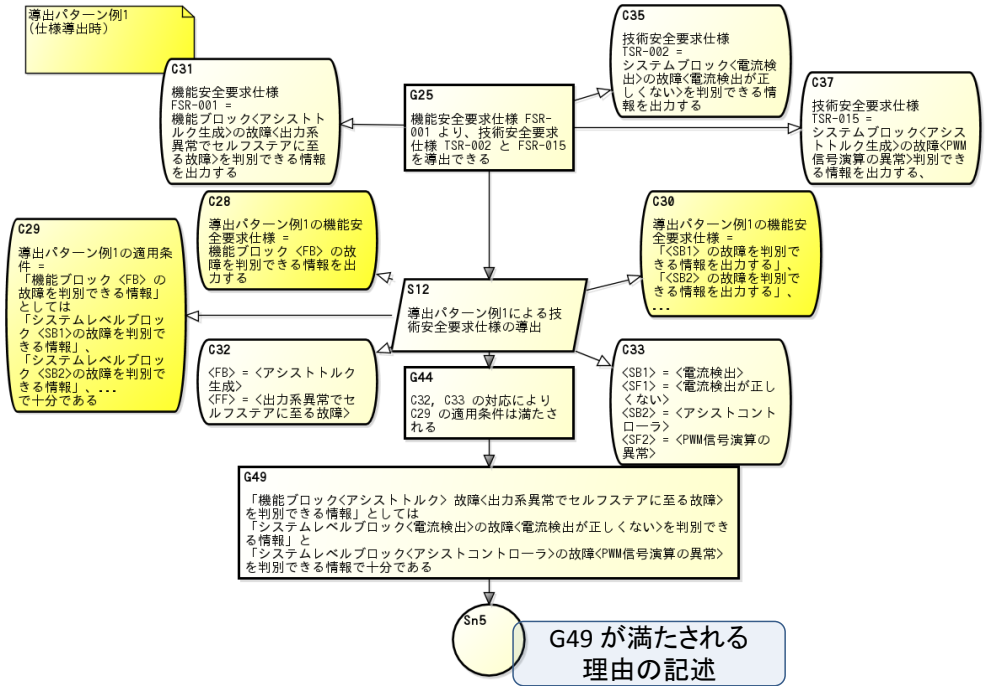

図 3.3-4 導出パターン適用例（仕様導出時）

(エ) 導出パターンの適用によるアシュランスケース

ゴール 「技術安全要求仕様 TSR-002，TSR-015 の達成は，機能安全要求仕様 FSR-001 の達成を導く」に対する議論は，導出パターンを用いると，次の 3 つのサブゴールを示せばよいことになる．

- 与えられたパラメータの代入により a) 機能安全要求仕様のパターンから FSR-01 が得られる
- 与えられたパラメータの代入により b) 技術安全要求仕様のパターンから TSR-02 と TSR-015 が得られる
- c) 導出パターン適用条件に，与えられたパラメータの代入によって得られた適用条件が満たされる

アシュランスケースは次のようになる．

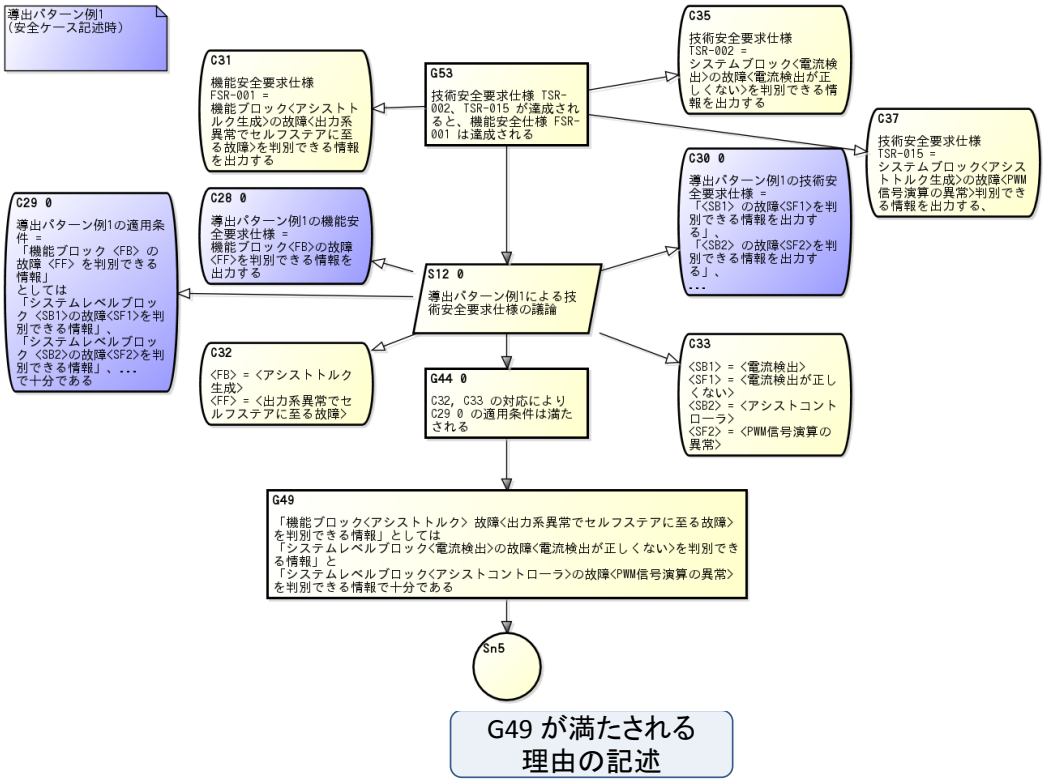

図 3.3-5 導出パターン適用例（アシュランスケース記述時）

(オ) 機能安全/技術安全 phase の「仕様設定の根拠」

導出パターンによる議論を用いることにより，機能安全/技術安全 phase の「仕様設定の根拠」についての議論は次のようにまとめられる．

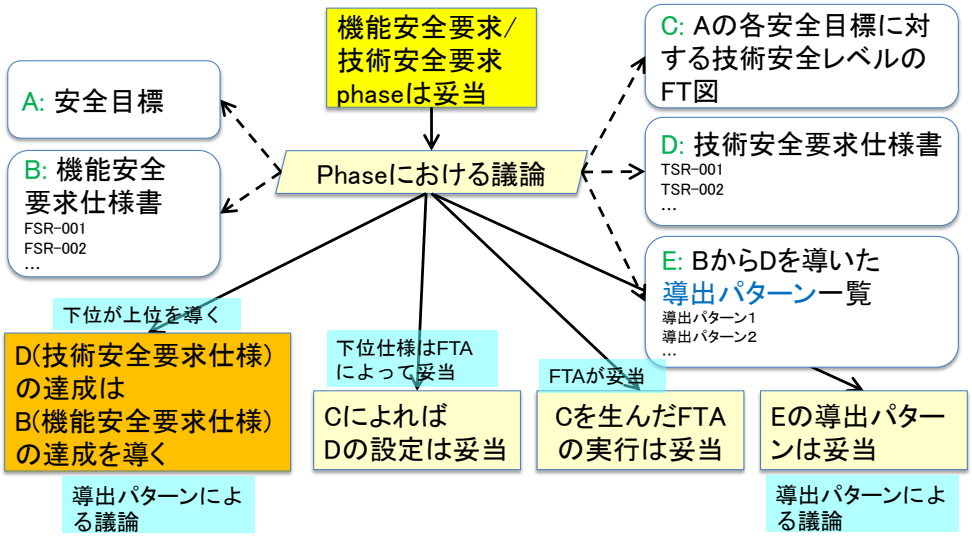

図 3.3-6 機能安全要求/技術安全要求 phase における「仕様設定の根拠」議論

(カ) 導出パターンの妥当性

導出パターン「例１」の妥当性に関しては，ほとんど自明である．しかし説得力を持って主張するためには，数学的な議論が必要となる．我々は，導出パターンの妥当性を数学的に示すために，機能安全要求仕様，技術安全要求仕様を数学的理論として捉え，「技術安全要求仕様の達成は機能安全要求仕様の達成を導く」ことを，理論間の形式的解釈(formal interpretation) として捉える手法を提案した．

5) FFO の有効性評価

導出パターンを用いた議論のフレームワークとその有効性について共同研究先と定期的に議論を行った．その結果，導出パターンにより作業量をそれほど増やさずに要求仕様の導出の妥当性について記述できること，数学的枠組みも与えられていることから，我々のフレームワークが有効であるとの評価を受けることができた．

② 防災システム

1) アシュランスケース記述事例研究相手先探索/事例研究協力者との契約

神奈川県平塚市の協力を得ることができ，同市の地域防災計画に規定されている防災業務を対象として防災 FFO を適用したアシュランスケース記述実験を行った．同市と共同研究を開始し，平塚市の防災担当部署である防災危機管理部災害対策課の担当者と定期的に打合せを実施し，防災に関する知識提供を受けつつ，研究成果に対するフィードバックを得た．

2) アシュランスケース記述対象の技術領域調査

平塚市地域防災計画および関連文書を調査して明らかになった事項のうち，アシュランスケース記述に関係する事項を以下に列挙する．

(ア) 平塚市地域防災計画の構成

3.2.2 で述べたように，地方自治体が行う防災関係業務は，「地域防災計画」という文書によって規定されている．平塚市の防災業務は，平塚市防災会議によって作成された「平塚市地域防災計画」[18]によって規定される．最新版である平成 27(2015)年 3 月改訂の平塚市地域防災計画の構成は，表 3.3-1 に示すとおりである．地震と風水害を中心に，いくつかの災害を想定して計画が記述されている．

表 3.3-1　平塚市地域防災計画の構成

冊子名	概要	ページ数
地震災害対策計画	地震・津波対策	196
風水害対策計画	台風・大雨対策	181
東海地震に係る地震防災強化計画	東海地震対策	24
特殊災害対策計画	海上災害・放射性物質災害・火山災害・鉄道災害・航空災害対策	33
資料編	各計画で利用する図表等	288

本研究では，事例研究としての適切さを考慮して地震災害対策計画を対象とした．その理由は，記述量が多い計画を対象とすることで，より大きなアシュランスケースの記述が可能になると考えたためである．

(イ) 防災計画間の階層構造

行政の防災業務には図 3.3-7 に示すような階層構造があり，下位の文書は上位の文書に対する整合性を保つことが求められる．具体的には，まず災害対策基本法に基づき，国が「防災基本計画」を制定している．神奈川県では，「神奈川県地域防災計画」を定めているが，これは災害対策基本法をはじめとする法律や，防災基本計画との整合性を保つことが要請される．同様に，平塚市地域防災計画も，その上位にあたる神奈川県地域防災計画および諸法律・防災基本計画との整合性を保つことが要請される．さらに，平塚市地域防災計画の記述では不足する部分については，市の各組織や避難所単位で「マニュアル」と呼ばれる文書を作成する．これらも当然，上位の計画との整合性を保つことが要請される．

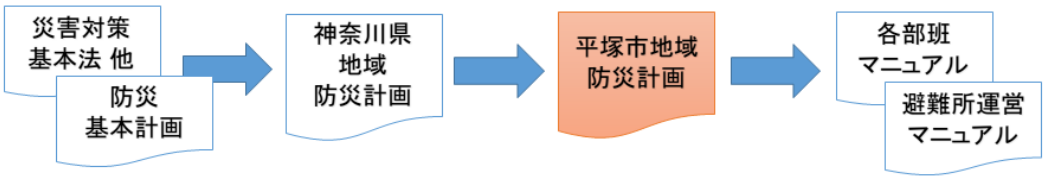

図 3.3-7　防災業務を規定する文書の階層構造（平塚市の場合）

本研究では，平塚市地域防災計画の記述で不足する部分については，マニュアルを参考としてアシュランスケースの記述を試みた．

(ウ) 地域防災計画とシステムライフサイクルモデルとの対応

地震災害を対象としたことから，表 3.3-1 のうち，地震編の構成を調査した．特に，FFO が依拠するシステムライフサイクルモデル（通常運用の他に，合意形成，障害対応，変化対応，説明責任の 4 つのプロセスビューを持つ SLC）と，章立てが対応することが明らかになった．その結果を

表 3.3-2 に示す．

表 3.3-2　平塚市地域防災計画地震編の構成

章	概要([ ]は SLC の対応先)	ページ数
第 1 章 地震災害対策の計画的推進	計画の位置づけ，地震ごとの被害想定	24
第 2 章 減災に向けたまちづくり	平常時におけるハード面（インフラ整備等）の対策[通常運用]	14
第 3 章 平常時の対策	平常時におけるソフト面（人的支援等）の対策[通常運用]	34
第 4 章 災害時の応急対策	発災時の災害対策本部の運営や各種業務[障害対応]	108
第 5 章 災害復旧・復興対策	応急対策後の長期的な防災業務[変化対応]	16

(エ) 応急対策業務の分類

表 3.3-2 が示す章で構成される平塚市地域防災計画だが，章以下の各節は必ずしも個々に独立していないことが明らかになった．すなわち，1 つの節が 1 つの業務を記述するとは限らず，1 つの業務が複数の節に分散して記述されている場合や，複数の業務が 1 つの節に混在して記述されている場合があった．

そこで，地域防災計画のうち，特に発災時の応急対策についての記述を調査し，応急対策業務を以下の 9 業務に分類した（表 3.3-3）．

表 3.3-3　応急対策業務の 9 分類

No.	名称	概要
1	物資供給	水，食料，その他物資の供給
2	避難支援	避難所運営，帰宅困難者対応
3	医療	病院，救護，要支援者への支援，遺体処理
4	救急	消火，救急（救難，救助）
5	インフラ復旧	水道管，道路復旧などの公共インフラの復旧
6	建物復旧	住宅復旧，仮設住宅，建築判定
7	衛生	廃棄物処理
8	応援	他自治体，ボランティア等の支援受け入れ
9	管理	情報収集・伝達，情報に基づく判断

この分類は，応急対策業務の記述から構築したものであるが，それぞれの業務に対して，平常時の業務・復興時の業務もあると考えられる．また，平塚市に限定した分類ではなく，他の自治体防災業務にも適用可能である．つまり，防災システムのアシュランスケース記述時に広く利用できる分類である．

(オ) 応急対策時の給水業務を事例として選択

アシュランスケースを記述するにあたり，最初の対象を応急対策時の給水業務とした（図 3.3-8）．これは，給水業務の記述が地域防災計画に充実しており（第 4 章第 9 節 1　給水対策として約 5 ページ），事例として適切と判断したためである．また，その下位文書にあたる給水業務のマニュアルを随時参照した．

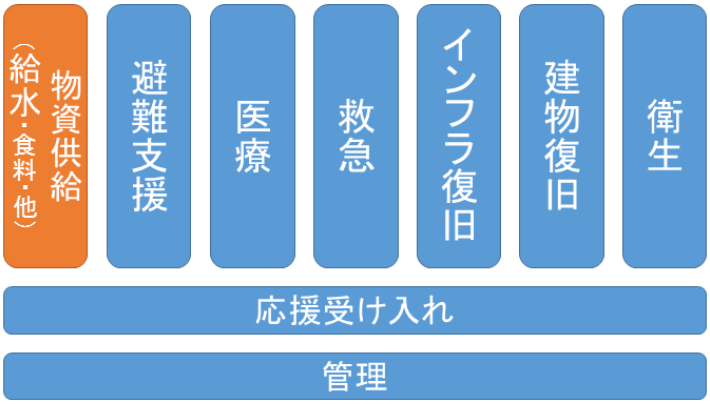

図 3.3-8　応急対策業務の 9 分類における給水業務の位置づけ

3) アシュランスケース記述実験

以上の調査結果をもとに，アシュランスケース記述を試みた．しかし，防災業務のシステムライフサイクルは，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．

これには大きく 2 つの理由がある．まず，地域防災計画に記述すべき事項は，防災基本計画等の上位文書によってある程度規定されている．しかし，地域防災計画に記述すべき事項をどのような様式で記述すべきかの規定はない．その結果，地域防災計画は長年の加筆修正に伴って複雑かつ膨大になり，容易には理解が困難な文書となっている．

また，防災業務がシステムであるという考えは，世間一般に認知されたものではない．そのため，地域防災計画の執筆者が，防災業務とはシステムであるとは考えておらず，システムライフサイクルについても考慮にないことは当然である．

この問題を解決するために，防災 FFO の 6W1H モデルを利用した．まず，事例として選んだ応急対策時の給水業務について，ISO/IEC/IEEE15288 に定める Tailored Process として「給水プロセス」を定義した（表 3.3-4）．（ISO/IEC/IEEE15288 には 30 のプロセスが定義されているが，適用するプロセスがない場合には，一定の基準に従ってプロセスを修正・新規作成することが許容されている．この基準に従って作成されたプロセスのことを Tailored Process と言う.）

これにより，DPP 議論モデルという新たな議論モデルを発見した．次に，給水プロセスのアシュランスケースを，DPP 議論モデルに従って記述した（図 3.3-9）．そして，防災 FFO の

形式アシュランスケース・フレームワークを適用して，平塚市の防災業務全体のアシュランスケースを記述した（図 3.3-10）.

図 3.3-9 に赤枠で示したのが，DPP 議論モデルに従ってアシュランスケースのゴールを 3 つのサブゴールに分割した部分である．また，図 3.3-10 に赤枠で示したのが，図 3.3-9 のアシュランスケース全体にあたるサブモジュールである.

このアシュランスケースでは，平塚市の防災業務全体のアシュランスケースのうち，障害対応の部分を想定される被害ごとに分割した．これにより，発災時に給水業務が実施される理由は，上水道の断水という被害（障害）に対応するためであるということが明らかになっている.

4) アシュランスケース記述の過程および結果を用いた FFO の有効性評価

アシュランスケースの作成にあたり，平塚市防災危機管理部災害対策課の担当者による定期的なレビューを受けた．その結果，以下の評価を得た.

- 6W1H モデルを利用し，地域防災計画の記述において必ずしも明らかではない業務の主体（Who）等を明確にすることは有効である.
- 6W1H モデルで業務の階層構造を明確にすることで，地域防災計画に記述すべき部分と，その下位文書の各マニュアルに記述すべき部分との区別が明確になる．(2015-07-31)
- 「決定，準備，実施」という枠組みを持つ DPP 議論モデルは，給水業務以外にも適用できると考えられる.
- 現在，地域防災計画の下位文書であるマニュアルについて，マニュアルごとに記述の粒度がばらついているという問題がある．アシュランスケース記述の枠組みづくりが，マニュアルの記述のばらつきを防ぐ統一的な枠組みを提供する可能性がある.
- 平塚市地域防災計画に計画された業務は，平塚市総合計画という文書で進捗管理される．しかし，平塚市地域防災計画と平塚市総合計画の間の業務の対応付けが必ずしも明らかではないという問題がある．アシュランスケースのうち，通常運用，変化対応の部分を掘り下げていくことで，この対応付けが明らかになる可能性がある
- フレームワークにおける「障害対応」は，災害による「被害への対応」と考え，平塚市地域防災計画に記載されている想定される被害ごとにアシュランスケースのゴールを分割することは自然である.

表 3.3-4　6W1H モデルによる給水プロセスの定義

ID	Who	What	Whom	When	Where	Why
P	市	飲料水等を供給	被災者等	発災時	避難所等	市民の生命・身体の保護
A1	市	給水の実施を判断	×	発災後	災害対策本部設置場所	必要とする市民への給水
T1-1	総務部被害調査班、特別調査班	水道の被害状況を調査	×	発災後	×	給水開始/終了の決定
T1-2	総務部被害調査班、特別調査班	水道の被害状況を報告	総合対策部総合調整班	T1-1の後	×	給水開始/終了の決定
T1-3	土木復旧部	交通の状況を調査	×	発災後	×	給水開始/終了の決定
T1-4	土木復旧部	交通の状況を報告	総合対策部総合調整班	T1-3の後	×	給水開始/終了の決定
T1-5	給水部	給水体制の進行状況等を調査	×	給水開始後	×	給水継続/終了の決定
T1-6	給水部	給水体制の進行状況等を報告	総合対策部総合調整班	T1-5の後	×	給水継続/終了の決定
T1-7	災害対策本部	給水の実施を判断	×	T1-2,4,6の後	災害対策本部設置場所	必要とする市民への給水
A2	市	給水業務を準備	被災者等	A1の後	市内各地	必要とする市民への給水
T2-1	総合対策部広報班	汲み置きを連絡	自主防災組織	A1の後	×	飲料水の確保
T2-2	自主防災組織	汲み置きを呼びかけ	被災者等	T2-1の後	×	飲料水の確保
T2-3	県企業庁平塚水道営業所	貯水量を確認	×	A1の後	平塚配水池	飲料水の確保
T2-4	県企業庁平塚水道営業所	貯水量を連絡	災害対策本部	T2-3の後	×	飲料水の確保
T2-5	協定締結事業者	飲料水の状況を確認	×	A1の後	事業所	飲料水の確保
T2-6	協定締結事業者	飲料水の状況を連絡	災害対策本部	T2-5の後	×	飲料水の確保
T2-7	給水部	非常用貯水タンクの状況を確認	×	A1の後	非常用貯水タンク所在地	飲料水の確保
T2-8	給水部	非常用貯水タンクの状況を連絡	災害対策本部	T2-7の後	×	飲料水の確保
T2-9	消防部	火災の状況を確認	×	A1の後	災害対策本部	飲料水の確保
T2-10	給水部	臨時給水栓の設置を協議	消防部	T2-9の後	×	飲料水の確保
T2-11	給水部、避難部	臨時給水栓を設置	×	T2-10の後	消火栓所在地	飲料水の確保
T2-12	避難部	ろ水機を移動	×	A1の後	保管場所から利用場所へ	飲料水の確保
T2-13	避難部	耐震性プールの水をろ過	×	T2-12の後	ろ水機利用場所	飲料水の確保
T2-14	県企業庁平塚水道営業所	配水管を復旧	×	A1の後	配水管故障個所	飲料水の確保
T2-15	県企業庁平塚水道営業所	応急給水栓を設置	×	T2-14の後	避難所等	飲料水の確保
T2-16	給水部	給水の計画を決定	×	A1の後	災害対策本部設置場所	必要とする市民への給水
A3	市	給水業務を実施	被災者等	A2の後	避難所等	必要とする市民への給水
T3-1	給水部	給水車を移動	×	A2の後	元の場所から平塚配水池へ	飲料水の供給
T3-2	給水部	飲料水を移送	×	T3-1の後	平塚配水池から給水車へ	飲料水の供給
T3-3	給水部	給水車を移動	×	T3-2の後	平塚配水池から目的地へ	飲料水の供給
T3-4	給水部	給水車を移動	×	A2の後	元の場所から各事業所へ	飲料水の供給
T3-5	給水部	飲料水を移送	×	T3-4の後	事業所	飲料水の供給
T3-6	給水部	給水車を移動	×	T3-5の後	事業所から目的地へ	飲料水の供給
T3-7	給水部	給水車を移動	×	A2の後	元の場所から非常用貯水タンク所在地へ	飲料水の供給
T3-8	給水部	飲料水を移送	×	T3-7の後	非常用貯水タンク所在地	飲料水の供給
T3-9	給水部	給水車を移動	×	T3-8の後	非常用貯水タンク所在地から目的地へ	飲料水の供給
T3-10	市民、事業所	水道水を汲み置き	×	A2の後	自宅、各事業所	飲料水の確保
T3-11	避難所運営委員会等	飲料水を供給	被災者等	T3-3,6,9の後	目的地	市民の生命・身体の保護
T3-12	避難所運営委員会等	飲料水を供給	被災者等	A2の後	消火栓所在地	市民の生命・身体の保護
T3-13	避難所運営委員会等	耐震性プールの水を供給	被災者等	A2の後	ろ水機利用場所	市民の生命・身体の保護
T3-14	県企業庁平塚水道営業所	飲料水を供給	被災者等	A2の後	応急給水栓所在地	市民の生命・身体の保護

※IDの頭文字P=プロセス、A=アクティビティ、T=タスク

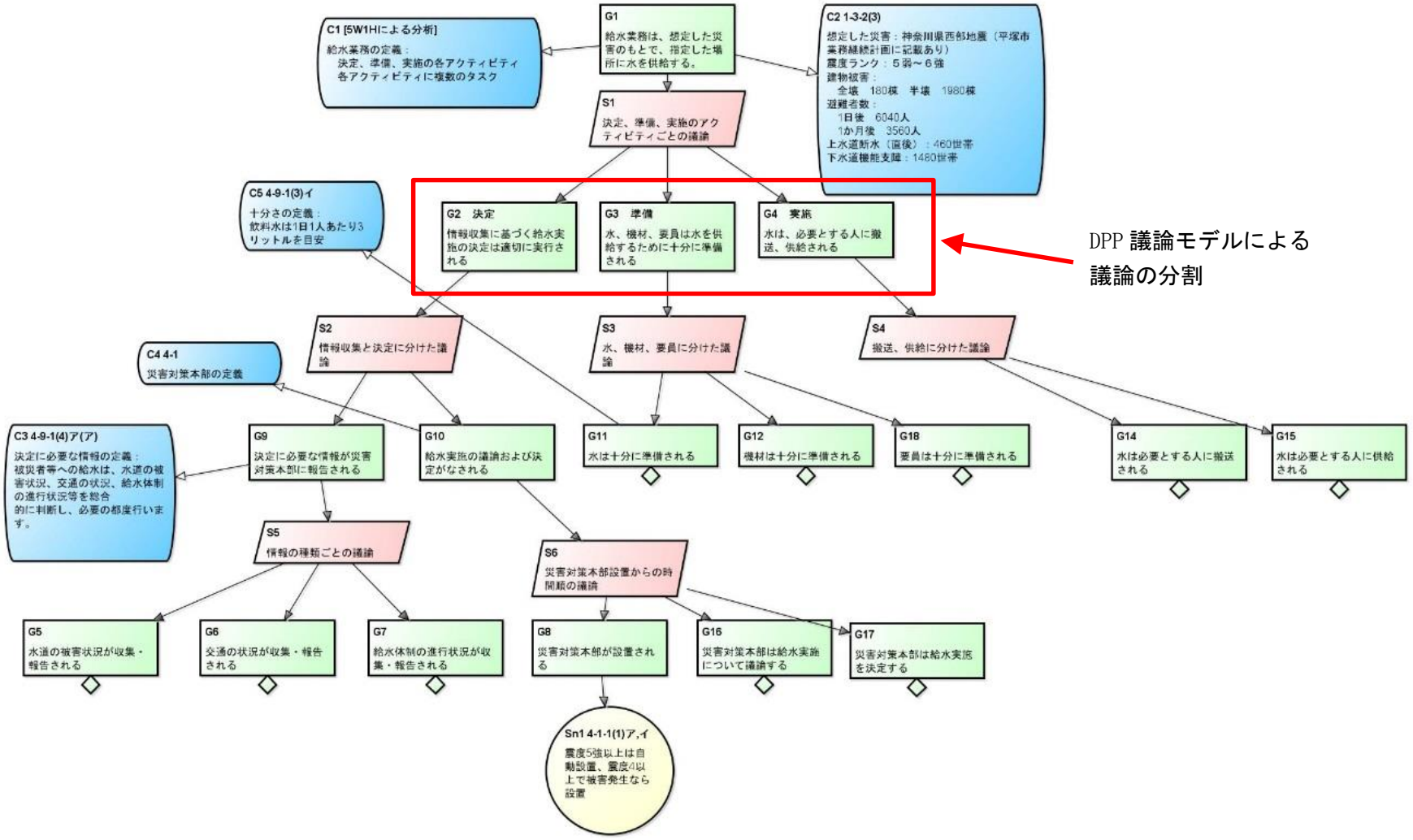

図 3.3-9　給水業務のアシュランスケース

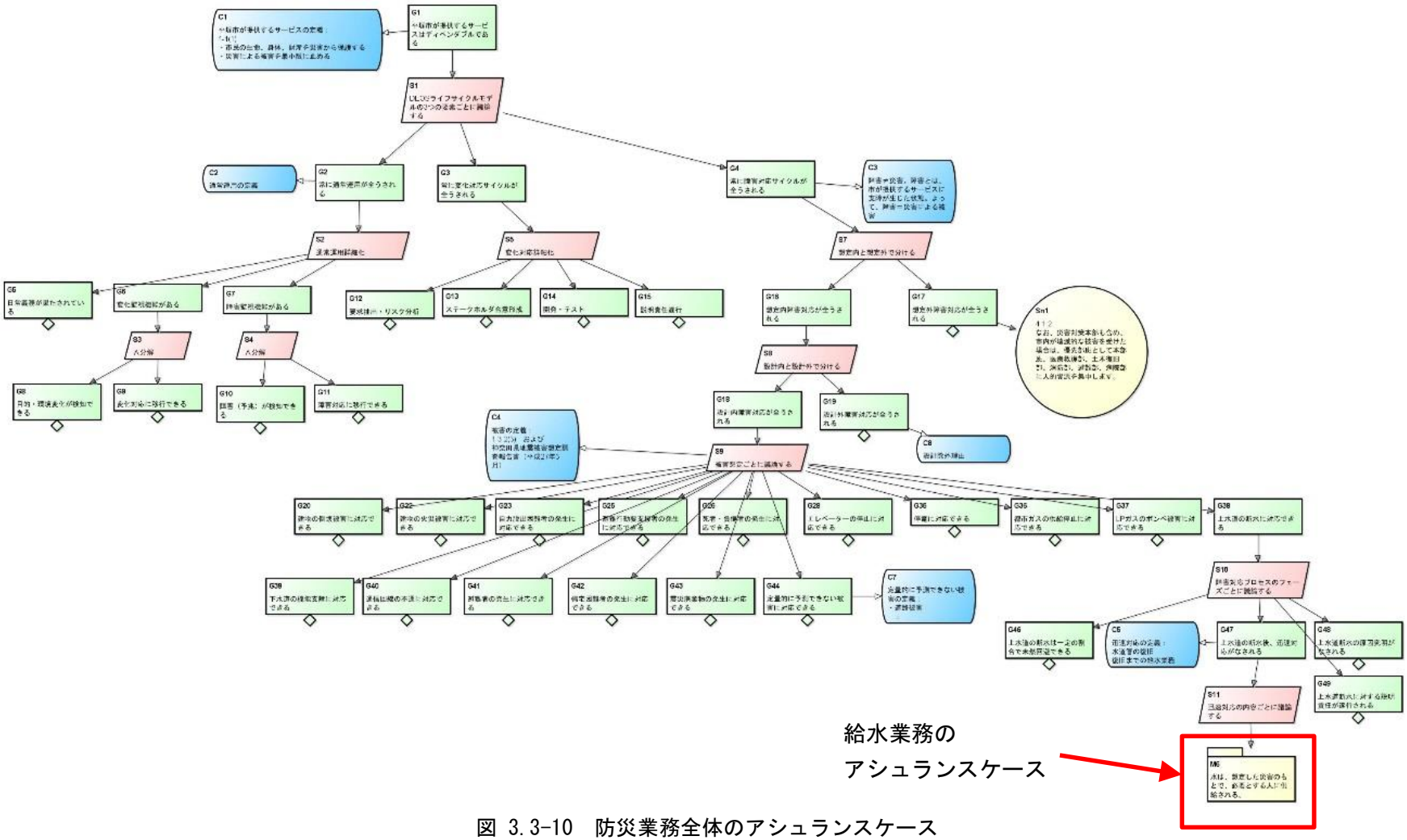

図 3.3-10　防災業務全体のアシュランスケース

3.3.3 発生した課題および今後の展望

(1) 発生した課題

① 対象システムの膨大さ

車載システム，防災システムともに，対象が大きく，本研究では全体をカバーする分野別 FF0 を作成することはできず，対象を一部分に絞った FF0 を作成することとした．車載システムでは，機能安全要求仕様と技術安全要求仕様，及びこの二つの間の関係の議論に対象を絞った．防災システムに関しては給水業務に的を絞った．

② 有効性評価の困難さ

記述実験は FF0 の有効性を対象技術領域の専門家によって評価してもらうためのものであったが，対象技術領域の専門家による本研究への協力は得られたものの，専門家がアシュランスケース，さらには形式アシュランスケースの知識をあまり持ち合わせない，あるいはまったく持ち合わせていなかったため，FF0 そのものの評価を求めることができなかった．車載システムの場合は，専門家がある程度アシュランスケースの知識を持っていたため，形式アシュランスケースのフレームワークである FF0 を GSN で構造化アシュランスケースとして表現したものを作り，それに対する評価を依頼することができた．防災システムの場合は，アシュランスケースについては，本研究に関連した知識を専門家が持つのみであったので，FF0 を他のアシュランスケース・フレームワークと比較して評価してもらうことが困難であった．

(2) 今後の展望

課題［(1)①対象システムの膨大さ］については，研究開始以前からある程度予測していたことであり，的をしぼることによって解決した．これは大規模システムを対象とする研究に共通する課題であるけれども，一般的にどのようにするべきなのかは明らかではない．

課題［(1)②有効性評価の困難さ］も，研究の方法論に関する課題である．また，その原因は，アシュランスケースに関する知識が対象技術領域に広まっていないことにある．この意味では，本研究項目の設定は野心的でありすぎたのかもしれない．今回，車載システムについておこなった手法が，方法論の一つのモデルになるかもしれない．つまり，最先端の手法（この場合形式アシュランスケース）を使って作ったものの評価をしてもらうために，そのもの本質を保ちながら state of the art の手法（この場合構造化アシュランスケース）を使って作ったものに変換あるいは翻訳し，それを評価してもらう，という研究法である．

3.4 研究目標 4「FFO が依拠するシステムライフサイクル概念の確立」

3.4.1 当初の想定

(1) 研究内容

FFO はシステムライフサイクルがオープンシステム・ディペンダビリティを達成することを主張するアシュランスケースのフレームワークである．しかし，システムライフサイクル周辺の用語定義は，産業界でも一つに収束しておらず，ディペンダビリティ周辺でもディペンダビリティライフサイクルプロセス(IEC 60300-1[13])とシステムライフサイクルプロセス(ISO/IEC/IEEE 15288[12])が必ずしも相互に整合的でない形で規定されている．

先行研究[16][17]では，オープンシステム・ディペンダビリティを達成するシステムライフサイクルモデルとして，DEOS プロセスが提唱された．しかし，これも既存のシステムライフサイクルと整合的な形で提唱されていない．例えば DEOS プロセスの「プロセス」の語の使い方はシステムライフサイクルに関する ISO の最上位標準[12]における使い方に一致していない．

そこで，システムライフサイクルに関連する既存の国際標準における用語定義を比較対照した上で，これらに矛盾しない形でオープンシステム・ディペンダビリティを達成するライフサイクルモデルとして提供する．

(2) 想定課題と対応策

ディペンダビリティライフサイクルとシステムライフサイクルが，国際標準において独立に規定されている．FFO はこの二つを参照する必要があるため，両者の用語の相違やずれをどのように扱うのかを決定しなければならない．そのため，これら二つの国際標準の比較対照を行った上で，両者と矛盾しない形でオープンシステム・ディペンダビリティライフサイクルの概念を確立し，FFO が依拠する概念として提示する．

3.4.2 研究プロセスと成果

(1) 研究プロセス

① ライフサイクルの記述

先行研究（[16][17]）における DEOS プロセスのグラフを解釈して，意図する遷移系をペトリネットを用いて導き出す方法を考案する．ディペンダビリティライフサイクルの定義を文書で明確に記述するための手法としてペトリネットによる記述を採用する．

② 国際標準との比較対照

ディペンダビリティライフサイクルの定義を記述し，IEC62853 Committee Draft[11] Annex B として IEC TC56 各国委員会に回覧し，既存のシステムライフサイクル概念との比較対象を行い，フィードバックを得てより洗練したものとした．回覧された草稿は国際標準として出版される予定である．

(2) 具体的な研究成果の内容

① ライフサイクルの記述

ライフサイクルモデルの記述では，プロセスを節にし，節の間の辺によってプロセスの遷移を表現するグラフがよく用いられる．先行研究で提案された DEOS プロセス（[16][17]）の記述もそのようなものである．しかし，このグラフを解釈して，意図する遷移系をどのように導き出すのかは自明ではない．我々は，ペトリネットを用いた解釈を提出した（図 3.4-1）．

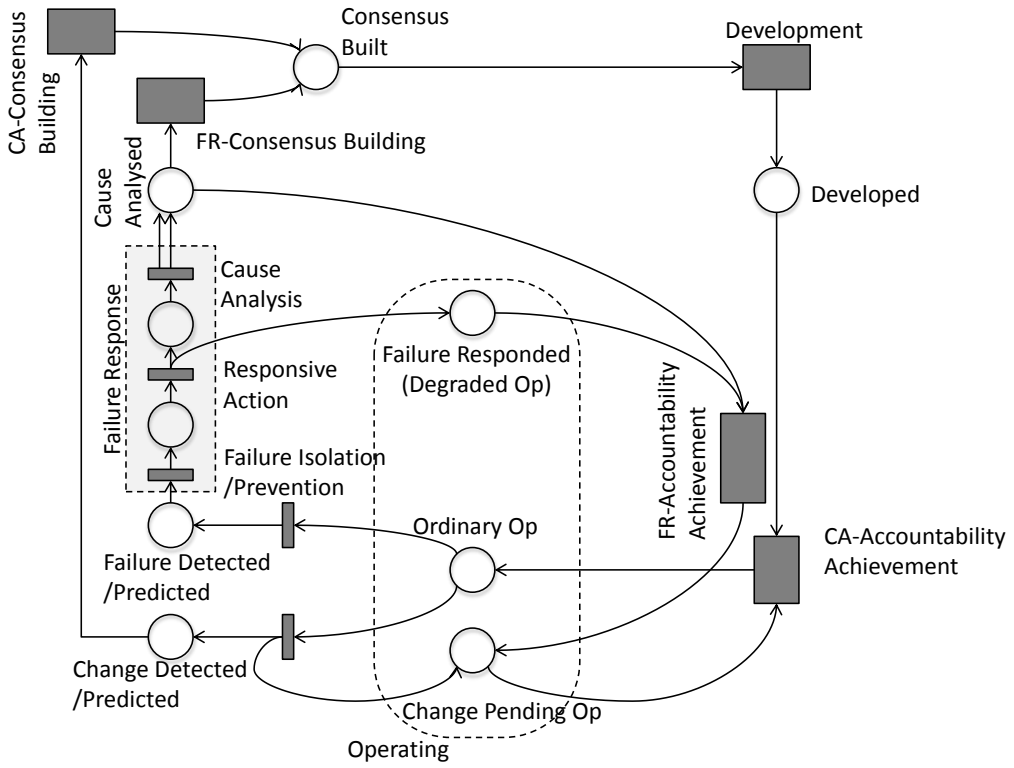

図 3.4-1 ペトリネットによる DEOS プロセスの表現

DEOS 基本構造では，DEOS プロセス（もの）と，それが持つ性質（こと）の混同がみられる．そこで我々は，DEOS プロセスを「もの」としてモデル化して，DEOS ライフサイクルモデルを定義し，「～は OSD を達成している」を DEOS ライフサイクルモデルが満たすべき性質（こと）として定式化した．

先行研究[16][17]では，DEOS プロセスをグラフによって図示している．しかし，このグラフが何を意味しているのかは自明ではない．グラフの節（ライフサイクルのステージ）が状態，辺が状態遷移を表す遷移系を DEOS プロセスとする，とみなす素朴なアプローチは不可能である．

その理由は，一つのサービスに関して，複数のステージが同時に進行し得るからである．例えば，あるサービスに関する変化予兆を検知した後，その変化に対応するシステムの新版に関する合意形成および開発とシステムの現行版の通常運用は同時進行する．この場合，合意形成，通常運用が同時進行するステージである．また，あるサービスに関する障害対応が終了した後，説明責任遂行のプロセスと，再発防止策に関する合意形成は同時進行する．この場合は，説明責任遂行と合意形成が同時進行するステージである．

また，複数のサービスに関して，それぞれ別のステージが同時に進行する場合のことも，素朴なアプローチでは説明しにくい．例えば一つのサービスの障害対応と他のサービスの通常運用は同時進行することを素朴なアプローチで説明するのは困難である．

この困難を解決するため，先行研究[16][17]でDEOSプロセスの説明のために用いられたグラフをペトリネットのネット構造として解釈することとした．グラフの節をペトリネットのプレースとし，辺をペトリネットの遷移とする．ペトリネットのトークンによってシステムが提供するサービスを表す．プレースはサービスが満たすべきいろいろな要件を表す．つまり，トークンがプレースにある，ということによって，そのサービスがプレースによって表される要件を満たす，ということを表す．したがって，トークンがあるプレースから別のプレースに移ることは，サービスがある要件を満たす状態から，別の要件を満たす状態に移ることを表す．

例として，図 3.4-1 における OrdinaryOp と FailureDetected/Predicted を考えよう．プレース OrdinaryOp にサービス s を表すトークンがある，ということは

i. サービス s は正常に機能している
ii. s はモニタリングされている
iii. 障害対応の準備は整っている
iv. 直近の障害・変化対応の説明責任は果たされている

などなどの要件がすべて満たされていることを示す，と見なす．また，プレース FalureDetected/Predicted はにサービス s を表すトークンがある，ということは

v. サービス s に関する障害，あるいはその予兆が検知されている
vi. s の障害対応に必要な情報が得られている

が共に満たされていることを示す，と見なす．さらに，あるサービス s が OrdinaryOp から FailureDetected/Predicted へ遷移するのは，

s が i.，ii.，iii.，iv.を満たす状態から，s が v.，vi.を満たす状態に移る

ということを意味する．

以上のようなペトリネットによって DEOS プロセスをモデル化する．このモデルのもとで「オープンシステム・ディペンダビリティが達成されている」という主張を定式化し，この枠組に基づいてアシュランスケースを記することができることがわかった．

しかし，FFO がこの枠組をサポートするまで詳細化されているわけではない．

② 国際標準との比較対照

ペトリネットによる定式化に加え，DEOS プロセスの各ステージと，ISO 国際標準体系においてシステムライフサイクルプロセスに関する最上位標準と位置付けられている ISO/IEC/IEEE 15288[12]が定義するシステムライフサイクルプロセス（図 3.2-13）との関

連を明示したライフサイクルモデルを DEOS ライフサイクルモデル（図 3.4-2）として本プロジェクトで考案し，その結果を，国際標準 IEC62853 策定の委員会（IEC TC56 PT4.8）に提出して提案した．ISO/IEC/IEEE 15288 のシステムライフサイクルプロセス（6.4.X の番号が付されているもの）が DEOS プロセスの各ステージに割り当てられている．詳細は，[11]に記されているとおりであるが，概要を以下に記す．

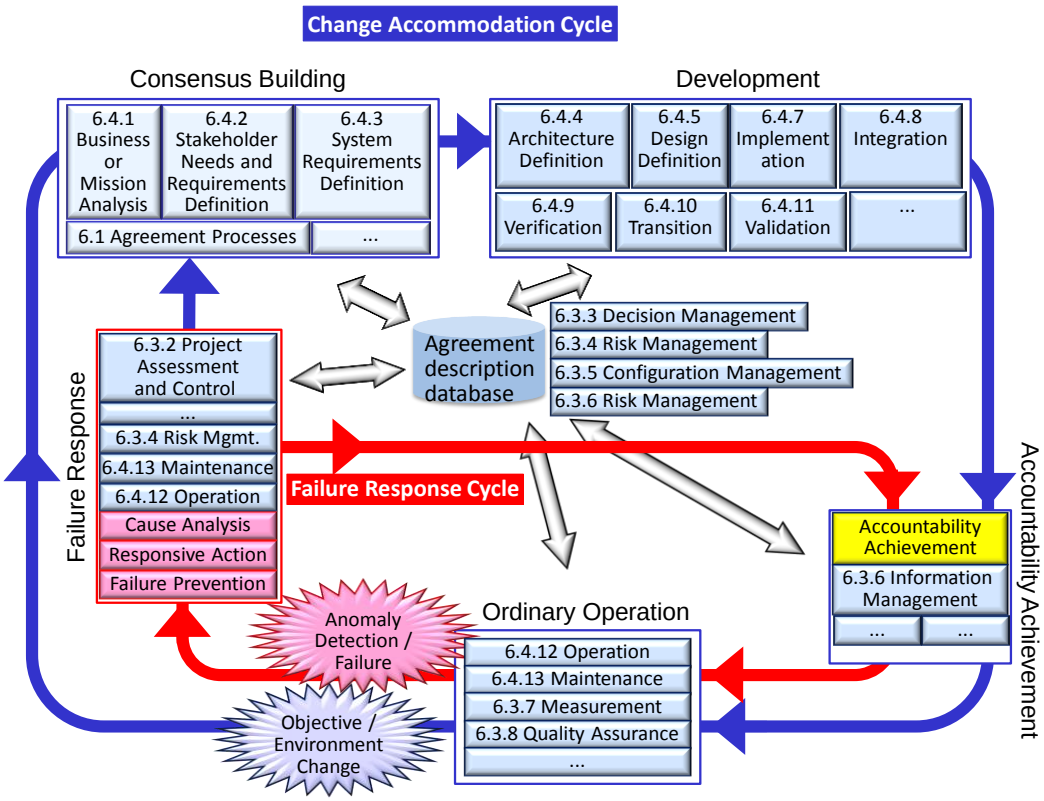

図 3.4-2 DEOS ライフサイクルモデル（[11], Annex B）

文献[17][16]で DEOS プロセスと呼ばれているものは，文献[11]では DEOS ライフサイクルモデルと呼ばれている．国際標準体系では「プロセス」の語が異なる意味に用いられているからである．以下では[11]Annex B に基づいて DEOS ライフサイクルモデルを紹介する．

DEOS ライフサイクルでは，システムの利害関係者として以下のものを考慮する．

- 提供されるサービスあるいは製品の利用者（システムがインフラストラクチャの一部である場合には，利用者は社会全体である）
- サービスあるいは製品の提供者
- サービスあるいは製品の認証者
- システムの提供者．これには以下が含まれる．
  - 設計者と開発者
  - 保守運用担当者
  - ハードウェアの提供者

DEOS ライフサイクルは五つのステージをもつ．合意形成ステージ，開発ステージ，説明責任遂行ステージ，通常運用ステージ，障害対応ステージである．また，これらのステージによって二つのサイクルが構成されている．変化対応サイクル（外ループ）と障害対応サイクル（うちループ）である．二つのサイクルとも，通常運用ステージから開始される．

各ステージは，次のライフサイクルプロセスから構成される．

- 合意形成ステージ：取得プロセス，供給プロセス，ビジネス・ミッション解析プロセス，ステークホルダニーズ・要求定義プロセス，システム要求定義プロセス
- 開発ステージ：アーキテクチャ定義プロセス，設計プロセス，実装プロセス，統合プロセス，検証プロセス，移行プロセス，妥当性確認プロセス
- 説明責任遂行プロセス：情報管理プロセス
- 通常運用プロセス：運用プロセス，保守プロセス，測定プロセス，品質保証プロセス
- 障害対応ステージ：プロジェクトアセスメント・制御プロセス，リスク管理プロセス，運用プロセス，保守プロセス

システムの目的，環境，性能などが変わるとシステムの変更が必要になる．このような変更が始まると，変化対応サイクルが開始される．このサイクルは，合意形成ステージ，開発ステージ，説明責任遂行ステージなどからなる．

システムの障害が発生すると，障害対応サイクルが開始される．このサイクルは，障害対応ステージと説明責任遂行ステージからなる．

障害の原因解析の結果によっては，障害対応サイクルが変換対応サイクルを開始する場合もある．これは同じ障害を繰り返さないためのシステム改修が必要になる場合である．

3.4.3 発生した課題および今後の展望

(1) 発生した課題

① ライフサイクルモデルへの時相導入

今回のペトリネットモデルでは，プレースがあらわす述語には時間の概念が自然な形では入っていない．しかし，オープンシステム・ディペンダビリティ達成の主張には，「そのうちに障害対応が終了する」あるいは「いつでも，システムの最新の状況に関するアシュランスケースが用意されている」など，時間の概念を含んだ主張が含まれている．時間の概念を時相論理の考えを使ってペトリネットモデルに導入することが必要であることがわかった．

② ライフサイクルモデルの標準化

IEC 62853 は，システムライフサイクルプロセスへの要件を規定するものであり，それらを組み合わせたライフサイクルモデルがどのようなものであるかを規定するには至っていない．本研究項目で我々が構築したペトリネットモデルをもとにして，オープンシステム・ディペンダビリティを達成するライフサイクルモデルへの要件を標準として制定する，という課題が発生している．

(2) 今後の展望

課題［(1)①ライフサイクルモデルへの時相導入］については，課題［3.1.3(1)③ペトリネットモデルの出現と反映］と同じく，時相論理による解決策が考えられるが，それによってモデルがどの程度，またどのように複雑になるのかは前もってはわからない．いずれにしろこの課題と課題 3.1.3(1)③とは連携して解決を図っていくのが適切であると考えられる．

課題［(1)②ライフサイクルモデルの標準化］については，課題(1)①の結果が出次第，IEC TC56 で国際標準制定作業を開始することができる．研究成果をすぐに国際標準制定につなげるのは，一般には容易なことではないが，本研究と先行研究（CREST/DEOS 利用者指向ディペンダビリティの研究）などにより，数年以上をかけてこのような人的体制を構築することができているのは特筆すべきことと考える．

4 考察

4.1 研究による効果や問題点等

本研究は，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ことを主張するアシュランスケースを書くためのツールを試作することが当初の目的であった．以下を当然のこととして（実施計画の想定仮説に明示せずに）想定して，本研究が開始された．

- アシュランスケースの書き方は，明らかである．さらに，
- オープンシステム・ディペンダビリティを要求するとはどういうことなのかが，すでに明らかになっている．

しかし，研究を開始すると，意外にも

- アシュランスケースの書き方についての課題が多数浮かび上がった．また，
- オープンシステム・ディペンダビリティがどういう属性なのか，明らかでない点が多々あった．

この結果，ツール試作と，オープンシステム・ディペンダビリティおよびアシュランスケースについてのより洗練された理解を得る作業が，相互に他を引き起こしつつ研究が進められた．研究による効果を考えるとき，アシュランスケースに関する知識が十分に広まっていない現状では，研究開始当初に目的としていたツールの存在そのものよりも， ツール試作を通して得られたオープンシステム・ディペンダビリティおよびアシュランスケースについての，より洗練された要件や概念が得られ，それが国際標準に反映されていることのほうが，産業を含めた社会に貢献する度合いが大きいと考えられる．

4.1.1 目標の達成程度

［到達目標 1 FFO 提供］は，FFO Basic Pattern 1 [15]として作成した．無償公開の予定であるが，現在神奈川大学内で知的財産公開の手続き中である．

［到達目標 2 オントロジー明示］については，FFO Basic Pattern 1[15]が提供するオントロジー（語彙定義）を明示することができた．

［到達目標 3 形式記述機構の提供］についても，FFO Basic Pattern 1[15]が提供する記述機構を提供することができた．

［到達目標 4 記述実験］について，車載システムに関しては実験そのものを行うことができなかったため，車載システムドキュメンテーションに関する業界標準に FFO に基づくアシュランスケースを加える案を提示し，協力企業から有効性評価を得ることとした．防災システムに関しては，災害時の給水業務についてのアシュランスケースを記述し，協力自治体から評価を受けた．しかし，当初想定していたような，対象技術領域の専門家（この場合自治体の関連部署職員）による FFO の詳細な有効性評価までは得られていない．0

［到達目標 5 国際標準との整合］について，FFO の現行版と IEC 62853 の最新草稿とのあいだには形式的なずれがあるものの，内容の整合性は確保されている．さらに，成果物（FFO）の現行版と国際標準の最新草稿との整合性にとどまらず，FFO の発展と国際標準の変化が整合的に進んでいく体制を整えたことは特筆に値すると考える．IEC 62853 の制定は本研究の研究チームからプロジェクトリーダおよびコンビナを出すなどリーダシップを確保しているからである．本研究の成果を社会に還元していくためには，このような長期的活

動が不可欠であり，研究プロジェクトの枠を超えた構想を支える研究支援体制が求められる．

［到達目標 6 トータルコスト半減］について．［到達目標 1 FFO 提供］により，システムのオープンシステム・ディペンダビリティ達成を主張するアシュランスケースの整合性検査や段階的詳細化が Agda システムを用いて計算機処理することが可能になった．このことと仮説 7 から，システムのオープンシステム・ディペンダビリティ達成を主張するアシュランスケースのライフサイクルにかかるトータルコストが半分以下に削減されるはずである．しかし，コストの実測には至っていないため，この到達目標は「仮説を前提として実現した」としかいえない．

4.1.2 達成できなかった目標とその理由

6 つの到達目標のうち，［到達目標 4 記述実験］以外は満足の行く程度に達成できたと考えている．

［到達目標 4 記述実験］については，二つの対象技術領域ともに，達成できなかった．

とくに，車載システムについては，実験に取り掛かることすら不可能であった．これには次のような理由があげられる．

- 対象領域の技術者に形式アシュランスケースを記述してもらうことがむずかしかった．アシュランスケースを理解している技術者ですら少人数でしかなく，本研究への協力グループの周辺には，記述実験に協力できる技術者がいなかった．
- アシュランスケースの技術がまだ広まっているとはいえない状況の中で，さらに特別な「形式アシュランスケース」の考えを対象領域の技術者に伝えるのは，長期間の協力作業が必要で，本研究のなかで割り当てることができる期間では短すぎた．
- 対象領域の技術者ではなく，我々アシュランスケースの研究者側がが実験的な記述をすることも考えたが，車載システムの業界は競争が激しく，アシュランスケース執筆に必要な機密情報に社外の者が触れるのは困難であった．また，対象領域の知識を得るのにも時間がかかるので，この方法をとったとしても，協力作業に必要な期間は短縮されなかったと思われる．

一方，防災システムについては，対象領域の職員ではないが，我々のチームの研究員がアシュランスケース記述実験を行うことができた．防災システムの性質上，機密情報はごく限られていることも幸いした．しかし，この記述実験の結果をもとに，FFO の有効性を対象領域の職員に評価してもらうことまではできなかった．そのような評価を下すには， FFO 以外の方法との比較が必要であり，アシュランスケースについてそこまでの知識を対象領域の協力者に要請することは，現状では無理であった．

［到達目標 4 記述実験］の不達成について総括すると，FFO の有効性評価のために記述実験を行うのは，当初想定したよりもはるかに大きな仕事であり，FFO の試作，開発のプロジェクトの一部分として簡単に行うことができるようなものではなかったと考えられる．また，アシュランスケースについての理解が十分に普及していない現状では，この到達目標の設定自体が，時期尚早であったと考える．

4.1.3 新たに見出された課題

(1) ライフサイクルモデルの発展

DEOS ライフサイクルモデルをペトリネットによる記述を含んだ形で発展させ，標準化する．

(2) FFO 普及・啓発

FFO による形式アシュランスケースを，形式論理学の知識を持たない者にも利用可能にするための，FFO 利用インターフェイスや教育メソッドの開発．

(3) 対象技術領域向けの標準・法律等

車載システムに関する本研究の知見をアシュランスケース執筆の社内標準あるいは業界標準に反映させる．また，防災システムに関する本研究の知見を活かして，防災計画のアシュランスケースを要求する標準あるいは法律などの制定につなげる．

(4) アシュランスケース執筆のトータルコスト

本研究では［到達目標 6　トータルコスト半減］は仮説 7 を前提として達成されたとしかいえていない．そこで，さらに仮説 7 自身の真偽を確かめたいところである．FFO の導入によってトータルコストが半減するかどうかの実測は将来の課題として残される．

4.1.4 他の類似研究と比べての特徴や優れているところ

(1) 構造化アシュランスケース

アシュランスケースは文書だから，もともとは英語や日本語の文章で記されるのみであった．しかし，アシュランスケースの文書構造を規定し，内容の理解をより容易にするために，図式を補助的に用いる構造化アシュランスケースの枠組みがいくつか提案され (Claims, Arguments and Evidence[1], GSN[14], D-Case[16] [17])，これらに適合するアシュランスケースの作成ツールも開発され，用いられている (ASCE[2], astah GSN[3])．

これらの構造化アシュランスケースに基づくツールと FFO を比べると，以下の得失がある．

- 整合性検査．アシュランスケースの内容に関する整合性検査は，構造化アシュランスケースでは困難だが，形式アシュランスケースで行うことができる検査がある．FFO はプログラミング言語 Agda を用いるので，構成的型理論に基づく強力な意味検査が可能である．
- 段階的詳細化．FFO に基づいて形式アシュランスケースを記すと，Agda システムが提供する機能を用いて，段階的詳細化を行うことができる．GSN などで未完のノードを表す◇に相当するプレースホルダを使用することができ，その詳細を後で埋めることができる．
- パラメータ化とモジュール化．構造化アシュランスケースのコミュニティでは，アシュランスケースのパラメータ化やモジュール化が大きな課題として研究されている．し

かし，これらはいずれも，2000 年までにプログラミング言語論の分野で大量の研究成果が得られたテーマである．プログラミング言語 Agda によって書かれる FFO 形式アシュランスケースには，Agda が提供する現代的かつ強力なパラメータ機構とモジュール機構を用いることができる．

(2) アシュランスケースの自動生成

データからアシュランスケースのエビデンス部分を自動生成しようとする試みが現れている（[4][5][6]）．これを発展させて，議論部分の一部を自動生成させることも可能であると思われる．しかしどのような議論でもすべて自動生成することは，理論的に不可能である．

一方，FFO は，自動生成せずに人間（あるいは機械）がアシュランスケースを書くが，書いたものの整合性を自動的に行う，というアプローチをとっている．

自動生成と FFO のアプローチは矛盾するものではなく，FFO に基づく形式アシュランスケースを自動生成することも十分に考えられる．FFO が重点を置くのはその整合性検査である．

4.1.5 論文発表等による外部の客観的評価

論文発表は未だ行われていないが，国際会議などにおける口頭発表は[2.2(2)③学会及び研究討論参加状況]に記した通りである．特に以下の発表は，国際会議への招待に基づいてなされたものであり，これらの発表がなされたこと自体が外部からの肯定的な客観的評価を表していると考える．

- Makoto Takeyama, “Formal Assurance Case in Agda (FACIA)”, Workshop on Logical Analysis of Descriptions and their Representations, NII Shonan Workshop, 2015.
- Yoshiki Kinoshita, “Formal Assurance Case”, Workshop on Logical Analysis of Descriptions and their Representations, NII Shonan Workshop, 2015.
- Shuji Kinoshita, “Towards Assurance Arguments of Local Disaster Management Plans”, ASSURE 2015, SAFECOMP Workshop, Delft, 2015.
- Yoshiki Kinoshita, “Open systems dependability standardization activity in IEC TC56”, WOSD (Workshop on Open Systems Dependability, IEEE ISSRE Workshop, 2015.
- Shuji Kinoshita, “Formal Assurance Case in Agda (FACIA)”, Demo, 2nd International Workshop on Argument for Agreement and Assurance, November 17, 2015 - Keio University.

4.2 国際標準化への貢献

［研究目標 4 FFO が依拠するシステムライフサイクル概念の確立］の活動として，オープンシステム・ディペンダビリティ，システムライフサイクルプロセス，およびアシュランスケースに関する国際標準化活動に参加した．オープンシステム・ディペンダビリティについては，IEC Technical Committee 56 (TC56) Dependability において，また，システムライフサイクルプロセスとアシュランスケースについては，ISO/IEC Technical Committee 1 (JTC1) Information Technology, SubCommittee 7 (SC7) Software and Systems Engineering において活動した．

4.2.1 IEC TC56 Dependability

この委員会では，主としてオープンシステム・ディペンダビリティを国際標準 IEC 62853 として制定する活動を行った．

(1) IEC TC56 WG4

IEC Technical Committee 56 (TC56) Dependability は本研究の対象であるディペンダビリティに関する国際標準を所掌する委員会であり，研究代表者はその Working Group 4 (WG4) Information systems aspects of dependability（情報システム関連のディペンダビリティ標準を所掌する）のコンビナ（convenor）を 2012 年以来務めている．また，TC56 と ISO/IEC JTC1 SC7（4.2.2 参照）の間のリエゾンオフィサー（双方向）をも務めている．

IEC TC56 は，製品標準よりも基本標準（[13]）とその周辺の概念標準（FTA や FMEA，Root Cause Analysis など）を中心に扱っており，企業間の営利的な論争は議論の中心には置かれないため，本研究の成果が IEC 62853 草稿作成に活かされただけでなく，草稿に関する PT4.8 における議論が本研究における問題を解決する，あるいは本研究では想定していなかった問題が指摘される場合もあるなど，この委員会での活動が本研究の活動に寄与した．

(2) IEC 62853 Open systems dependability

この WG に設けられた Project Team 4.8 (PT4.8) （プロジェクトリーダは研究代表者）において現在，国際標準 IEC 62853 Open systems dependability[11]の制定がすすめられている（2017 年発行予定）．

この国際標準は，ハードウェア，ソフトウェア，人的側面を含む製品，システム，プロセスあるいはサービスあるいはこれらの組み合わせに対して適用可能である．この国際標準はオープンシステムのディペンダビリティを改善するために用いることができる．また，オープンシステム特有のプロセスビューが，求められるアウトカムを達成することの確信（アシュランス）を与えるために用いることもできる．また，オープンシステムのディペンダビリティ目標を達成するために必要なアクティビティやタスク（ディペンダビリティに関する意志疎通，ディペンダビリティのアセスメント，システムライフサイクル全体にわたるディペンダビリティの評価などを含む）を組織（営利会社，政府，NGO，学校など）が定義する際の指針をこの国際標準は提供する．

草稿[11]は報告書執筆現在，Committee Draft 3 が回覧されている状態であり，未だ出版されていないが，本研究からの寄与を中心に，その内容の概略を説明する．

IEC 62853 はオープンシステム（範囲や機能，構造が時間とともに変化し，また，視点によってさまざまに認識されるシステム）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に，そのシステムライフサイクルプロセス（[16]に基づく）に要求される要件を規定するものである．オープンシステム・ディペンダビリティ達成のためには，次の 4 つの「プロセスビュー」（[16]のシステムライフサイクルプロセスを実体とする，バーチャルなプロセス）が正しく実現されていることが必要であるとし，それぞれのプロセスビューに対して，その outcome を規定している．

- 合意形成プロセスビュー
  要件定義及びその実現に関する合意形成

- 説明責任遂行プロセスビュー
  システムについての説明責任遂行

- 障害対応プロセスビュー
  障害への短期的対応

- 変化対応プロセスビュー
  故障への中長期的な対応，環境変化への対応など

そして，システムがこの標準に適合していることを主張するためには，これら 4 つのプロセスビューが規定された outcome を確かに達成することを議論するディペンダビリティケース（アシュランスケース）が用意されることが必要十分である，としている

4.2.2 ISO/IEC JTC1 SC7 Software and systems engineering

この委員会では，主としてシステムライフサイクルプロセスの国際標準 ISO/IEC/IEEE 15288 の改定における議論と最新草稿情報の収集およびアシュランスケースの国際標準 ISO/IEC 15026-2 の保守の活動を行った．

(1) ISO/IEC JTC1 SC7 WG7

ISO/IEC Joint Technical Committee 1 (JTC1) Information Technology SubCommittee 7 (SC7) Software and Systems Engineering はディペンダビリティの対象であるシステムライフサイクルプロセスに関する国際標準を所掌する委員会であり，研究代表者はその Working Group 7 (WG7) Life cycle management（システムライフサイクル管理に関する標準を所掌する）の国内委員を務めている．WG7 での活動の一環として ISO/IEC 15026-2 Assurance case のエディタを務め，現在はこの標準の保守を担当している．さらに，TC56 と ISO/IEC JTC1 SC7（4.2.2 参照）の間のリエゾンオフィサー（双方向）をも務めている．

ISO/IEC JTC1 SC7 WG7 も，IEC TC56 と同様に，製品標準よりも基本標準（[12]）とその周辺の概念標準を中心に扱っており，企業間の営利的な論争は議論の中心には置かれないため，この委員会での活動が本研究の活動に寄与した．

本プロジェクトの研究目標４「FFO が依拠するシステムライフサイクル概念の確立」達成で得られた成果を反映させて国際標準 ISO/IEC 15026-4 Assurance in the life cycle を改定することとなり，2016 年中に NWIP (New Work Item Proposal) を ISO/IEC JTC1 SC7 に提出する予定である．承認された場合，2019 年には同標準の改訂版が発行される見込みである．

4.2.3 FFO 研究と標準化活動の相互作用

(1) 適合性

FFO によって記述したアシュランスケースは，そのままの形で IEC 62853 への適合性を主張するためのディペンダビリティケースとして用いることができる．このような整合性が得られたのは，本研究と IEC TC56 における標準制定活動が連携して進められた結果である．

(2) システムライフサイクルプロセスとの関係の明確化

FFO が対象とするシステムライフサイクルプロセスは ISO/IEC/IEEE 15288[12]に規定されているものである．しかし，IEC 62853 を制定している IEC TC56 では，ディペンダビリティが対象とするシステムライフサイクルプロセスの定義は明確にされていなかった．[12]に基づく FFO の枠組みを IEC 62853 がとりいれたことから，IEC TC56 における他のディペンダビリティ標準でも[12]を参照する流れができた．この意味で，FFO 研究がディペンダビリティ標準におけるシステムライフサイクルプロセスの概念を収束させる素因を作ったと考える．

① 想定外の変化やニーズへの対処法

オープンシステム・ディペンダビリティを要求するうえで本質的に重要なのは，環境や要求の変化や，システムへのニーズや立場のすべてをあらかじめ想定しておくことができない，という前提に立たなければならないことである．

想定外の変化やニーズのもとでもディペンダビリティを担保する準備を要求することには，矛盾がある．想定外の変化に備える，ということは，その変化をあらかじめ想定していないとできないのではないかと考えられるからである．想定外の変化やニーズがないよう，あらゆる変化やニーズを考えておくのも不可能である．一方で，変化やニーズをできるだけ広い範囲であらかじめ考えておくことは求められる．しかし，「できるだけ広い範囲」をどのように規定すべきかは自明ではない．

本研究から生じたこの難しい問題について，IEC 62853 制定の議論において，次のような解決策が提出された．それは，「できるだけ広い範囲の変化やニーズを考えた」ということを主張するアシュランスケースをオープンシステム・ディペンダビリティ達成の主張者に提出させよう，というものである．どんな変化やニーズを考えたか，を議論するだけでなく，変化やニーズを考えだす過程が妥当なものである，という，メタな議論を付け加えることにより，ディペンダビリティのための考慮が必要な範囲で合理的になされたことを主張することができることがわかった．

本項目のような，オープンシステム・ディペンダビリティ達成の最も困難な点についての解決法が，FFO 研究と標準化活動の連携から生まれたことを特筆しておきたい．

4.3 産業界への展開と今後の研究の進め方

4.3.1 研究成果の産業界への展開

本研究では，オープンシステム・ディペンダビリティ達成の形式アシュランスケースを書くためのツール FFO を開発した．また，その過程で，オープンシステム・ディペンダビリティ達成のための要件を精査して 2017 年発行予定の国際標準 IEC 62853 とした．

アシュランスケースの技術が広まっていない現状では，IEC 62853 の策定が，本研究の成果のなかでもっともはじめに寄与するものと考えられる．この標準が利用されることにより，適合性のために要求されているアシュランスケースがより広く使われるようになれば，アシュランスケースの執筆や認証の効率化が課題として認識され，本研究の本来の目的である形式アシュランスケースの価値が産業界の中でも広く理解されるようになる．車載シ

ステムの機能安全に関する ISO26262 が果たした役割を，一般的システムのオープンシステム・ディペンダビリティに関して IEC62853 が担うものと考えられる．電気・電子機器の機能安全に関する IEC 61508 は，アシュランスケース（安全ケース）を要求しているわけではないが，一般のシステムのドキュメンテーション一般について，これと類似の役割を果たした．

しかしながら，アシュランスケースの普及が 10 年以上先行している欧州では，すでにアシュランスケースの整合性や執筆の効率化の問題が広く理解されており，4.1.5 に記したように，本研究の方向の価値も認められている．我が国の産業界でのアシュランスケースへの理解が欧州の現状のレベルに達するまでに数年程度の時間が必要であると考えられるが，その間に，次節に記すような形式アシュランスケースの研究を進めることによって，欧州に対して後の先をとることができると考える．

4.3.2 今後の研究の進め方

残された到達目標で最も大きいものは，対象技術領域の専門家が FFO を使って形式アシュランスケースを記述する［到達目標 4 記述実験］であった．4.1.2 に記したように，この到達目標設定は時期尚早であったと考えられる．本到達目標に再び挑戦するには，産業界におけるアシュランスケースへの理解が経営層を含めて全体的に十分浸透し，対象技術領域の専門家が FFO を使う環境が整うまで，時期を待つ必要がある．

新たに見出された課題には以下のようなものがある．

1. DEOS ライフサイクルモデルをペトリネットによる記述を含んだ形で発展させ，標準化する．
2. FFO による形式アシュランスケースを，形式論理学の知識を持たない技術者などにも利用可能にするための，FFO 利用インターフェイスや教育メソッドの開発．
3. 車載システムに関する本研究の知見をアシュランスケース執筆の社内標準あるいは業界標準に反映させる．
4. 防災システムに関する本研究の知見を活かして，防災計画のアシュランスケースを要求する標準あるいは法律などの制定につなげる．

4.3.3 産業界への要望

オープンシステム・ディペンダビリティの研究に限らず，現代の大規模で複雑なシステムの問題に対しては，いわゆる技術的な観点だけではなく，管理，経営，倫理などから総合的に取り扱う観点が必要である．産業界から官界，学界へ働きかけて，このような総合的研究活動のイニシアティブをとっていただきたい．

謝辞

AFSCF 議論モデルの研究にあたって（株）デンソーからご協力をいただいた．また，6W1H モデルと DPP の研究にあたって平塚市防災危機管理部災害対策課からご協力をいただいた．ここに記して深甚の感謝の意を表する．

参考文献

[1] Adelard, Safety Case Structuring: Claims Arguments and Evidence, http://www.adelard.com/services/SafetyCaseStructuring/

[2] Adelard, ASCE, http://www.adelard.com/asce/choosing-asce/cae.html.

[3] チェンジビジョン, http://astah.change-vision.com/ja/product/astah-gsn.html

[4] Denney, E., Pai, G.: A lightweight methodology for safety case assembly. In: Proc. 31st Intl. Conf. Comp. Safety, Reliability and Security (SafeComp). pp. 1-12. (Sep 2012)

[5] Denney, E., Pai, G., Pohl, J.: Automating the generation of heterogeneous aviation safety cases. Tech. Rep. NASA/CR-2011-215983, NASA Ames Research Center (Aug 2011)

[6] Denney, E., Pai, G., Pohl, J.: AdvoCATE: An Assurance Case Automation Toolset. In: 31st Intl. Conf. Comp. Safety, Reliability and Security Workshops. pp. 8-21. (Sep 2012)

[7] ISO 26262 Road Vehicles -- Functional Safety. ISO Standard (2011)

[8] (社)JASPAR, 機能安全テンプレート, (2013), URL: https://www.jaspar.jp/outcome/1307_index.html

[9] J.-C. Laprie, From Dependability to Resilience, The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) 2008.

[10] 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.

[11] IEC 62853 Open systems dependability (unpublished Committee Draft).

[12] ISO/IEC/IEEE 15288:2015 Systems and software engineering - System life cycle processes

[13] IEC 60300-1:2014 Dependability management - Part1: Guidance for management and application.

[14] Origin Consulting Limited, GSN Community Standard, Version 1, York, 2011.

[15] M. Takeyama, FFO Basic Pattern 1. 神奈川大学サイトより無償公開の手続き中.

[16] M. Tokoro (ed.), Open systems dependability, CRC Press, 2nd edition, 2015.

[17] 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.

[18] 平塚市地域防災計画, 平塚市防災会議, URL: http://www.city.hiratsuka.kanagawa.jp/bousai/plan.htm (2015)

[19] 災害対策基本法 最終改正：平成 27 年 9 月 11 日法律第 66 号 URL: http://law.e-gov.go.jp/htmldata/S36/S36HO223.html