ソフトウェア組込み製品における品質向上のための障害分析手法の研究

三瀬　敏朗

目 次

図 目 次

表 目 次

第1章　序論

住宅，車，オフィスビルなどの生活空間における，家電製品や，ホームオートメーシュン，ビルオートメーションなどの機器制御・監視設備製品などのソフトウェア組込み製品(以後，製品という)は，我々の日常生活において様々な環境の下で使われている．そのため，その製品の開発者は，製品が想定使用期間を越えて使用されても，また想定していない方法で使用されても故障による障害から利用者を守らなければならない．さらに，利用者に支障の少ない軽微な故障であれば，安全性を確保しつつ，製品機能を維持する利便性も満足することが必要である．製品では，軽微な部品故障や運用ミスなどが常に起こり得る環境にあり，このような状況が発生した場合に，製品の全ての機能が正しく動作する必要はない．しかし，そのような状況が発生しても可能な限り利用者への影響を最小限に止めなければならない．また，その状況が取り除かれた場合には，影響が残らず正しい動作に復帰することが望まれる．

これらの製品は，近年，複雑化や多様化が進み，ソフトウェア要求仕様の正確な記述が困難となってきている．そのため，製品のテスト工程において，ソフトウェア要求仕様の記述の曖昧さに起因する障害が数多く発生している．その結果，ソフトウェア開発の手戻りが発生し，ソフトウェア構造の修正が多発している．これによりソフトウェア構造の堅牢性が低下する．また，ソフトウェア構造の堅牢性の低下は，ソフトウェアを組込んだ製品の品質を低下させる．同時に，テスト段階での不具合が多発することは，市場への不具合流出リスクを高める．そのため，ソフトウェアの要求仕様曖昧さを削減し，ソフトウェアの品質を向上するためのしくみが要望されている．

そこで本研究は，家電製品などの製品における，信頼性や可用性などの製品とソフトウェアの品質を向上することを意図し，ソフトウェア要求仕様を，より確実に定義するための有効な手法を構築することとした．

そのために，本研究では，筆者の勤務する社内の製品の開発事例を調査し，システムテストによる不具合の中で，仕様への手戻りが発生した不具合についての原因を分析した．その分析の結果，着目すべき不具合の原因は，製品の正常な振る舞いからの逸脱に対する設計者の考慮不足であった．この製品の正常な振る舞いからの逸脱は，開発する製品のソフトウェア以外のハードウェアや周辺機器，あるいは利用者などからの様々な刺激に起因していた．

製品は，正常な振る舞いから逸脱せざるを得ない過負荷や異常環境の下で，安全を確保しつつ，出来る限りの機能要求や非機能要求の維持が期待される．このような，製品が正常な振る舞いから逸脱せざるを得ない状況において，製品が利用者へ最大価値を提供することも非機能要求として扱わなければならない．そのため，製品が正常でも異常停止でもない状況における振る舞いは，正常な場合における振る舞いと同様に要求仕様に定義されなければならない．しかし，調査した製品開発の要求仕様において，正常でも異常停止でもない状況に対する記述は十分ではなかった．また，要求工学の分野において，製品が正常な振る舞いから逸脱せざるを得ない状況に対する非機能要求の研究は進んでいない．そのため，この正常でも異常停止でもない状況に対する非機能要求の曖昧性を削減するための方法について検討を行った．その結果，システムアーキテクチャを設計し，ソフトウェア要求仕様定義を行う前に，製品が正常な振る舞いから逸脱することによる障害分析を行うことが必要であると判断した．

そこで，本研究の具体的な対象を，ソフトウェア組込み製品の開発に適した障害分析手法に焦点をあてることにした．ただし，本研究として扱う障害とは，利用者の製品に対する些細な不満も障害と扱う．これは，製品の品質向上を本研究の目的としており，利用者の不満が起こる事象をできる限りなくすことを意図するためである．

製品の障害分析手法の検討に際し，社内調査における仕様への手戻りが発生した不具合要因をもとに必要な要件を抽出した．そして，既存の障害分析手法や先行研究について抽出した要件の充足性を調査した．しかし，既存の手法においては，ソフトウェアを具体的な対象にしていないため，時間変化に対応した分析の要件が満足していなかった．また，先行研究において，製品のような，正常ではない状況で使用され続けることを配慮されていないため，複合要因によ

る障害分析を明示的に分析する研究はされていない．さらに，ソフトウェアの振る舞いを中心にした分析においては，ソフトウェア要求仕様作成後の，ソフトウェアアーキテクチャに対する分析手法が多く，本研究対象のソフトウェア要求仕様定義前とは，手法適用フェーズが異なっていた．そこで，製品に適用するための新たな障害分析手法を開発することにした．

本研究により開発した障害分析手法は，筆者らが実務で行っていた方法に理論的な考察を加え体系化したものである．本研究において，分析対象を製品利用環境を含んだ状態遷移モデルとし，理論的なモデルから簡略化した状態遷移表である非正常系分析マトリクスと，これを適用した障害シナリオ抽出手法ESIM(Embedded System Improving Method)を提案する．また，提案した障害分析手法を，筆者の所属する会社における実際の製品開発に適用した実験を行い，有効性と実用性があることを確認する．

本研究による障害分析手法ESIMは，4つの特長を持つ．

1番目は，障害に至る時間変化に対する影響を陽に含むシナリオ過程を分析できることである．このため，ESIMでは，理論的な並列分散処理であるcommunicating state machinesの動的モデルから意味のある状態のみを抽出した状態遷移モデルとして非正常系分析マトリクスを作成する．これにより，システム内の構成要素間の逸脱から波及するシナリオや，タイミングに依存する連鎖が抽出できることを，事例や適用実験において確認する．

2番目は，複合的な要因が相互に影響する障害の分析が行えることである．このため，ESIMでは，分析者の観察の視点から記述を行い，逸脱した非正常状態と，逸脱した状態に遷移したという非正常イベントの定義を行う．また，すべての逸脱要因を１つの非正常系分析マトリクスに記載する．そして，適用実験において，複合要因による障害シナリオが抽出できていることと，非正常系分析マトリクスの規模と分析時間が実用的な範囲で実施できることを確認する．

3番目は，アーキテクチャ設計段階で予見できなかった障害の抽出ができることである．ESIMでは，逸脱要因の抽出手順により，システム構造図の構成要素が追加される．また，障害シナリオ抽出手順で，非正常系分析マトリクスの分析過程で新たな構成要素間の関連が発見される．さらに，逸脱を発見することにより状態やイベントを追加していくため，その状態の粒度を細分化し，逸脱に

ついてより詳細な分析を行い，早期に課題を発見することができる．これらについて，適用実験により，ESIMを使わなかった場合よりも，多くの障害シナリオを発見できることを確認する．

4番目は，部品故障や誤操作など広く要因を捉えた分析ができることである．このため，ESIMは，製品だけでなく，人，環境，周辺ハードウェアなども分析対象とする．そして，初期の逸脱要因の抽出として，故障モードとガイドワードをハードウェア，人の運用，環境などに拡張する．また，分析対象に連続的な振る舞いと，離散的な振る舞いの特性が混在するため，同値分割法を用いて離散的な特性に統一した状態とイベント記述を定義する．

このESIMは，システムアーキテクチャを設計し，ソフトウェア要求仕様を作成する前に適用する．ESIMは，アーキテクチャ設計段階で予見できなかった未知の状態やイベントを抽出し，想定していなかった障害を抽出することにより仕様の妥当性の確認を行うことが可能となり，製品の品質とソフトウェアの品質を向上させる．

以下，各章を概観する．

製品の要求を明確にするために，2章では，製品の特徴を分析し，さらに実際の開発商品を通じて製品の正常な振る舞いからの逸脱という課題を抽出し，その課題に対応するために障害分析手法に研究の焦点を絞り，それに必要な要件の抽出を行う．その要件に対し，3章において，既存の手法や先行研究の問題点を明らかにし，本研究の位置付けを行う．4章では，障害分析手法の概念を明確にするため，製品を環境も含めたモデル化を行い，障害に至る過程について論じる．製品の未知の障害を発見するための障害分析を行うためには，障害要因がどのように利用者に影響を与えていくかという原因から結果への方向の分析が必要になる．そのため，最初の逸脱要因の抽出と逸脱から障害に至る連鎖の分析という2つの手順が重要となる．そこで，前者を5章で，後者を6章で論じ，それらをまとめた一連の障害分析手法ESIMの手順を7章で論じる．次に，ESIMを用いた事例を8章で論じ，ソフトウェア要求仕様の曖昧性を削減する手法としての有効性を確認する．さらに，9章では，実務への適用実験にもとづく有効性と実用性を確認する．最後に10章においてまとめを行い，11章に障害シナリオ抽出手法としての今後の課題について述べる．

第2章　障害分析の重要性と必要要件

本章では，研究の目的である製品の品質向上に対し，具体的な課題を明確にし，研究対象として障害分析手法に焦点をあてた経緯につて述べる．そのために，最初に研究の対象となる家電製品などの製品の特徴と，製品に対する社会的な安全性要求について述べる．次に，ソフトウェア開発のプロセスとしてソフトウェア要求仕様書が重要であり，非機能要求が課題であることについて述べる．また，ソフトウェア要求仕様の詳細な要因を把握するために，筆者の社内における製品開発を対象に行った調査の結果について述べる．その調査結果の分析から，ソフトウェア要求仕様を定義する前に障害分析を行うことが重要であることについて述べ，障害分析が製品の品質向上に対して有効な範囲について述べる．最後に，製品の障害分析に必要な要件について述べる．

2.1　対象とする製品の特徴

近年，製品は，様々な分野で用いられ，分野毎に使われる環境や要求される品質が異なる．本研究の対象は，住宅，車，オフィスビルなどの生活空間における，家電製品や，ホームオートメーシュン，ビルオートメーションなどの機器制御・監視設備製品である．これらは，自然現象や人為的な製品を取り巻く環境の中で安全性を保証しつつ，耐久性，利便性，可用性などの品質を実現する必要がある．以下に，それら製品が使用される環境や要求される品質の特徴について述べる．

1)　専任オペレータの不在

多くの製品の利用者は，専任オペレータではない．利用者は，製品の技術的なしくみや振る舞いについての知識がなく，製品のマニュアルに記載された操作手順に従って操作されるとは限らない．そのため，利用者に対して製品の操作の方法や運用手順を制限することができない．たとえば，利用者は，使用方法が分からなくなった場合に，パニック的な操作を行ってしまう場合がある．また，利用

者は，イタズラ操作や誤った運用を行う場合もある．

さらに，製品は，コンピュータとしては扱われない．そのため，ソフトウェアの不具合が発生した場合において，利用者にソフトウェアの再起動などの回復操作を期待することはできない．

2)　制限できない使用環境

利用者に対して，製品の動作に影響を及ぼす恐れのある外乱のない安定した環境でのみ製品を使用することを制限できない．そのため，製品が，温度，電波環境，湿度，振動などにおける様々な外乱のある環境下で使用されることを想定しなければならない．また，製品に接続される周辺機器について，利用者が，指定された機器のみ接続するとは限らない．さらに，利用者により接続された機器が，常に正常に動作するとは限らない．

3)　部分的な劣化や故障時における使用

多くの製品では，製品の利用が生活に密着している．このため，たとえ製品の些細な部分が誤動作や故障を起こしても容易に停止せず，可能な限り機能を維持し続けることを利用者は期待している．そのため，利用者は，主要な構成要素ではない製品の一部が故障することにより，製品の主要機能が停止することに不満をもつ．さらに，製品は，使用される期間の制限を行うことが出来ない．このため，劣化に伴う様々な部品が故障している製品の寿命末期において，製品が使用され続けることも想定しなければならない．

4)　ハードウェアの適正コスト

多くの製品において，利用者は，頻度の高い使用状況における高い経済性を期待する．このため，製品は，利用される頻度の高い使用状態の負荷に対して十分な性能が実現できるハードウェアとして構成される．したがって，例外的な使用に対応するための冗長的なハードウェアは組込まれていない．

このため，利用される過酷な負荷や異常な環境などの利用頻度が低い状況において，製品は，完全な機能や十分な性能を要求されない．しかし，製品は，可能な限り機能や性能を維持すると同時に，利用者を製品による障害から守らなければならない．さらに，その過酷な負荷や異常な環境の状況から回復した場合に，製品は，その状況が発生する前の状態に復帰することが要求される．

5) リアルタイムシステム

製品は，処理要求の発生に対して，即座に処理を実行して結果を返す必要がある．このため，製品は，ハードウェア処理や製品の機能を実現する処理をリアルタイムに並行処理する．このような並行処理を行うシステムでは，システム内のタイミング的な組み合わせが複雑になり，システム全体の正確な状態が把握しにくい．

2.2 安全性に対する品質要求

2.1節において，製品は，様々な使い方をされ，その状況の中でも安全性が要求されることについて述べた．本節では，安全性について社会的にどのように扱われているかについて述べる．

近年，製品は，高機能化や多様化した要求が高まり，ソフトウェアが複雑化，大規模化している．同時に，これらの製品に障害が発生した場合のリスクも年々増加している．一方，社会の要求により様々な法律などで製品が規制されている．これらの法律により，利用者の使用ミスや悪環境での使用などに起因する製品の不安全はメーカの責任とされている．このため，一度，重要な事故が発生すると企業ブランドは失墜してしまう．

消費生活用製品安全法や電気用品安全法においては，重大製品事故が発生した場合には，発生の事実を知った日から起算して10日以内に経済産業省に報告しなければならない[2, 3, 4, 5, 6]．また，海外においても同様の法律により利用者を保護している[7, 8]．ここで，重大製品事故とは，一般消費者の身体に対する重大な危害を及ぼす場合，あるいは，消費生活用製品が滅失やき損した事故で，一般消費者の生命または身体に対する重大な危害が生ずるおそれのあるものを示す．その中で，製品事故と扱われない場合は，一般消費者による明確な目的外使用や重過失と考えられる場合だけであり，製品の想定可能なリスクに対する安全配慮が求められる．さらに，製品の寿命末期などの経年劣化による事故を防止することも求められる．これらの法律により，万が一重大な事故が発生した場合において，消費者庁は，企業名を含め事故情報を迅速に公表し，徹底した回収も義務付けている[5]．また，重大製品事故以外の製品事故であっても，独立行政法人製品評価技術基盤機構に報告が必要であり，その製品による事故報告の内容により行政指導が行われる．

一方，電気・電子・ソフトウェアの機能安全の国際規格が，IEC 61508を基本的

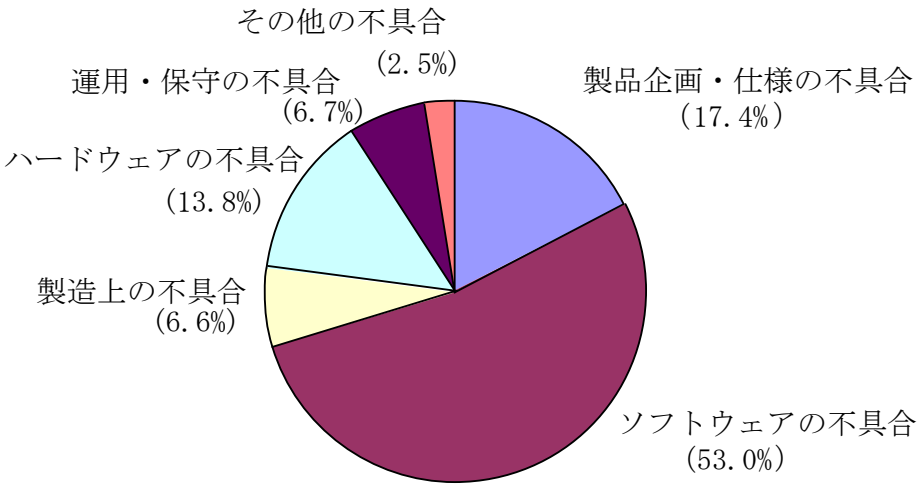

図 2.1: 製品出荷後の不具合原因比率

な上位規格とし，車や医療品などの製品カテゴリごとの規格として定められている[9, 10, 11, 12]．これらの規格は，安全な製品を開発するために有効と考えられる管理や開発手法の適用を定めている．これらの規格では，電気，電子，ソフトウェアを含む制御対象や制御機器において，システムの設計初期段階において起こり得る障害リスクを抽出し，許容できるリスク値まで下げるための安全装置や安全機能を設計仕様として設定し，製品に組込むことを要求している．

2.3 製品における開発の課題

2.2節までに，製品の特徴や製品に対して安全性を中心とした品質が要求されることについて述べた．本節では，製品開発プロセスの課題について述べる．

2009年の経済産業省の調査[14]によると，図 2.1に示すように，ソフトウェア組込み製品における市場出荷後の全不具合の中で53%の不具合がソフトウェア起因であると述べられている．さらに，全不具合の中で製品の企画・仕様の不具合が17.4%を含めており，ソフトウェアおよびその上流起因が全体の70%を占めている．このため，ハードウェア設計起因や製造上の不具合起因等と比較して，ソフトウェアの品質が，製品の品質を決める最大の要因といえる．また，同調査において，図 2.2に示すように，プロジェクトの遂行に一番重要となる項目について，技術者のスキルの次に，要求仕様が挙げられている．このことから，ソフトウェ

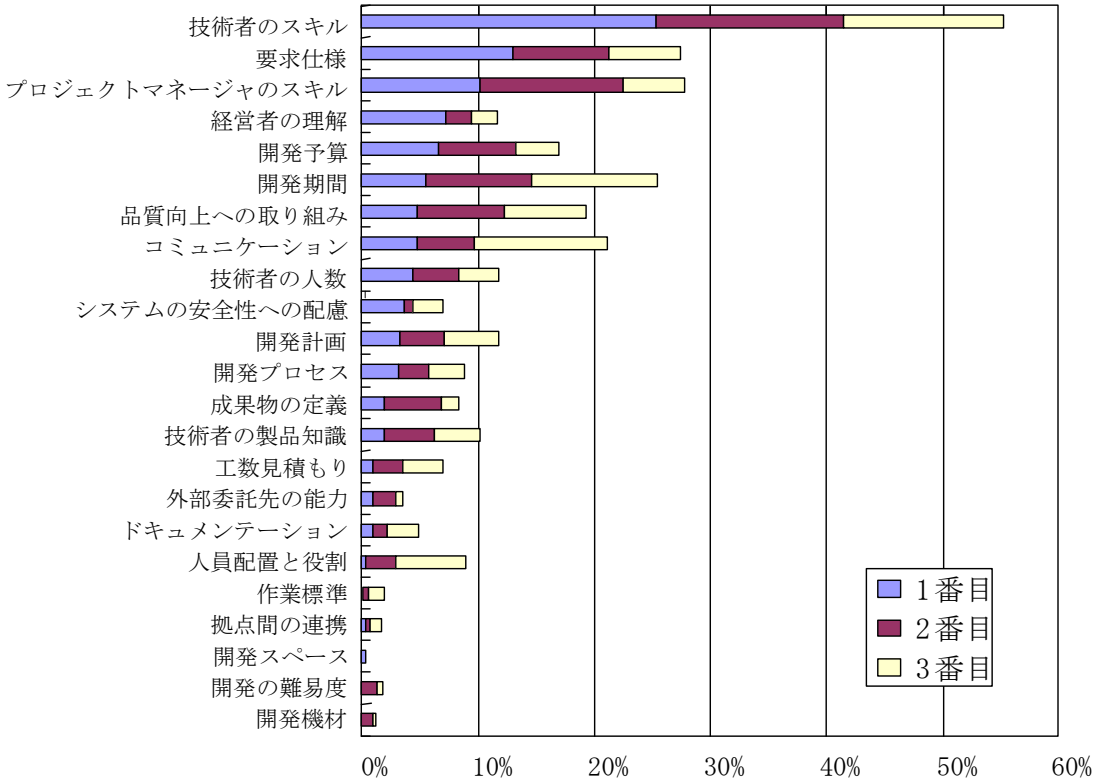

図 2.2: プロジェクトの遂行に重要となる項目

ア開発において要求仕様の定義が一番重要な課題であることが言える．これらの傾向は，毎年の調査でも同様の傾向を示している [13].

製品を開発する現場では，その製品テストの実施時に，ソフトウェア要求仕様に起因する不具合が発見される．そのため，ソフトウェア開発の手戻りが発生し，最初のソフトウェア構造が崩れ品質が低下する．また，このような商品化の水際における不具合の発見は，市場に不具合を流出させるリスクを高める．このため，ソフトウェアの要求仕様の精度を高め，ソフトウェア開発の手戻りを削減し，製品の品質を確保することが求められている．

2.4 ソフトウェア要求仕様における非機能要求

2.3節において，ソフトウェア要求仕様定義が重要であることを述べた．本節では，ソフトウェア要求仕様定義における課題として非機能要求が重要であることについて述べる．

経済産業省による「情報システムの信頼性向上に関するガイドライン第２版」では、非機能要求を明確に定義することが情報システムの信頼性を向上する上で重要であると指摘している[15]．また，情報処理推進機構における要求工学・設計開発技術研究部会の非機能要求アーキテクチャワーキンググループは，非機能要求の検討を行った[16, 17]．この報告書において，非機能要求は，安全性などの製品の品質要求や制約条件とされ，主観的であり，要求する品質項目間に相関性があるとしている．機能要求は，製品の動作に対する要求であるため，ソフトウェア要求仕様に記載する製品の振る舞いに変換しやすい．たとえば,「車のブレーキをかけると減速あるいは停止する」という要求は，ブレーキ操作信号により，ブレーキ制御メカニズムを制御するという振る舞いに変換できる．一方，非機能要求は，振る舞いに変換できない．たとえば,「ブレーキが安全に働く」という非機能要求は，安全という記述が明確でないため，振る舞いには変換できない．Nick Rozanskiは，複雑な機能的特徴と品質特性を単一の包括的モデルとして捉えることはできないと述べている[23].

このため，非機能要求がソフトウェア要求仕様における曖昧性の主要な要因であり，ソフトウェア要求仕様における曖昧性を低減させるためには，非機能要求を明確にすることが必要である．

2.5　ソフトウェア要求仕様における詳細要因の調査

2.3節において，ソフトウェア要求仕様が重要であることを述べた．本節では，実際の製品開発を調査し，さらに詳細な要因を追求する．

製品のソフトウェア要求仕様における課題を明確にするために，筆者の所属する社内における製品開発について調査を実施した[20]．調査対象とした製品の開発プロジェクトは，システムテストにおいて849件の不具合報告書が存在した．調査のための情報として，システムテスト不具合報告書，ソフトウェア要求仕様定義書，ソフトウェア設計書，ソフト開発委託先と社内担当者との約2000件のメールや打ち合わせ議事録を参照した．このシステムテスト不具合報告書に記載された不具合修正内容を基点として，入手した情報から真の要因を分析した．そして，849件のシステムテスト不具合報告書の中で単純なプログラムバグや試験手順や環境・ハードウェアを除き，ソフトウェア要求仕様書に絡む211件に着目

比率		仕様変更要因
20 %		仕様の記述ミス
	20 %	基本的な仕様の曖昧な記述，記述漏れ
40 %		正常な動作の詳細な振舞いに対する検討不足
	16 %	機能間の詳細な関係の検討不足
	15 %	詳細な機能，通信の詳細検討不足
	6 %	性能を考慮した処理の検討不足
	2 %	ハードウェアの詳細な特性に対する動作
	1 %	製造施工・メンテナンス時の検討不足
40 %		正常な動作から逸脱した場合の振舞いの検討不足
	22 %	停電，過負荷などの例外的な環境時の検討不足
	9 %	例外操作・異常な設定・運用に対する検討不足
	9 %	部分的な障害発生時の処理や障害回復後の検討不足

表 2.1: ソフトウェアテスト不具合における仕様変更理由

した．この211件について，さらに原因分析を行った結果を表2.1に示す．この211件中，20%は，基本的な仕様の記述漏れや曖昧な記述であり，ソフトウェア要求仕様の記述方法が要因であった．また，40%については，商品関連知識や基本技術の要因であった．これらの60%については，教育などに対策により対応が可能と判断し，本研究の対象からは，除外した．

残された40%については，製品の正常な振る舞いではなく，何らかの正常な振る舞いからの逸脱が原因となっていた．この正常な振る舞いからの逸脱について，ソフトウェア要求仕様書に明確に振る舞いが規定されていなかった．そのため，テスト実施者は，利用者視点による妥当性確認として不具合を指摘していた．このように，正常な振る舞いからの逸脱については，テスト実施者の試験項目やテスト合否基準が明確ではなく，システムテストで確実にテストされる保証がない．また，この不具合項目に対する具体的な改善対策はなかった．

そこで，本研究では，この不具合項目に焦点を当てることにした．そのため，この40%の不具合要因に対して，さらに詳細に不具合要因について分析を行った．その結果，40%の不具合要因における製品の正常な振る舞いから逸脱を発生させた原因として考慮すべき特徴を，以下に挙げる．

1) 例外的な環境などの開発製品以外の要因

利用者の矛盾処理，操作放置，周辺機器の停止や例外的な動作，周辺装置の例外的な過負荷，断線，電源断や再投入，周辺機器の障害や誤動作，施工ミスを想定した周辺機器の異常動作など，CPUが直接的には介在出来ないところに起因する要因が多い．これは，40%の不具合要因である正常な振る舞いからの逸脱のほぼすべてが当てはまる．

2) 複合的な要因

要因としては，希に発生する特殊なものではなく，複数の要因の組み合わせにより不具合が発生している．また，これらの複数の要因が発生した場合に常に起こるのではなく，特定のタイミングにのみ発生する．これらの個々の要因は些細であり，複合した場合の特定の状況における影響を見過ごしている．これは，表 2.1 に示した正常な振る舞いからの逸脱の約 30%が当てはまる．

3) 最初の要因に連鎖した結果としての障害

ある要因による正常な動作からの逸脱が他の要素への逸脱を誘因し，さらにそれが他の要素に影響するといった一連の連鎖であるシナリオの結果として不具合が発生していた．この連鎖は，40%の不具合要因である正常な振る舞いからの逸脱のほぼすべてが当てはまる．

ひとつの逸脱事象は，それ自体で製品への影響が大きいとは判断しにくい．しかし，その影響が他の逸脱事象に連鎖していき，結果として障害に至る．この連鎖に対する配慮が漏れており，起こりうる障害リスクが想定できず，そのリスクに対応した設計が行えていなかった．

以上のプロジェクトの分析結果に対して，実際の社内の市場流出不具合，社会的に公表されているソフトウェアによる障害として問題となった事故の不具合解析に関する文献などを調査し，同様の原因が関係していることを確認した [21, 22]．たとえば，文献に記載されていた事例として，下記があった．

インターネットによる無料通信サービスである Skype が，2007 年に 2 日間のサービス停止となる障害が発生した．その原因は，パソコン OS である Windws の定期的なアップデートによるコンピュータの再起動が短時間に集中したことが原因となり，Skype のネットワークに打撃を与えた．さらに，ログインが集中し，ネットワークのリソース不足と合わせ，決定的な打撃となる連鎖反応を招いた．

この例においても，開発対象以外からの要因であり，Windws の定期的なアッ

プデートによるコンピュータの再起動とネットワークのリソース不足といった複合的な要因により，ログインが集中する連鎖から障害が発生している．

2.6 障害分析の必要性

本節では，2.5節において述べた，製品の正常な振る舞いからの逸脱に対する対策として，障害分析の有効性について述べる．また，障害分析を行う際にアーキテクチャ設計と分離し，ソフトウェア要求仕様定義の前に障害分析を行うことの必要性について述べる．

本研究の目的は，家電製品などの製品における，信頼性や可用性などの製品とソフトウェアの品質を向上することであり，利用者の不満が起こる事象をできる限りなくすことである．そこで，本研究で扱う障害は，安全性だけでなく，すべての非機能要求に関わる利用者の不満となる事象すべてを含めるものとする．

非機能要求の多くの要素である品質要求は，ソフトウェア品質モデルISO25010[25]において，機能性，信頼性，使用性，効率性，保守性，移植性，有効性，生産性，安全性，満足度などが定義されている．これらは，運用における利用者からの視点と捉えることができる．一方，アーキテクチャの品質項目として，ロバスト性（堅牢性）がある．このロバスト性は，様々な環境や使われ方，システム内の一部の故障などに対してシステムが障害にならない性質を示している．ロバスト性に脆弱性があった場合，システムは，様々な環境において顧客の期待する品質要求から逸脱する．たとえば，センサが故障しているにもかかわらず間違い値を正常のように表示していれば，信頼性の障害であり，操作ミスを放置してしまった場合、重要データが誰にでも見ることができる状態になってしまうのは，セキュリティの障害である．また，末端の一部の部品が故障していたため、主要機能が停止するのは，使用性の障害であり，同様に末端の一部の部品が故障していたため、全体のスピードが低下すれば効率性の障害である．さらに，一部のデータが壊れたが、全てのデータが代替装置に入らないことは，保守性の障害であり，一部のセンサが壊れていたことにより，電源を入れた途端に駆動部が急速回転を始めることは，安全性の障害である．

近年，製品におけるロバスト性や利用者視点の品質項目を統合したディペンダビリティという品質項目が提唱されている[18, 19]．ディペンダビリティは，製品

が様々な環境下で，様々な使われ方をされたとしても安全かつ利便性などの品質を総合的に保たれる性質を示す．また，ディペンダビリティは，安全性，信頼性，保全性，可用性などを統合した信頼性として定義され，「頼りがい」とも呼ばれている．しかし，ディペンダビリティは，品質項目を統合しているために，その意味として含まれている安全性や可用性などが相互に関係し，トレードオフの関係を持つ曖昧性を有している．

製品の利用価値は，2.1節で述べたように，価格に対して最大の価値を提供することであり，製品は，すべての状況において，カタログなどに記載された機能や非機能要求が満足することを要求されていない．そのため，製品は，どのような事態に陥っても全体としての機能を失わないようにするフォールトトレランスが要求されているのではない．可能な範囲で稼動させるフェールソフト，システムが誤動作をしたり部品が故障したりしても，安全側に制御するフェールセーフ，操作や手順を間違えても，危険を招かないように設計するフールプルーフにより，製品は，利用者に安全性と高い可用性を提供する必要がある．

このため，ディペンダビリティを明確にするためには，様々な環境や使われ方，システム内の一部の故障などを把握しておく必要がある．さらに，様々な環境や使われ方，システム内の一部の故障などを通じて，製品がどのように振る舞い，それにより，利用者にどのような影響があるのかを把握する必要がある．そして，それらの起こりうる状況の中で利用者の影響として許容できない状況がない製品の振る舞いを定義し，要求仕様に明確に定義されなければならない．

以上のことから，障害分析は，ロバスト性を確保するための分析であるが，ロバスト性の低さは，利用者から見たすべての非機能要求に関わる欠陥となりうる．そのため，正常な振る舞いから逸脱せざるをえない状況において，何が利用者から見た障害であり，どの品質要素が優先されるべきであるかは，要求として定義されるべきである．このような要求も非機能要求の一部であり，障害分析を行うことにより明確にすることが可能である．

非機能要求が障害分析を介して機能要求に変換できることについて，自動車におけるブレーキの例を用いて説明する．

自動車の運転におけるブレーキ操作に関する様々な動作シナリオを抽出しても，「ブレーキが安全に働く」という非機能要求に対応する安全なブレーキの振

る舞いを定義できない．そのため，動作シナリオをもとに，ブレーキの操作に関する様々な逸脱可能性を抽出する障害分析を行う．たとえば，衝突しそうになり運転手がパニックになった場合に，アクセルとブレーキを同時に踏む可能性がある．また，アクセルのメカニズムにおいて，アクセルがフロアーマットに絡むことによりアクセルが戻らなくなった場合，車を止めるためにブレーキを踏むという状況が想定できる．これらの場合，アクセルとブレーキが同時に踏まれている．この結果，先行した操作を優先するという振る舞いである場合には，ブレーキが働かず車が衝突するという障害事象が抽出できる．

この障害事象を抽出したことにより，非機能要求の具体的な振る舞いの定義が可能になる．すなわち，非機能要求である安全性を確保するための機能として，アクセルとブレーキを同時に踏んだ場合には，ブレーキを優先するというブレーキの振る舞いを定義できる．

以上のことから，非機能要求を明確にするためには，以下のプロセスが必要となる．図 2.3に非機能要求から非機能要求を保証するための機能要求に変換するプロセスを示す．はじめに，図 2.3の(A)に示す機能要求から製品の振る舞いに変換する．次に，その振る舞いに対して，(B)に示す製品が使われる悪環境や誤操作などの逸脱要因を想定し，障害分析を行う．障害分析は，システムの振る舞いが逸脱した結果として起こりうる障害に至る例外的な振る舞いを抽出する．さらに，(C)に示すように，例外的な振る舞いが，非機能要求の視点から許される範囲かの妥当性を検証する．その逸脱が許容できない場合は，障害と見なされる．そのため，(D)に示す障害に対する回避対策を検討し，それらを非機能要求を保証するためのための障害回避機能として機能要求に組込み，様々な逸脱に耐えうるかの検証をもう一度行う．この理由は，非機能要求の項目間に相互関係があるため，他の非機能要求に影響がないことを確認する必要があるためである．

また，Dev Rahejaらは，システム安全の新しいパラダイムとして，システムが複雑化・ネットワーク化し，システムとしての安全性を考える必要があること，もはや部品から発生する故障以外の原因が多彩にあること，さらにソフトウェアは，故障の前兆を監視し，障害から回避することが可能であり，設計段階で十分にデザインすることが必要であることを述べている [24].

現在，ソフトウェア開発工程において，正常な製品の振る舞いの機能分析と障

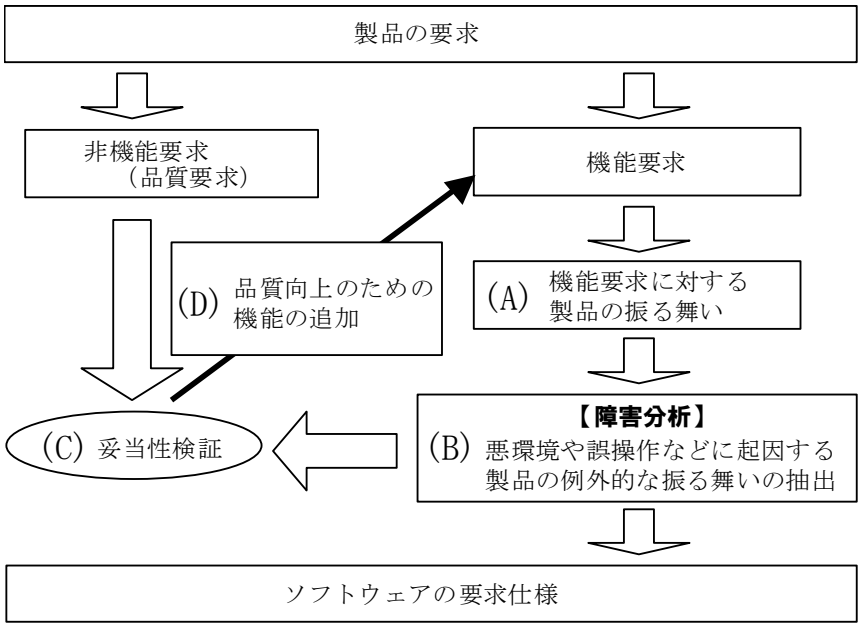

図 2.3: 非機能要求の明確化フロー

害分析が明確に分離されていない場合が多い．一般的には，システムアーキテクチャ設計を行い，ソフトウェアが果たすべき要件の定義を行う過程の中で，障害原因となる例外事象の分析を含めた製品の振る舞い分析を行い，障害を抽出することが行われている．しかし，システム分析中に全ての例外処理を検討することは困難とされている．たとえば，Bruce P. Douglassは，リアルタイムシステムのアーキテクチャ分析において，全ての例外処理を分析の中に取り込むことは，必要以上に分析を複雑にするだけであり，必要な例外処理に限定するべきであると論じている[27]．しかし，製品開発では多くの例外事象である逸脱に対して検討しなければならない．そのため，基本的な動作の分析と障害分析を分離することが望ましいと考える．

2.7 障害分析により明確化可能な非機能要求の領域

2.6節において，障害分析の必要性について述べた．本節では，本研究の対象である障害分析により明確にすることができる非機能要求の範囲について述べる．

製品が使用される環境と製品に要求される要件について図 2.4に示す．

図 2.4の領域Aにおいて，カタログに記載された使用環境において，製品は機能要求と非機能要求を満足されなければならない．この領域における非機能要求の定義については，要求工学の分野での研究が数多く進められている．また，製品を製造する企業においては，機能性については，商品企画部門，使用性については，デザイン部門，保守性や移植性などについては，カスタマサービス部門など専門部署が存在する．そして，製品の開発において，企画から開発，製造，販売，顧客サービス部門が一体となり，設計の確認が行われ，保守マニュアルや施工マニュアルなどに展開されている．このため，保守性や施工性などの必要な非機能要求は考慮されている．さらに，社内規程などにより非機能要求を明確にされるしくみが存在する．

また，領域Cにおいて，製品が重大な故障や製品の環境に重大な問題があった場合は，直ちに安全に停止しなければならない．このため，この領域における非機能要求は，明確である．

これらに属さない領域Bは，製品の環境や製品自体に軽微な問題があるが，重大な問題ではないという領域，あるいは，システム内の一部に重要な問題が発生し，システムとして問題を把握し，領域Cに至るための過渡状態である．この領域では，カタログに記載されたすべての機能を要求されてはいないが，できる限りの機能や非機能を継続し，最大価値を提供することが要求される．また，製品の環境や製品自体の問題がなくなった場合には，速やかに前の状態に復帰することが期待され，逆に，問題が深刻になった場合には，速やかに領域Cに遷移することが期待される．このような，すべての機能やすべての非機能を部分的にしか満足できない状況において，使用者に最大の利用価値を与えることも非機能要求である．つまり，利用者は，正常なカタログに記載された振る舞いだけではなく，これらから逸脱した製品の振る舞いに対しても製品の品質と見なす.

本研究は，この領域Bにおける要求仕様の曖昧性を削減することに焦点をあて，領域Aにおける非機能要求は対象としない．したがって，2.5節で述べた，商品知識や設計技術が要因とした40%については，図 2.4に示す領域Aにおけるカタログなどに記載された機能要求と非機能要求の定義に関する不具合である．本研究は，領域Bにおける正常な振る舞いからの逸脱である40%を対象とする．

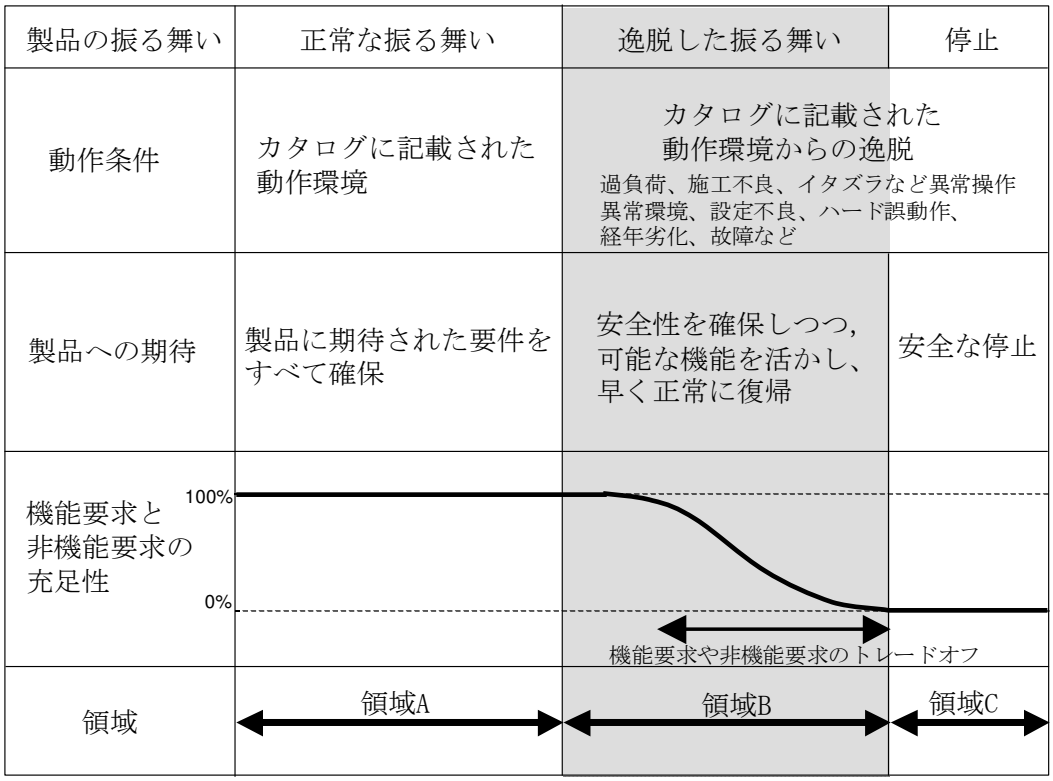

図 2.4: 製品が使用される環境と要求される要件

2.8 障害分析手法に必要な要件

前節までの製品の特性と製品開発における課題の分析結果より，システムアーキテクチャ設計フェーズ完了時の障害分析手法の要件は，下記が挙げられる．

[要件1] 製品だけでなく，環境や利用者等も分析対象として含むこと

2.5節において，仕様に影響する不具合は，製品のソフトウェア以外が要因になっていることを述べた．製品は，過負荷や異常な環境での使用，ハードウェアの劣化や誤動作，部品の故障，誤った運用など，利用中にほとんど発生しない状況においても徹底した安全性が確保された上で，可能な限りの可用性が要求される．このため，製品だけではなく，製品が使われる様々な環境や使われ方がどのように製品に影響するかの分析が必要となる．

Jacksonは，問題フレームを定義するにあたり，制御システムと制御システム外の世界を切り分け，その間の接続に注目した．その問題フレームでは，各接続で考えるべき関心事が提示され，これが要求抽出の指針となっている [26].

Jacksonは，問題フレームにおける制御システムを取り巻く外界は物理ドメインと呼び,「顧客が観察可能な効果をチェックする実世界の一部」と定義している．このように，製品では，制御システムからコントロールできない物理ドメインの影響への配慮が重要な品質要件となる．

[要件2]　複合的な要因による障害を分析できること

2.5節において，複合的な要因が課題であることについて述べた．製品開発において，単独の要因では障害に至らず，複数の要因が複合した場合のみ障害が発生する場合についての検討が漏れやすい．製品は，安全性を保ちながら，可能な限り機能を継続させる必要がある．このため，製品は，部分的に正常な振る舞いから逸脱した状態で使用される状況も多く，複合的な要因が発生する確率が高い．

[要件3]　障害が発生するシナリオ過程の分析が行えること

2.5節に述べたように，ある要因からの連鎖が発生し，結果として障害に至ることを述べた．この場合，特定の要因の発生が必ずしも障害に至るのではなく，特定の状況においてのみ連鎖が発生し，結果として障害が起こる．このようなタイミングに依存した連鎖の結果として発生する障害は，分析者の検討から漏れやすい．したがって，その連鎖を時間的な考慮を含むシナリオとして抽出する必要がある．

2.9　まとめ

本節は，本研究の目的である製品とソフトウェアの品質を向上させるための，研究対象の絞込みについてまとめる．

製品は，利用者の生活に密着し，様々な環境で様々な使い方をされる．その中で，安全性を確保しつつ，機能や性能などの品質要求を，できる限り利用者の状況に合わせて確保する必要がある．しかし，筆者の社内の製品開発における調査において，利用環境や運用などが製品に想定された範囲を超え，製品のカタログに記載された機能を実現するための振る舞いから逸脱した振る舞いに対する仕様定義に課題があることを抽出した．その課題解決のためには，起こりうる逸脱とそれが利用者へ及ぼす影響を把握しなければならない．その影響が製品として許容できない品質であるならば，その回避のための機能を追加することにより，曖昧な非機能要求を明確な機能要求に変換することができる．そのための方策として，ソフトウェア要求仕様書を作成する前に障害分析が必要ある

ことを提案した．そして，筆者の社内の製品開発における調査結果から，製品の特徴にもとづいた障害分析を行うための要件として，製品ソフトウェア以外の要因を含んだ障害分析を行うこと，複合要因を分析できること，障害に至る連鎖を考慮できることの3つの障害分析のための要件の抽出を行った．

第3章　既存の障害分析手法と先行研究

本章では，障害分析手法について，現在用いられている既存の手法と，先行研究において提案されている手法ついて紹介し，本研究における製品の障害分析の観点から捉えた課題を明らかにすることにより，本研究の位置付けを明確にする．そのため，それぞれの障害分析に関する手法が，2.8節に述べた要件に対して充足しているかの確認を行う．

3.1　既存の障害分析手法における要件の充足性

本節では，現在，製造業の開発現場で使用されている障害分析手法についての課題を明らかにする．現在，一般的に用いられている障害抽出分析手法としては，機能安全IEC61508[9]の推奨手法に記載されているFTA，FMEA，HAZOPが挙げられている．これらを中心にした既存の障害分析手法について述べる．

3.1.1　FTA(Fault Tree Analysis)

FTAは，分析対象であるシステムに起こり得る危険事象を想定し，それをトップ事象に置き，これを発生させるシステムの下位の発生要因に順次展開することにより，障害と要因の因果関係を明らかにする手法である[28, 29, 30]．このとき，上位事象と下位事象の関係をブール論理（主として論理的AND，論理的OR）を用い，図3.1で示すフォールトツリー図として表現する．フォールトツリー図は，分析対象である特定の危険事象をトップ事象とし，これを発生させる原因事象に展開する．さらに，この原因事象の原因となる事象というように展開を繰り返し，根本事象となる基本事象まで分解していく．

FTAでは，トップ事象として起こりうる障害をあらかじめ決めておく必要があり，想定できない障害に対しては，分析できない．FTAは，2.8節で述べた要件1に関しては，分析範囲を規定していない．また，要件2については，プール論理

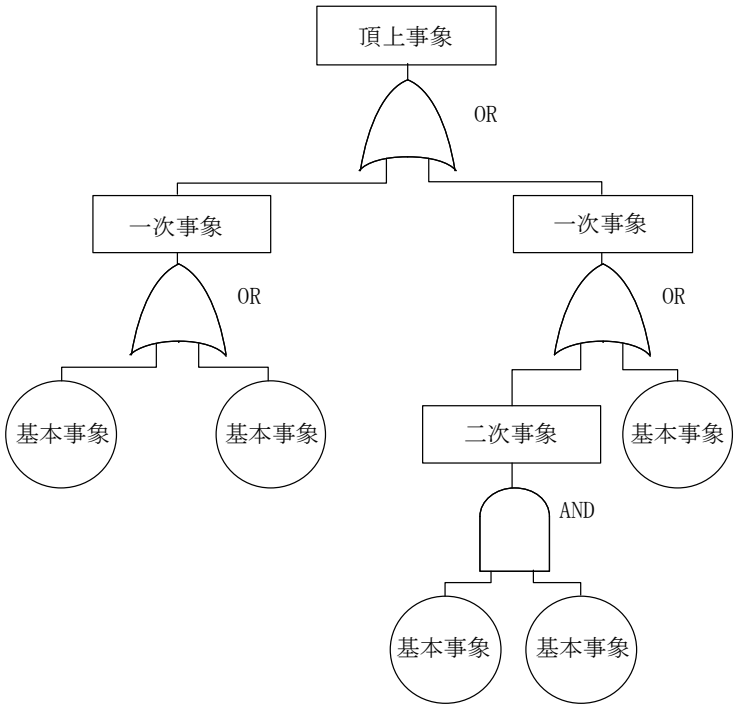

図 3.1: フォールトツリー図

のorを用い，複合要因を抽出できるが，要件3の時間を考慮したシナリオの検討を支援する明示的な手順はない．

3.1.2 FMEA（Failure Mode and Effects Analysis)

FMEAは，設計の不完全や潜在的な欠点を見出すために，構成要素の故障モードとその上位アイテムへの影響を解析する手法である [31, 32, 33]．FMEAは，表 3.1で示すFMEAチャートを用い，対象となるシステムに対して各部品，構成要素に対して故障モードを当てはめ，それらに起こりうる故障を抽出する．この故障モードは，類似部品などに共通した故障要因を抽象化したものである．そして，抽出した故障に対してシステムが起こす可能性のある障害を抽出する．

FTAがトップダウンの分析手法であるのに対し，FMEAはボトムアップの分析手法という違いがある．FMEAは，最初のアイテムとして製品のハードウェア構成をもとに部品故障などを想定している．そのため，一般的に誤使用などのヒューマンエラーや環境条件を考慮しにくいという点がある．

アイテム	機能	故障モード	故障の影響	故障原因	重要度	対策内容	対策結果

表 3.1: FMEA チャート

FMEAは，要件1に関しては，分析範囲を規定しておらず，要件2に対しては，ひとつの故障をもとに上位事象を想定していくため，複合要因を検討しにくく，要件3の時間的要因に対しての明示的な支援方法はない．

3.1.3 ETA（Event Tree Analysis）

ETAは，図 3.2に示すイベントツリー図（ET，事象の木）と呼ばれる樹形図が用いられる [29, 30, 32]．これは，左端に起点として分析対象となる「初期事象（発生した事故事象）」を取り上げ，その初期事象に始まる事故の影響の有無，事故からの進展状況および進展状況に関係する設備等の関係を明らかにし，ツリーとして表す．ETAは，初期事象から右に向かって時系列に事象の進展や対策などを展開していき，その対策が成功（適切な対応）した場合と失敗（不適切な対応，誤作動など）した場合を上下に分岐させる．ETAは，原因となる初期事象がどのような過程で危険事象に進展・拡大するかを時系列に示すもので，初期事象から最終事象までの各段階における対策の問題点を評価するのに有効である．しかし，対応策の効果を成功・失敗の二次元で扱うために，部分的な故障や事故のような，曖昧な事象は考慮できない．また，事故の進展状況を検討する手法であるため，分析対象全体のリスクを把握するのは困難である．ETAは，要件1に関しては，分析範囲を規定しておらず，要件2，要件3も分析の対象としていない．

3.1.4 HAZOP（Hazard and Operability Studies）

HAZOPは，システムの状態を規定する流量や圧力といったプロセスパラメータの目標値や目標状態からの逸脱を想定し，その逸脱の起こる原因と発生する危険事象を解析し，さらにその原因から危険事象に進展することを防護する機

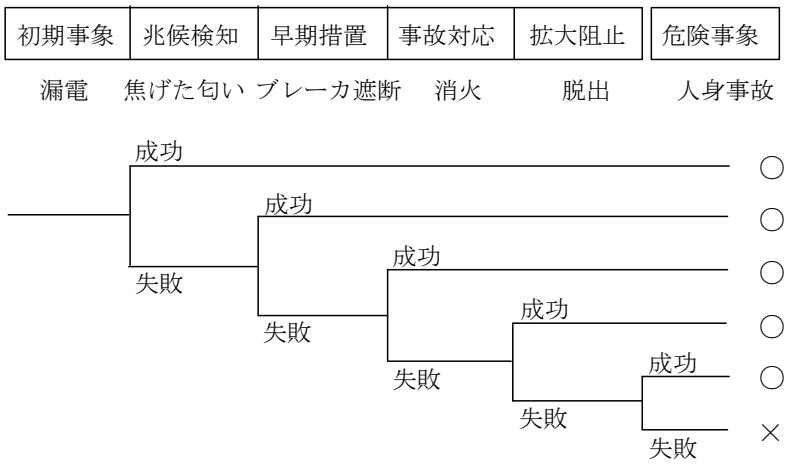

図 3.2: イベントツリー図

能を評価し，対策を検討する手法である[34, 35]．表 3.2に示すHAZOPワークシートを用い，分析点におけるパラメータに対し，正常な値からの逸脱を想定し，想定した逸脱が起こりうるかを検討する．起こりうる要因がある場合は，その逸脱が発生した場合に起こりうる影響を分析し，障害を抽出する．その際，目標値からの逸脱を想定するために，表 3.3に示す「ガイドワード」を用いる．HAZOPは，原因の推定と起こり得る障害の推定を含んでいるが，主に化学プラントにおける薬品などの流れを対象としている．そのため，ガイドワードは，ソフトウェアに着目した分析に向いていない.

HAZOP の応用として，SHARD（Software Hazard Analysis and Resolution in Design）は，HAZOPをもとにガイドワードをソフトウェア向けに整理した手法である[36]．この手法はガイドワードをOmission(サービスが存在しない)，Commission(不要なサービスが存在する)，Early(サービスのタイミングが早い)，Late(サービスのタイミングが遅い)，Value(値の間違い)の5つに限定している．これはBondavalliらによるソフトウェアの故障分類[37]にもとづき考案されたものである．SHARDは，ガイドワードを単純化しており抽象的なため応用範囲は広いが，ガイドワードの解釈のバリエーションを考慮しながら分析を進める必要がある．SHARDは，要件1に関しては，分析範囲を規定していない．要件2に対しては，支援する明示的な手順はない．また，要件3について，逸脱の想定においてのタイミングは考慮されているが，障害に至るシナリオの明示的な手順

分析箇所	考慮する逸脱	原因	起こりうる 結果	防護機能	改善策

表 3.2: HAZOP ワークシート

ガイドワード	意　味	説　　明
no、not	前提の否定	入力が発生しない，応答がない，止まるなど正常な状態を否定する
more	量的増加	期待以上にデータが通過するなど過負荷などを考える
less	量的減少	期待した量のデータが来ないなど軽負荷などを考える
as well as	同様に	想定している入力と類似した入力があった場合など，振る舞いの追加を考える
other than	以外に	想定している入力以外の入力があった場合，対応できないため判断が異常になる場合などを想定する
part of	一部の	設定された情報が不完全など，一部の振舞いが正常ではない場合を考える
reverse	逆	正常な振舞いと逆のことが起こることを考える
early	早く	タイミングが正常な状態から早まった場合について考える
late	遅く	タイミングが正常な状態から遅れた場合について考える
before	前に	順番が早くなった場合について考える
after	後に	順番が遅くなった場合について考える

表 3.3: HAZOP ガイドワード

は，提示されていない．

3.1.5　TRIZ-FP

TRIZ（Theory of Inventive Problem Solving）は，発明・問題解決の定石や，それら定石をどう適用すべきかについて考察する分析手法をまとめたものである [38, 39]．TRIZは，過去の250万件の特許を調査分析し,特許を体系化することにより，難しい技術的な問題の解決を支援するツールのナレッジマネジメントからなる．

TRIZ-FP（Theory of Inventive Problem Solving - Failure Prediction）は，その中のひとつの不具合予測手法として作成された [40, 41]．TRIZ-FPは，システムおよび

その環境にある資源などの不具合を発見するための発想法として，逆転の発想や増幅の発想を用いている．その逆転の発想により「何を発生させることができるか」と思考することで不具合予測を行い，解決案を創り出す手法である．TRIZ-FPは，分析手順のナレッジを示したものである．要件1，要件2,について具体的に述べられていない．また，要件3の障害シナリオの抽出といった動的な分析の支援はない．

3.1.6 SSM（Stress-Strength Model）

SSMは，対象分野のナレッジを作成するための手順であり，トラブル知識のデータベースを構築することにより，不具合予測を行うものである[42, 43]．SSMは，自然科学的なメカニズムの組合せ（因果連鎖）にもとづき，不具合の最終的な形態は異なっていても，その因果連鎖に含まれるメカニズムの本質部分の多くは様々なアイテムに共通している点に着目した．SSMは，製品や工程に起こりうるトラブル発生メカニズムの知識を将来の設計・計画に再利用できるように構造的に表現したモデルである．SSMは，要件3の障害に至るシナリオについて考慮されているが，静的な構造劣化などの変化の追跡であり，分析者の知見に依存し，動的な分析の支援は行わない．また，要件1，要件2,について具体的に述べられていない．

3.2 先行研究

本節では，3.1節で述べた既存手法の応用研究を含めた先行研究における課題を明らかにし，本研究の位置づけを明確にする．

3.2.1 安全分析手法

STPA（STAMP - Based Process Analysis）は，宇宙航空研究開発などのハザード分析として，STAMPをもとにした分析プロセスを定義したハザード解析とモデリングの手法である[44, 45, 46]．STAMPは，安全性のための分析手法のため，事前に重要安全障害となりうる事象を特定する．これをもとに，システムの制御構造図における要素間の相互作用による不適切な制御アクションを，４つの着

目キーワード”Not provided”,”Incorrectly provided”, ”Provided too early, too Late, or out sequence”,”Stopped too soon”を用いてハザードシナリオを識別し，そのシナリオにつながる潜在原因の識別を行うという手順で分析が行われる．STPAは，重要安全を対象とした分析を行うために，起こりうる障害を特定できるため，トップダウン的な分析手法を行う．本研究の目的は，未知の障害を抽出することを目的とするため，STPAのアプローチとは異なる．STPAは，ネットワークを含む広域システムなどの大規模系システムを中心に考えられており，要件3の詳細なタイミングや状態変化などの分析支援は行っていない．また，要件1，要件2，について具体的に述べられていない．

3.2.2 時間や状態遷移に着目した研究

紫合は，システムを構成する制御モジュールの状態モデルに対して分析を行い，状態モデル上には現れない状態に遷移する可能性を調べることにより障害を抽出する方法を提案した[47]．しかし，個別のソフトウェアモジュールを分析対象としており，システム全体を分析対象とはしていない．この研究は，要件3は満たしているが，限定的なソフトウェアモジュールを対象としているため，要件1を満たしていない．また，1つの逸脱を対象とし，要件2を満たしていない．

SMHA（State Machine Hazard Analysis）は，状態遷移図を対象に，初期状態から遷移可能なすべてのパスを網羅的に探索し，深刻な危害をもたらす状態に遷移するパスを見つけ出す手法である[48]．これらの手法は，状態遷移図に，あらゆる障害が発生した場合の状態と，その状態へ遷移するパスが網羅的に記述されていることを前提としている．本研究の目的は，1章で述べたように，想定できていない障害リスクを抽出することが目的である．そのため，障害が起こる状態や遷移するパス自体を抽出することが課題である．したがって，本研究目的の想定していない障害リスクを発見するための手法ではない．

SASTD（Safety Analysis method based on State Transition Diagram）は，状態遷移図を対象に，状態遷移図の各状態で満たされるべき性質が満たされないという逸脱と，状態が遷移する際に実行されるべき処理が実行されないという逸脱を，HAZOPのガイドワードを用いてより網羅的に列挙するための手法である[49]．また，SAHSTD（Safety analysis method based on hierarchical state transition diagram)

は，状態遷移を階層化することにより分析を効率化した[50]．しかし，SASTD及びSAHSTDは，要件3の障害シナリオの分析を追うことはできるが，要件2の複合的な障害分析の支援は行っていない．また，要件1については，言及していない．

ペトリネットは,非同期的でかつ並列的にふるまうシステムに対して,その中の情報の流れや制御を記述し解析するために考えだされたものである．ペトリネットは，離散事象システムを条件(condition)と事象(event)を基本としてモデル化し,数学的解析を可能にする[51, 52]．このペトリネットを用いた障害分析手法が研究されている[53]．Frederickらは，確率ペトリネット（Stochastic Petri Net）を用い，初期の故障から，ペトリネットに沿ってどのような障害がどの様な確率で起こり得るかを分析した[54]．Rezaらは，FTAを用い，トップダウン的に障害要因を抽出する方法と，ペトリネットを用いてボトムアップ的に障害の可能性を抽出することを組み合わせることにより障害分析を行った[55]．ペトリネットを応用した研究は，並行的に動作する構成要素間の故障連鎖についての検討を行えるため，要件3を満足している．しかし，１つの事象を中心に分析するため，要件2の複合的な障害分析の分析に対しての支援は行っていない．また，要件1については，言及していない．

3.2.3 FTA，FMEA，HAZOPを応用した研究

システムの障害を分析しソフトウェアに対する要件を抽出するために，HAZOPを用いた障害分析の手法が研究されている．平山らは，システムの障害要因を抽出することを目的とし，ソフトウェアを設計する段階において，データフロー図及びUMLにHAZOPのガイドワードを適用した事例を研究している[56]．また，Redmilらは，これらのガイドワードを，ソフトウェアの設計に適用した事例を紹介している[57]．この手法は，ユースケースにおける事前条件や，ガード条件，事後条件に対してガイドワードを適用することと，ユースケースの記述を用いて障害を表現することが特徴である．これらHAZOPを用いた分析手法では，ガイドワードを対象に適用した場合に分析者が設計意図からの逸脱を想定できることを前提としている．そのため，分析者の能力によって分析の精度が大きく異なる．また，ガイドワードを適用した結果，想定した逸脱が発生する可能性や，想定した逸脱が開発対象システムに与える影響については別の方法を用

いて検証する必要がある．これらのFTA，FMEA，HAZOPを応用した研究では，それぞれ要件2の複合的な障害分析や要件3の障害シナリオの分析に対しての支援は行っていない．また，要件1については，言及していない．

また，FTAを用いた安全分析の研究も行われている[58]．Fengらは，ソフトウェアを設計する段階においてFTAとFMEAを用い，トップダウンとボトムアップの両方向からの安全分析を行っている[59]．Fengらの手法は，障害のイベントをトップ事象とし，FTAを用いることにより下位のイベントを抽出する．また，Fengらの手法は，FMEAを用い，構造図とシーケンス図におけるデータやイベントに対して故障モードを用いて障害要因を抽出し，起こり得る障害を抽出する．その際，期待するデータからの逸脱を想定するために，データとイベントに対する故障モードを定義した．この定義されたデータの故障モードとして，「incorrect value」,「absent value」,「wrong timing」や「duplicated value」を用いる．また，定義されたイベントの故障モードとして,「halt/abnormaltermination」,「omission」,「incorrect logic/event」や「timing/order」を用いる．Fengらの手法は，FTAとFMEAのそれぞれの欠点を改善できるが，要件2と要件3の逸脱の連鎖の過程で複数の要因が結合することによる障害の支援は行っていない．また，要件1については，言及していない．

3.2.4 コンピュータによるモデル検証

コンピュータを用いたモデル検証により，安全性を検査する手法が研究されている[60, 61, 62]．モデル検証は，モデル記述言語を用い，分析対象のモデル化を行う．このモデルについて，定義された反例に相当する振る舞いを網羅的に探索し，モデルの欠陥を発見する．モデル検証において抽出される障害は，モデルの構造の欠陥であり，たとえば，ある処理に対するガード条件の矛盾やガード条件が抜けているなどである．このため，モデル検証では，モデルに定義されていない障害を発見することを対象とはしていないため，モデル検証を行うためには，あらかじめ，障害と定義する反例を与えておく必要がある．一方，本研究は，分析者が想定していない製品の障害を発見することが目的であり，モデル検証と目的が異なり，手法の適用フェーズが異なる．そのため，本研究の結果として得られた障害を反例としてモデル検証に適用することは可能である．

3.2.5 情報ダイアグラムを用いた研究

新屋敷らは，情報フローダイアグラムという障害リスク抽出のための手法を研究している[63, 64, 65, 66]．この研究は，製品内の装置や，動作環境に存在する利用者などのオブジェクトに対し，その物理的なつながりと，情報のつながりを統合した情報フロー・ダイアグラムの記法を用いて分析する．このダイアグラムを用いれば，製品の例外条件として大きな位置を占める物理的要因が，製品のプロセスに与える影響を形式的に追跡できる．しかし，1つの障害シナリオごとに分析シートを作成する必要があり，シート間の関係の分析は困難である．そのため，新屋敷らの手法は，要件2の複合要因による障害を抽出する支援は行わない．また，要件1は満たしているが，要件3の時間的な変化を含む障害シナリオを抽出する支援も行わない．

3.2.6 セキュリティ分析を中心とした研究

Ian Alexanderらは，ミスユースケースで非正常な使い方をする利用者などの開発対象システムに対する敵対的なアクタをネガティブエージェントとして定義し，逸脱を与える攻撃の振舞いをUMLのユースケースを用いて記述し，その影響を分析する[67, 68, 69, 70, 71]．それにより，許容できないリスクに対する対応方法をソフトウェアの要求として定義する．Ian Alexanderらの研究は，要件1を満足し，正常な入力以外の入力による影響の与え方とシステムの欠陥を総合的に分析している．しかし，セキュリティの脅威や弱点は静的に捉えており，要件3障害シナリオを明示的に支援する手順はない．また，要件2の複合要因による障害を抽出する支援も行わない．

3.3 考察

既存の分析手法は，ソフトウェアを含む製品の特徴である離散的な振る舞いを行う対象を意図して開発された手法ではない．このため，時間的な特性変化などの要因を明示的に扱った分析手法はない．既存の手法では，障害あるいは，故障要因を起点に，トップダウン的，あるいはボトムアップ的な手順を用いて段階

的に影響の範囲を特定し分析を進めているものが多い．障害に関わる検討事項の範囲が広いため，このような段階的な手順は有効と考える．

先行研究において提案されている手法の中には，時間的な遷移を考慮した分析も行われているものがあり，状態やイベントに着目している．これらの研究では，FMEAで用いられている故障モードの考え方や，HAZOPのガイドワードの考え方を応用し，UMLの状態遷移図やシーケンス図などから逸脱を抽出している．このため，ソフトウェア設計図面の精度が高くなった時点での分析を対象としているため，本研究の目的としているソフトウェア要求仕様書作成前の設計フェーズと異なっている．

また，安全性分析においては，重要な安全を対象とした分析であり，原子力や自動車などにおける人命に直接関わる障害を想定した分析が多い．このようなシステムでは，逸脱の発生は，重大なリスクを伴うため，故障即安全停止として扱われる．そのため，これらのシステムにおける環境や運用は整備されており，システムが正常な環境でのみ使用される．

しかし，本研究対象の製品は，重大な安全にかかわらない障害と利便性や製品コストのトレードオフが重要となる．このようなシステムでは，悪環境下で異常な運用などの逸脱が頻繁に発生し，複合した逸脱要因が絡みあう確率も高い．また，リアルタイムシステムとして，特に詳細な時間的変化による影響が大きい．そのため，製品では，過負荷，誤使用，部分故障などの正常でも異常でもない逸脱した状況での使用確率が高く，故障連鎖や複合障害に対する発生確率を無視することができない．このような，逸脱した状態による使用を明示的に扱った製品向けの障害分析研究は行われていない．

そこで，我々の提案する障害分析手法は，2.8節に述べた要件をもとに，製品向けの障害分析手法として，以下の従来にはない特徴を持たせる．

特徴1: 分析対象を状態遷移モデルとして捉え，動的に障害に至る過程を分析することによりタイミングに依存する障害の抽出を容易にし，時間変化に対する影響の検討を陽に含む．

特徴2: システムを構成する要素の状態が相互に影響して障害を発生させることに着目した手法として，複合的な要因による障害を動的な分析により容易

に抽出することを可能する.

特徴3: アーキテクチャ設計段階で予見できなかった未知の状態やイベントを抽出し, 想定していなかった障害を抽出することにより仕様の妥当性の確認を行うことを可能とする.

特徴4: 製品であるがゆえに部品故障や誤操作など広く影響要因を捉え, 更に他の要因との関わり合いの中で障害が起こる可能性を抽出することを可能とする.

第4章　分析対象の概念モデル

本章では，本研究で提案する障害分析手法に用いる概念を明確にするため，2章で述べた製品の特徴をもとにした概念モデルについて述べる．はじめに，分析対象の境界について述べ，次に静的構造モデル，論理的な動的モデルについて述べる．さらに，障害分析を行うために，開発ソフトウェアと製品及び周辺環境をひとつのモデルで表すための変換について述べ，最後に障害シナリオの記述について述べる．

4.1　分析の範囲

2章において，製品は，様々な環境や周辺機器からの影響を受けることについて述べた．そのため，2.8節で述べた要件1より，分析対象は，製品と製品に影響を与える装置，人，動作環境全てを含ものとする．以下，分析対象範囲全体をシステムと呼ぶ．これにより，システムは，外部からは影響を受けない閉空間のシステムと見なすことができる．

しかし，製品と製品に影響を与える装置，人，動作環境の範囲は，一律に決定できるものではなく，分析範囲に入るか否かの境界となるシステム化境界を考慮しなければならない．このシステム化境界には，図 4.1に示す2つの境界がある[72].

1つ目は，製品の機能要求の視点からの基本要求に対するシステム化境界であり，システムが基本的な機能を実現するための入力と出力および，それに影響する環境がシステム化境界の範囲に入る．製品開発において，利用者に提供するための機能とそのための製品の振る舞いが決定される．その製品の振る舞いにより，製品が受ける入力と製品が与える出力が決定される．この入力に関わる要素と出力により影響を受ける要素からシステム化境界が決定できる．

2つ目のシステム化境界は，製品の機能を実現するために考慮すべきシステム化境界ではなく，製品の非機能要求を保証するために配慮しなければならないシス

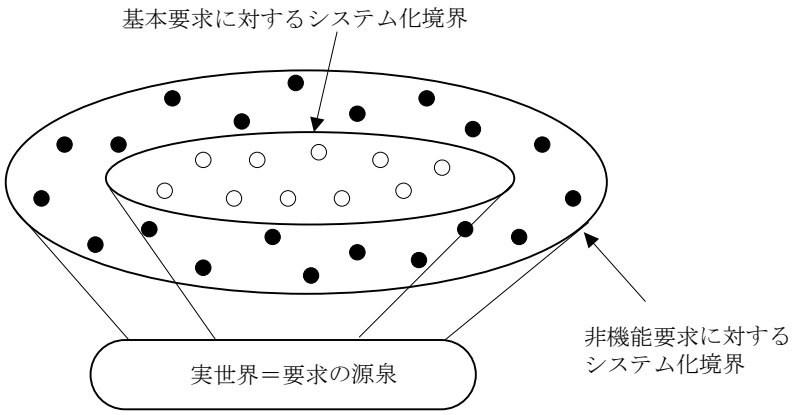

図 4.1: 二つのシステム化境界

テム化境界である．製品には，期待している入力対象ではない要素からの期待していない大きさ，順序，量などの入力が発生する．また，製品からの出力は，期待しない対象に出力してしまう可能性がある．この期待していない対象へ出力することにより，安全性などの非機能要求に影響を与える可能性がある．したがって，製品の障害分析を行う際，このシステム化境界を考慮しなければならない．

しかし，期待しない対象の存在は，製品の機能を実現するためのシステム化境界が定義されることによって，初めて期待されない対象の識別することができる．そこで，このシステム化境界の広がりを把握するために障害要因の分析過程において，システム化境界に立ち戻って検討する必要があることを考慮しておく必要がある．

4.2 静的構造モデル

静的構造モデルとして，システム構造図を図 4.2に示す．システム構造図は，分析対象となるシステムを構成する要素と各要素間の関係を示す．分析対象となるシステムは，機能ブロックである構成要素の集合からなる．この各構成要素は，システムに定義された要件を実現するために構成要素に割り当てられた目的を実現するための機能を持つ．また，システム化境界内の分析対象であるシステムの各構成要素は，他の構成要素と相互に影響している．この構成要素間の相互影響関係は，通信路により接続され，構成要素間の影響は，この通信路を

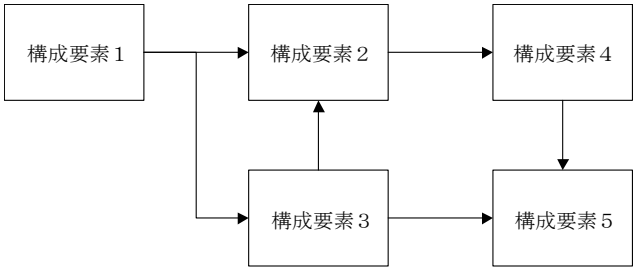

図 4.2: システム構造図

通じて伝えられると捉える．この通信には，メッセージの集合，電気信号，及び温度，圧力などの影響を含む．したがって，システム構造図は，構成要素をノード，通信路をアークとする有向グラフとして表現される．

このシステム構造図の1つの構成要素は，構成要素の外部から観察して1つのまとまった役割を持ち，1つの振る舞いとして表現できるものとする．したがって，1つの構成要素は，1つの状態遷移と見なすことができる．

4.1節で述べたように，2.8節で述べた要件1より，分析対象は，製品と製品に影響を与える装置，人，動作環境全てを含む非機能要求に対するシステム化境界を分析範囲とする．

また，分析対象の規模が多い場合，分析可能な構成要素の数に絞るために，構成要素の集合を1つの構成要素と見なす．この場合，構成要素の粒度について，新規開発要素の場合は，粒度を細かくとり，既存や派生開発による再利用の構成要素は，粒度を荒くすることができる．本分析において，粒度を揃える必要はない．

4.3 論理的な動的モデル

2.8節の要件1より，分析対象は，システム内の個々の構成要素に限定せずにシステム全体とする．また，システムは，周辺機器や利用する人も加えており，各構成要素は，それぞれ独立した状態遷移を持ち，独立して動作する．さらに，要件3より，製品は，同時並行処理のリアルタイムシステムであるため，システムを動的モデルとして捉え，ここでは，システム全体の状態遷移モデルを考える．

動作環境の変化や利用者の操作，構成要素の故障によりいずれかの構成要素

に状態遷移が生じると，システム全体の状態にも遷移が生じる．この構成要素の状態遷移は，通信路により接続される他の構成要素にイベントとして伝播し，その構成要素の状態遷移を引き起こす．これによりシステム全体の新たな状態遷移が発生する．イベントは，構成要素の内部に発生するイベント（以降，内部イベントと呼ぶ）と，通信路により接続された構成要素より伝播するイベント（以降，伝播イベントと呼ぶ）に分類される．4.2節のシステム構造図において，動作環境や人も構成要素としたので，動作環境の変化や利用者による操作は内部イベントと捉える．また，構成要素の故障も内部イベントと捉える．

すなわち，図4.3に示すようにシステム内の各構成要素が固有の状態遷移モデルを持つ状態機械とし，それらが伝播イベントにより通信を行うcommunicating state machinesとして，システムを捉える[73, 74, 75, 76].

そこで，システムの構成要素は独立した状態空間を持ち，システム全体の状態空間は各構成要素の状態空間の直積として表す．即ち，ある時点でのシステムの状態をS_aとし，そのときの各構成要素U_iのx番目の状態をs_{ix}とするとき，

$$S_a = [s_{1p}, s_{2q}, ..., s_{nr}] \tag{4.1}$$

と表される．ここで，nはシステムの構成要素数である．

しかし，communicating state machinesとして定義した形式的な状態遷移モデルにおいて，システムの状態を定義しようとした場合，システムの状態空間を個々の構成要素の状態空間の直積であるため，状態空間が膨大となる．たとえば，各々が4状態を持つ構成要素10個からなる単純なシステムでさえ，式(4.1)のシステムの状態表現（ベクトル）の長さは10となり，状態空間内には4^{10}，すなわち約100万の状態が存在する．分析者がこれら全ての状態を把握し，追跡することは不可能である．

しかし，製品である同時並行に動作するcommunicating state machinesを設計するシステムアーキテクトは，システム全体像を把握しながら設計を行っている．これは，システムアーキテクトがシステム全体像を把握し，構成要素間のインタフェースの情報にシステムの動作シナリオを対応付けた上で，そのインタフェースと構成要素の動的な振る舞いを局所的に検討しているためである．システムアーキテクトが全体の動作を概念的に把握できる理由は，実システムに

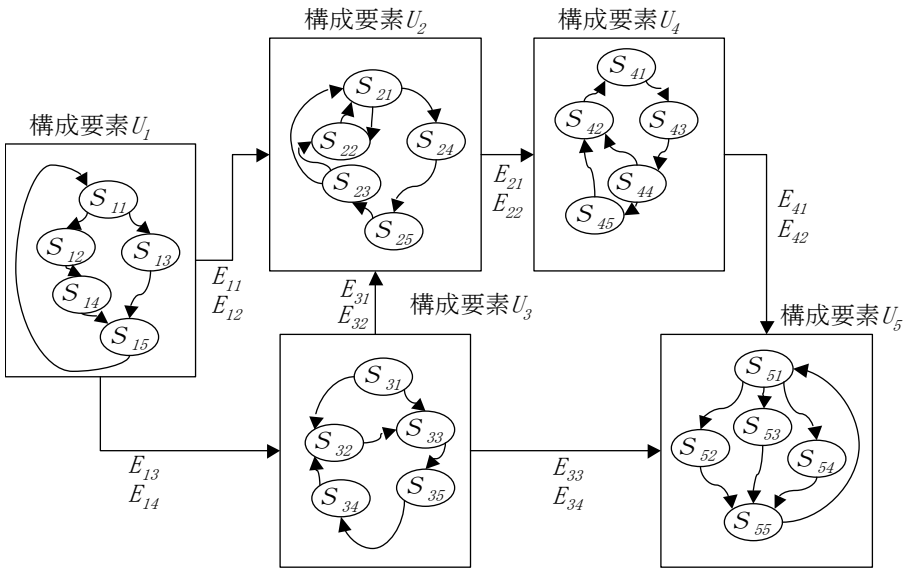

図 4.3: communicating state machines

おける正常な状態が全状態空間中，ほんのわずかであり，また，分析者が全ての観点を同時に分析しているのではなく，着目している局所的な領域における関心のある状態，状態遷移，イベントもわずかであるためである．

大部分の状態は，分析者にとって関心が無いか，把握する必要の無いものである．たとえば，正常な振る舞いを把握していることを前提としたとき，ガス給湯器において,「ガスが点火している」場合，ガスが供給されており，水道の蛇口が開かれ，湯が出ていることが想像できる．

以上から，システムアーキテクトは，正常系の状態遷移は把握していることを前提に，着目している構成要素の状態で，システムの状態を代表するものとする．これにより，この構成要素の状態で代表したシステムの状態と構成要素間の伝播イベントにより，システムの状態遷移表が成り立つ．ただし，記載上の省略であり，以降での状態の検討時には，システムの状態として，その代表した状態を取り得るシステムのすべての状態を想定しているものとする．

そこで，状態及び状態遷移の意味的側面を捉え，膨大な状態空間から分析者に関心のある意味のある状態のみを抽出し，それに他の状態を統合，あるいはその状態で状態の集合を代表させ，状態空間全体の規模を縮小させる．具体的には，正常なシステムの振る舞いにおいては，構成要素の状態でシステムの状

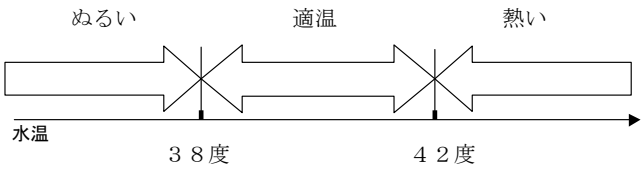

図 4.4: 風呂の湯の同値分割

態を代表する．この考え方にもとづき，システムの動的モデルとして，システムの状態を列に，構成要素間の伝播イベントを行にした状態とイベントのマトリクスを考える．このシステムの状態は，1つの構成要素の状態によりシステムの状態を代表したものである．また，状態とイベントの交点のセルには，発生する伝播イベントと遷移する状態を記述する．

4.4 周辺環境の動的モデルへの統合

要件1より，分析対象は，製品だけでなく，ハードウェアや利用環境，利用者等も含むこととした．この分析対象である開発対象のソフトウェアは，離散的な特性を持つ．一方，分析対象のハードウェアや周辺環境には，温度や圧力などの連続的な特性を持つ対象も存在する．したがって，システムには離散的な特性と連続的な特性が混在している．そこで，システムを動的モデルに統合するために，連続的な特性を持つ構成要素の振る舞いを離散化して捉える必要がある．

そのため，同値分割法を用いて，連続的な特性を持つ構成要素の振る舞いを，離散化した状態遷移で表す[77, 78]．また，連続した特性を持つ構成要素間の通信に対しては，離散化された状態間の遷移を契機とした他の構成要素に伝播する一過性のイベントと扱う．

同値分割法は，対象となる連続的な特性に対して構成要素の外部特性からの視点，すなわち，対象構成要素外から捉えた影響の視点により，同一の特性と見なせる入力定義域や出力定義域の領域を同値領域とする手法である．これにより，連続的な特性値の領域全体を，同等と扱う有限個の領域に分割する．そして，各領域からの代表値で領域を示す．

たとえば，風呂の温度について図 4.4に示すように,「水温がぬるい状態」,「適

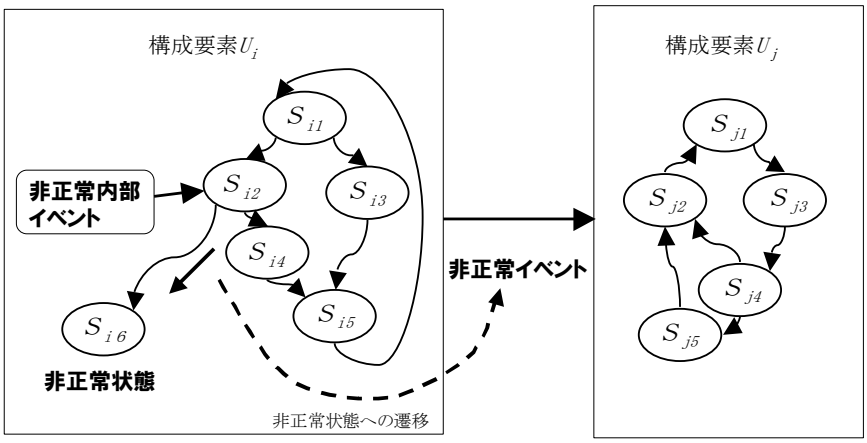

図 4.5: 逸脱の連鎖モデル

切な状態」,「熱い状態」と捉え，これらの状態の変化時に,「ぬるくなった」,「適切な温度になった」,「熱くなった」というイベントが発生したと捉える．

4.5 逸脱から障害に至る振る舞いの動的モデルへの記述

本節では，逸脱から障害に至るモデルを示し，動的モデルへの記述について述べる．次に，逸脱の特性を示し，逸脱を記述するために，特別な扱いをしなければならない記述について述べる．

分析対象を製品に関わる構成要素を全て含む閉空間と捉えた場合，最初の逸脱は，分析対象の構成要素内に発生する．この最初の逸脱の要因は，ハードウェアの欠陥や劣化や故障，利用者の誤操作やいたずら，異常環境などである．この逸脱要因が，逸脱の発生する構成要素の機能を遂行する機能単位の能力の縮退や喪失などとして現れる．さらに，構成要素の逸脱による特性の変化は，他の構成要素に影響を与える．その影響は，さらに他の構成要素に連鎖していく．システムとして捉えた場合，システム内の要因により，逸脱した動作が発生し，その逸脱は，次の逸脱の要因となる．このように，逸脱が連鎖を発生させ，その逸脱シナリオの一部が，利用者に好ましくない影響を与える障害に至る障害シナリオを形成する．

本論文では，このような逸脱を契機とした障害に至る経緯を以下のように扱う．

図 4.5に逸脱の連鎖モデルを示す．正常な振る舞いから逸脱する基点として，

ある構成要素の機能単位の能力の変化や故障，利用者の誤操作などの構成要素内の逸脱要因の発生を構成要素の内部イベントと捉え，これを非正常内部イベントと呼ぶ．さらに，非正常内部イベントは，その構成要素の状態に応じた逸脱した振る舞いを発生させる場合がある．構成要素の逸脱した振る舞いにより，構成要素から他の構成要素へ逸脱した影響を伝播する場合，その構成要素の逸脱した状態はシステムの状態として捉える必要があり，これを非正常状態と呼ぶ．また，構成要素の逸脱した振る舞いにより，他の構成要素に影響を及ぼす場合や，他の構成要素への影響が変化する起こす場合において，それが想定している振る舞いから逸脱している場合，これを非正常イベントと呼ぶ．非正常イベントは，さらに他の構成要素に伝播し，その構成要素の非正常状態への遷移をもたらす場合がある．また，非正常状態は，他からのイベントにより，新たな非正常状態や非正常イベントを発生させる場合がある．このように，非正常イベントや非正常状態は，逸脱を連鎖させていく．

この非正常イベントの通信路には，設計者が予め明示的に意図した通信路以外に，たとえば，ある構成要素の異常な発熱が物理的に隣接する他の構成要素に伝わるなど，後の障害分析により始めて分析者により認識される暗黙的な通信路を含む．

つぎに，逸脱の記述について述べる．逸脱は，システム内の相互関係による逸脱を含めて特定できる記述をしなければならない．正常な振る舞いからの逸脱は，IEEE Std 1044（2002）において，要求仕様，設計書，ユーザ文書，標準，または誰かの認識もしくは体験など，これらにもとづく期待から乖離した状態と定義されている[80]．このように，逸脱は，振る舞いに対して想定している正常な範囲からの大きさ，速度，方向，時間，回数，順序などからの相対的な変化として捉えられる．相対的であるため，システム全体の状態との関係で逸脱が定義される．そのため，逸脱は，単独の構成要素の状態やイベントでは表されない．たとえば，電気ポットにおいて,「ヒータは停止している」という状態は，正常な振る舞いの場合,「ポット容器に水が入り，水温が保温温度にあり，ヒータは停止している」を示している．しかし，正常な振る舞いから逸脱している場合には,「温度センサが故障し，水温が低下しているが，ヒータは停止している」という場合や,「ポット容器内に水がないため，ヒータは停止している」といった場合が存在する．

さらに，逸脱の扱いにおいて，逸脱は，分析者として認識できる事象であるた

め，逸脱の表現は，分析者の認識として扱う．

たとえば逸脱には，途絶えるといった逸脱がある．このため，分析モデルとしては，イベントが途絶える逸脱について,「イベントが発生しない」という分析者の認識としての非正常イベントとして扱う．同様に,「イベントが発生しない」というイベントを受けた場合,「イベントを受信できない」という状態を扱う．

また，逸脱には，時間前後の相互関係により逸脱と判断できる場合がある．たとえば，イベントが，正常な前後の時間間隔を空けずに連続した場合，分析者の認識として，これらの時間的に前後するイベント群をまとめて「発振した」というイベント,「発振している」という状態として捉える．

以上のように扱うことにより，動的モデルに逸脱から障害に至る経緯を記述し，システムの状態遷移とイベントの系列を時系列として逸脱の連鎖を追跡することができ，構成要素の故障等から製品の障害に至る過程を表現することが可能になる．

4.6 考察

本節では，本章に提案した，製品の逸脱分析のための概念モデルを考察する．具体的には，2.8節で述べた，障害分析に求められる3つの要件について，本章に提案した概念モデルが満足しているかについて考察する．

要件1として，製品だけでなく，環境や利用者等も分析対象として含む必要性から，非機能要求に対応したシステム化境界として，利用者や動作環境を分析対象に含める．この結果，システム化境界の中に連続的な特性と離散的な特性が混在することになり，これらの統合が必要となった．このため，同値分析法を用いて連続的な特性を離散的な特性に変換し，分析対象を状態とイベントで表す定義を行った．

また，要件2として，複合的な要因による障害を分析できる必要がある．本概念モデルにおいて，初期の逸脱を非正常内部イベントとした．また，非正常内部イベントを起点とした逸脱の連鎖として，非正常状態と非正常イベントを表した．そこで，分析モデルの中に，複数の非正常内部イベントを組込み，各非正常内部イベントから連鎖した非正常状態や非正常イベントを展開していく．それにより，異なった要因による非正常状態と非正常イベントの組み合わせの分析が行え，複合要因による障害を抽出することができる．

さらに，要件3として，障害が発生するシナリオ過程の分析が行えることが必要である．逸脱は，構成要素相互関係により定義する必要があるため，動的モデルにおける状態は，構成要素単独ではなく，システムの状態であり，イベントもシステムとしてのイベントでなければならない．しかし，システムの状態遷移は，論理上膨大な数に及ぶ．ただし，現実のシステムの振る舞いにおいて，正常な動作における状態の組み合わせは，僅かであり，構成要素の状態で代表できることを述べた．これにもとづき，システム全体を状態とイベントの動的モデルとして記述でき，その中に非正常状態や非正常イベントを含めることにより，状態遷移の時系列として障害シナリオをモデルに記述することが可能となる．

以上から，本章に提案した概念モデルが，2.8節で述べた障害分析に求められる3つの要件について満足している．

第5章　逸脱要因の抽出

4.5節において，障害は，構成要素の最初の逸脱を基点とした連鎖により発生することを述べた．そこで，本研究の目的である未知の障害を抽出するために，逸脱要因からのボトムアップ分析を中心とした障害分析手法とすることにした．そのため，障害の起点となる逸脱要因の抽出と，逸脱からの連鎖の分析という2つの手順が必要となる．本章では，逸脱要因の抽出について，6章に逸脱からの連鎖の分析を行うために考案した非正常系分析マトリクスについて述べる．そして，それらを踏まえて，7章に一連の障害分析手法であるESIMの手順を述べる．

5.1　故障モードとガイドワードの拡張

構成要素の最初の逸脱要因は，構成要素の物理的な誤動作や故障，人による異常な運用方法，異常な環境など様々である．そのため，これらの具体的な逸脱要因は，構成要素の固有の動作特性に依存する．したがって，逸脱要因を抽出するためには，それぞれの構成要素に対する固有の知識があることを前提とする．しかし，構成要素の固有知識だけでは，逸脱要因を抽出することができない．その理由は，逸脱した振る舞いが，システム全体の動作としての相対的な振る舞いを含むために，構成要素単独の動作視点だけでは，逸脱要因を認識されない場合があるためである．このため，逸脱要因を抽出するために，逸脱要因の発見を支援する手順が必要となる．

既存の逸脱要因を抽出する手法として，3.1節で述べたFMEA，HAZOP，FTAが利用できる．FMEAは，故障モードを用いて故障を抽出する．この故障モードは，故障状態の形式による分類であり，故障そのものではなく故障をもたらす不具合事象の分類である．たとえば，剥離，抜け，緩み，詰まり，外れ，断線，短絡，折損，摩耗などの物理や化学的な変化を故障モードと呼ぶ．このように，故障モードは，ハードウェアの具体的な同一形状や同一材料などの共通要素を持

つ部品に対して定義される．このため，抽象度が低く，部品などの故障を抽出しやすいが，新規の部品には当てはまらない．また，操作ミスなど物理的な部品や機器の故障以外の抽出が困難とされている．

一方，HAZOPは，ガイドワードを用いて設計図面の具体的なパラメータの逸脱事象を抽出し，その逸脱事象から故障を抽出する．HAZOPのガイドワードは，すべてのパラメータに対応できるため，応用範囲が広い．また，抽象度が高いために抽出できる故障は，ハードウェアの部品から，人の振る舞いや環境などの抽出が可能である．しかし，ガイドワードは，抽象度が高いために，具体的な故障の想定には，分析者の知識に依存する．また，想定した逸脱事象から故障を抽出することも分析者の知識に依存する．ただし，逸脱事象から故障への分析に関しては，逸脱事象をトップ事象としたFTAを用いることができる．

本研究では，FMEAやHAZOPを用いて構成要素の故障を逸脱要因として広く捉えることにより，逸脱要因を抽出する．そこで，故障モードの適用範囲をハードウェアから人や環境などのハードウェア以外の構成要素に拡張した．また，故障の抽出から逸脱要因の抽出として利用するために，構成要素を特性ごとの分類に拡張した故障モードを抽出した．構成要素の分類に，ハードウェア構成要素として，CPU周辺回路，電気回路，機構などを抽出した例を表5.1に示す．また，環境や利用者なども一つの部品として扱った故障モードの例を表5.2に示す．

また，ガイドワードについては，英語の文法としてnotなどの抽象的な記載であり，分析者が意味を十分に把握しにくい面がある．そこで，日本語文法の表現形式として，着眼した特性を明確にした言葉に変換した．HAZOPのガイドワードをベースに，変換したガイドワードの例を表5.3に示す．しかし，HAZOPのガイドワードやそれらを応用した研究も含め，ガイドワードをシンタックスとして捉えている．シンタックスと捉えた場合，適用範囲は広いが，そこから抽出される逸脱は広がりすぎる．そこで，ガイドワードをセマンティックスとして捉えたガイドワードの例を表5.4に示す．ガイドワードをセマンティックスとして捉えた場合，逸脱理由を考慮し，意味的な側面から逸脱を考える．このため，逸脱の想定が，あらかじめ発生する理由を含め考えることができ，効率的に抽出できる．ただし，分析者の熟練度が低い場合には，抽出漏れが発生する可能性があるため，分析者により選択する．

種別	項目	故障モード
CPU周辺	割り込み	割り込み消失，多重割り込み，誤割り込み，ノンマスカブル割り込み
	レジスタ、メモリ	レジスタデータ化け，誤書き込み
電気回路	デジタル回路	ラッチミス，データ化け，ポートリセット，チャタリング，停止
	アナログ回路	アンダーフロー，オーバーフロー，発振，劣化による値ずれ，劣化遅延
	ＬＳＩ	モードリセット，誤動作，無応答、
	コネクタ	短絡，断線，接触不良，アース開放
	スイッチ	ＯＮ故障，ＯＦＦ故障，中間位置，チャタリング
	センサ	更新中読み出し，オーバーフロー，アンダーフロー，誤読出，停止
	周辺機器	機器故障無応答，通信中のリセット，ローカルモード
機構	駆動部	詰まり，挟まれ，バックラッシュ，こすれ
	移動体	衝突，スリップ，ブレーキ故障，初動負荷
	筐体	汚れ，割れ，ずれ
	動力伝達	たるみ，ねじれ，グリス切れ，ベルト切れ，磨耗
	接続部	緩み，ずれ，接触不良，固着

表 5.1: ハードウェアの故障モード

5.2 逸脱要因の抽出手順

最初にシステム構造図における構成要素間の通信であるイベントに着目する．各イベントに対し，表5.3に記載したガイドワードをそれぞれ当てはめ，逸脱事象を想定する．イベントには，イベントの内容が持つパラメータとして大きい，小さい以外に，イベントの数が多い，少ない，イベントの順序が，前に，後に，また，イベントを発生している構成要素が正しい，偽りなど多方面に検討する．次に，各イベントに対し，表5.4に記載したガイドワードをそれぞれ当てはめる．このガイドワードでは，具体的な運用を想定しながら起こりうる逸脱事象を想定する．

イベントにガイドワードを当てはめた結果，想定した逸脱事象に対して，FTAのトップ事象に記載したフォールトツリー図を作成し，物理的に起こりうる逸脱要因を抽出していく．FTAの分析の中で，複合要因の抽出が行える場合もあるが，それらも独立して逸脱要因として列挙する．

次に，構成要素に対し，表5.1あるいは，表5.2から構成要素の種類にあった故障モードを当てはめる．故障モードは，共通的な用語として記載されているため，実施の構成要素に合わせた具体的な逸脱要因を抽出する．

種別	項目	故障モード
初期化	周辺機器同期	周辺機器と接続遅延，初期化中の通信
操作	入力操作	無効操作，連続操作，同時操作（ﾘﾓｰﾄ/ﾛｰｶﾙ，多箇所，自動/手動）
	いたずら操作	連続操作，スイッチ（二度押し・渡り押し・同時押し），矛盾操作
	登録・設定	登録箇所漏れ，重複登録，無効設定，範囲外設定，矛盾値設定
	操作対象機器	機器異常，未接続，誤接続，誤機種接続，範囲外登録，矛盾設定
負荷	過負荷	急負荷，連続過負荷，限界をわずかに超えた過負荷，異常過負荷
	軽負荷	負荷はずれ，スリップ
通信	通信機器	接続台数オーバー，接続先停止，応答拒否，異機種接続
	プロトコル	ネゴシエーションリセット，未定義プロトコル，通信中中断
	通信線	通信高負荷，断線，接続不安定，
	無線	ハンドオーバー
使用	用途	適正外用途による使用
ﾒﾝﾃﾅﾝｽ	周辺機器	機器整備不良（プリンタ紙切れなど）
環境	電源	瞬時停電，連続瞬時停電，長期停電，広域停電，電圧低下
	加速度	ゆれ，振動，衝撃
	温度	高温，低温，急激な温度変化
	電磁波	ノイズ，強電磁波，放送設備近隣，違法移動無線局
	湿度	結露，水蒸気
	天候	雨，雷，ヒョウ
	虫	ハエ，蚊，蟻，蜘蛛
	障害物	壁，段差，鉄製ドア
施工	配線	誤結線，重複アドレス，アドレス設定ミス，
	設置	位置ずれ，ゆがみ，方向ずれ

表 5.2: 運用に関する故障モード

最後に，ガイドワードから抽出した逸脱要因と故障モードから抽出した逸脱要因の重複を省いた表を作成する．ただし，抽出した逸脱要因の中には，構成要素から外部へ影響を及ぼさない対策が既存の技術として確立しており，実用実績があるものに対しては，表から削除する．

たとえば，スイッチ入力に対して，チャタリングというスイッチON時の信号電圧のバウンディングにより，10msec以内のON／OFFが繰り返される現象が発生する．しかし，入力チャタリング処理手順は，従来から対策されている処理であり実績もあり，かつ障害例もない場合，逸脱要因として抽出しない．

着眼パラメータ	ガイドワード
有無	有／無
大きさ	強／弱　大／小　厳しい／緩い　激しい／穏やか
速度	急／ゆっくり
精度	精／粗
時間	長／短
方向	正／逆
範囲	余分／不足
相対時間	遅く／早く
発生時間	同期／非同期
順序	前に／後に
回数	多く／少なく
対象	正／偽
データ構造	一致／不一致

表 5.3: 拡張ガイドワード（シンタックス）

5.3　考察

本章の逸脱要因の抽出の手順が，2.8節に記載した障害分析の要件に充足しているかについて考察する．

逸脱要因の抽出の手順に関係する要件は，要件1の分析対象として，製品だけではなく，環境や利用者も分析対象として含むという項目である．逸脱要因は，製品内だけではなく，環境や利用者にも存在し，ハードウェアの誤動作や故障から，使われ方や環境など様々である．特に運用面の障害要因は，製品の目的や機能，利用される状況など範囲が広く，共通的でかつ抜け漏れのない手順が困難である．そのため，扱われる商品形態ごとに逸脱要因の抽出のための知識を蓄積していけることが重要となる．

製品のハードウェアに関しては，FMEAにおける故障モードが当てはまり，表5.1に例を示している．しかし，人という物理的な構造ではなく，運用的な逸脱要因が存在する．そこで，故障モードと同様に，運用面における共通的なキーワードとして，負荷的側面，手順的側面，施工的側面などを洗い出し，運用面の故障モードとし，表5.2に運用面から捉えた故障モードを示した．ただし，

ガイドワード	
境界値	断線/復帰
同時・多重	電源変動・瞬停
競合・衝突	初期化
優先	放置
公差	設定変更
無効	順序変更
輻輳	範囲外
構成変更	重複
過負荷・集中	再起動
例外・意地悪	モード移行
失敗	処理切替
中断	意図外

表 5.4: 拡張ガイドワード（セマンティックス）

FMEAの故障モードと同様の欠点を持ち，わかりやすい反面，適用範囲が狭い．そのため，すべての製品の特性を網羅することができない．そのため，扱われる製品の特性ごとに，故障モードの拡張を行っていく必要がある，

一方，ガイドワードは，抽象的であるため，ハードウェアの故障に制限されず，運用面を含む多面的な逸脱要因の抽出が行える．しかし，逸脱要因の抽出の網羅性は分析者の知識に依存し，ガイドワードとして知識の支援を行うことが困難である．そこで，シンタックス的なガイドワードに対し，セマンティック的なガイドワードを追加した．セマンティック的なガイドワードでは，パラメータが逸脱する可能性のある例外的な運用などが発生した場合に陥り易い障害要因を抽象化してまとめたものである．これにより，分析者の知識に依存せず，知識の蓄積と支援が行えることを配慮している．

以上の故障モードによる逸脱要因を直接想定する方法と，ガイドワードを用いた逸脱という現象の想定から逸脱要因を想定する2通りの手順の相互補完により，ハードウェア，環境，運用などの幅広い構成要素に発生する逸脱要因の抽出の抜け漏れを削減している．

第6章　非正常系分析マトリクス

本章では，4章で述べた概念モデルをもとにした障害シナリオ抽出のための非正常系分析マトリクスの概要とその構造，分析手法について述べる．この非正常系分析マトリクスは，5章で述べた分析手順により抽出した逸脱要因をもとに障害シナリオを抽出する．

6.1　非正常系分析マトリクスの概要

非正常系分析マトリクスは，製品の逸脱した動作の抽出のために，筆者らが実務的に行っていた手法を，4章に述べた周辺環境や逸脱した振る舞いの記述を統合した動的な概念モデルをもとに，理論的に再構築したものである．

非正常系分析マトリックスは，状態遷移表を拡張した手法である．一般的な状態遷移表は，状態欄とイベント欄のマトリクスであり，有限オートマトンで記述可能な記述対象の振る舞いを表したものである．状態遷移表の状態は，記述対象が取りうる状態を示し，イベントは，記述対象内，あるいは分析対象外から発生するイベントを示す．状態遷移表の状態とイベントの交点には，状態欄に示された状態において，イベント欄で示されるイベントを受けた場合に，発生する振る舞いと遷移する状態を記述する．状態遷移表は，有限オートマトンで記述可能な記述対象の振る舞いを表現するために用いられる．

一方，非正常系分析マトリクスは，4.2節で述べたcommunicating state machinesにおいて，着目する構成要素の状態で代表したシステムの状態を状態欄とし，構成要素内部及び構成要素相互間の伝播イベントをイベント欄とした表である．非正常系分析マトリクスは，構成要素の状態に着目しながら，システム全体を捉えた状態と構成要素間のイベントによる動作を分析する．また，非正常系分析マトリクスは，4.4節で述べた連続的特性を持つ構成要素の振る舞いも離散的な状態とイベントに変換し記述する．さらに，非正常系分析マトリクスは，想定

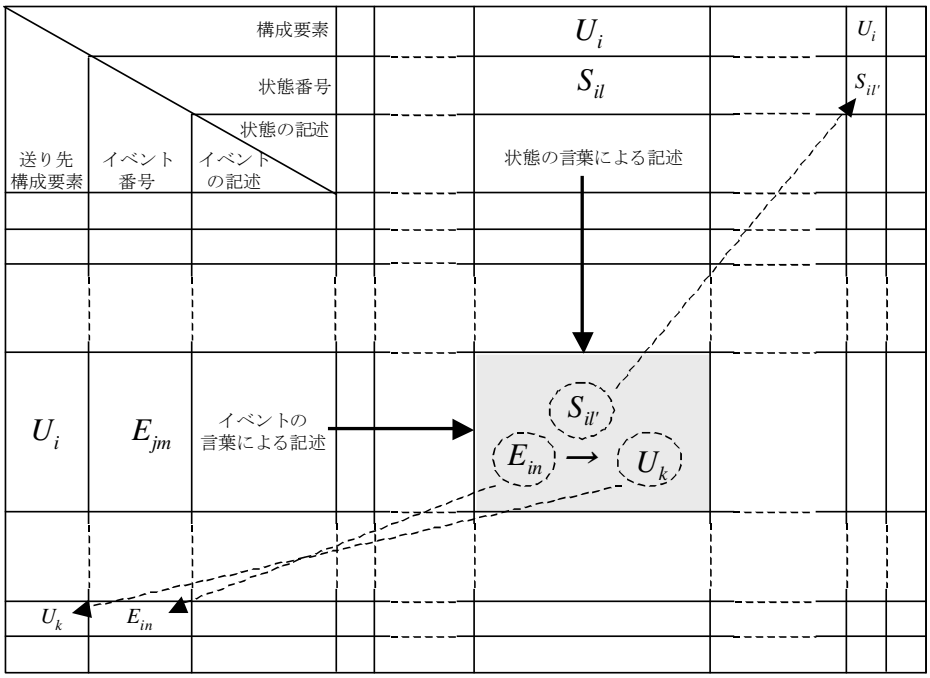

図 6.1: 非正常系分析マトリクス

されていないイベントや状態を発見するための分析者の認識を記述した表である．そのため，4.5節で述べた，分析者の認識である「イベントが発生しない」というイベントを含む正常な振る舞いからの逸脱状態や逸脱イベントも記載する．この非正常分析マトリクスを用いて想定していない状態やイベントを発見した場合，状態欄やイベント欄を拡張していく．このため，分析者の関心により状態の抽象度に差異を設けた状態遷移表である．即ち，正常状態については，粒度の大きな抽象的な状態として捉え，非正常状態は，細かく分析するために求められるレベルまで具体化した小粒度の状態として捉える．

6.2 非正常系分析マトリクスの構造と記述

4章で述べた論理的動的モデルを用いて作成した非正常系分析マトリクスの構造を図 6.1に示す.

まず，マトリクスの列は，構成要素U_iの状態のひとつに着目したシステム状態S_{il} を表現する．以降，マトリクスの列を状態欄と呼ぶ．状態の記述は，分析者の観点にもとづく自然言語でなされる．ここで,「U_iの状態のひとつに着目した

システム状態」とは，分析の主対象となる構成要素の観点から，その構成要素の状態を含むシステム全体の状態を表現したものである．これは，非正常状態の場合，障害の要因となる非正常内部イベントからこの構成要素における非正常状態への状態遷移が集約された記述である．

たとえば，ポット内の湯の温度を温度センサにより検知し，ヒータを制御し，一定温度範囲に保温する電気ポットにおいて,「湯」,「温度センサ」,「ヒータ」という構成要素を持つとする．そこで,「ヒータ」が「温めていない状態」を考える．この場合,「ヒータ」という構成要素そのものの状態は,「温めていない状態」である．しかし，システムとしての「温めていない状態」は,「既に保温温度を保持しているため，温めていない状態（正常状態)」や,「湯が保温温度を下回る温度の状態であるが，センサが故障しているため状態変化の信号を受信せず，温めていない状態」,「ヒータそのものの故障により温めていない状態」など複数の状態が含まれている．そのため，これらの状態は，異なった状態として明確に記述しなければならない．

ここで,「湯が保温温度を下回る温度の状態にあり，センサが故障しているためヒータが温めていない状態」は,「（ヒータ停止から時間が経過し）湯が保温温度を下回る温度の状態にあり，センサが故障しているため（状態変化の信号を送らない状態にあり)，ヒータが（センサが保温温度を上回る温度の信号を最後に受け付けた状態を保持し）温めていない状態」と言うシステム全体の状態記述において，括弧内を省略した短い記述に集約された形式とする．これが,「ヒータ」の「温めていない状態」に着目したシステムの状態である．短い記述に集約された形式にする目的は，分析上，理解できる範囲の省略した記述により，分析を効率的に進めるためである．このような状態記述は，構成要素の非正常状態への状態遷移や非正常イベントを追跡することにより可能となる．

次に，マトリクスの行は，送信側の構成要素U_jにおいて発生した状態遷移により，通信路で接続された受信側の構成要素U_iに伝播するイベントE_{jm}を表す．また，構成要素の逸脱要因の影響として発生する構成要素U_hの非正常内部イベントも併記する．非正常内部イベントの記述は，E^i_{hn}とし，E^iは非正常内部イベントと表す．非正常内部イベントの送信先構成要素は，それ自体の構成要素である．以降，マトリクスの行をイベント欄と呼ぶ．イベントの記述も，分析者の観

点にもとづく自然言語でなされる．

更に，状態とイベントの交点であるセルには，ある構成要素U_iがそのl番目の状態S_{il}にあるとき，その送信側構成要素U_jからそのm番目のイベントE_{jm}を受け取ったときに遷移する状態$S_{il'}$と，その受信側構成要素U_kに伝播するイベントE_{in}と伝播先U_kを記述する．また，複数の状態や複数のイベントが発生する場合には，セルの中に併記する．U_jからU_iに通信路がない場合，このセルは空白となる．

6.3 分析手順

1) 分析開始時の非正常系分析マトリクスの作成

分析開始時の非正常系分析マトリクスとして，各構成要素のすべての正常状態を状態欄に記載する．また，構成要素間のすべての正常イベントをイベント欄に記入する．さらに，構成要素が持つ固有の故障や利用者の誤操作などの5章で述べた手順により抽出したすべての逸脱要因は，4.5節で述べた非正常内部イベントとしてイベント欄に記入する．

非正常系分析マトリクスにおける正常状態は，構成要素U_iの正常状態を含み，他の構成要素$U_j(j=1,2,...)$が正常状態s_{j1}にあると仮定したシステムの正常状態$[s_{11}, s_{21}, ..., s_{i1}, ...]$であるが，$U_i$の正常状態の記述をもってシステムの正常状態を表現する．

2) 非正常系分析マトリクスにおける逸脱連鎖の分析

分析開始時の非正常系分析マトリクスの作成後，各構成要素内での非正常内部イベント，及び送信側構成要素より伝播するイベントによる状態遷移を分析する．なお，非正常内部イベントは，5章で述べた分析手順により抽出済である．すなわち，前項で記入した各構成要素U_iの各非正常内部イベントE^i_{ix}の発生，及び送信側の構成要素U_jから構成要素U_iに伝播するイベントE_{jx}について，状態S_{il}から派生する新たな振る舞いの逸脱を分析する．分析結果は，状態とイベントの交点のセルに，この新たな振る舞いの逸脱を非正常状態$S_{il'}$への遷移として記入する．

ここで新たに遷移する状態の記述も,「構成要素U_iの状態のひとつに着目したシステム状態$S_{il'}$」となる．ただし，U_iがS_{il}の状態にいる時間帯と，E_{jx}が生じる時間帯が異なれば，U_iはE_{jx}の影響を受けないので，セルは空欄とする．

このように，新たに遷移する状態は非正常状態となる．この非正常状態が，ここまでに分析者により把握されていない状態，すなわち，マトリクスの状態欄に記入のない状態であれば，新たにこの状態欄を含む1列を非正常系分析マトリクスに追加し，その状態を記入する．

次に，この状態遷移$S_{il} \rightarrow S_{il'}$について，構成要素外への影響の変化を検討し，ここで発生する非正常イベントE_{in}とその伝播先U_kを記入する．

また，この新たな状態$S_{il'}$が異常発熱や機械的異常振動のように物理的に接する構成要素に非正常イベントとして伝播する場合，あらたな通信路が把握できたことになる．

この伝播イベントE_{in}がここまでに分析者により把握されていないイベント，すなわち，イベント欄に記入のないイベントであれば，新たにこのイベント欄を含む1行をマトリクスにを追加し，そのイベントを記入する．このように，分析が進むに従って，マトリクスの列と行, すなわち，状態欄とイベント欄が追加されて行く．

セルを分析した結果，セルに発生する非正常状態が回避すべき障害であると判断した場合には，そのセルにおける分析はそこで終了し，その後に続く状態遷移を想定することは不要である．ここで，障害と見なされたセルへ至るイベントの系列が障害シナリオとなる．

以降，この手順を再帰的に繰り返す．非正常イベントあるいは非正常状態が関わるセルすべてにおいて，新たに抽出する非正常状態がなくなった時点で分析を終了する．

非正常系分析マトリクスにおいて，記載する状態やイベントは，構成要素間に影響を与える構成要素の状態と構成要素間のイベントのみを扱っている．ただし，非正常系分析マトリクスのセルにおける検討においては，構成要素内に閉じた状態やイベントであっても分析の対象としなければならない．これは，構成要素内に閉じた状態やイベントが逸脱の連鎖の影響により，想定していなかった影響が起こる場合があり，分析過程で，構成要素内に閉じた状態やイベントが，逸脱を契機に構成要素間に影響を与える状態やイベントに変化する場合も存在するためである．

たとえば，制御指示を受けて起動あるいは停止する構成要素において，初期起動時の状態は，他の構成用要素に影響を与えないと判断し，構成要素内に閉

じた状態としていた．この理由は，起動開始から安定状態に入るまでの初期起動状態に他の構成要素から処理要求が発生しないという判断のためであった．ただし，その構成要素は，初期起動時の状態において処理が行えない．しかし，分析過程において，この構成要素に対し，起動開始直後に処理を要求するという逸脱した振る舞いがあることが判明した場合，この構成要素の初期起動時の状態を非正常系分析マトリクスに追加する．

また，セルの分析において，抽出できた逸脱事象は，機能が実現できないあるいは，動作が遅いや精度が悪いなどの非機能要求が満足できない事象になっている可能性がある．しかし，これらの機能要求や非機能要求が満足できなくとも，その事象に至った経緯により，必ずしも障害とは見なされない．それは，利用者側の原因が明確であり，かつ避けられない場合などにおいて，完全に停止するよりは，望ましい事象である場合などである．したがって，逸脱している事象が障害と判断するか否かは，事象ごとに判断しなければならない．

3) 障害シナリオの抽出

非正常系分析マトリクスには，障害の要因となる全ての故障や誤操作から製品の障害に至るまでの非正常イベントや非正常状態の連鎖が記載されている．非正常状態や非正常イベントは，必ずそれを発生させるもとになった非正常状態や非正常イベントが存在するため，製品の障害と判断した構成要素の非正常状態から，分析とは逆順で追跡することにより，障害の要因となる非正常内部イベントにたどり着くことができる．すなわち，構成要素の故障や利用者の誤操作などの逸脱の要因となる非正常内部イベントを起点とした製品の障害に至るイベントの系列として障害シナリオを記述することができる．

6.4 考察

非正常系分析マトリクスの実用可能性について考察する．

3章において，既存の障害分析手法や研究中の手法において，故障要因から段階的に分析を進める手法が多いことについて述べた．これは，障害分析を行う際に，障害に関して検討しなければならない要因が多く，思考範囲が広いことによる分析の困難さを回避するためである．そこで，非正常系分析マトリクスの思考範囲について考察する．

障害は，構成要素間の相対的な逸脱を含むため，システム全体として判断する必要がある．しかし，多くの構成要素の変化を同時に分析することは困難である．そこで，非正常系分析マトリクスは，ひとつの構成要素に焦点を当てている．非正常系分析マトリクスの状態は，代表する構成要素の１つの状態とその構成要素に送られるひとつのイベントである．逸脱は，システムの各構成要素の相互関係において発生するため，システムとして記述されるが，非正常状態や非正常イベントの記述の簡略化と考慮する範囲の絞込みにより対応している．非正常系分析マトリクスの状態とイベントの交点であるセルの分析からの結果は，次の構成要素への逸脱の伝播である．これにより，セルの分析において新たに発見された非正常状態と非正常イベントにより，非正常系分析マトリクスを展開していくことは，障害シナリオを断片的に追っていることになる．

これにより，非正常系分析マトリクスは，空間的には，1構成要素に焦点を当て，時間的には，ある構成要素のある状態という特定された時間に絞った分析が行えている．そして，セルの分析を再帰的に行うことにより，段階的にシナリオを構築していく．

また，非正常系分析マトリクスに記載する状態やイベントは，構成要素間に影響を与える状態やイベントのみを扱っている．ただし，非正常系分析マトリクスのセルにおける検討においては，構成要素内に閉じた状態やイベントも検討している．非正常系分析マトリクスの記述において，状態やイベントを絞ることにより，記述を簡略化し，検討の負荷を軽減している．

非正常系分析マトリクスは，以上の分析手順の配慮により，3章における既存の障害分析手法や研究中の手法よりも，分析を細分化している．このため，思考範囲を狭めることによる分析の容易性を配慮した．一方，分析を細分化したことによる分析時間は増加する．この分析時間が実用的な時間で実施できることについては，9章において検証する．

第7章　障害分析手法ESIM

ESIMは，4章で述べた概念モデルを用い，5章で述べた最初の逸脱の抽出と6章で述べた非正常系分析マトリクスによる障害シナリオ抽出の一連の手法である．

本章では，5章と6章を前提に，逸脱要因の発見から障害シナリオ抽出までの一連の分析手順について述べる．なお，本章において，手順に焦点を当てるため，単一の逸脱要因に限定した極めて簡単な事例を用いて説明する．このため，2.8節で述べた要因２の複合要因に対する分析については，8章の具体的な事例の中で述べる．

7.1　事前条件とESIM実施後のプロセス

ESIMは，製品の正常系のシステムアーキテクチャを設計した後，システムに発生する可能性のある障害リスクを把握するために障害シナリオを抽出する手法である．ESIMによる障害分析を行うために，前提とする事前条件とESIMによる分析後の想定するプロセスについて述べる．

（事前条件）

ESIMを実施する前のシステムアーキテクチャ設計完了までに以下のことは既知である．

- システムの目的達成のために定義されている機器の正常機能
- 構成要素ごとの正常な動作における状態遷移モデル
- システムの構成とその構成要素間の接続関係
- 構成要素間の正常な通信の内容と意味
- 構成要素の状態と構成要素間の通信の正常な場合の組み合わせ
- 構成要素における固有の故障，及び利用者の誤操作の種類と性質

（ESIM適用後のプロセス）

ESIMの分析結果として幾つかの障害シナリオが抽出される．その発見した障害シナリオに対して，発生する確率，発生した障害の影響度を配慮し対策を行うか否かの検討を行う．その検討には，機能の縮退，安全性，信頼性，利便性，製品価格などの観点で顧客に最適なトレードオフを検討する．新たに対策機能を追加する場合は，障害回避機能として機能仕様に追加する．しかし，この回避機能が実施されている状態という新たな状態が発生するため，新たなリスクを発生させる可能性がある．そこで，この機能が有効であるかについて，この機能を追加したことによる正常機能の追加や差分をESIMに適用し，変更した機能に抜け落ちがないかを確認する．

7.2 ESIMの分析手順

7.1節で述べた事前条件のもとに，障害シナリオを抽出する手順について述べる．

7.2.1 正常機能にもとづくシステム構造図の作成

アーキテクチャ設計による情報にもとづき分析を開始する．この時点において，既に障害を意識し障害回避機能も一部的に組み込まれているが，それらは，障害回避という正常な機能として扱う．

はじめに，4.2節で述べた図4.2に示すシステム構造図を作成する．システム構造図は，システムを構成する要素と各要素間の関係を示し，四角形で囲んだ構成要素と構成要素間の通信を矢印で示した有向グラフとして表現する．

システム構造図に記載するシステム化境界は，製品に影響を与える，あるいは製品から影響を受ける環境や周辺機器，利用者を含む．これらの構成要素は，アーキテクチャ分析時に検討されたハードウェア構成や環境から抽出する．なお，新規開発のために既存情報の少ない構成要素の粒度は細かく，既存や派生開発の構成要素は，粒度を荒くすることができる．本分析は，障害抽出を目的としているため，分析対象の構成要素の粒度を揃える必要はない．

ESIM手順を述べるために単純な保温機能のみを持つ仮想の電気ポット事例を用いる．電気ポットのシステム構造図を図7.1に示す．電気ポットは，電気ポット

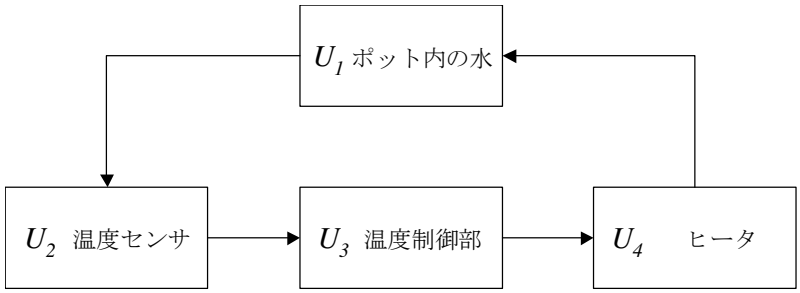

図 7.1: 電気ポットのシステム構造図

容器内の水を温度センサで検知し，その情報から温度制御部が，ヒータに指示を出し，ポット容器内の水を一定範囲の温度に保つものである．なお，処理を簡単にするため，沸騰開始後の一定時間沸騰を続けた後に保温温度になるような機能は省略し，一定温度に保つだけの機能とする．開発対象は，温度制御部のソフトウェアであるが，温度制御部に影響を持つ周辺ハードウェアとして，温度センサ，ヒータ，ポット容器内の水も分析対象とする．

以下に，電気ポットの構成要素の役割と構成要素間の関係を示す．

U_1 ポット内の水：

電気ポット内に収容されている水を示す．ヒータにより加熱され，温度センサに温度を検知される．

U_2 温度センサ：

電気ポット内に収容されている水の水温を計測し，計測結果を温度制御部に伝える．

U_3 温度制御部：

温度センサから水温情報を得て，保温するために加熱制御の判断を行う．判断結果より，ヒータに加熱指示を行う．

U_4 ヒータ：

温度制御部からの指示により，電気ポットの水を加熱する．

7.2.2 正常状態，正常イベントの抽出

このシステム構造図のひとつの構成要素は，外部から観察してひとつのまとまった役割を持ち，ひとつの振る舞いとして表現でき，ひとつの状態遷移と見な

構成要素		正常状態		正常イベント	
U_1	ポット内の水	S_{11}	保温温度を下回る温度	E_{11}	保温温度を下回る温度になった
		S_{12}	保温温度	E_{12}	保温温度になった
		S_{13}	保温温度を上回る温度	E_{13}	保温温度を上回る温度になった
U_2	温度センサ	S_{21}	保温温度を下回る温度を検知している	E_{21}	保温温度を下回る温度を検知した
		S_{22}	保温温度を検知している	E_{22}	保温温度を検知した
		S_{23}	保温温度を上回る温度を検知している	E_{23}	保温温度を上回る温度を検知した
U_3	温度制御部	S_{31}	保温温度内と判断し，加熱制御を停止している	E_{31}	加熱停止を制御指示
		S_{32}	保温温度を下回ったと判断し，加熱制御を実行している	E_{32}	加熱を制御指示
U_4	ヒータ	S_{41}	加熱している	E_{41}	加熱した
		S_{42}	加熱を停止している	E_{42}	加熱を停止した

表 7.1: 電気ポットの正常状態と正常イベント

すことができる．そこで，システム構造図に記載された各構成要素における状態とイベントを抽出する．正常状態において，システムの状態は，代表する構成要素の状態構成要素の状態とするため，抽出した正常状態は，システムの状態と見なす．構成要素内の状態遷移や，構成要素の外部から観察した役割や振る舞いについては，前提条件として既に得られている．

事例において,「ポットの水の温度」という特性は，連続的特性であるが，同値分割法を用い離散的特性に変換しなければならない．そこで,「ポットの水の温度」に対して「保温を行う」という役割に着目し，同様の性質を持つ特性領域に分割する．その結果,「保温温度を下回る温度」,「保温温度」,「保温温度を上回る温度」の3状態を抽出する．また，これらの状態の間を遷移するイベントとして,「保温温度を下回る温度になった」,「保温温度になった」,「保温温度を上回る温度になった」を抽出する．以上をもとに抽出した電気ポットの各状態とイベントを表 7.1 に示す.

7.2.3 逸脱の要因となる非正常内部イベントの抽出

構成要素間の各イベントに対し，5.1節で述べた表5.3と表5.4のガイドワードを適用し逸脱事象を想定する．しかし，ガイドワードを適用して逸脱事象を抽出したとしても，実際に，そのような現象が生じないのであれば，分析対象として取り上げる必要はない．

構成要素「ポットの水」から構成要素「温度センサ」に送られるイベントである「保温温度になった」に表5.3のガイドワードを適用した例を表 7.2に示す．イベントの持つパラメータに対して，ガイドワードの各項目を当てはめ，そこから想定できる逸脱事象を列挙する．

列挙した想定した逸脱事象は，FTAを適用し，逸脱要因の抽出を行う．そのため，ガイドワードを適用して想定した逸脱事象を頂点事象としたFTA分析により，起こり得る構成要素の逸脱要因を検討する．図 7.2に表 7.2で想定した逸脱事象である「温度が急激に上昇する」をトップ事象にしたFTA分析の例を示す．この結果,「高温の水を投入」と「水がない」が抽出できる．

次に，システム構造図の構成要素に対して，構成要素がハードウェアの場合には表5.1を，人などの運用の場合には表5.2の対応する故障モードの表を割り当て，逸脱要因を抽出する．表 7.3に表5.1の故障モードを割り当てた表を示す．

イベントに対して，ガイドワード用いる手順と，構成要素に対して故障モードを用いる手順は，それぞれの手順の特徴の相互補完であるため，二つの手順から抽出された逸脱要因は，重なる場合が多い．二つの手順から抽出された逸脱要因は，構成要素の非正常内部イベントとして以降の分析に用いる．

電気ポットの逸脱要因の事例として，構成要素「温度検出部」における回路故障による「温度検出停止」「異常高温を検出」「異常低温の検出」などが抽出される．しかし，本節における事例説明の簡単化のために，構成要素「ポットの水」の「水がなくなる」という逸脱事象のみが発生するものとする．この「水がなくなる」という逸脱事象を構成要素「ポットの水」の非正常内部イベントとする．

なお，4.1節において述べたように，分析が進むにつれ，システム化境界が明確になっていく．そのため，この逸脱の要因を抽出する手順の中で，当初検討出来ていなかった構成要素の存在や，構成要素間のイベントのつながりを発見する場合がある．この場合，システム構造図の作成に戻り，再度全体構成から再検討を行う．

イベント	ガイドワード		抽出した逸脱
E_{12} 保温温度になった	有無	有，なし	温度が変化しない
	符号反転	正，負	————————
	速度	速く，遅く	温度が急激に変化する
	幅	長く，短く	————————
	同期／非同期	同期，非同期	————————
	間隔	長く，短く，同時に	————————
	回数	多く，少なく	————————
	順序	前に，後に	————————
	量	大きく，小さく	温度が異常高温になる
	送信元	異なった	水温以外の温度を受ける
	タイミング	矛盾した	————————
	データ構造	不一致部分	————————

表 7.2: ガイドワードの適用事例

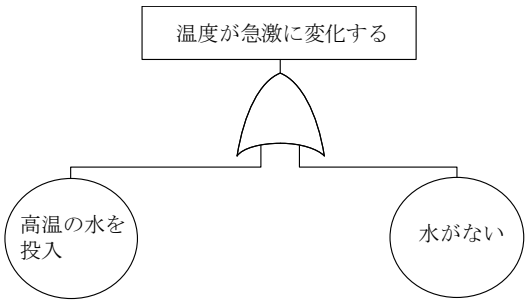

図 7.2: FTA の適用事例

7.2.4 分析開始時の非正常系分析マトリクスの作成

前節で抽出した，正常状態，正常イベント，非正常内部イベントをもとに分析開始時の非正常系分析マトリクスを作成する．

最初に，非正常系分析マトリクスの状態欄を記述する．記載する状態はシステムの状態であるが，記述が複雑になるため，正常状態においては，構成要素の状態を代表し，システムの状態とする．たとえば，U_4のヒータにおいて,「加熱している」という正常状態は,「U_1のポットの水が保温温度を下回り，U_2の温度センサから，保温温度を下回っているという情報を受け，U_3の温度制御部が保温する判断を行いヒータに対して加熱の指示を出している」という状態であるこ

No	構成要素	種別	項目	故障モード	逸脱要因
U_1	ポット内の水	負荷	過負荷	急負荷，連続過負荷，限界をわずかに超えた過負荷，異常過負荷	水が溢れる
			軽負荷	負荷はずれ，スリップ	水がなくなる
U_2	温度センサ	電気回路	センサ	更新中読み出し，オーバーフロー，アンダーフロー，誤読出，停止	誤った値を読み出す 故障により異常に低い値を示す 故障により異常に高い値を示す センサ停止
U_3	温度制御部	CPU周辺	割り込み	割り込み消失，多重割り込み，誤割り込み，ノンマスカブル割り込み	センサ値を読み飛ばす
			レジスタ， メモリ	レジスタデータ化け，誤書き込み	誤ったセンサ値を書き込む
U_4	ヒータ	電気回路	アナログ回路	アンダーフロー，オーバーフロー，発振，劣化による値ずれ，劣化遅延	ヒータの加熱が低下する

表 7.3: 故障モードの適用事例

とは，一義的に想定出来る．

そのため，システム構造図の各構成要素から抽出した状態をシステムの状態として，状態欄に記載する．また，着目している構成要素の記号を記載する．さらに，構成要素記号とその構成要素における状態の添字を付けた状態記号も記載する．構成要素記号は，U_mとし，mは構成要素の添字とする．また，状態記号はS_{mn}とし，mは代表する構成要素の番号，nは，その構成要素が持つ状態の添字とする．

次に，イベント欄に正常イベントと非正常内部イベントを記載する．イベントは，システム構造図における構成要素間の伝播イベントを記載する．イベント記号はE_{mn}とし，mは，代表する構成要素の番号，nはその構成要素から発生するイベントの添字とする．非正常内部イベントには，構成要素の逸脱要因をもとにイベントの表現として記載する．非正常内部イベントの記述は，E^i_{op}とし，E^iは非正常内部イベントを示し，oは代表する構成要素の番号，pはその構成要素内の非正常内部イベントの添字とする．非正常内部イベントの送信先構成要素は，それ自体の構成要素である．

表 7.4に分析開始時の非正常系分析マトリクスを示す．7.2.2項で抽出した正常状態を状態欄に，7.2.2項で抽出した正常イベントと7.2.3項で抽出した非正常内部イベントである構成要素「ポット内の水」におけるE^i_{11}「水が空になった」をイベント欄に記載した．

送り先構成要素	イベント番号	構成要素 / 状態番号 / 状態の説明 / イベントの説明	ポット内の水 U_1			温度センサ U_2			温度制御部 U_3		ヒータ U_4	
		状態番号	S_{11}	S_{12}	S_{13}	S_{21}	S_{22}	S_{23}	S_{31}	S_{32}	S_{41}	S_{42}
		状態の説明	保温温度を下回る温度	保温温度	保温温度を上回る温度	保温温度を下回る温度を検知している	保温温度を検知している	保温温度を上回る温度を検知している	加熱制御を停止している	加熱制御を実行している	加熱している	加熱を停止している
U_2	E_{11}	保温温度を下回る温度になった										
	E_{12}	保温温度になった										
	E_{13}	保温温度を上回る温度になった										
U_3	E_{21}	保温温度を下回る温度を検知した										
	E_{22}	保温温度を検知した										
	E_{23}	保温温度を上回る温度を検知した										
U_4	E_{31}	加熱停止を制御指示										
	E_{32}	加熱を制御指示										
U_1	E_{41}	加熱した										
	E_{42}	加熱を停止した										
	E^i_{11}	水が空になった										

表 7.4: 分析開始時の非正常系分析マトリクス

7.2.5 非正常系分析マトリクスにおける逸脱連鎖の分析

分析開始時の非正常系分析マトリクスにおける，状態とイベントの交点であるセルの分析を行い逸脱連鎖を抽出する．非正常系分析マトリクスにおける状態欄の各状態は，特定の構成要素U_xの状態で代表している．また，イベント欄の各イベントは，送信先の構成要素U_yが定まっている．非正常系分析マトリクスにおけるセルを示すイベントの送信先構成要素が，状態を代表している構成要素と一致している場合のみ，イベントを受け取ることが可能であり，考慮しなければならないセルである．すなわち，状態欄を代表している構成要素U_xのxと送信先の構成要素U_yのyが等しいセルだけが考慮しなければならないセルである．

正常な状態と正常なイベントは，結果として正常な状態に遷移し，正常なイベントを発するため，新たな状態やイベントは発生せず，最初に記載したどこかの状態やイベントが選ばれる．イベントあるいは状態の少なくともいずれかが非正常イベントあるいは非正常状態である場合には，新たな非正常状態あるい

は新たな非正常イベントが発生する可能性がある．この非正常状態が，ここまでに分析者により把握されていない状態，すなわち，マトリクスの状態欄に記入のない状態であれば，新たにこの状態欄を含む一列を非正常系分析マトリクスに追加し，その非正常状態を記入する．また，非正常イベントが，ここまでに分析者により把握されていないイベント，すなわち，マトリクスのイベント欄に記入のない状態であれば，新たにこのイベント欄を含む一列を非正常系分析マトリクスに追加し，その非正常イベントを記入する．

なお，状態欄を代表している構成要素U_xのxと送信先の構成要素U_yのyが等しくとも，既にイベントを受けた状態にあり，同一のイベントが無意味である場合や，発生したイベントと受け取る状態の時間帯が矛盾する場合には，考慮する必要はなく，セルに－を記入する．

以降，この手順を再帰的に繰り返す．なお，セルにおいて想定された構成要素の振る舞いの逸脱が製品の障害と見なされた場合，対策が必要となるため，そのセルに対してはそれ以降の分析は行わない．ここで，セルが製品の障害と見なされれば，そこへ至るイベントの系列が障害シナリオとなる．非正常イベントあるいは非正常状態が関わるセルすべてにおいて新たに抽出する非正常状態がなくなった時点で分析を終了する．

表 7.5 に事例の非正常系分析マトリクスを示す．

逸脱の非正常内部イベントE^i_{11}は，自らの構成要素「ポット内の水」U_1の状態に作用する．E^i_{11}が，S_{11}の場合には，非正常状態「保温温度を下回る温度において水がない」S_{14}に遷移する．また，S_{12}の場合には，非正常状態「保温温度範囲において水がない」S_{15}に遷移し，S_{13}の場合には，非正常状態「保温温度を上回る温度において水がない」S_{16}に遷移する．

構成要素「ヒータ」の正常イベント「加熱した」E_{41}は，構成要素「ポット内の水」U_1のS_{14}とのセルにおいて，非正常イベント「急激に温度が上昇した」E_{14}を発生する．また，構成要素「ヒータ」の正常イベント「加熱を停止した」E_{42}は，構成要素「ポット内の水」U_1のS_{16}とのセルにおいて，非正常イベント「急激に温度が下降した」E_{15}を発生する．

非正常イベントE_{14}は，構成要素「温度センサ」U_2の正常状態「保温温度を下回る温度を検知している」状態S_{21}とのセルにおいて，非正常状態「急激な温度

上昇を検知している」S_{24}と非正常イベント「急激な温度上昇を検知した」E_{24}を発生させる．また，非正常イベントE_{15}は，構成要素「温度センサ」U_2の正常状態「保温温度を上回る温度を検知している」状態S_{23}とのセルにおいて，非正常状態「急激な温度下降を検知している」S_{25}と非正常イベント「急激な温度下降を検知した」E_{25}を発生させる．

非正常イベントE_{24}は，構成要素「温度制御部」U_3の正常状態「加熱制御を実行している」S_{32}とのセルにおいて，非正常状態「急激な温度上昇を検知し，加熱を停止している」S_{33}を発生させる．非正常イベントE_{25}は，構成要素「温度制御部」U_3の正常状態「加熱制御を停止している」S_{31}とのセルにおいて，非正常状態「急激な温度下降を検知し，加熱している」S_{34}を発生させる．

水がない場合，ポット内の水は，加熱を停止することにより急激に温度が下降し，加熱することにより急激に温度が上昇する．このため，非正常イベント「急激な温度下降を検知した」E_{25}と非正常状態「急激な温度上昇を検知し，加熱を停止している」S_{33}の交点において，保温制御が発振するという事象を障害と見なし，このセルの分析は中止する．

以上において，各イベントが発生した場合に，セルにおいてどのようなことが発生するかについての知識，あるいは，水がない場合，ポット内の水は，加熱を停止することにより急激に温度が下降し，加熱することにより急激に温度が上昇することについては，知識として持っていることを前提としている．しかし，構成要素の固有知識だけでは，逸脱の連鎖を把握することはできない．それは，システム全体の動作としての逸脱であり，構成要素単独の視点では，逸脱と認識されない場合や，些細な逸脱が，システムとしてどのような影響を受けるか把握してないために，逸脱の連鎖として認識されない場合があるためである．そのため，非正常系分析マトリクスにおいて，検討範囲を限定し構成要素間の逸脱情報を与えることにより，逸脱の発見が可能になる．

7.2.6 障害シナリオの抽出

表 7.5におけるU_2の非正常状態「急激な温度上昇を検知し，加熱を停止している」S_{33}が非正常イベントE_{25}を受け取ることにより発生した「制御が発振」という障害は，回路に劣化を与えると同時に，不要な電力を消費させる．E_{25}とS_{33}のたセ

ルを起点に，分析と逆方向で，非正常イベントあるいは非正常状態を追跡し，最初の逸脱要因である非正常内部イベントまで遡る．この障害シナリオを図 7.3に示す.

ESIMは，communicating state machinesをモデルとし，個別の構成要素の状態に着目しながら，システムとしての状態を分析している．このため，障害シナリオは，単純な非正常イベントの系列で示すことができない．そこで，図 7.3に示すように，個別の構成要素のイベント系列を横断するイベント系列として示している．円は，構成要素の代表による状態を示し，矢印は，内部イベントあるいは，構成要素間の伝播イベントを示す. 矢印の上のスラッシュの左側には，矢印が示すイベント，スラッシュの右側には，そのイベントにより，その代表する構成要素に接続される構成要素へ送られる伝播イベントを示す. また，その伝播イベントが送られる先を点線の矢印で示す.

ESIMにおける障害シナリオは，一般的なシナリオにおけるイベント系列とは異なるが，システム内に発生する逸脱が，構成要素間をどのように連鎖していくかについての時間的な経過の表現が行える．

障害シナリオを抽出できたことにより，急激な温度変化が発生した場合，製品を異常と判断し加熱を停止する処理を入れることができる．この処理は，加熱制御の発振を停止させることにより，耐久性と省エネルギー性を向上させる．

7.3 考察

ESIMが障害分析手法として2.8節に記載した3つの要件に適合していることについて考察を行う.

(要件1)

開発対象は，温度制御部のソフトウェアであるが，周辺ハードウェアとして，温度センサ，ヒータ，ポット容器内の水も分析対象とした．このため，ポット容器内の水は，連続的な特性を持ち，同値分析により,「保温温度を下回る温度」,「保温温度」,「保温温度を上回る温度」の状態と「保温温度を下回る温度になった」,「保温温度になった」,「保温温度を上回る温度になった」というイベントに変換することにより分析を行えることを述べた．この結果，開発対象ではない構成要素「ポットの水」からの逸脱により,「制御が発振」という障害が抽出できることを述べた．このように，ESIMは，周辺環境を分析対象として取り込み，それ

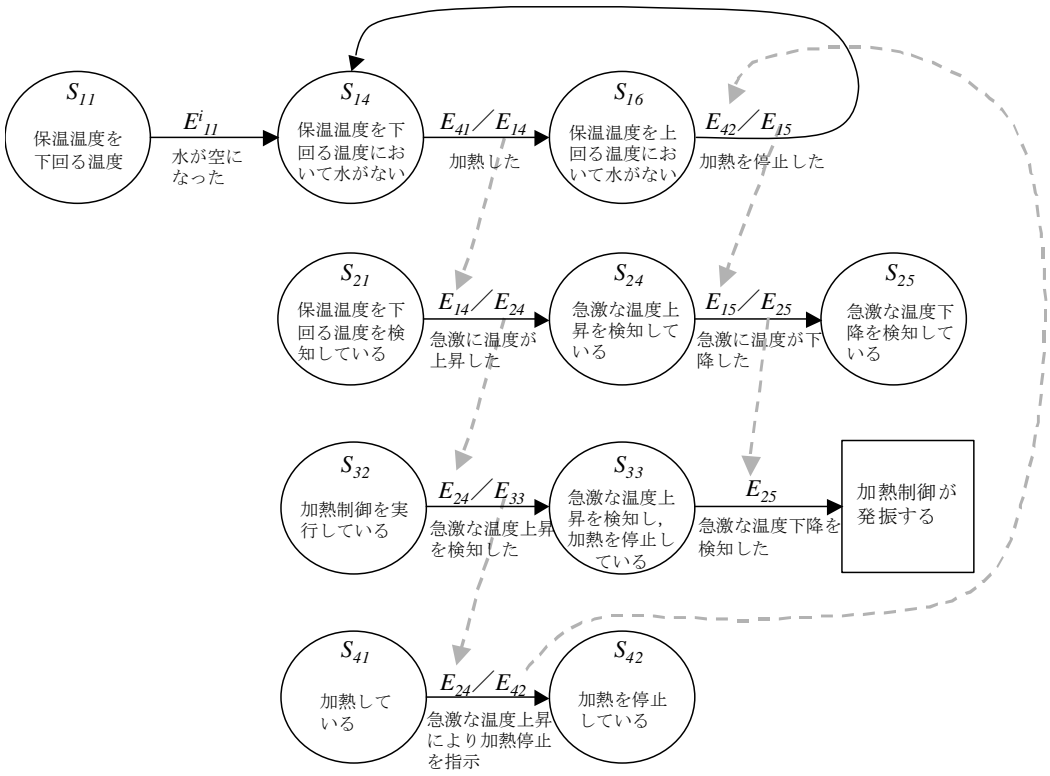

図 7.3: 事例の障害シナリオ

により課題となる連続的な特性と離散的な特性の混在に対して明確な手順を持ち，障害抽出が行えることを示した．

また，ESIMは，ハードウェア，人，環境などに関わる初期の逸脱要因抽出のために，FMEAやHAZOPから拡張した故障モードとガイドワードを使用している．これにより，開発対象以外の逸脱要因を加えて分析を行えている．

(要件2)

本章では，逸脱要因を一つとしたため，複合要因が分析できる事の確認は，次章にて行う．

(要件3)

非正常系分析マトリクスは，初期の逸脱要因を非正常内部イベントとし，システムの状態遷移を構成要素の状態に着目しながら，イベントの発生を発見し，さらにそのイベントによる影響を確認するという再帰的な手順でシナリオを追跡している．事例において，逸脱した状態や「ポット内の水」の非正常内部イベント「水が空になった」を起点に状態とイベントを追跡していくことにより,「制

御が発振」という障害に至るまでの障害シナリオを抽出できた．

さらに，ESIMが，要件 2 の複合要因が分析できる事について考察する．

ESIMは，逸脱要因である非正常内部イベントをすべてひとつの非正常系分析マトリクスに組込む．これらの非正常内部イベントは，構成要素間に影響を与える非正常状態や非正常イベントに展開される．

このとき，一過性の非正常系イベントや他の構成要素に影響を与えない非正常状態の場合を除き，ひとつの逸脱要因からは,「逸脱している」とういう非正常状態に遷移し，その結果「逸脱している状態に遷移した」という非正常イベントが発生する．この非正常状態と非正常イベントは，両方とも非正常系分析マトリクスに展開される．他の逸脱要因からも同様に，非正常状態と非正常イベントが非正常系分析マトリクスに展開される．これらの非正常状態と非正常イベントは，さらに連鎖し，新たな非正常状態と非正常イベントを発生させる．

このため，非正常分析マトリクス上には，一方の逸脱要因から連鎖した非正常状態ともう一方の要因から連鎖した非正常イベントのセルが存在する．また，同様に，非正常分析マトリクス上には，一方の逸脱要因による非正常イベントともう一方の要因による非正常状態のセルという状態とイベントの逸脱要因が逆になったセルも存在する．これは，一方の逸脱要因による非正常状態が発生している状態において，もう一方の逸脱要因による非正常イベントが後から発生したというセルと，逆の順序のセルが存在することになる．逸脱要因の発生順序に影響のない場合は，同一の事象を発生させるが，同一でない場合は，異なった事象の発生が起こりうる事を確認できる．このように，非正常系分析マトリクスは，逸脱要因の発生順序を含めた複合的な要因による障害の分析を行うことができる．

表 7.5: 事例の非正常系分析マトリクス

送り先構成要素	イベント番号	構成要素 / 状態番号 / 状態の説明 / イベントの説明	ポット内の水 U_1			温度センサ U_2			温度制御部 U_3		ヒータ U_4		ポット内の水 U_1			U_2	U_2	U_3	U_3
		状態番号	S_{11}	S_{12}	S_{13}	S_{21}	S_{22}	S_{23}	S_{31}	S_{32}	S_{41}	S_{42}	S_{14}	S_{15}	S_{16}	S_{24}	S_{25}	S_{33}	S_{34}
		状態の説明 / イベントの説明	保温温度を下回る温度	保温温度範囲の温度	保温温度を上回る温度	保温温度を下回る温度を検知している	保温温度を検知している	保温温度を上回る温度を検知している	加熱制御を停止している	加熱制御を実行している	加熱している	加熱を停止している	保温温度を下回る温度において水がない	保温温度範囲において，水がない	保温温度を上回る温度において水がない	急激な温度上昇を検知している	急激な温度下降を検知している	急激な温度上昇を検知し，加熱を停止している	急激な温度下降を検知し，加熱している
U_2	E_{11}	保温温度を下回る温度になった				—	S_{21}	S_{21}									—		
						—	$E_{21}\rightarrow U_3$	$E_{21}\rightarrow U_3$								$E_{21}\rightarrow U_3$	—		
	E_{12}	保温温度になった				S_{22}	—	S_{22}								S_{22}	S_{22}		
						$E_{22}\rightarrow U_3$	—	$E_{22}\rightarrow U_3$								$E_{22}\rightarrow U_3$	$E_{22}\rightarrow U_3$		
	E_{13}	保温温度を上回る温度になった				S_{23}	S_{23}	—								—	S_{23}		
						$E_{23}\rightarrow U_3$	$E_{23}\rightarrow U_3$	—								—	$E_{23}\rightarrow U_3$		
U_3	E_{21}	保温温度を下回る温度を検知した							S_{32}	—								S_{32}	—
									$E_{32}\rightarrow U_4$	—								$E_{32}\rightarrow U_4$	—
	E_{22}	保温温度を検知した							—	—								—	—
									—	—								—	—
	E_{23}	保温温度を上回る温度を検知した							—	S_{31}								—	S_{31}
									—	$E_{31}\rightarrow U_4$								—	$E_{31}\rightarrow U_4$
U_4	E_{31}	加熱停止を制御指示									S_{42}	—							
											$E_{42}\rightarrow U_1$	—							
	E_{32}	加熱を制御指示									—	S_{41}							
											—	$E_{41}\rightarrow U_1$							
U_1	E_{41}	加熱した	S_{12}	S_{13}	—								S_{16}	S_{16}	—				
			$E_{12}\rightarrow U_2$	$E_{13}\rightarrow U_2$	—								$E_{14}\rightarrow U_2$	$E_{14}\rightarrow U_2$	—				
	E_{42}	加熱を停止した	—	S_{11}	S_{12}								—	S_{14}	S_{14}				
			—	$E_{11}\rightarrow U_2$	$E_{12}\rightarrow U_2$								—	$E_{15}\rightarrow U_2$	$E_{15}\rightarrow U_2$				
	E^i_{11}	水が空になった	S_{14}	S_{15}	S_{16}								—	—	—				
			—	—	—								—	—	—				
U_2	E_{14}	急激に温度が上昇した				S_{24}	S_{24}	—								—	S_{24}		
						$E_{24}\rightarrow U_3$	$E_{24}\rightarrow U_3$	—								—	$E_{24}\rightarrow U_3$		
	E_{15}	急激に温度が下降した				—	S_{25}	S_{25}								S_{25}	—		
						—	$E_{25}\rightarrow U_3$	$E_{25}\rightarrow U_3$								$E_{25}\rightarrow U_3$	—		
U_3	E_{24}	急激な温度上昇を検知した							—	S_{33}								—	制御が発振［障害］
									—	$E_{33}\rightarrow U_1$								—	
	E_{25}	急激な温度下降を検知した							S_{34}	—								制御が発振［障害］	—
									$E_{34}\rightarrow U_1$	—									—
U_4	E_{33}	急激な温度上昇をにより加熱停止を指示									S_{42}	—							
											$E_{42}\rightarrow U_1$	—							
	E_{34}	急激な温度下降により加熱を指示									—	S_{41}							
											—	$E_{41}\rightarrow U_1$							

第8章　具体的な事例

7章は，手順の説明を目的とし，単純な1つの非正常内部イベントによる事例を用いた．本章では，事例による手法の検証を目的とし，複数の非正常内部イベントを持つ事例を用い，7章において確認できなかった障害分析手法に対する要件が充足していることを確認する．

8.1　事例の障害分析

8.1.1　システム構造図の作成

事例として取り上げる道路灯の要求仕様は,道路の最低照度を維持し，道路の安全を確保することである．この道路灯は，筐体の受光窓より周辺環境の光量を取り込み，照度計測部により照度を計測し，その計測値を昼夜判断部で判断する．この判断結果にもとづき，照明の点灯あるいは消灯を行い，道路環境の安全な明るさを保持する．

図 8.1 に，道路灯のシステム構造図を示す.

U_1：太陽光

道路環境を照らす光源であるを太陽光を示す.

U_2：太陽光以外の光源

道路環境を照らす太陽光以外の光源を示す．本事例では，車のヘッドライトのみを対象とする．

U_3：受光窓の周辺環境

受光窓周辺の道路環境とする．受光窓周辺には，光を遮るものがないことを正常とする．

U_4：筐体

道路灯装置を保管する筐体であり，環境の光量を取り入れるための受光窓を持つ．施工時の注意書により，照明器具の光は，直接受光窓に入らないように施

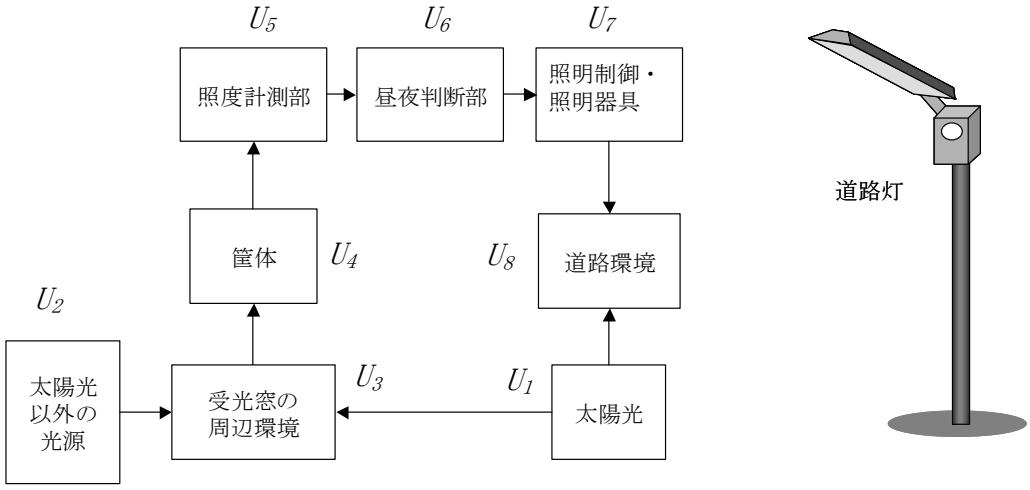

図 8.1: 道路灯のシステム構造図

工されているものとする．受光窓は，汚れていないことを正常とする．

U_5：照度計測部

昼夜判断部から，測定の指示を受けた場合，受光窓から取り入れた光量を測定し，測定結果の照度を昼夜判断部に送る．

U_6：昼夜判断部

周辺道路環境の最低限の明るさを確保するため，定期的に照度計測部から照度の情報を取得し，取得した照度の値から昼あるいは夜を判断する．その判断した結果から，照明制御・照明器具に対して点灯あるいは消灯の指示を送る．

U_7：照明制御・照明器具

昼夜判断部からの指示により，照明の点灯や消灯を制御する．照明器具は，蛍光灯を想定しているため，短時間の点灯と消灯の繰り返しは，照明器具の寿命を大きく低下させる．

U_8:道路環境

道路灯周辺の道路環境を示す．道路灯の照明により，最低限の明るさが保持されている必要がある．

U_1からU_8の構成要素の中で，次に述べる構成要素は，後工程である逸脱要因の分析時において，新たな構成要素の発見を行い追加したものである．まず，太陽光以外の光源U_2は，太陽光以外の光源という逸脱分析から追加した．次に，

構成要素番号	構成要素名	状態番号	状態詳細
U_1	太陽光	S_{11}	夜の明るさである
		S_{12}	うすくらい昼の明るさである
		S_{13}	昼の明るさである
U_2	太陽光以外の光源	S_{21}	光を出す物はない
U_3	受光窓の周辺環境	S_{31}	受光窓を遮るものがない
U_4	筐体の受光窓	S_{51}	受光窓が汚れていない
U_5	照明計測部	S_{51}	信号を計測している
U_6	昼夜判断部	S_{61}	夜の判断をしている
		S_{62}	昼の判断をしている
U_7	照明制御部・照明器具	S_{71}	照明を点灯している
		S_{72}	照明を消灯している
U_8	道路環境	S_{81}	照明が点灯し通行出来る明るさを保持している
		S_{82}	照明は点灯してないが通行出来る明るさを保持している

表 8.1: 道路灯の正常状態

受光窓の周辺環境U_3は，受光窓周辺の環境で受光窓を遮るものがあることから，道路環境と分離した．さらに，筐体U_4は，逸脱の原因分析時に受光窓の汚れが発生することから追加した．

8.1.2 正常状態，正常イベントの抽出

構成要素「太陽光」は，夜から白昼まで明るさには連続的な特性を持つ．そのため，同値分割法を持ちいて，道路灯を点灯させる境界値を中心に同一の明るさの特性と認識できる領域に分割する．同値分割法により，太陽光は,「昼の明るさ」,「うす暗い昼の明るさ」,「夜の明るさ」という3つの状態を持つものとする．また，イベントは，この状態領域に遷移したという「昼の明るさになった」,「うす暗い昼の明るさになった」,「夜の明るさになった」とする．表 8.1に正常状態の表を，表 8.2にイベントの表を示す．ただし，逸脱要因の非正常内部イベントE^i_{21}，E^i_{31}，E^i_{41}，E^i_{51}は，8.1.3項から抽出したものである．

8.1.3 逸脱の原因となる非正常内部イベントの抽出

道路灯の正常イベントに対してガイドワードを当てはめ，逸脱事象を抽出する．太陽光の受光イベントに対してガイドワードを当てはめた例を表 8.3に示

構成要素番号	イベント番号	イベント詳細
U_2	$E^i{}_{21}$	夜間にヘッドライトが照射
U_3	E_{11}	夜の明るさになった
	E_{12}	うす暗い昼の明るさになった
	E_{13}	昼の明るさになった
	$E^i{}_{31}$	街路樹が風でなびく
U_4	E_{31}	夜の明るさになった
	E_{32}	うす暗い昼の明るさになった
	E_{33}	昼の明るさになった
	$E^i{}_{41}$	受光窓が汚れる
$U5$	E_{41}	夜の明るさを通過させた
	E_{42}	うす暗い昼の明るさを通過させた
	E_{43}	昼の明るさを通過させた
	$E^i{}_{51}$	信号計測機能の停止
U_6	E_{51}	夜の明るさを計測した
	E_{52}	うす暗い昼の明るさを計測した
	E_{53}	昼の明るさを計測した
U_7	E_{61}	夜の明るさを判断した
	E_{62}	昼の明るさを判断した
U_8	E_{11}	夜の明るさになった
	E_{12}	うす暗い昼の明るさになった
	E_{13}	昼の明るさになった
	E_{71}	照明を点灯した
	E_{72}	照明を消灯した

表 8.2: 道路灯のイベント

す．また，抽出された逸脱事象に対してFTAを当てはめた例を図 8.2に示す．さらに，道路灯の各構成要素に故障モードを当てはめた例を表 8.4に示す．この抽出した逸脱要因のうち，既知の対策があるものや施工マニュアルなどで対策できるものを除き，初期の非正常内部イベントは，下記4つとする．

E^i_{21}：夜間にヘッドライトが照射

道路灯は，道路脇に設置されており，道路灯筐体に直接あるいは，周辺の金属やガラスに反射することにより，車のヘッドライトによる強い光を受ける可能性がある．

E^i_{31}：街路樹が風でなびく

時間帯により，太陽光が直接受光窓に入る場合，道路周辺の街路樹の木の葉に

イベント	ガイドワード		抽出した逸脱
	着眼属性	拡張ガイドワード	
E_{13} 昼の明るさになった	有無	有，なし	明るくならない
	符号反転	正，負	――――――
	速度	速く，遅く	朝になってもすぐに明るくならない
	幅	長く，短く	短時間の明かりを受ける
	同期／非同期	同期，非同期	――――――
	間隔	長く，短く，同時に	一日以上明るさが続く
	回数	多く，少なく	明暗が繰り返される
	順序	前に，後に	――――――
	量	大きく，小さく	強い明るさを受ける
	送信元	異なった	太陽光以外から光を受ける
	タイミング	矛盾した	夜に明るくなる
	データ構造	不一致部分	――――――

表 8.3: 道路灯のガイドワード適用例

より陰になる場合がある．街路樹は，風でなびき，受光窓の光をチラつかせる可能性がある．

E_{41}^{i}：受光窓が汚れる

道路灯は，屋外に露出しているため，筐体は，時間経過により砂埃などによる汚れが発生する．

E_{51}^{i}：信号計測機能の停止

照度計測部の回路故障により，昼夜判断部からの読み出しの際に，照度計測部が応答しなくなる故障が考えられる．

8.1.4 非正常系分析マトリクスを用いた逸脱分析

8.1.3項までの正常状態，正常イベントおよび非正常内部イベントをそれぞれ，状態欄，イベント欄に記載し，非正常系分析マトリクスを作成する．次に，非正常系分析マトリクスにおける有効なセルを分析し，非正常系分析マトリクスを展開していく．

展開したマトリクス表のU_1からU_4を表 8.5，U_5を表 8.6に，U_6を表 8.7に，U_7を

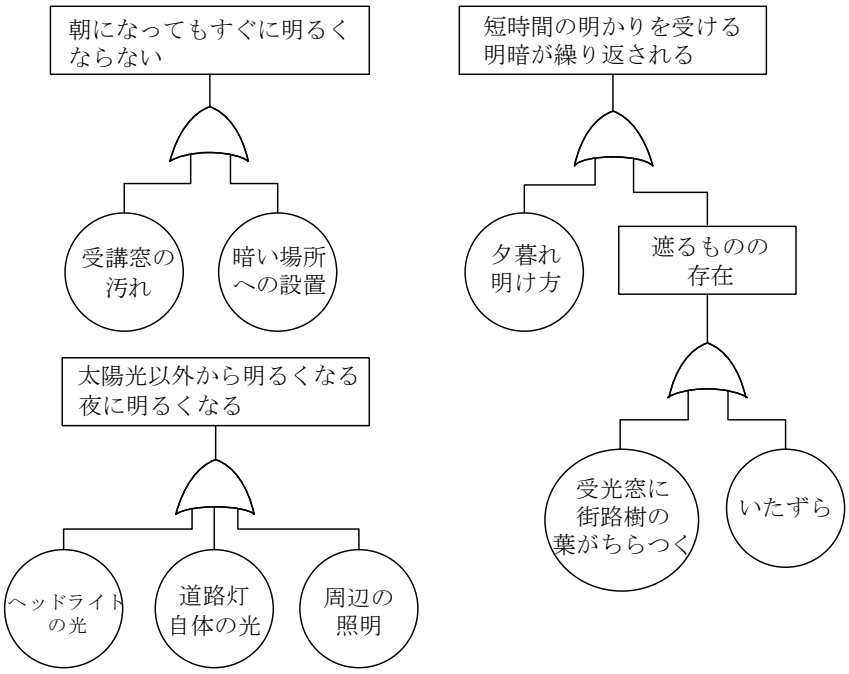

図 8.2: 道路灯の FTA 適用例

表 8.8 に, U_8 を表 8.9 にそれぞれ示す. それらを組み合わせることにより, 非正常系分析マトリクス図全体を示している.

以上の分析により, 表 8.9 のセルに 6 種類の障害が認識できた. この中で,「昼間に照明の点灯と消灯を繰り返すことによる照明器具の寿命低下」障害の障害シナリオを図 8.3 に示す. また,「夜間に通行できる明るさが保持できない」障害の障害シナリオを図 8.4 に示す.

以下に, 障害の内容, 逸脱要因, 障害シナリオの概要を記述する.

(安全性に影響を与える障害)

[障害 1] 夜間に通行できる明るさが保持できない.

逸脱要因: 信号計測機能の停止 E_{51}^{i}

昼間に照度計測部 U_5 が故障し, 昼夜判断部 U_6 に, 照度情報を伝えることを停止したため, 昼夜判断部は, 照度計測部が故障する直前に判断した「昼の判断」を固定する状態 S_{64} になる. この状態で夜になったため, 夜間に通行できる明るさが保持できない.

[障害4] 夜間に通行できる明るさが保持できない.

逸脱要因： 夜間にヘッドライトが照射E_{21}^{i}

夜間に車のヘッドライトが道路灯の受光部を照射したため，昼夜判断部U_6は，昼の判断を行い，照明を消灯する．道路灯受光部は，ヘッドライトの光が照射されているが，道路灯により明るくする領域すべてが明るい状態ではない.

[障害5] 夜間に照明の点灯と消灯を短時間に繰り返す.

逸脱要因： 夜間にヘッドライトが照射E_{21}^{i}，街路樹が風でなびくE_{31}^{i}

夜間に車のヘッドライトが道路灯の受光部を照射し，さらに，そのヘッドライトの光が，道路周辺の街路樹によりちらついた場合，昼夜判断部U_6は，昼の判断と夜の判断を繰り返すため，道路灯は，夜間に照明の点灯と消灯を短時間に繰り返す．夜間に，照明が点灯と消灯を繰り返した場合，通行者は，消灯している以上に道が見えにくいため危険となる.

（省エネルギーに関する障害）

[障害2] 昼間に照明を点灯させる.

逸脱要因： 信号計測機能の停止E_{51}^{i}

夜間に照度計測部U_5が故障し，昼夜判断部U_6に，照度情報を伝えることを停止したため，昼夜判断部は，照度計測部が故障する直前に判断した「夜の判断」を固定する状態S_{63}になる．この状態で昼になっても照明が消灯せず，電力を無駄に消費する.

[障害3] 昼間に照明を点灯させる.

逸脱要因： 受光窓が汚れるE_{41}^{i}

筐体U_4の受光窓が汚れたため，照度計測部U_5に送られる光量が減衰していくため，道路灯を消灯するべき夕方や早朝に道路灯を点灯させるため，電力を無駄に消費する.

（耐久性に関する障害）

[障害6] 昼間に照明の点灯と消灯を繰り返すことにより，照明器具の寿命を低下させる障害となる.

逸脱要因：　街路樹が風でなびくE^i_{31}，受光窓が汚れるE^i_{41}

「街路樹が風でなびく」という非正常内部イベントE^i_{31}から,「直射光が街路樹の葉でちらつく」という非正常イベントE_{34}に至る．道路灯が夜を判断する明るさは，太陽が沈んだ後の夕方の明るさである．このため，太陽の直射光がちらついても，太陽が沈んでいない日中では，十分な明るさを有しているため，道路灯の昼と夜の判断に影響することはない．

一方,「道路灯の受光窓が汚れる」という内部イベントから，筐体U_4が「受光窓が汚れている」という非正常状態S_{42}となる．非正常イベントE_{34}が非正常状態S_{42}に送られることにより，複合した逸脱要因による非正常イベントE_{49}「街路樹の葉が風でなびき，チラついた直射光が減衰して通過した」という非正常イベントが発生する．

この受光窓が汚れることによる減衰した光量である直射光のチラツキは，道路灯の昼と夜の判断値を交互に繰り返してしまう．このため，最終的に照明制御・照明器具U_7は，昼に照明の点灯と消灯を繰り返し，照明器具を劣化させてしまう．

8.2　考察

7.3節において，ESIMの手順を明確にするために，逸脱要因を1つとした．そのため，ESIMが障害分析手法として2.8節に記載した要件2に適合していることについて確認できなかった．そこで，本節において要件2に適合していることについて確認する．また，障害シナリオ分析をソフトウェア仕様定義前に行うことにより，ソフトウェア要求仕様の曖昧性を削減することができることについて考察する．

項8.1.4において6種類の障害を抽出した．

障害5の「夜間に照明の点灯と消灯を短時間に繰り返す」は，逸脱要因「夜間にヘッドライトが照射E^i_{21}」と,「街路樹が風でなびく」E^i_{31}という2つの逸脱要因の組み合わせであった．これは，逸脱要因E^i_{21}からの非正常イベントE_{35}が，別の逸脱要因E^i_{31}からの非正常状態S_{32}に入ることにより逸脱要因が複合している．また，障害6の「昼間に照明の点灯と消灯を繰り返すことにより，照明器具の寿命を低下させる障害となる」は，逸脱要因「街路樹が風でなびく」E^i_{31}と,「受光

窓が汚れる」E^i_{41}という2つの逸脱要因の組み合わせであった．これは，逸脱要因E^i_{31}からの非正常イベントE_{34}が，別の逸脱要因E^i_{41}からの非正常状態S_{42}に入ることにより逸脱要因が複合していいる．このように，障害要因の複合を確認できることは，要件2を満足している．

また，障害1の「夜間に通行できる明るさが保持できない」と，障害3の「昼間に照明を点灯させる」は，同一の逸脱要因「信号計測機能の停止」E^i_{51}に起因している．これは，逸脱要因E^i_{51}が発生したときの昼夜判断部U_6の状態が,「昼を判断している」か「夜を判断している」かにより，障害の種類が異なった．

ESIMでは，システムの状態に着目しながら，個別の構成要素の状態でシステムの状態を代表している．これにより，ある構成要素の逸脱が，離れている構成要素には影響が到達していないという過程を詳細に追っている．そのため，逸脱が伝播していく過程において，様々な状態ごとに影響がどの様に伝播していくかを確認することができる．

また，抽出した障害シナリオは，安全性，耐久性，省エネルギー性という多彩な品質要求に関わる非機能要求に関係していた．これらの障害を回避する仕様をソフトウェア要求仕様に組込むことにより，非機能要求の曖昧さを軽減できる．これらのソフトウェア要求仕様は，正常な振る舞いを詳細に検討しても抽出することはできない．また，逸脱要因の抽出に際して，意図的に安全性，耐久性，省エネルギー性といった品質要求の抽出を考慮したものではない．抽出した障害シナリオは，障害分析を行った結果として発見できたものである．

このように，障害シナリオが発見できれば，具体的に品質を保証するための機能を定義することが可能になり，結果として，品質要求を具体的な振る舞いに変換できる．これは，2章で述べた非機能要求を明確にするための障害分析の必要性に合致する．

No	構成要素	種別	項目	故障モード	逸脱要因
U1	太陽光	負荷	過負荷	急負荷，連続過負荷，限界をわずかに超えた過負荷，異常過負荷	————————
			軽負荷	負荷はずれ，スリップ	————————
U2	太陽光以外の光源	負荷	過負荷	急負荷，連続過負荷，限界をわずかに超えた過負荷，異常過負荷	ヘッドライトの光，広告灯
			軽負荷	負荷はずれ，スリップ	————————
U3	受光窓の周辺環境	環境	電源	瞬時停電，連続瞬時停電，長期停電，広域停電，電圧低下	————————
			加速度	ゆれ，振動，衝撃	————————
			温度	高温，低温，急激な温度変化	————————
			電磁波	ノイズ，強電磁波，放送設備近隣，違法移動無線局	————————
			湿度	結露，水蒸気	————————
			天候	雨，雷，ヒョウ	積雪
			虫	ハエ，蚊，蟻，蜘蛛	————————
			障害物	壁，段差，鉄製ドア	街路樹
U4	筐体	機構	筐体	汚れ，割れ，ずれ	受光窓の汚れ
		施工	配線	誤結線，重複アドレス，アドレス設定ミス，	
			設置	位置ずれ，ゆがみ，方向ずれ	受光窓に道路灯の光が直射
U5	照度計測部	電気回路	センサ	更新中読み出し，オーバーフロー，アンダーフロー，誤読出，停止	回路故障
U6	昼夜判断部	CPU周辺	割り込み	割り込み消失，多重割り込み，誤割り込み，ノンマスカブル割り込み	誤読み出し
			レジスタ，メモリ	レジスタデータ化け，誤書き込み	————————
U7	照明制御・照明器具	電気回路	アナログ回路	アンダーフロー，オーバーフロー，発振，劣化による値ずれ，劣化遅延	————————
			コネクタ	短絡，断線，接触不良，アース開放	————————
U8	道路環境	環境	電源	瞬時停電，連続瞬時停電，長期停電，広域停電，電圧低下	————————
			加速度	ゆれ，振動，衝撃	————————
			温度	高温，低温，急激な温度変化	————————
			電磁波	ノイズ，強電磁波，放送設備近隣，違法移動無線局	
			湿度	結露，水蒸気	————————
			天候	雨，雷，ヒョウ	————————
			虫	ハエ，蚊，蟻，蜘蛛	————————
			障害物	壁，段差，鉄製ドア	————————

表 8.4: 道路灯の故障モード適用例

構成要素記号	イベント記号	イベント / 状態	U_2	U_3	U_4	U_2	U_3	U_4
		構成要素名	太陽光以外の光源	受光窓の周辺環境	筐体の受光窓	太陽光以外の光源	受光窓の周辺環境	筐体の受光窓
		状態記号	S_{21}	S_{31}	S_{41}	S_{22}	S_{32}	S_{42}
		状態	光を出す物はない	受光窓を遮るものがない	受光窓が汚れていない	夜間にヘッドライトの光が照射している	受光窓の前で街路樹の葉が風でなびいている	受光窓が汚れている
U_2	E_{21}	夜間にヘッドライトが照射	S_{22} $E_{21}\rightarrow U_3$			— —		
U_3	E_{11}	夜の明るさになった		— $E_{31}\rightarrow U_4$			— —	
	E_{12}	うす暗い昼の明るさになった		— $E_{32}\rightarrow U_4$			— —	
	E_{13}	昼の明るさになった		— $E_{33}\rightarrow U_4$			— $E_{34}\rightarrow U_4$	
	E_{31}	街路樹が風でなびく		S_{32} —			— —	
U_4	E_{31}	夜の明るさになった			— $E_{41}\rightarrow U_5$			— —
	E_{32}	うす暗い昼の明るさになった			— $E_{42}\rightarrow U_5$			— $E_{44}\rightarrow U_5$
	E_{33}	昼の明るさになった			— $E_{43}\rightarrow U_5$			— $E_{45}\rightarrow U_5$
	E_{41}	受光窓が汚れる			S_{42} —			— —
U_3	E_{21}	夜にヘッドライトの光を照射した		— $E_{35}\rightarrow U_4$			— $E_{36}\rightarrow U_4$	
U_4	E_{34}	直射日光が街路樹の葉でなびく			— $E_{46}\rightarrow U_5$			— $E_{49}\rightarrow U_5$
	E_{35}	夜間にヘッドライトの光が照射された			— $E_{47}\rightarrow U_5$			— $E_{4a}\rightarrow U_5$
	E_{36}	夜に街路樹の葉が風でなびき，チラついたヘッドライトの光が通過した			— $E_{48}\rightarrow U_5$			— $E_{4b}\rightarrow U_5$

表 8.5: 道路灯の非正常系マトリクス（構成要素1から4の状態）

構成要素記号	送り先記号 イベント	イベント	構成要素記号	U_5	U_5
			構成要素名	照明計測部	照明計測部
			状態記号	S_{51}	S_{52}
			状態	信号を計測している	信号を計測していない
U_5	E_{41}	夜の明るさを通過させた		ー $E_{51}\rightarrow U_6$	ー ー
	E_{42}	うす暗い昼の明るさを通過させた		ー $E_{52}\rightarrow U_6$	ー ー
	E_{43}	昼の明るさを通過させた		ー $E_{53}\rightarrow U_6$	ー ー
	E_{i51}	信号計測機能の停止		S_{52} $E_{54}\rightarrow U_6$	ー ー
U_5	E_{44}	うす暗い昼の明りが減衰して通過した		ー $E_{55}\rightarrow U_6$	ー ー
	E_{45}	昼の明るさが減衰してうす暗い昼の明るさになり通過した		ー $E_{56}\rightarrow U_6$	ー ー
	E_{46}	街路樹の葉が風でなびき，チラついた直射光が通過した		ー $E_{53}\rightarrow U_6$	ー ー
	E_{47}	夜にヘッドライトの光が通過した		ー $E_{57}\rightarrow U_6$	ー ー
	E_{48}	夜に街路樹の葉が風でなびき，チラついたヘッドライトの光が通過した		ー $E_{58}\rightarrow U_6$	ー ー
	E_{49}	街路樹の葉が風でなびき，チラついた直射光が減衰して通過した		ー $E_{59}\rightarrow U_6$	ー ー
	E_{4a}	減衰したヘッドライトの光が通過した		ー $E_{51}\rightarrow U_6$	ー ー
	E_{4b}	街路樹の葉が風でなびき，チラついたヘッドライトの光が減衰して通過した		ー $E_{51}\rightarrow U_6$	ー ー

表 8.6: 道路灯の非正常系マトリクス（構成要素5の状態）

構成要素記号	送り先記号	イベント	構成要素記号	U_6		U_6					
			構成要素名	昼夜判断部		昼夜判断部					
			状態記号	S_{61}	S_{62}	S_{63}	S_{64}	S_{65}	S_{66}	S_{67}	S_{68}
		イベント	状態	夜の判断をしている	昼の判断をしている	夜の判断を固定している	昼の判断を固定している	昼を判断するうす暗い明るさを夜と判断している	夜にヘッドライトの光のために, 昼を判断している	夜に昼と夜の判断が振動している	昼に, 昼と夜の判断が振動している
U_6	E_{51}	夜の明るさを計測した		— —	S_{61} $E_{61}\rightarrow U_7$	— —	— —	S_{61} $E_{61}\rightarrow U_7$	S_{61} $E_{61}\rightarrow U_7$	S_{61} $E_{61}\rightarrow U_7$	S_{61} $E_{61}\rightarrow U_7$
	E_{52}	うす暗い昼の明るさを計測した		S_{62} $E_{62}\rightarrow U_7$	— —	— —	— —	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$
	E_{53}	昼の明るさを計測した		S_{62} $E_{62}\rightarrow U_7$	— —	— —	— —	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$
U_6	E_{54}	信号を計測しなくなった		S_{63} $E_{63}\rightarrow U_7$	S_{64} $E_{64}\rightarrow U_7$	— —	— —	S_{63} $E_{63}\rightarrow U_7$	S_{64} $E_{64}\rightarrow U_7$	S_{63}/S_{64} $E_{63}/E_{64}\rightarrow U_7$	S_{63}/S_{64} $E_{63}/E_{64}\rightarrow U_7$
	E_{55}	うす暗い明りが減衰して通過したため, 夜の明るさを計測した		— —	S_{65} $E_{65}\rightarrow U_7$	S_{65} $E_{65}\rightarrow U_7$	S_{65} $E_{65}\rightarrow U_7$	— —	S_{65} $E_{65}\rightarrow U_7$	S_{65} $E_{65}\rightarrow U_7$	S_{65} $E_{65}\rightarrow U_7$
	E_{56}	昼の明りが減衰して通過したため, うす暗い明るさを計測した		S_{62} $E_{62}\rightarrow U_7$	— —	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$	S_{62} $E_{62}\rightarrow U_7$
	E_{57}	夜にヘッドライトの光のために, 昼を計測した		S_{66} $E_{66}\rightarrow U_7$	— —	S_{66} $E_{66}\rightarrow U_7$	S_{66} $E_{66}\rightarrow U_7$	S_{66} $E_{66}\rightarrow U_7$	S_{66} $E_{66}\rightarrow U_7$	S_{66} $E_{66}\rightarrow U_7$	S_{66} $E_{66}\rightarrow U_7$
	E_{58}	夜に街路樹の葉が風でなびき, チラついたヘッドライトの光が通過したため, 昼と夜の計測が振動した		S_{67} $E_{67}\rightarrow U_7$	— —	S_{67} $E_{67}\rightarrow U_7$	S_{67} $E_{67}\rightarrow U_7$	S_{67} $E_{67}\rightarrow U_7$	S_{67} $E_{67}\rightarrow U_7$	— —	S_{67} $E_{67}\rightarrow U_7$
	E_{59}	街路樹の葉が風でなびき, チラついた直射光が減衰して通過したため, 昼の判断と夜の判断を振動した		— —	S_{67} $E_{67}\rightarrow U_7$	S_{68} $E_{68}\rightarrow U_7$	S_{68} $E_{68}\rightarrow U_7$	S_{68} $E_{68}\rightarrow U_7$	S_{68} $E_{68}\rightarrow U_7$	S_{68} $E_{68}\rightarrow U_7$	— —

表 8.7: 道路灯の非正常系マトリクス（構成要素6の状態）

構成要素記号	送り先	イベント記号	イベント	状態	U_7		U_7				
			構成要素名		照明制御部・照明器具		照明制御部・照明器具				
			状態記号		S_{71}	S_{72}	S_{73}	S_{74}	S_{75}	S_{76}	S_{77}
					照明を点灯している	照明を消灯している	照明の点灯を固定している	照明の消灯を固定している	昼と判断するうす暗い明るさを夜と判断し照明を点灯している	夜にヘッドライトの光のために，照明を消灯している	夜に照明の点灯と消灯を繰り返している
U_7		E_{61}	夜の明るさを判断した		— —	S_{71} $E_{71}\rightarrow U_8$	S_{71} $E_{71}\rightarrow U_8$	S_{71} $E_{71}\rightarrow U_8$	S_{71} $E_{71}\rightarrow U_8$	S_{71} $E_{71}\rightarrow U_8$	S_{71} $E_{71}\rightarrow U_8$
		E_{62}	昼の明るさを判断した		S_{72} $E_{72}\rightarrow U_8$	— —	S_{72} $E_{72}\rightarrow U_8$	S_{72} $E_{72}\rightarrow U_8$	S_{72} $E_{72}\rightarrow U_8$	S_{72} $E_{72}\rightarrow U_8$	S_{72} $E_{72}\rightarrow U_8$
U_7		E_{63}	夜の判断を固定した		S_{73} $E_{73}\rightarrow U_8$	— —	— —		S_{73} $E_{73}\rightarrow U_8$	S_{73} $E_{73}\rightarrow U_8$	S_{73} $E_{73}\rightarrow U_8$
		E_{64}	昼の判断を固定した		— —	S_{74} $E_{74}\rightarrow U_8$		— —	S_{74} $E_{74}\rightarrow U_8$	S_{74} $E_{74}\rightarrow U_8$	S_{74} $E_{74}\rightarrow U_8$
		E_{65}	昼と判断するうす暗い明るさを夜と判断した		— —	S_{75} $E_{75}\rightarrow U_8$	S_{75} $E_{75}\rightarrow U_8$	S_{75} $E_{75}\rightarrow U_8$	S_{75} $E_{75}\rightarrow U_8$	S_{75} $E_{75}\rightarrow U_8$	S_{75} $E_{75}\rightarrow U_8$
		E_{66}	夜にヘッドライトの光のために，昼を判断した		S_{76} $E_{76}\rightarrow U_8$	— —	S_{76} $E_{76}\rightarrow U_8$	S_{76} $E_{76}\rightarrow U_8$	S_{76} $E_{76}\rightarrow U_8$	S_{76} $E_{76}\rightarrow U_8$	S_{76} $E_{76}\rightarrow U_8$
		E_{67}	夜に昼と夜の判断が振動した		S_{77} $E_{77}\rightarrow U_8$	S_{77} $E_{77}\rightarrow U_8$	S_{77} $E_{77}\rightarrow U_8$	S_{77} $E_{77}\rightarrow U_8$	S_{77} $E_{77}\rightarrow U_8$	S_{77} $E_{77}\rightarrow U_8$	— —
		E_{68}	昼に昼と夜の判断が振動した		照明器具の点灯と消灯を短時間に繰り返すことにより照明器具が劣化する **［障害6］**	照明器具の点灯と消灯を短時間に繰り返すことにより照明器具が劣化する **［障害6］**					

表 8.8: 道路灯の非正常系マトリクス（構成要素7の状態）

構成要素記号	送り先記号	イベント	構成要素記号	U_8		U_8	
			構成要素名	道路環境		道路環境	道路環境
			状態記号	S_{81}	S_{82}	S_{83}	S_{84}
			状態	照明が点灯し通行出来る明るさを保持している	照明は点灯してないが通行出来る明るさを保持している	照明の点灯を固定しているが，通行出来る明るさを保持している	照明の消灯を固定しているが通行出来る明るさを保持している
U_8	E_{11}	夜の明るさになった		ー ー		ー ー	通行できる明るさが保持できない **［障害1］**
	E_{12}	うす暗い昼の明るさになった		ー ー	ー ー	不要な照明を点灯している **［障害２］**	ー ー
	E_{13}	昼の明るさになった		ー ー	ー ー	不要な照明を点灯している **［障害2］**	ー ー
	E_{71}	照明を点灯した		ー ー	S_{81} ー		
	E_{72}	照明を消灯した		S_{82} ー	ー ー		
U_8	E_{73}	照明の点灯を固定した		S_{83} ー	ー ー	ー ー	
	E_{74}	照明の消灯を固定した		ー ー	S_{84} ー		ー ー
	E_{75}	昼と判断するうす暗い明るさを夜と判断し照明を点灯した		ー ー	明るい状態で照明を点灯し電力を浪費する **［障害3］**		
	E_{76}	夜にヘッドライトの光のために，照明を消灯した		足元の明るさを保てない **［障害4］**	ー ー		
	E_{77}	夜に照明の点灯と消灯を繰り返した		足元の明るさを保てない **［障害5］**			

表 8.9: 道路灯の非正常系マトリクス（構成要素 8 の状態）

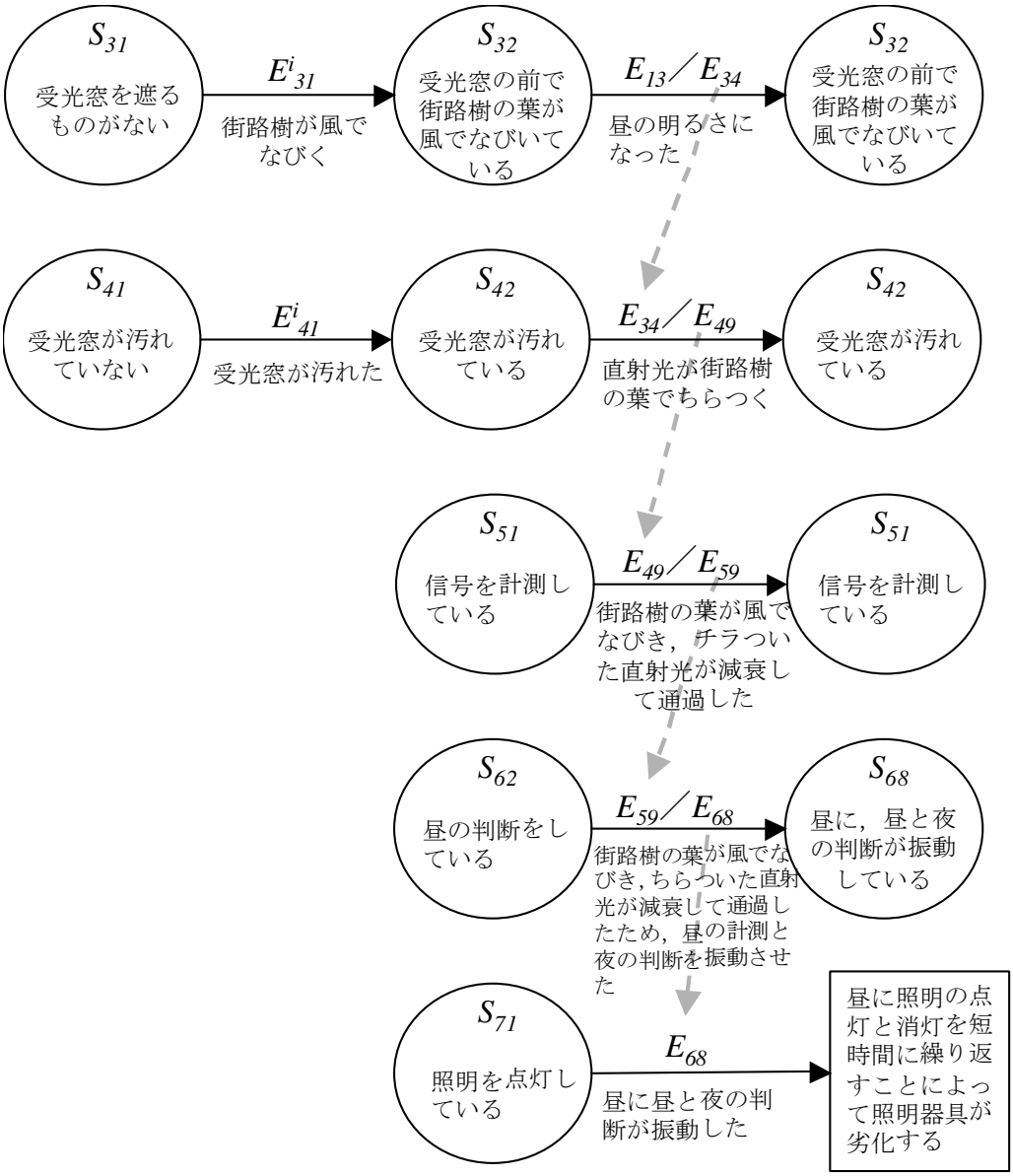

図 8.3: 道路灯の耐久性に関する障害シナリオ例

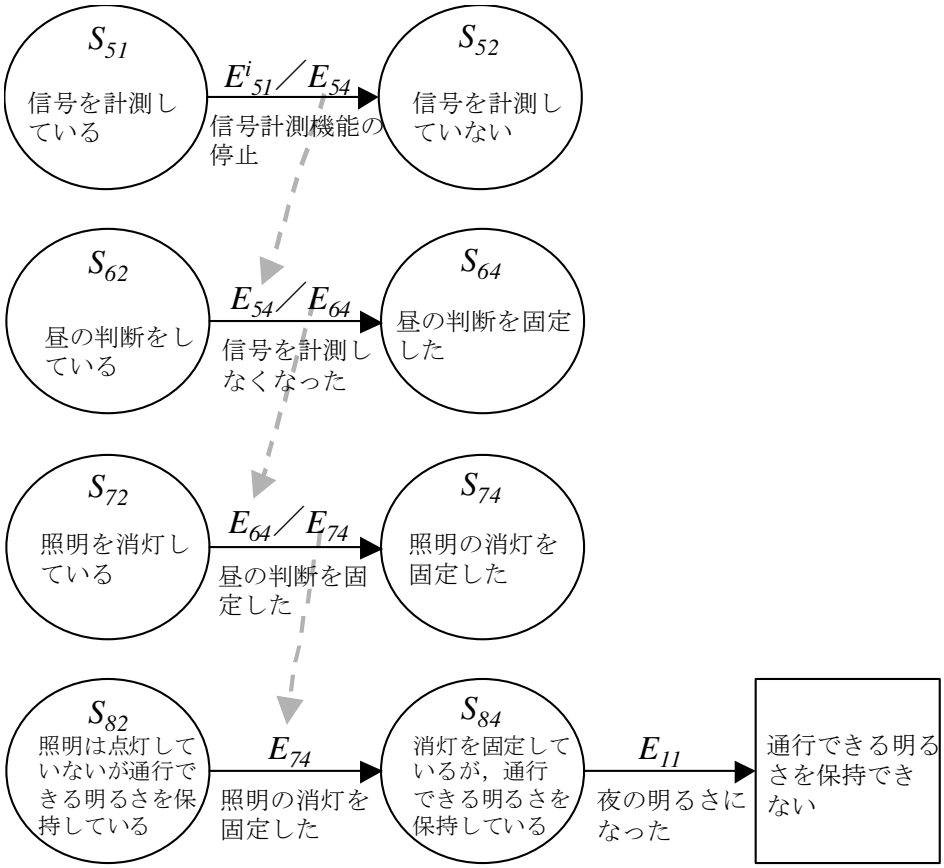

図 8.4: 道路灯の安全性に関する障害シナリオ例

第9章　適用実験

ESIMの有効性と実用性を確認するため，実験として筆者の所属する会社における実際の製品開発にESIMを適用した．有効性については，製品のプロトタイプ開発工程において抽出できた障害シナリオと，ESIM を適用することによって抽出できた障害シナリオの過不足を確認する．また，実用性については，ESIMを用いた分析に要した工数が，実際の開発において許される範囲かを確認する．

9.1　評価方法

評価対象の製品は，図 9.1の右端に示す様々な機器を制御する新規開発の製品である．それらの機器の中で，全体制御ユニットと表示操作系ユニットが今回の開発対象であり，他は，既存のものであった．開発したソフトウェアは，C言語を用い50Kステップの規模であった．

適用実験においては，表 9.1に示すA とB の2 チームを比較する．図9.2にAチームとBチームの開発プロセスを示す．Aチームは，要求仕様書第一版をもとに，プロトタイプ開発を行う．そのプロトタイプ開発の後，要求仕様書を更新する．更に，品質管理部の担当者がレビューし，その結果を含め，要求仕様書第二版を作成する．一方，Bチームは，要求仕様書第一版をもとにESIMを適用して障害シナリオを抽出する．ここでのAチームによる障害シナリオ抽出作業は，プロトタイプ開発におけるソフトウェア要求分析時のFMEA等による障害分析や，設計レビュー，テストも含むものとする．

ESIMの有効性を確認するため，AチームとBチームが抽出した障害シナリオの個数を比較する．そのため，Aチームに対して，要求仕様書第一版の変更履歴書とレビュー報告書を確認する．また，開発技術者へのヒアリングを実施し，障害シナリオと，その抽出過程を記録する．Bチームに対しては，ESIMを適用して抽出した障害シナリオの個数を記録する．また，ESIMの実用性を確認するた

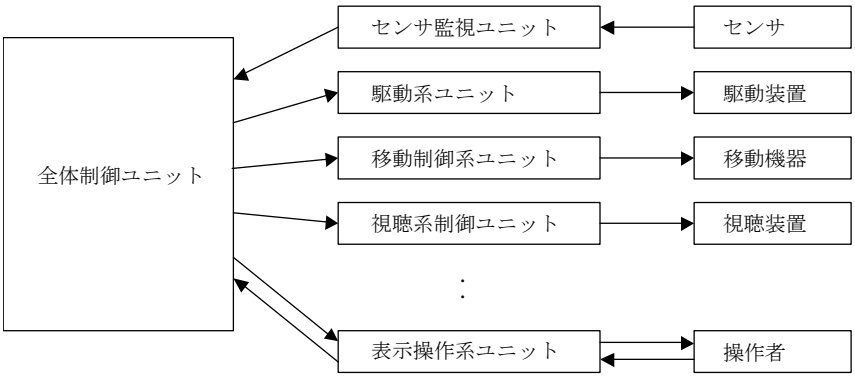

図 9.1: 実験対象のシステム構造

項目	Aチーム	Bチーム
要員構成	システム開発技術者3名 （経験8年以上） プログラマ最大6名 品質管理部レビュー担当者3名 （内2名は経験20年以上）	ESIM実施者1名 （経験20年以上） ESIM分析結果のレビュー実施者1名 (経験15年）
工数	プロトタイプ開発53人　月 レビュー総工数　36人時	ESIMによる分析　24人時 その分析結果のレビュー　8人時

表 9.1: AチームとBチームの比較

め，Bチームの分析に要した工数と，Bチームの技術者のスキルを記録する．

9.2 実験結果

実験の結果，Aチームが抽出した障害シナリオは全て，Bチームも抽出しており，表 9.2に示すように，障害シナリオの抽出数は，AチームよりもBチームの方が8件多かった．なお，ESIMは，システム・アーキテクチャ設計の後に用いる障害分析手法のため，Aチームによるプログラム構造に起因する障害シナリオは除いて評価した．

この中で，Bチームが抽出し，Aチームは抽出できなかった8件の障害シナリオは以下のような，複合要因による障害シナリオであった．

原因1 利用者が設備を少しだけ移動させるため操作部より駆動部に制御開始指

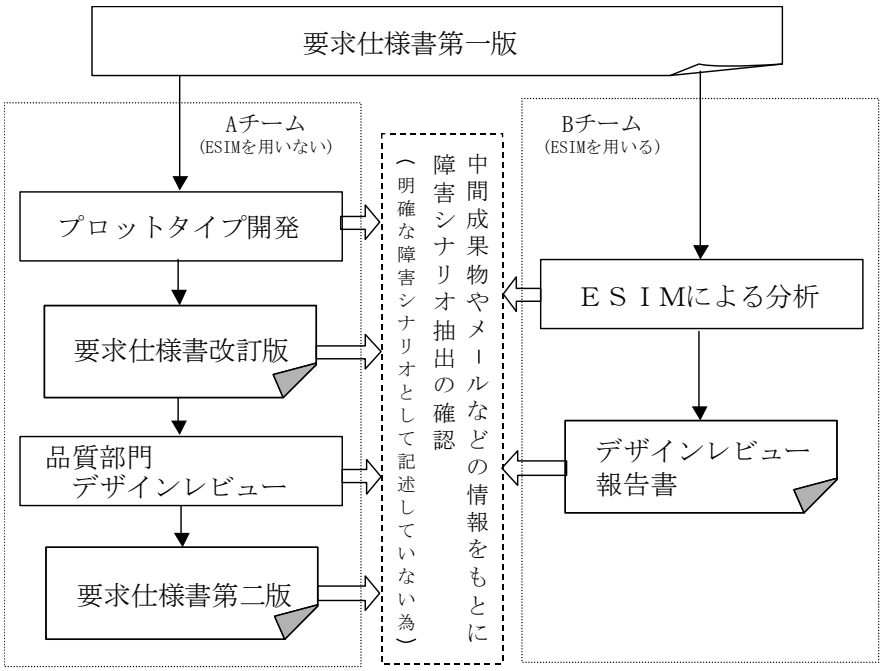

図 9.2: AチームとBチームの開発プロセス

令を送り,

原因2 駆動部が,操作部からの指令を受け取り,設備の移動が開始した時に

原因3 通信が不能,あるいは操作部が機能停止した場合,

結果 駆動部は,操作部から停止指令を受けることができない状態に陥り,利用者が設備に挟まれる.

複合要因による障害が発生する確率は低く,プロトタイプ開発,仕様書の分析,デザインレビューにおいても,このような起こり得る組み合わせを確認することは困難である.

また,BチームがESIMによる分析に要した工数は,表 9.1に示すように32人時であった.これは,本実験対象と同規模の開発において,障害を未然に防ぐための品質保証活動に用いる工数として問題のない工数であった.なお,ESIMはソフトウェア開発者が使用することを想定しているため,製品仕様の理解に要する時間は,前述の32人時に含めていない.

次に,Bチームが作成した分析マトリクスの規模を表 9.3に示す.マトリクス

計測対象	Aチーム	Bチーム
ソフトウェアの設計開始までに抽出した件数	22	48
プロトタイプ開発のテスト完了までに抽出した件数	14	—
品質管理部担当者のレビューより抽出した件数	4	—
合計	40	48

表 9.2: AチームとBチームの抽出した障害シナリオの件数

計測対象	個数
正常状態	25
非正常状態	22
正常イベント	20
非正常イベント	24
マトリクス内のセル	2068
状態遷移のあったセル	448
同一の性質を除いた障害	80
複合要因による障害	11

表 9.3: 分析マトリクスの規模

のセルは，2068個に及ぶが，状態遷移が現れたセルは448個であった．この状態遷移から障害発生のみを抽出し，複数のセルに現れる重複した障害発生を整理すると，障害の種類としては80種類であった．分析者は，開発対象の構成要素の状態の単位を大きく捉え，分析量を可能な数に抑えており，実用的な分析時間で実施出来ていた．

なお，ESIMが分析者の個人の特別なスキルに依存していないことを確認するため，実験完了後に追加実験を行った．この追加実験は，初期段階の非正常系分析マトリクスから得られる分析開始時の分析マトリクスの一部を用い，ESIMによる分析をAチームのレビュー担当者に依頼した．この実験の結果，同担当者は，4時間の分析の間にAチームが抽出できていなかった障害シナリオ2件を抽出することができた．このように，レビューを担当できる者であれば，特別なスキルはなくともESIMを使用できることが確認できた．

9.3 考察

ESIMは，製品を分析の対象としているため，2.8節の要件1 に示す通り，ハードウェアや利用環境，利用者も分析の対象に含めなければならない．これは，4章で述べた通り，外部環境や利用者も構成要素と捉えシステムの一部とし，外部環境変化や利用者の誤操作も障害の要因となる内部イベントと捉えている．これらにより，ESIMは製品に有効な方法となっている．

また，ESIMでは，要件2に示す通り，複合的な要因による障害を分析するために，システムを分割せずに全体として扱う状態モデルを採用した．しかし，この形式的な状態モデルは，状態の表現が複雑となり，状態空間も膨大となる．そこで，ESIMでは，4章や6章で述べた通り，分析中のある時点で対象としている構成要素の状態に着目したシステムの全体の状態を記述することで，分析者にとって関心のある状態のみを抽出した．これにより，分析範囲を局所化することにより思考範囲を限定し，分析者の能力への負担を軽減している．このことは，適用実験において，32人時の工数によって障害シナリオを抽出できたことで確かめられた．実際に，状態とイベントの組み合わせによって生じる分析マトリクスの交点のうち状態遷移の記入のあったセルは，448個と多くはなかった．

更に，要件3に挙げた通り，障害発生を時系列に沿ったシナリオとして動的に分析するために，システムの状態遷移モデルを採用し，ある種の状態遷移表である非正常系分析マトリクスを導入した．これにより，障害の要因となるハードウェアの故障や利用者の誤操作等から障害の発生に至る時系列が追跡でき，それを障害シナリオとすることができた．たとえば，節9.2に示した適用実験から得られた障害シナリオの例において，通信部の機能停止直前に操作部からの操作指令を受けるという時間的な要因にもとづく障害が抽出できていた．

また，分析が進むに従って，システムアーキテクチャの設計時には把握していなかった非正常状態や非正常イベントを把握した場合，このマトリクスに，状態欄やイベント欄を追加することができる．これにより，障害分析開始以前に全ての障害を定義しておく必要はない．このことは，適用実験において，実際のプロトタイプ開発中に抽出されていた41件の障害シナリオに加えて，その設計レビューやテスト工程においても見落とされていた複合要因による障害シナリオ

を，8件新たに抽出することができたことにより確認できた．

以上により，ESIMの有効性が認められ，前章の追加実験において述べたようにレビューのできる者であれば，ESIMを初めて使用する場合であっても障害分析できたことから，ESIMは実用性を備えていると認められる．

第10章　まとめ

本研究は，家電製品などの製品における安全性，信頼性，可用性などの品質を向上することが目的である．そこで，開発段階における製品の品質を向上するために，ソフトウェア開発における要求仕様の曖昧性を低減するための有効な手法として，障害分析手法ESIMを提案した．ESIMは，安全性だけでなく，些細な利用者の製品に対する不満も障害と扱う．そのため，ESIMは，製品，利用者，環境を含めた製品が運用されている状況を詳細に分析する手法である．

本研究の課題を明確にするために，筆者の勤務する社内の実製品開発事例を調査した．その結果，カタログに記載された要件を満足できない過負荷，異常環境，例外的な使われ方などにおける製品の振る舞いに対するソフトウェア要求仕様の定義の曖昧さが課題であることが判明した．非機能要求には，カタログに記載された要件を満足するための非機能要求と，カタログに記載された要件が守れない状況において，機能や安全性などの品質を最大限維持するという非機能要求がある．調査結果，後者の検討が十分でないことがソフトウェア要求仕様を曖昧にしている注目すべき課題であることがわかった．また，この非機能要求に関して要求工学の分野でもほとんど研究されていなかった．

そこで，本研究の対象をカタログに記載された要件が守れない状況において，機能や安全性などの品質を最大限維持するという非機能要求を明確にすることに絞った．そこで，具体的な方策の検討を行い，ソフトウェア要求仕様定義前の障害分析手法を行う必要性を明確にした．さらに，社内の製品開発事例の調査結果と製品の特徴とから3つの要件を抽出した．この要件をもとに，従来技術や先行研究を調査したが，本研究対象の家電製品などの製品の特徴であるカタログに記載された要件が守れない状況下で動作することを前提とした障害分析が行われていないことを確認した．

そのため，逸脱から障害に至る分析対象の理論的モデルの検討を行い，分析

対象となるシステム化境界を明確にし，システム構造図という静的な構造モデルと，communicating state machinesをもとにした実用可能な理論的な動的モデルを定義した．その理論的モデルにもとづいて，筆者らが実施していた手法を再構築し，非正常系分析マトリクスと，それをもとにした障害分析手法ESIMの提案を行った．そして，ESIMに対して，実際の製品開発による適用実験を行い，有効性と実用性があることを確認した．

3.3節において，4つのESIMの特徴を記述した．1番目は，分析対象を状態遷移モデルとして捉え，動的に障害に至る過程を分析することによりタイミングに依存する障害の抽出を容易にし，時間変化に対する影響の検討を陽に含むことである．

このために，分析対象を状態遷移モデルとして捉えようとしたが，理論上のシステムの状態遷移は，膨大な規模になる．しかし，意味のある状態は僅かであることは，筆者らの実施していた手法から確認ができた．そこで，意味のある状態に着目するために，システムの状態を構成要素の状態で代表させた．また，逸脱については，システム内の相対的な関係による逸脱が発生する．このため，非正常状態や非正常イベントの記述は，逸脱を特定できる記述を含むものと定義した．これにより，システム全体を捉え，逸脱を特定できる状態遷移マトリクスの記述を定義することができた．

また，膨大な規模の状態遷移から実用的な状態遷移の記述への削減については，適用実験により，分析工数やセルの数などを測定し，実用可能であることを確認することができた．

2番目は，システムを構成する要素の状態が相互に影響して障害を発生させることに着目した手法として，複合的な要因による障害を動的な分析により容易に抽出することを可能することである．

このために，最初の構成要素の逸脱を非正常内部イベントとし，非正常状態と非正常イベントの記述を分析者のシステムに対する観察の視点から行うようにした．たとえば，逸脱が発生し，構成要素の機能が停止した場合，イベントが発生しないというイベントを定義した．このように，分析者の観察の視点で記述することにより，1つの逸脱は，新たに遷移した逸脱状態という非正常状態と逸脱状態に遷移したという非正常イベントを発生させる．非正常系分析マトリクスは，すべての逸脱の起因である非正常内部イベントを１つの非正常系分析

マトリクスに記載するため，逸脱の連鎖を分析する過程において，異なった逸脱要因から発生する非正常イベントと非正常状態の結合を確認できる．これにより，複合的な要因による障害を動的に分析することができる．

複合要因によるシナリオが実際に抽出できることについては，具体的な事例により確認でき，また適用実験により，ESIMを使わないチームでは発見できない複合要因による障害シナリオが発見できていることを確認した．

3番目は，アーキテクチャ設計段階で予見できなかった未知の状態やイベントを抽出し，想定していなかった障害を抽出することにより仕様の妥当性の確認を行うことを可能とすることである．

このために，非正常系分析マトリクスは，発展型の状態とイベントのマトリクスとしている．非正常系分析マトリクスの状態欄とイベント欄は，新たな逸脱を発見した場合，非正常イベントや非正常状態として欄を追加していく．このことは，逸脱を発見することにより，その状態の粒度が細分化されていくことになる．このため，非正常系分析マトリクスの状態やイベントの粒度は不均一を前提としている．逸脱におけるシステムの振る舞いに関して詳細に追跡できることにより，想定していなかった障害を抽出できた．また，非正常系分析マトリクスの分析から新たな逸脱を発見することにより，新たな通信路が発見され，システム境界は広がった．このように，想定していない障害の発生を確認することにより，適用実験の製品の仕様に対して妥当性があるかを確認することができた．

4番目は，製品であるがゆえに部品故障や誤操作など広く影響要因を捉え，更に他の要因との関わり合いの中で障害が起こる可能性の抽出を可能とすることである．

障害の抽出には，構成要素の固有の要因である初期の逸脱要因の抽出が必要となる．その抜け漏れを削減するために，既存の障害分析手法であるFMEA, HAZOP，FTAを利用した，まず，具体的なハードウェアである電気回路や機構などの構成要素の逸脱を抽出するために，故障モードを作成した．次に，利用者の運用や環境などハードウェアではない構成要素からの逸脱要因を抽出するための故障モードを作成した．これらの故障モードは，技術的な知見や開発経験により充実させていくことができる．また，故障モードでは把握できない開発対象固有の逸脱要因を抽出するために，逸脱分析の補完としてガイドワードを拡張した．この拡張した故障モードとガイドワードを用いた逸脱要因の抽出手順を作

成することにより，部品故障や誤操作など広く影響要因を捉えることを行った．

一方，これらの利用者の運用や環境を分析対象とするため，分析対象には，連続的な振る舞いの特性を持つ構成要素と，離散的な振る舞いの特性を持つの構成要素が混在した．そこで，連続的な特性をもつ構成要素に対して，同値分割法を用い，離散的な特性に統一する状態とイベントの記述を定義した．

これらの本研究の成果により，製品における障害がソフトウェア要求仕様の定義前に明確になり，その対策である仕様を定義ことにより，ソフトウェア要求仕様の曖昧さが軽減できる．仕様の曖昧性が軽減されることにより，ソフトウェア開発の手戻りが削減され，ソフトウェア構造の劣化が防げ，製品の品質を保つことができる．また，ソフトウェア要求仕様の定義の前に障害分析を行うことにより，想定外の製品の振る舞いを削減することができ，製品の品質を保証することができる．

第11章　今後の課題

本章では，ESIMを活用していく上での今後の課題と対策について述べる．

11.1　ESIM適用者の制約の軽減

ESIMを適用するためには，熟練度の高い技術者を必要とする．ESIMにおいては，アーキテクチャレベルの情報はシステム構造図や分析マトリクスにドキュメント化するが，集約された状態の記述を分析者の観点に基づいて行うなど，量が多い詳細レベルの情報は，熟練技術者のスキルを活用し，ドキュメント化を省略している．これにより，システムに起こりうるすべての，逸脱シナリオを1つのシート上で分析可能にし，複合要因による障害シナリオの発見を可能にしている．また，逸脱要因の発見や，非正常系分析マトリクスにおける新たな逸脱の発見についても，熟練者のスキルを必要とする．

しかし，組込みソフトウェアの開発量は今後も増加していくのに対し，熟練技術者を開発量の増加に比例して確保していくのは困難と予想される．そのため，より熟練者の負担を軽減するためのESIMの拡張を行って行きたい．

そのためには，分析知識の蓄積と活用により，熟練者の知識をデータベース化し活用していく必要がある．

障害分析の起点となる構成要素の逸脱要因を抽出するために，拡張したガイドワードと拡張した故障モードを作成した．この拡張は，筆者らの過去の経験にもとづいて抽出したものであり，対象範囲が限定されている．このため，今後，知見を集め，様々な製品の特性に合わせたガイドワードや故障モードを拡張し，逸脱要因の分析漏れを削減するための知識の蓄積と活用を図って行きたい．

また，ESIMは，障害シナリオを断片的に構築していく手法であり，状態やイベントの展開という局所的な変化は，共通的なパターンを有している．そのため，ESIMの分析対象の製品や製品の使われれ方は様々であるが，非正常系分析マトリ

クスの障害シナリオを断片的に捉えると類似の傾向が見られる．ESIMの分析データを多く蓄積し，この状態とイベントの展開していくパターンを整理することにより，それらの再利用が可能であると考える．ESIMによる分析事例の再利用が可能になれば，製品開発に熟練していない者への利用も容易になると考える．

11.2 ESIMのツール化による効率の向上

本論文に記載した適用事例では，セルの数が2068であったが，ソフトウェア規模の大きい事例では，セルの数は1万を超える．そのため，対応する構成要素，状態，イベントの確認に時間を要している．ESIMは，着目しているセルの状態，イベント，状態やイベントの履歴，関連する構成要素の局所的な分析である．そこで，非正常系分析マトリクスにおける着目しているセルに関連した項目だけを，自動的に抽出し，表示することにより，分析時間を短縮することができる．

また，非正常系分析マトリクスは，システム構造図による構成要素間の関係から，着目している構成要素に送信されるイベントや関連している構成要素の関係は既知である．この情報をもとに，自動的に関連項目の抽出を行うことが可能である．さらに，非正常系分析マトリクスの状態欄やイベント欄が分析の過程で追加されていくため，検討漏れのセルを自動で抽出し表示することも分析の効率化につながる．

このように，ESIMの支援ツールを構築することにより，ESIMの適用を容易にすると同時に，ESIM記述の抜け漏れを削減し，より確実に分析を行えるようにして行きたい．

11.3 製品以外のへ障害分析手法の展開

家電製品などの製品では，例外的な負荷の発生や例外的な操作など製品のカタログに記載された機能や性能などが達成できない状況下で運用される確率が高いことを述べた．そのため，本研究は，複合障害にも着目した．社会システムでは，製品のように逸脱した状況の下で使用されることが少ないことから，複合的な要因についての検討手法があまり議論されていない．しかし，社会的システムにおいて発生している障害の中にも，想定外の事象が原因となり，複合的

な要因や障害の連鎖が原因となっている．このため，社会システムなどにおける障害分析手法として展開するために，ESIMを拡張することも検討して行きたい．このような大規模系の業務システムなどへの展開を進めていくためには，分析の分割や階層化などが有効と考える．

11.4 コンテクスト・アウェアネス分析への展開

本研究では，製品の品質向上を目的として，製品に起こりうる障害を抽出する手法としてESIMを作成した．しかし，その障害分析の過程において，いくつかの入力の影響シナリオを抽出する際に，入力の情報から，本来の目的ではない分析ができることが多かった．

たとえば，電気ポットの障害分析の例示において，利用者が水位を満水量以上に入れ，さらに，ポットを傾斜して置いた場合，水位センサは満水を判断できず，お湯を沸かし，沸騰することにより，蓋から熱湯が出てくる可能性を抽出した．この際，水位センサは，満水を検知できなかったが，ポットを傾斜して置いた場合に，水温上昇が水位センサから予測できる水温上昇速度より遅いという変化を抽出していた．この水温上昇が水位センサから予測できる水温上昇速度より遅いという情報は，温度センサが本来の温度を測定する目的ではない，しかし，水位の測定や傾斜の測定に使えることを示している．このような特徴を利用することにより，ユビキタス社会における各種センサが，実環境で得られる情報の可能性を分析できる．

コンテクスト・アウェアネスであるコンピュータとは，たとえば物体の動作，電灯の点灯・消灯，あるいは気温の変化といった，さまざまな事柄に関する状況の変化を，コンピュータがネットワークを用いて自動的に認識し，ユーザにサービスを提供するすることのできるようなコンピュータであるとされる[79]．ESIMを用いて，人の行動シナリオと，シナリオ過程にある各種のセンサの状況を把握することにより，人の行動の予知のシステム開発に役立つ可能性を検討して行きたい．

謝辞

本研究を進めるにあたり，多くの方々にご指導，ご協力，ご支援を頂きました．得られた知見と成果は，パナソニック株式会社エコソリューションズ社内の商品開発学部システム商品開発学科の学科長として，多くの研修指導に役立たせて頂いています．ここに謹んで感謝の意を表します．

九州工業大学大学院情報工学研究院情報創成工学研究系　橋本正明名誉教授には，元企業出身者として，企業と大学のあり方のご指導から，製品ソフトウェア開発に関わる全般のご指導とご助言を賜るとともに，本研究の開始からご指導とご鞭撻を賜り，深く感謝申し上げます．

九州工業大学大学院情報工学研究院情報創成工学研究系　吉田隆一教授には，本研究の理論的な再構築から本論文をまとめるまでのご指導を賜り，厚く御礼申し上げます．

筑波大学大学院ビジネス科学研究科　中谷多哉子准教授には，本研究全般のご指導を頂くとともに，研究会やワークショップなど色々な人脈のご紹介を頂き，幅広い知見を得ることができました．心より感謝申し上げます．

九州工業大学大学院情報工学研究院情報創成工学研究系　温暁青教授，久代紀之教授，九州工業大学大学院情報工学研究院電子情報工学研究系　梶原誠司教授には，本論文をまとめるにあたりご指導いただき，厚く御礼申し上げます．

九州工業大学大学院情報工学研究院情報創成工学研究系　片峯恵一准教授には，本研究のご指導とご鞭撻を賜り，心より感謝申し上げます．

九州大学 大学院システム情報科学研究院 情報知能工学部門　鵜林尚靖准教授には，関連技術の紹介と本研究のご指導を賜り厚く御礼を申し上げます．

また，パナソニック株式会社エコソリューションズ社　新屋敷　泰史氏には，共同研究者としてご助言を頂き，また，適用実験にご協力を頂くなど，本研究の実現に協力して頂き，深く感謝申し上げます．

参考文献

[1] IEC, IEC61508 First edition: Functional safety of electrical/ electronic/ programmable electronic safety-related systems, (1998)

[2] 通商産業省産業政策局消費経済課編: 製造物責任法の解説, 通商産業調査会(1994)

[3] 経済産業省: 消費生活用製品のリコールハンドブック (2007)

[4] 経済産業省: 消費生活用製品向けリスクアセスメントのハンドブック（第一版）（2010）

[5] 経済産業省: ～事業者用ハンドブック～製品安全法に基づく製品事故情報報告・公表制度の解説 (2009)

[6] JIS Q2001, リスクマネジメントシステム構築のための指針 (2001)

[7] 英国貿易産業省: Consumer Product Recall -A Good practice guide-(1999/11)

[8] 米国消費生活用製品安全委員会: Recall Handbook(1999/5)

[9] IEC 61508（JIS C 0508）,Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems,(1998)

[10] IEC 61511,Functional safety-Safety instrumented systems for the process industry sector(2003)

[11] IEC 62061,Safety of machinery-Functional safety of safety related electrical,electronic and programmable electronic control systems (2001)

[12] ISO13849-1,Safety of machinery,Safety related parts of control systems,General principles for design(1999)

[13] 経済産業省:2007 年版組込みソフトウェア産業実態調査報告書 (2007/06)

[14] 経済産業省:2009年版 組込みソフトウェア産業実態調査:経営者及び事業責任者向け調査 (2009/06)

[15] 経済産業省:情報システムの信頼性向上に関するガイドライン第 2 版(2009/03)

[16] 情報処理推進機構 要求工学・設計開発技術研究部会:非機能要求とアーキテクチャWG 2006 年度活動報告書 (2007/8)

[17] 独立行政法人情報処理推進機構 ソフトウェア・エンジニアリング・センター エンタプライズ系ソフトウェア開発力強化推進委員会 非機能要求とアーキテクチャ分析 WG:非機能要求とアーキテクチャ分析 WG 報告書 (2011/3)

[18] A. Avizienis, J.-C. Laprie and B. Randell: Fundamental Concepts of Dependability. Research Report No 1145, LAAS-CNRS(2001/4)

[19] Brian Randell:Software Dependability: A Personal View, In the Proc of the 25th International Symposium on Fault-Tolerant Computing (FTCS-25), California, USA, pp 35-41(1995/6)

[20] 三瀬 敏朗, 新屋敷 泰史, 中谷 多哉子, 片峯 恵一, 鵜林 尚靖, 橋本 正明: 高品質組込みソフトウェア設計における非機能要求に着目したプロジェクトマネジメント,プロジェクトマネジメント学会2008年度春季研究発表大会 pp.221-216(2008)

[21] 科学技術振興機構:JST失敗知識データベース http://www.sozogaku.com/fkd/index.html

[22] 日経BP社:IT Pro 相次ぐシステム障害,http://itpro.nikkeibp.co.jp/trouble/index.html

[23] Rozanski, Nick: Software systems architecture : working with stakeholders using viewpoints and perspectives ,Eoin Woods Upper Saddle River, NJ , Munich [u.a.]: Addison-Wesley(2012)

[24] D. Raheja and B. Moriarty: New paradigms in system safety, Journal of System Safety, vol. 42, no. 6(2006/11)

[25] ISO/IEC 25010,Systems and software engineering -Systems and software Quality Requirements and Evaluation (SQuaRE) -System and software quality models(2011/3)

[26] M. Jackson: Problem Frames, Addison-Wesley(2001)

[27] Bruce Powel Douglass :Real-Time UML: Developing Efficient Objects for Embedded Systems (2nd Edition) Pearson Education(1999/10)

[28] IEC 61025. Ed.2,Analysis technique for system reliability - Procedure for fault tree analysis (FTA),(2006)

[29] Center for Chemical Process Safety (CCPS): Guidelines for Hazard Evaluation Procedures 3rd Edition Wiley-AIChE, PP158-PP166,(2008/4)

[30] Sam Mannan: Lees' Loss Prevention in the Process Industries, Third Edition: Hazard Identification, Assessment and Control volume 1,Butterworth Heinemann,(2005/1)

[31] IEC 60812 Ed.2,Analysis technique for system reliability - Procedure for failure mode and effects analysis (FMEA),(2006)

[32] N.G. Leveson:SAFEWARE System safety and computers,Addison-Wesley(1995)

[33] Pumfrey, D. J.: The Principled Design of Computer System Safety Analyses, PhD Thesis, The University of York.pp129-pp174(1999)

[34] Brian Tyler, Frank Crawley,Malcolm Preston :HAZOP Guide to Best Practice, 2nd Edition (2008/4)

[35] Felix Redmill,Morris Chudleigh,James Catmur:System Safety HAZOP and Software HAZOP Wiley(1999/6)

[36] Pumfrey, D. J.: The Principled Design of Computer System Safety Analyses, PhD Thesis, The University of York.pp129-pp174(1999)

[37] Bondavalli A. and Simoncini L.:Failure Classification With Respect to Detection, in First Year Report Task B: Specification and Design for Dependability. ESPRIT BRA Project 3092: Predictably Dependable Computing Systems(1990)

[38] TRIZ 研究会メンバー:TRIZ の理論とその発展 システマテック・イノベーション 産業能率大学出版部 (2003/4)

[39] 澤口学:VEとTRIZ 革新的なテクノロジーマネジメント手法入門 同友館(2002/3)

[40] 三菱総合研究所 知識創造研究部:革新的技術開発の技法 図解 TRIZ 日本実業出版社 (1999/7)

[41] 津波古和司,大塚正樹,鈴木悟士,水本直志:KT法TRIZのコラボレーションによるハ-ドディスクドライブ信頼性向上の実務適用例第7回日本TRIZシンポジウム 2011(2011/8)

[42] 構造化知識研究所:SSMとは http://www.ssm.co.jp/ssm/index.html

[43] Samuel Kotz, Yan Lumelskii, Marianna Pensky:The Stress-Strength Model and Its Generalizations: Theory and Applications,World Scientific Publishing Co Pte Ltd (2003/03)

[44] Ishimatsu, Takuto, Leveson, Nancy, Thomas, John, Katahira, Masa, Miyamoto, Yuko, Nakao, Haruka:Modeling and Hazard Analysis using STPA. Proceedings of the International Association for the Advancement of Space Safety Conference, Huntsville, Alabama(2010/5)

[45] C. W. Johnson and C. M. Holloway: The ESA/NASA SOHO mission interruption: Using the STAMP accident analysis technique for a software related ‘ mishap:, Software: Practice and Experience, vol. 33, pp. 1177-1198(2003)

[46] N. Leveson:A new approach to hazard analysis for complex systems, in Proceedings of the 21st International System Safety Conference (ISSC). Ottawa, Canada: Systems Safety Society(2003/8)

[47] 紫合治:組込みソフトウェアにおける非正常処理の抽出, 組み込みソフトウェアシンポジウム 2005(ESS2005) 論文集,pp.144-147(2005)

[48] N.G. Leveson: SAFEWARE System safety and computers, Addison-Wesley(1995)

[49] 金周慧, 松原豊, 高田広章:階層型状態遷移図に着目した安全分析手法 9th Workshop on Critical Software System(2011/11)

[50] 金周慧,松原豊,高田広章: 状態遷移図に着目した安全分析手法, 電子情報通信学会論文誌A(2012/2)

[51] 青山幹雄,平石邦彦,内平直志:ペトリネットの理論と実践,朝倉書店（1995）

[52] 熊谷貞俊,薦田憲久:ペトリネットによる離散事象システム論,コロナ社（1995）

[53] Nancy G. Leveson and Janice L. Stolzy: Safety analysis using petri nets, IEEE Transaction on Software Engineering, SE-13(3), 386-397.(1987/3)

[54] Frederick T. Sheldon Stefan Greiner Matthias Benzinger:Specification, Safety and Reliability Analysis Using Stochastic Petri Net Models,Proceedings of the Tenth International Workshop on Software Specification and Design (IWSSD'00) IEEE(2000)

[55] Reza, H. Krishna, V. ,Hildle, J. :A Safety Analysis Method Using Fault Tree Analysis and Petri Nets,Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on Date of Conference: 27-29(2009/4)

[56] 平山雅之,岸本卓也,水野修,菊野亨:Use-Caseを利用したソフトウェアフォールトに対するSS-FTAの提案,信学技報,Vol.SS99, No.53, pp.25-32(2000)

[57] F. Redmill, M. Chudleigh, J. Catmur:System safety Hazop and software hazop, John Wiley Sons(1999)

[58] J. Dehlinger and R.R. Lutz:Software fault tree analysis for product lines, Proc. of the 8th International Symposium on High Assurance Systems Engineering, pp.12-21(2004)

[59] Feng, Q. and R. R. Lutz: Bi-Directional Safety Analysis of Product Lines, In Journal of Systems and Software, Vol. 78, pp.111-12(2005)

[60] 斉藤直希,小川清,水口大知,菊池達也,大西秀一,長谷部浩二, 堀武司:コンポーネントソフトウェアに対するハザード分析手法の検討,第五回システム検証の科学技術シンポジウム（SSV2008） 講演論文集,算譜科学研究速報 AIST-PS-2009-001,pp. 53-64,(2009)

[61] 堀武司,岡村真吾,服部智幸,斉藤直希,小川清: 機能安全対応組込ソフトウェア開発におけるB メソッド導入の試み,第五回システム検証の科学技術シンポジウム（SSV2008） 講演論文集,算譜科学研究速報AIST-PS-2009-001,pp. 123-129(2009)

[62] Sergiy A. Vilkomir, Jonathan P. Bowen and Aditya K. Ghose: Formalization and assessment of regulatory requirements for safety-critical software Innovations in Systems and Software Engineering Volume 2, Numbers 3-4,165-178(2006)

[63] 新屋敷泰史,三瀬敏朗,橋本正明,片峯恵一,鵜林尚靖, 中谷多哉子:情報フロー・ダイアグラムによる組み込みソフトウェア非正常系の要求分析の一手法, ” Vol.48, No.9, 情処学論,pp.2894-2903(2007)

[64] 畑中久典,新屋敷泰史,三瀬敏朗,亀谷秀洋,橋本正明,鵜林尚靖,片峯恵一,中谷多哉子:組込みソフトウェア非正常系の概念モデルによる情報フロー・グラフの解析,電子情報通信学会技術研究報告(知能ソフトウェア工学),Vol.104, No.431, 電子情報通信学会,pp.19?24, (2004)

[65] Yasufumi, S., Toshiro, M., Masaaki, H., Keiichi, K., Naoyasu, U., and Takako, N.: Enhancing the ESIM (Embedded Systems Improving Method) by Combining Information Flow Diagram with Analysis Matrix for Efficient Analysis of Unexpected Obstacles in Embedded Software, Proceedings of the 14th Asia-Pacific Software Engineering Conference (APSEC ’07), pp.326-333, (2007)

[66] Hidehiro, K., Yasufumi, S., Toshiro, M., Masaaki, H., Naoyasu, U., Keiichi, K., and Takako, N.: Information Flow Diagram and Analysis Method for Unexpected Obstacle Specification of Embedded Software, The Proceedings of the Knowledge-Based Software Engineering JCKBSE 2006, pp.115?124, (2006)

[67] J. McDermott: Abuse-Case-Based Assurance Arguments. In 17th Annual Computer Security Applications Conference (ACSAC’01), pp. 366-374, New Orleans, Lousiana, (2001/12)

[68] Guttorm Sindre and Andreas L. Opdahl: Eliciting security requirements with misuse cases. Requirements Engineering, Vol. 10, No. 1, pp. 34 - 44(2005/1)

[69] Ian Alexander: Misuse Cases: Use Cases with Hostile Intent. IEEE Software, Vol.20, No. 1, pp. 58-66(2003/1)

[70] Donald Firesmith: Security Use Cases. Journal of Object Technology, Vol. 2, No. 3, pp. 53-64,(2003/5)

[71] L. Lin, B. Nuseibeh, D. Ince and M. Jackson: Using abuse frames to bound the scope of security problems, Proc. of the 12th International Requirements Engineering Conference, pp.354-355(2004)

[72] 中谷 多哉子, 三瀬 敏朗, 新屋敷 泰史, 片峯 恵一, 鵜林 尚靖, 橋本 正明:実世界分析に基づくシステム化境界の定義,日本ソフトウエア科学会 第12回ソフトウェア工学の基礎ワークショップ (FOSE 2005), pp.221-226 (2005).

[73] Shaw, A.C. :Communicating real-time state machines Software Engineering, IEEE TransactionsVolume: 18,Issue: 9 805 - 816(1992/9)

[74] Gouda, M. Yao-Tin Yu: Communications, IEEE Transactions Vol.32,Issue.7 pp.779 - 788(1984/7)

[75] Rajeev Alur, Sampath Kannan and Mihalis Yannakakis: Communicating Hierarchical State Machines ICALP'99 LNCS 1644, pp.169-178(1999)

[76] Chris Barrett,Harry B.Hunt,Madhav V.Marathe, S.S.Ravi,Daniel J.Rosenkrantz and Richard E.Stearns: Analysis Problems for Sequential Dynamical Systems and Communicating State Machines Foundations of Computer Science 2001,Vol.213,159-172(2001/6)

[77] Burnstein, Ilene: Practical Software Testing, Springer-Verlag, p. 623(2003)

[78] Glenford J. Myers,Tom Badgett,Corey Sandler:The Art of Software Testing,Wiley, (2011/11)

[79] Dey, A. K. and Abowd,G. D.: Toward a Better Understanding of Context and Context-Awareness, In Proceedings of the CHI2000 Workshop on The What, Who, Where, and How of Context-Awareness,(2000)

[80] IEEE Std 1044-1993(R2002)IEEE Standard Classification for Software Anomalies (2002)