ま　え　が　き

本書は安全分析、アシュアランスケース、モデルベースシステムズエンジニアリングを解説した本です。

2022 年 10 月

松野裕, 高井利憲, 岡本圭史

1 イントロダクション

1.1 システムのディペンダビリティ：基礎概念と現代的課題

1.1.1 システムとは何か

私たちの日常生活において、「システム」という言葉をよく耳にします。しかし、この概念を正確に定義するのは簡単ではありません。システム工学の観点から見ると、システムとは「私たちの身の回りにある、何らかのサービスを提供する物や人を抽象化した概念」と定義できます。

システムは常に特定の環境に置かれており、その環境との相互作用を通じて機能します。例えば、自動車というシステムは道路という環境の中で機能し、気象条件や交通状況などの環境要因の影響を受けます。

システムが提供する「サービス」とは、システムによって私たちに提供される機能や便益のことを指します。私たちユーザーは、このサービスを通じてシステムと関係を持ちます。例えば、スマートフォンというシステムは、通話、メッセージング、インターネット閲覧などのサービスを提供し、私たちはこれらのサービスを通じてスマートフォンと関わっています。

1.1.2 ディペンダビリティの概念

ディペンダビリティ（Dependability）は、直訳すると「依存可能性」となりますが、システム工学では「総合信頼性」と訳されることが多い重要な概念です。この概念の起源は、あるシステムが別のシステムに依存（Depend）するこ

とができる、そのシステムの性質を表すことにあります。

例えば、運転手（システム A）が自動車（システム B）に依存する場合を考えてみましょう。自動車がディペンダブル（依存可能）であるためには、どのような性質を持つべきでしょうか？安全であること、運転しやすいこと、目的地まで確実に到達できることなどが挙げられるでしょう。このように、ディペンダビリティは利用者や環境によって求められる性質が異なる、相対的な概念です。

システムがディペンダブルであるためには、以下の条件を満たす必要があります：

- 利用者や環境において望まれる性質を持ち続けること
- サービスを継続的に提供すること

さらに、システムがその性質を失った場合（つまり、ディペンダブルでなくなった場合）には、速やかに復旧を行い、サービスを継続する能力も求められます。

1.2　ディペンダビリティの体系と用語

ディペンダビリティの概念は、1980 年代から国際的な研究グループ（IFIP WG 10.4 ”Dependable Computing and Fault Tolerance”など）によって体系化され、用語の整理が行われてきました。当初は「耐故障性 (Fault Tolerance)」研究から発展し、近年ではセキュリティの概念も含めて議論されています。

ここでは、Avizienis et al. (2004) による体系に基づいて、ディペンダビリティの主要な概念を紹介します。

1.2.1　ディペンダビリティ属性

ディペンダビリティ属性は、システムがディペンダブルであるために持つべき特性を表します。主な属性には以下のものがあります：

- 可用性（Availability）：正しいサービスの即応性

- 信頼性（Reliability）：正しいサービスの継続性
- 安全性（Safety）：利用者と環境へ破壊的影響をもたらさないこと
- 一貫性（Integrity）：不適切なシステム変更がないこと
- 保守性（Maintainability）：変更と修理を受け入れられること

これらの属性は相互に関連しており、時には相反する関係にあることもあります。システム設計者は、対象システムの要求に応じてこれらの属性のバランスを取る必要があります。

1.2.2　ディペンダビリティへの脅威

システムのディペンダビリティを脅かす要因は、以下の 3 つの概念で整理されています：

- 欠陥（Fault）：エラーの原因となるとみなされる、あるいは推定されるもの、こと
- 誤り（Error）：障害が起こりうるシステムの状態（ただし、エラー状態になったからといって、必ずしも障害が起こるとは限らない）
- 障害（Failure）：サービスが正しいサービスから逸脱する出来事

これらの概念は因果関係にあり、欠陥がエラーを引き起こし、エラーが障害につながる可能性があります。

1.2.3　ディペンダビリティへの脅威に対処する手段

ディペンダビリティへの脅威に対処するため、以下の 4 つの手段が提案されています：

- 欠陥防止（Fault Prevention）：欠陥の導入や発生を防ぐ
- 耐故障性（Fault Tolerance）：欠陥がある中で障害を防ぐ
- 欠陥除去（Fault Removal）：欠陥の数や深刻度を減らす
- 欠陥予測（Fault Forecasting）：欠陥の現在の数、今後の障害、影響などを予測する

これらの手段を適切に組み合わせることで、システムのディペンダビリティを

向上させることができます。

1.3　現代のシステムとディペンダビリティの課題

1.3.1　情報システムの規模拡大とネットワーク化

近年、情報システムは急速に規模を拡大し、ネットワーク化が進んでいます。この変化は以下のような段階を経ています：

- 単機能システム
- 複合機能システム
- ネットワーク化されたシステム
- サービスポータル化されたシステム

この変化に伴い、IT システムは生活・社会インフラとしての役割を担うようになり、組込みシステムもポータル化が進んでいます。これにより、システムの複雑さと重要性が増大し、ディペンダビリティの確保がより困難かつ重要になっています。

1.3.2　大規模なシステム障害のリスク

システムの大規模化と複雑化に伴い、以下のような要因により大規模なシステム障害のリスクが高まっています：

- プログラムサイズの増大と多機能化
- ブラックボックス化したコンポーネントの増加
- 技術進化のスピードの加速
- 接続システムの多様化
- 利害関係者の変化と要求の頻繁な変更

これらの要因により、システムの完全な理解と制御が困難になっています。

1.3.3　オープンシステム（開放系）の問題

現代のシステムは、多くの場合オープンシステム（開放系）として設計され

ています。これにより、以下のような新たな課題が生じています：

- 仕様/実装の不完全さ：要求、仕様、設計、実装、テストの各段階での不完全さが避けられない
- システムの完全理解の困難さ：構成要素の論理的不透明さ（複雑化、巨大化、ブラックボックス化）により、システム全体の挙動予測が困難
- 使用環境の変化に伴う不確実さ：要求事項・レベルの変化、想定外の使われ方、ネットワークを介しての構成要素の変化
- セキュリティリスクの増大：ネットワークを介した外部からの意図的な攻撃のリスク

1.4　これからのディペンダビリティに向けて

1.4.1　不完全・不確実なシステムへの対応

従来の形式手法やテストなどの手法に加え、不完全・不確実なシステムがディペンダブルであるための新たなアプローチが必要とされています。しかし、完全に障害を排除することは不可能であり、深刻な障害が起こる可能性は以前よりも高まっています。

1.4.2　システム保証（System Assurance）の重要性

このような状況下で重要になってくるのが「システム保証」の概念です。システム保証とは、システムがどの程度ディペンダブルか、あるいはディペンダブルでないか、リスク分析などをもとに利用者などのステークホルダーに説明し、納得してもらうプロセスです。

絶対に安全である、あるいは完全にディペンダブルであることは不可能であるという事実を、ステークホルダーに理解してもらうことが重要です。

1.4.3　説明責任の必要性の高まり

近年、システムの安全性や信頼性に関する説明責任の重要性が高まっていま

す。例えば、2009-2010 年のトヨタ プリウスの北米大規模リコール問題では、原因説明への準備不足が指摘されました。

また、自動車の機能安全規格である ISO 26262 の制定により、自動車メーカーはより説明しやすい形で安全性の根拠を示すことが求められるようになりました。

1.4.4　AI 技術とディペンダビリティ

最近では、AI（人工知能）技術を用いたシステム、特に自動運転システムなどのディペンダビリティが重要な課題となっています。AI 技術の不確実性や説明可能性の問題は、従来のシステムとは異なる新たなディペンダビリティの課題を提起しています。

システムのディペンダビリティは、もはや単なる技術的な問題ではなく、社会的、倫理的な問題としても捉えられるようになっています。システムの複雑化、不確実性の増大、そして AI 技術の台頭により、従来のディペンダビリティの概念や手法だけでは不十分になってきています。

これからのディペンダビリティ確保には、技術的な対策に加えて、システム保証と説明責任の遂行が不可欠です。また、不完全性や不確実性を前提としたシステム設計と運用の考え方を確立していく必要があります。

システム開発者、運用者、そして利用者を含むすべてのステークホルダーが、これらの課題を理解し、協力してディペンダブルなシステムの実現に取り組むことが求められています。

2 安全分析の基本手法: FTA と FMEA

2.1 未然防止の手法

障害への対応法は、応急処置、再発防止、および未然防止がある。応急処置、再発防止は発生した障害に対応する事後解析であり、リアクティブな方法（Reactive Approach）と呼ばれる。未然防止は、障害が発生してから対策を取るのではなく、計画段階や設計段階の生産の源流において、将来起こりうる障害を洗い出して、それらに対策を講じてしまうことである。潜在的な障害に対応する事前解析であり、プロアクティブな方法 (proactive approach) といわれる。未然防止の手法には、FMEA(Failure Mode and Effect Analysis), FTA(Fault Tree Analysis), ETA(Event Tree Analysis), 良品解析などがある。信頼性工学や安全性工学では良さ加減を増強するよりも、悪さ加減を減少させる方法がとられることが多い。

未然防止技術は多くある（多変量解析、品質機能展開、品質工学、実験計画法、WCA、FEM、信頼性試験、故障解析、信頼性データ解析、リスク・アセスメント、ライフサイクル・アセスメント（LCA））。本講義では、最も代表的な FTA, FMEA を扱う。

2.2 FTAとFMEA

FMEA, FTA は対象とするシステム（製品、設備、プロセスなど）の故障、不具合、欠陥などの「悪さ加減」を論理的に洗い出して、内在する問題点を発見する解析手法である。見出された問題点に対しては、実際の技術活動や管理活動を通じて対策や是正措置が取られる。

FTA と FMEA の形態を示す。図図 2.1、図図 2.2 は、工作用の洋ばさみについての FTA と FMEA である。(a) ははさみの部分と名称を示したもので、

図 2.1　FTA の形態（工作用はさみの事例）

内刃、外刃、留ねじの 3 つの部品からなる。(b) ははさみに対して実施した設計 FMEA を示す。部品の故障モードを洗い出して、システム（この場合ははさみ）への影響を表形式で解析する手法である。原因系から結果系を予測する方法と言える。(c) は、はさみで紙が切れないという結果の事象を取り上げて、

図 2.2　FMEA の形態（工作用はさみの事例）

その原因を探る FTA の一部を示す。FTA は木構造の解析手法である。主な論理ゲートは AND ゲート (論理積) と OR ゲート (論理和) である。

一般にシステムはサブシステムに、サブシステムはコンポーネントに、というように、順次、下位の構成要素に分解される。最終的にこれ以上分解できないレベルにいたる。システムとしては、階層的に分解できる構造であれば、ハードウエアでも、ソフトウエアでも、プロセスでも、あるいはそれらの複合でもよい。FMEA は下位の階層の悪さ加減が上位の階層の悪さ加減にどのように影響するかを表形式上で論理的に解析する手法である。FTA は逆に上位の階層の悪さ加減の原因となる下位の階層の悪さ加減を木構造で論理的に解析する手法である。FMEA は単一の原因が及ぼす複数の結果を網羅的に洗い出すのに優れ

た手法であるが、複数の原因によって及ぼされる結果の洗い出しは難しい。一方、FTA は複数の原因によって及ぼされる結果も表現できるが、網羅性が十分ではない。両者は相補的かつ相乗的に用いられている。

2.3 FTA(Fault Tree Analysis、故障の木解析)

FTA は、「なぜなぜ」を繰り返すことで、重大な故障やトラブルの発生要因を下方に向かって木構造として展開し、網羅した要因の中から重要な要因を抽出する。1979 年にスリーマイル島で発生した原子力事故の解析の際、マサチューセッツ工科大学教授の Rasmussen が原因の特定に使用したことでその有効性が評価され広まった手法である。

2.3.1 FT 図を読む

FT 図は、視覚的に故障に至るメカニズムが大変わかりやすく表現されている。FT 図の記号は、イベントを示す事象記号と、それらの間の因果関係を示す論理記号とに分けられる。表表 2.1 にある 4 つの記号の意味を理解できれば、ほとんどの FT 図を読むことができる。さらに、表表 2.2 のノードも用いられ

表 2.1 FTA で用いる基本的な記号

る。FT 図の例を図図 2.3 に示す。トップ事象である欠報の原因には「火災の

表 2.2 FTA で用いる便利な記号

検知信号が届かない」不具合と「ブザーが鳴らない」不具合とがあり、いずれか一方の原因で欠報に陥ることが示されている。さらに、その 2 つ以外に原因がないこと、あるいは他の原因は無視してよいことを示している点が重要である。この必要十分性は、熟練者でもしばしば見落とす点である。一方の AND

ゲートは、すべての下位事象が同時に発生するときに上位事象が発生することを示す記号で、並列型に対応する。「火災の検知信号が届かない」事象は、冗長に設置された２つのセンサ系 A,B が同時に故障しているときのみ発生する事象となる。FTA には、

図 2.3　FT 図の例

- トップ事象と基本事象との因果関係が視覚的に示され、多様な事象間関係を把握しやすい。
- AND ゲートにより多重故障を解析できる。

などの特徴がある。

2.3.2　FT 図 の 作 成

FTA の実施は、FT 図の作成と FT 図の解析の２つのステップで構成される。ここでは作成までの手順を説明する。

1. FTA 実施の準備。対象製品を熟知する技術者と品質保証担当者を含めた 3 名から 6 名程度のメンバーで実施する。設計資料、図面、材料部品リストや想定使用状況、関連するトラブル、クレーム情報、特に不具合に関する情報を集める。
2. 解析対象の機能の理解。FTA で解析する対象製品の構造や機能について、参加者全員で十分理解する。自分の専門とする部分や分野に対象を限定することなく、周辺との関係なども理解することが重要である。
3. トップ事象の選定。信頼性や安全性を損なうような「発生することが望ましくない」トップ事象を注意深く選定する。その際、1. 明確に定義できる事象、2. 多くの下位事象の結果として発生する事象、3. 設計の中で技術的に対処できる基本事象が予想される事象、であることが望ましい。1 が最も重要な要件であり、「明確」なとは、その事象発生の有無の判断が人によって異なることはない、との意味である。例えば「エアバッグが開かない（不動作故障)」、「エアバッグが不要のときに開く（誤作動故障)」など

は、トップ事象としてふさわしい。しかし「***の満足度が低い」、「***の回転が不安定」などは、その範囲（基準値）が不明瞭であり適さない。また一見良さそうに見える「排気ガス規制の基準値を満たしていない」のような表現も避けるべきである。ガスの種類や基準値は時代と共に変化するので、「排気ガス CO の規制基準値**ppm を満たさない」などの具体的な基準値を明記する必要がある。2. は、FTA 解析はマンパワーが必要となるので、できるだけ重要なトップ事象を扱う、という意味である。3. は、設計時に、FT 図作成に関わる技術者が基本事象まで書ききることができるようにするためである。

4. トップ事象の 1 次要因への展開。トップ事象の 1 次要因を、製品の構造や機能、手順 1 で準備した情報などを基に列挙し、論理記号を用いて因果関係を明確に図示する。

   展開する方法は大きく分けて 2 つある。

   (a) 構造（信頼性ブロック図）からの作成。あるシステムが信頼性ブロック図で構造が示される場合、直列系の部分を OR ゲートに、並列型部分を AND ゲートに対応させることで、FT 図を容易に作成できる。

   (b) 機能を考えて作成。実際には、信頼性ブロック図を基に FT 図全てを作成できるケースは多くない。構成要素の機能に着目して、トップ事象の直接の原因である 1 次要因を抽出し、さらにそれらの原因である 2 次要因を抽出するという具合に、意味を考えてトップダウンに作成することになる。

   1 次要因への展開は、最も頭を悩ませるステップだが、重要な箇所であり、時間をかけるべき手順である。システムを構成するサブシステムごとに空間的に分割し、それぞれを解析するとの方針がとられることが多いが、それよりも、エネルギーの流れに注目するなど、機能的な側面から 1 次要因を分解すると、装置間の相互作用などを見失うことが少なく、効果的な木になることが多い。

5. トップ事象の 2 次要因以下への展開。展開可能な 1 次要因に関して、さ

らになぜなぜ分析を続け、2 次要因、３次要因を列挙、基本事象または非展開事象に至るまで論理記号を用いて展開する。

例題 図図 2.4 に示されるような、２つのセンサが並列に設置された自動照明器で、「照明が点灯しない」をトップ事象とする FT 図を作成せよ。

図 2.4　照明設備の回路図と信頼性ブロック図

手順 5 までで FT 図ができあがるが、効果的な FT 図を作成するためのコツがある。

i)　事象発生の有無が明確な表現にすること。トップ事象にかかわらず、中間事象や基本事象でも、事象発生の有無が人によりかわることのない、明確な表現にする。「**が弱い」、「**が不安定」などの表現はさけるべきである。

ii)　基本事象では、事象を一意に特定できるように表現すること。基本事象レベルでの「接点故障」などの表現は不適切であり、それらの状況により対策が異なる。基本事象では、FT 図の作成者に聞くことなく、その状況を一意に把握できる表現まで分解することが必要である。

iii)　必要で十分な要因を漏れなく列挙しながらトップダウンで作成すること。各レベルでの要因抽出時に、思いつく要因の候補をあげればよい、という発想という発想では適切な FT 図は得られない。特に OR ゲートの下位事象では、必要十分な要因を漏れなく列挙することが求められる。

iv)　横のレベルをそろえながら分解すること。自分の得意な部分になると、いきなり詳細の部位の不具合要因がならぶことがよくある。1 次、2 次、とトップダウンで作成するが、それぞれのレベルに並ぶ事象の階層を合わせることが、見やすい FT 図を作るコツである。

2.4 FT 図で定量的に解析する

FT 図を作成すると、トップ事象を発生させる要因、およびその発生経路が明らかになる。それらを基に、解析のステップに移ることができる。

手順 6 では、トップ事象に対して改善効果が高いことが見込まれる基本事象を抽出するために、定量的評価法または定性的評価法を適用する。定量的評価法では、トップ事象の発生確率を求め、さらに、その確率を下げるためにはどの基本事象の発生確率を下げることが効果的かを特定する。そのためには、すべての基本事象の発生確率がわかっていることが前提となる。

手順 7 では、手順 6 で抽出された重要要因について、対策事項、対策方法を決定し、担当部署を決める。そして、対策実施後のフォローアップを確実に行う。

手順 6 におけえる定量的評価法では、トップ事象の発生確率を求め、さらにその確率を下げるためにはどの基本事象の発生確率を下げることが効果的かを特定する。そのためには、すべての基本事象の発生確率がわかっていることが前提となる。

(i) トップ事象の発生確率。トップ事象の発生確率は、基本事象の発生確率から容易に推定することができる。論理記号に着目して、次の方法で上位事象の確率を算出していく。

(a) OR ゲート $\rightarrow$ 下位事象の発生確率の和

(b) AND ゲート $\rightarrow$ 下位事象の発生確率の積

例えば、図図 2.5(b) で基本事象である各パソコンの故障の確率を 0.001, プリンタの故障確率を 0.0001 とする。トップ事象の発生確率を推定する場合、パソコン故障の発生確率は、AND ゲートであるから、

$$Pr = 0.001 \times 0.001 = 1.0 \times 10^{-6}$$

となり、トップ事象の発生確率は、OR ゲートで結ばれているため、

$$Pr = 1.0 \times 10^{-6} + 1.0 \times 10^{-4} = 0.000101$$

となる。以下に注意したい。

- 事象の発生確率とは不信頼度 F のことであり、故障率ではない。故障率は単位時間当たりの故障発生回数で定義されるので、OR ゲートの場合は各構成要素の故障率の和で定義できるが、AND ゲートでは面倒な計算が必要となる。
- OR ゲートでの計算方法は近似式である。
- AND ゲート、OR ゲートでの計算の独立性が仮定されている。

(ii)　同一事象の排除。FT 図においては、同一の基本事象、中間事象が 2 箇所に現れても構わない。ただし、定量的解析を行う場合、同一事象を一つにまとめた FT 図に変形してから解析を進めなければ、誤った結果を導いてしまうことに注意すべきである。同一事象が複数含まれる場合には、事象間の独立性が失われるからである。図図 2.5 では、Tree としては同値だが、トップ事象の発生確率を求める場合、同一の基本事象を一つにまとめた (b) 図を用いなければならない。より深刻な例として、2000 年 12 月の京

図 2.5　同じ意味の 2 つの FT 図

福電鉄の正面衝突事故がある。常用ブレーキも非常ブレーキも効かなかった事故であるが、通常、常用と非常用のブレーキは冗長化構造になっており、図図 2.6 の (a) の FT 図が想定される。並列の 2 つのブレーキがあるにもかかわらず同時に故障に陥る確率は、1 次要因が AND ゲートで結ばれているため、積により小さな値になることが容易に想像できる。しかしこの車両のブレーキ装置では、常用と非常用で同一のブレーキレバーの角度の違いで操作する形式であり、ブレーキ制御部は同じものを使用していた。二重化されていたのは、圧縮空気の送風装置だけである。このため (b) の FT 図でトップ事象の発生確率を算出する必要があった。

図 2.6　常用ブレーキと非常ブレーキの多重故障の解析

(iii) トップ事象に大きな影響を与える基本事象の抽出。トップ事象の発生確率が推定されると、次に、どの基本事象の発生を抑えることがトップ事象の発生確率を下げるために効果的かを知ることができる。OR ゲートの場合、トップ事象の発生確率は（重複事象がなく独立性が成立していれば）基本事象の発生確率のトータル和となるため、発生確率の最も高い基本事象から対策を講じれば良い。しかし AND ゲートが含まれると容易ではない。

2.5 FT 図で定性的に解析する

多くの場合、基本事象の発生確率を推定することは難しい。基本事象の発生確率を前提とせず、トップ事象への影響の大きな基本事象を特定するための方法として、最小カット集合を利用した方法と構造重要度を利用した方法がある。

2.5.1 最小カット集合を利用した方法

最小カット集合とは、トップ事象を発生させ得る最小の基本事象の組み合わせである。例えば図 2.7 の Tree では、$\{A, B\}, \{A, C\}$ の 2 つが最小カット集合になる。この時、最小カット集合に共通の基本事象 A が存在することから、A の発生を確実に抑えることができれば、トップ事象を回避することができる。すべての最小カット集合に共通の基本事象が存在するとは限らないが、一般には、できるだけ少ない基本事象の組み合わせで、すべての最小カット集合の発生を防止できるような組み合わせを探し出し対策を講じることで、トップ事象を防ぐことが可能となる。例えば最小カット集合が $\{A, B\}, \{B, C, D\}, \{B, E, F\}, \{D, E, F, G\}, \{G, H\}$ ならば、B と G の 2 つの事象に確実な対策を施せばトップ事象を回避することができる。最小カット集合を得るためには、FT 図の AND ゲートを積、OR ゲートを和で表し、ブール代数を用いて積で表した項の和 (加法標準形) で全体を表現できれば、それら各項が最小カット集合である。ブール代数において、0 は事象未発生、正常、1 は事象発生、故障とする。図図 2.7 では

$$T = A \cdot (B + C) = A \cdot B + A \cdot C$$

となり、$\{A, B\}$, $\{A, C\}$ の 2 つの最小カット集合が得られる。

2.5.2 構造重要度を利用した方法

構造重要度は、一つの基本事象に着目し、その事象が正規した時にトップ事象が発生する割合 (他の事象の組み合わせ増加数) を表す。図図 2.7 の例で説明する。

図 2.7 FT 図とその基本事象とトップ事象

(i) 最小カット集合を求める。$\{A, B\}, \{A, C\}$ が最小カット集合である。

(ii) 最小カット集合に基づき、基本事象とトップ事象との関係を表す真理表を作成する。$A = B = 1$(故障)、$A = C = 1$ のとき $T = 1$ とすればよい。

(iii) A に関する構造重要度 $I_S(A)$ を下記で求める。

- $X =$ 事象 A が故障時のトップ事象発生の組み合わせ数、
- $Y =$ 事象 A が正常時のトップ事象発生の組み合わせ数としたとき、

$$I_S(A) = \frac{X - Y}{2^{n-1}},$$

すなわち $I_S(A) = \dfrac{3-0}{4} = \dfrac{3}{4}$ となる。また $I_S(B) = I_S(C) = \dfrac{2-1}{4} = \dfrac{1}{4}$ となり、A の構造重要度は B,C のそれよりも 3 倍となる。よって A を重点的に対策を取れば良いといえる。すべての基本事象の構造重要度を算出することで、トップ事象の回避への寄与度を知ることができるが、この値の算出は意外と面倒である。FTA 解析用のソフトウエアでは自動的に計算され便利である。

2.6 FTA 実施上の留意点

FTA を実施する際に注意すべき点は下記の通りである。

(i) 全ての基本事象の発生確率が必要であること。

(ii) 創発故障への対応。部品間の相互作用による創発故障を見落とさないためには、１次の分解で、構造ではなく、機能に着目して分解することに注意するとよい。

(iii) 動的変化への対応。FTA は基本的に静的な解析であり、動的変化を解析しにくい。トップ事象の発生確率も動的に変化するため、一定期間後のある時点での値で評価することに限定される。

(iv) 事後解析での活用。未然防止での利用を念頭に説明してきたが、故障解析や事故解析などの事後解析では確率値を利用した解析は行わない。実際に発生している真の原因を、多数の可能性がある要因の中から絞るために FT 図を利用するものであり、最小カット集合の中から、新の発生要因を絞り込むことが可能になる。

参 考 文 献

本資料は「システムの信頼性と安全性、田中健次、朝倉書店」、「新 FMEA 技法、益田昭彦、高橋正弘、本田陽広、日科技連」を基にしている。

3 安全分析手法STAMP/STPA

4 モデルベースシステムズエンジニアリング

5 アシュアランスケースとGSN

人工知能の研究開発は加速的に進み始めている。2022年に登場したChatGPTは、誰でも簡単にウエブ上で質問をすることができるチャットボットであるが、その回答の詳細さと自然さに、多くの人が驚いた。また歩行者や標識を自動認識する人工知能を持つ自動運転車は、アメリカのWaymo社や日本のTier4社など、多くの企業が開発競争を繰り広げており、アメリカではすでにカリフォルニア州において自動運転タクシーが実用化されている。しかしながら、チャットボットや自動認識を行う人工知能は、100%正しい出力をするわけではない。であるにも関わらず、その圧倒的な利便性から、人工知能が組み込まれたシステムが加速的に普及していくことはより確実になってきている。そのような状況おいて、我々を取り巻くシステムが安心して利用できるものなのか、改めて社会的な合意形成が必要な時期になっている。

5.1 アシュアランスケースとは

アシュアランスケース（Assurance Cases）は、システムまたは製品の特性（安全性、セキュリティ、信頼性など）に関して、構造化された議論を明示的に示すドキュメントです。このドキュメントは、最上位の主張を下位の証拠および前提条件に結びつける形で構成されます。

特に安全性に焦点を当てたアシュアランスケースは、セーフティケース（Safety Case）と呼ばれます。セーフティケースは、機能安全や自動運転開発の分野で

広く推奨されており、以下のような規格で言及されています：

- ISO 26262（自動車の機能安全規格）
- SOTIF（ISO/PAS 21448, UL4600）

5.2 MISRA Safety Case Guideline

MISRA（Motor Industry Software Reliability Association）は、ISO 26262 に準拠した Safety Case を作成するためのガイドラインを提供しています。このガイドラインでは、安全性の議論を以下のような層構造で捉えています：

(i) Core（核心）：適切な要求事項を得たか
(ii) Layer 1（第 1 層）：要求事項を満たしたか
(iii) Layer 2（第 2 層）：適切な手段を用いたか
(iv) Layer 3（第 3 層）：適切な環境で開発したか

この構造に基づいて、自動車の安全性に関する議論は以下のように展開されます：

- 自動車は完全で正しい安全ゴールの通りに動作する
- 自動車はハザードイベント i を低減する安全ゴール i の通りに動作する
- 安全ゴール i の妥当性
- 自動車の安全ゴール i への準拠性
- 自動車の安全ゴール i の達成手段
- 自動車の安全ゴール i の開発手段

[MISRA Safety Case Guideline の図を挿入]

5.3 セーフティケースの重要性

セーフティケースの重要性は、過去の事例からも明らかです。例えば、2009 年から 2010 年にかけて発生したトヨタ自動車のスロットル制御システムの問

題では、安全性に関する明確な説明や証拠の提示が不足していました。この事例は、「それらを明確に用意できていれば、もっと簡単に解決していた（かもしれない）」という教訓を残しました。

このような経験から、システムの安全性や信頼性に関する説明責任の重要性が高まっています。アシュアランスケースは、この説明責任を果たすための効果的なツールとなります。

5.4　D-Case の概要と基本原則

5.4.1　D-Case と は

D-Case は、JST CREST DEOS プロジェクトの D-Case コアチームによって 2009 年から 2013 年にかけて開発された手法です。名称の「D」は「Dependability（信頼性）」を表しています。

D-Case の主な目的は以下の通りです：

- 合意形成のための手法・ツールの提供
- 開発・運用を通じたアシュアランスケースによるディペンダビリティの合意形成

5.5　D-Case の目的: ミニマムの合意形成

D-Case の核心的な目的は、異なる立場や背景を持つステークホルダー間での「ミニマムの合意形成」を実現することです。以下の図は、この概念を視覚的に表現しています：

[D-Case の合意形成の図を挿入]

この図が示すように、D-Case は以下のプロセスを促進します：

(i)　異なる立場・関心・目的、経験・価値観を持つステークホルダーを特定する

(ii) それぞれのステークホルダーの前提・主張を明確にする

(iii) 共通の目的・前提を設定し、合意形成を図る

5.5.1 D-Case の基本的な考え方

D-Case を効果的に活用するための基本的な考え方は以下の通りです：

- GSN（Goal Structuring Notation）自体は基本的な構造にとどめ、大きすぎないようにする
- コンテキストには要求分析結果、安全分析結果、テスト結果などの詳細なドキュメントを配置する
- GSN 自体は様々なドキュメントを紐付ける論理的な骨組みとして機能させる
- 他のドキュメントが充実していれば、GSN 自体は迅速に作成できるようにする

この考え方により、D-Case は複雑なシステムの信頼性を効率的に議論し、合意形成を促進するツールとなります。

5.6 GSN（Goal Structuring Notation）

5.6.1 GSN の 概 要

GSN（Goal Structuring Notation）は、アシュアランスケースを視覚的に表現するためのグラフィカル記法です。GSN は以下のような特徴を持ちます：

- 議論のモデル化を可能にする
- 様々な目的で利用できる柔軟性がある
- 元々はシステムの安全性を保証するためのセーフティケースを記述する目的で開発された
- D-Case において中心的な役割を果たす

5.7 GSN の基本要素

GSN は以下の基本的なノードを使用して構成されます：

ゴール（Goal） ステークホルダ間で合意したい主張

戦略（Strategy） 上位のゴールの分解の仕方を説明

前提（Context） 議論の前提となる情報

証拠（Evidence） ゴールが達成できていることを示す証拠（テスト結果など）

未達成（Undeveloped） まだ具体化できていないゴールや説明であることを示す

[GSN の基本要素の図を挿入]

5.7.1 GSN のノード接続ルール

GSN のノードを接続する際は、以下のルールに従います：

- 「前提」に接続する場合はコンテキストリンク（点線）を使用
- それ以外の接続にはサポートリンク（実線）を使用
- 終端は必ず「証拠」か「未達成」
- 「ゴール」は「戦略」に基づきサブゴールに分解する

[GSN のノード接続ルールの図を挿入]

5.7.2 GSNの作成例

以下に、GSN の簡単な作成例を示します。この例では、システムの安全性を主張するための GSN を構築しています。

[GSN の簡単な例の図を挿入]

この例では、以下の要素が含まれています：

- トップゴール：「システムは安全である」

- 戦略：「ハザードごとに議論する」
- サブゴール：「ハザード A に対処できる」「ハザード B に対処できる」
- 前提：「ハザードリスト A,B」
- 証拠：「テスト結果」

このように、GSN を用いることで、システムの安全性に関する議論を構造化し、視覚的に表現することができます。

5.8 D-Case ステップ

5.8.1 D-Case ステップの概要

D-Case ステップは、システム開発や日常の場で異なるステークホルダが手軽に合意形成を行うためのプロセスです。このプロセスは、システム開発における様々なミスコミュニケーションを減らし、ディペンダビリティを向上させることを目的としています。

D-Case ステップは以下の 3 つのステップから構成されます：

(i) ステークホルダの設定
(ii) D-Case の記述
(iii) 合意形成の実施

5.8.2 ステップ 1：ステークホルダの設定

このステップでは、プロジェクトや議論に関わる全てのステークホルダを特定します。ステークホルダには、開発者、利用者、運用者など、システムに関わる全ての人々が含まれます。

ステークホルダを明確化することで、以下の利点があります：

- 各ステークホルダの立場・関心・考え・経験・価値観を理解できる
- ステークホルダ間の潜在的な対立や誤解を事前に特定できる
- 合意形成のプロセスをスムーズに進められる

5.8.3 ステップ 2：D-Case の記述

D-Case の記述は、以下の 3 段階で行います：

(i) 「前提」とトップの「ゴール」を設定する

(ii) 「戦略」を設定し、トップゴールを分割してサブの「ゴール」を設定する

(iii) それぞれの最終ゴールのための「証拠」（または「未達成」）を設定する

この過程で、以下の点に注意します：

- これまでの D-Case があれば参照する
- 設定したステークホルダの情報を考慮する
- 論理的な構造を保ちながら、詳細化していく

5.8.4 ステップ 3：合意形成の実施

合意形成の実施には、以下の 2 つの場合があります：

ステークホルダ全員の場合 プロジェクタ等で D-Case を表示しながら、合意ができるか議論する

一部のステークホルダのみの場合 D-Case 記述不参加のステークホルダにもわかるよう合意形成を行う。必要であれば D-Case と同等の情報量を持つ絵や文章を用意する

5.8.5 D-Case の評価基準

作成した D-Case は、以下の 3 つの観点から評価します：

前提の妥当性 前提が過不足なく配置されているか

議論の妥当性 議論が論理的でステークホルダが理解できるか

規模の妥当性 ステークホルダが理解できる規模か

評価の結果、改善が必要な場合は、これらの基準に基づいて D-Case を修正します。一般的に、スライド 1 枚で見えるくらいの規模が目安となります。

5.9　事例紹介：自動運転システム

5.9.1　レベル 4 自動運転システムの事例

ここでは、レベル 4 自動運転システムを継続的に保証するための枠組みを提案した事例を紹介します。この事例は、SafeComp 2024 で発表された「A Case Study of Continuous Assurance Argument for Level 4 Automatic Driving」に基づいています。

主な特徴：

-
- 塩尻駅から塩尻市庁舎の周回コース (2km) を対象とする
- 特に市役所へ入るための右折にフォーカス
- STAMP/STPA の分析結果などをもとに GSN を記述

この事例研究では、自動運転シャトルバスが直面する様々な信頼性の課題に焦点を当てています。例えば、道路上の物体に対する過度に保守的な安全マージンは、車両が無期限に右折できなくなる可能性があり、交通システム全体の可用性を低下させる可能性があります。

5.9.2　継続的アシュアランスの重要性

この研究では、静的なアシュアランスケースだけでなく、継続的なモニタリングデータを組み合わせた動的なアプローチの重要性を強調しています。UL4600（自動運転車の国際標準）で導入されている安全性能指標（SPIs）は、この考え方を反映したものです。

SPIs は、設計、シミュレーション、テスト、展開の各段階で安全性主張が反証されていないかを検出する手段を提供します。著者らが提案するモニタリングシステムを含むツールチェーンは、この SPI メカニズムの一例と言えます。

5.9.3 アシュアランスケースのトップレベル構造

レベル 4 自動運転システムのアシュアランスケースのトップレベル構造は、以下のような要素で構成されています：

- システムの安全性に関する最上位の主張
- 運用条件、環境条件などの前提
- サブシステムごとの安全性主張
- 妥当性検証と評価に関する主張

[アシュアランスケースのトップレベル構造の図を挿入]

5.9.4 特定のユースケースの妥当性検証

この事例研究では、塩尻市での特定のユースケース（市役所への右折）に焦点を当てた妥当性検証の GSN 図も提示されています。この図は、以下のような要素を含んでいます：

- ユースケースの安全性に関する主張
- 安全性要求事項の充足性
- テストシナリオの網羅性
- 実環境でのテスト結果

[特定のユースケースの妥当性検証の GSN 図を挿入]

この詳細な GSN 図は、特定のユースケースに対する安全性の議論を構造化し、必要な証拠と論理的つながりを明確に示しています。

5.10 ま　と　め

本書では、システムのディペンダビリティを確保するための重要なツールであるアシュアランスケースと GSN について学びました。主な内容は以下の通りです：

- アシュアランスケース：システムが信頼できることを示すドキュメント
- GSN (Goal Structuring Notation)：グラフィカルなアシュアランスケー

スの表記法、モデル

- D-Case：ステークホルダ間の合意形成を促進するための手法
- 自動運転システムなどの最近のシステムでの実践例

これらの手法と概念は、今日の複雑化するシステム開発において、信頼性、安全性、セキュリティを確保するために不可欠なものとなっています。特に、自動運転技術や AI システムなど、新しい技術の導入に伴い、システムの振る舞いの予測が困難になる中で、アシュアランスケースの重要性はますます高まっています。

今後のシステム開発者は、これらの手法を効果的に活用し、システムの信頼性を体系的に示す能力を磨くことが求められます。同時に、継続的なモニタリングと評価を通じて、システムの安全性と信頼性を維持・向上させていく必要があります。

アシュアランスケースと GSN は、単なる文書化ツールではなく、システム開発のプロセス全体を通じて、安全性と信頼性に関する思考を構造化し、ステークホルダ間のコミュニケーションを促進する強力な手段です。これらを適切に活用することで、より安全で信頼性の高いシステムの開発が可能となるでしょう。

6 総合演習

7 まとめ