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

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[†††]

本論文は，組み込みソフトウェアの品質向上を目的として，ソフトウェア仕様化時に想定が困難な組み込みシステムの障害発生可能性を発見するための手段として，情報フロー・ダイアグラムと，それを適用した要求分析手法を提案する．近年，組み込みソフトウェアは大規模化し複雑化しているので，ソフトウェアの開発期間は長くすることが望ましい．しかし，現実は逆に短くなっている．このような状況は，ソフトウェアの品質確保をますます，困難にしている．ところで，高品質な組み込みソフトウェアを開発するには，障害発生可能性を，システムの分析・設計時に十分検討しておくことが重要である．そこで，本論文の手法を提案し，事例に適用して，その妥当性を確認した．また，情報フロー・ダイアグラムについては，実際の製品に適用して記述実験を行い，組み込みソフトウェア熟練技術者の指導の下，未熟練者が記述できることを確認した．

A Requirements Analysis Method of Unexpected Obstacles in Embedded Software by Applying Information Flow Diagram

Yasufumi Shinyashiki,[†,††] Toshiro Mise,[†,††]
Masaaki Hashimoto,[††] Keiichi Katamine,[††] Naoyasu Ubayashi[††]
and Takako Nakatani[†††]

In order to improve the quality of embedded software, this paper proposes Information Flow Diagram and a requirements analysis method by applying the diagram for finding unexpected obstacles, in the software, which are difficult to be assumed in software specification. Recently, embedded systems have become large in scale and complicated. Therefore, the development cycle of the systems is desirable to be lengthened. However, the cycle is required to be shortened on the contrarey. This trend of industry troubles the quality of embedded software. In order to improve the quality of the software, unexpected obstacles should be carefully analyzed in its specification and design processes. Therefore, we propose the analysis method and describe a case study. Moreover, we outline a description experiment of Information Flow Diagram, by using an actual product, which proved that novice engineers can make Information Flow Diagram under the lead of embedded software expert engineers.

1. はじめに

計算機システムを組み込んだ家電製品等は，組み込みシステムの存在も意識しない様々なユーザによって，様々な環境の下で使用される[1),2)]．しかも，長期にわたって，安心して，安全に使用できることが期待される．そのような優れた製品の品質を得るため，組み込みシステムのソフトウェアは，ソースコード量にしてその 70%強が，製品の装置の誤作動や，利用者の誤操作，動作環境の悪影響等の例外条件へ対処するために費やされている．ところで，組み込みソフトウェアは大規模化し複雑化しているので，開発期間は長くするのが望ましい．しかし，開発期間は逆に短くなっている[3)]．そのため，例外条件を漏れなく想定することは，ますます困難となっている．しかも，組み込みソフトウェアの例外条件を想定するには，ソフトウェアの知識だけでなく，装置や，利用者，動作環境に関するソフトウェア以外の知識も必要である[4)]．そのため，ソフトウェア技術者は，例外条件を，ソフトウェア仕様から漏らしてしまうことも起きる．現実に，そのよう

† 松下電工株式会社
Matsushita Electric Works, Ltd.
†† 九州工業大学
Kyusyu Institute of Technology
††† 筑波大学
University of Tsukuba

な仕様漏れが，製品に障害を発生させ，設計やり直しによる開発遅れを起こしている．そのため，例外条件を正確に仕様化すれば，組み込みソフトウェアの品質と開発効率が改善されることが期待される．

そこで，筆者らは，組み込みソフトウェアの例外条件のうち，特に仕様から漏れやすいものに着目して，それをまとめて非正常系と呼び，その漏れを防ぐための要求分析手法を研究している[5)~9)]．本論文においては，正常系および非正常系を次のように定義する．正常系は，ソフトウェア使用マニュアルに記載されるような，設計工程開始時にすでに定義されているソフトウェアの振舞いを指す．一方，非正常系は，正常系から逸脱した例外条件による振舞い，たとえば装置の軽微な故障や，過負荷，利用者の誤操作，環境の影響等がいくつか重なって起きる障害を指す．そのような非正常系を漏れが少なく想定することは，現実には組み込みソフトウェアの熟練技術者にしかできない．そこで，筆者らは，熟練者が従来，整理はせずに用いてきた要求分析方法を，手法として体系化するために研究している．

非正常系に関する熟練者の要求分析方法の中に，動作環境も含めた組み込みシステムの中を流れる情報に着目して，その情報が，ソフトウェア技術者の意図から逸脱して伝達されることによって，障害に至るという考え方がある．さらに，その分析の際，従来は装置の障害分析に適用してきたFTA（Fault Tree Analysis）[10)]や，FMEA（Failure Modes and Effect Analysis）[11)]，HAZOP（the HAZard and OPerability studies）[12)]の考え方も用いている．

そこで，本論文は，動作環境も含めた組み込みシステムにおける情報の流れを表すための情報フロー・ダイアグラムIFD（Information Flow Diagram）を提案し，そのダイアグラムを用いて，FMEAとFTA，HAZOPの考え方を統合的に適用して非正常系の要求分析を行うための手法も提案する．本提案のIFDの特徴は，組み込みシステムの非正常系に装置が大きく影響するので，装置とその接続を表すためのデバイス・ダイアグラムを，IDEF0（Integrated DEFinition）[13)]を適用したプロセス・ダイアグラムに組み合わせたことである．トップダウンに分析するFTAや，ボトムアップに分析するFMEAやHAZOPを個別にソフトウェアに適用した研究[14)~16)]はすでにある．また，要求工学の分野におけるミス・ユースケース[17)]や，アブユース・フレーム，ゴール指向の例外処理[18)]の研究も，主にゴール指向のトップダウンの分析方法をとっている．そこで，FMEAとFTA，HAZOPの考え方をIFDの上で統合することにより，非正常系の分析漏れの防止を図ることも，本提案の特徴である．以下，2章に非正常系の要求分析手法の要件を述べる．3章ではIFDを述べ，4章でIFDを用いた非正常系の要求分析手法を述べる．その適用事例は5章に述べ，6章で考察する．

2. 非正常系の要求分析手法の要件

組み込みソフトウェアの非正常系を対象にした要求分析手法を提案するにあたり，その要件を以下に述べる．業務系システムは，CPUや，メモリ，ハード・ディスク等の限定された種類の装置を用いて動作する．一方，組み込みシステムは，センサやコントローラ等の様々な種類の装置を用いて動作し，その装置種別も製品種別ごとに異なる．しかも，その装置の例外条件が，非正常系の振舞いを起こす大きな原因となっている．また，組み込みシステムは，利用者による操作のほか，動作環境からくる光や温度等の情報もトリガとなって動作する．しかも，その利用者や動作環境の例外条件も，非正常系の振舞いを起こす大きな原因となっている．そのため，装置に着目するとともに，利用者と動作環境にも着目することが求められる．

ところで，障害が起きるには，その原因が存在し，その原因から障害へ至る因果関係がある．しかも，原因を起こした装置と，障害が認知された装置が異なる場合が多い．その異なる装置の間を流れる情報が，正常な内容から逸脱することによって，前述の因果関係が伝達される．そのため，組み込みシステムの中の情報の流れと，その情報の流れを作り出す情報処理プロセスに着目することも求められる．

装置故障等の重大な例外条件が，唯一の原因となって発生する障害は，想定が困難ではない．一方，装置の劣化や，過負荷，利用者の操作タイミングのずれ，環境の影響等は，個々に見れば重大ではない．しかし，それがいくつか重なって起きる障害は，想定が困難である．この例外条件の重なりは，前述の情報処理プロセスが，軽微に逸脱した情報をいくつか入力して処理した結果，重度に逸脱した情報を出力することによって起きる．実際，組み込みソフトウェアの熟練技術者は，軽微なものも含めた例外条件の重なりを障害シナリオとしてとらえ，それを丹念に追うことによって，見落としやすい障害も想定できる．そのため，例外条件の重なりを表すための障害シナリオに着目することが求められる．

ところで，組み込みソフトウェアの熟練技術者がFMEAの考え方を適用する際は，装置の種別ごとに

パターン化された故障に最初に着目して分析する．その故障の例外条件発生は，障害シナリオの始点となる．また，FTA の考え方を適用する際は，製品の品質仕様に反する障害に最初に着目して分析する．その障害の例外条件発生は，障害シナリオの終点となる．また，HAZOP の考え方を適用する際は，組み込みシステムの中を流れる情報の逸脱に最初に着目して分析する．その情報逸脱の例外条件発生は，障害シナリオの中間点となる．このように，上記の 3 手法をまとめてみると，最初に着目する障害シナリオの中の箇所が相互に異なるという利点を持つ．しかし，その 3 手法を個別に適用すると，その 3 手法にまたがって判明する例外条件の重なりを見逃しやすいという欠点がある．この欠点を補完して例外条件の重なりの分析漏れを防止するには，その 3 手法の考え方を統合する分析手法が望まれる．

3. 情報フロー・ダイアグラム

非正常系の要求分析に用いるための IFD を，本章において提案する．前章の要件に述べたように，非正常系の要求分析は，正常系の設計において決められた装置と情報処理プロセスに着目することが求められている．そのため，IFD は，以下の各節に述べるデバイス・ダイアグラム DD（Device Diagram）とプロセス・ダイアグラム PD（Process Diagram）を組み合わせて構成する．

3.1 デバイス・ダイアグラム（DD）

DD は，設計において決められた装置と，その装置の接続を図示する．前章の要件に述べたように，製品内の装置のほかに，製品の利用者と動作環境にも着目することが求められている．そのため，DD は製品内の装置のほかに，製品の利用者を含めた動作環境内の対象物も表すものとする．たとえば，光センサを用いて昼夜判断を行うシステムにおいては，光を発する太陽も，装置として表すことができる．

DD 中に装置を図示するには，**図 1** に示すように四角形を用い，その四角形を三分して，順に装置と属性と故障パターンのそれぞれの名前を記述する．故障パターンは，その装置において発生し得るノイズや，劣化，断線等の故障の種類を示す．属性は，非正常系の分析に必要なもの，たとえば，故障と認識されるノイズのレベルや，装置劣化が起きる経過時間等を表すのに用いる．

ところで，情報は，装置の間を，電気や，光，音等の物理媒体によって伝えられる．しかも，その物理媒体に減衰や断絶等の故障が起きれば，情報の流れが阻害され，障害に至ることがある．そのため，非正常系の要求分析にあたっては，情報を伝えるための物理媒体に着目することが重要である．そこで，その物理媒体をキャリアと呼ぶことにし，キャリアが装置を接続しているものと見なす．DD には，**図 2** に示すようにキャリアを菱形と矢印を用いて図示し，その矢印が 2 つの装置を接続する．矢印の方向は，情報が流れる方向を示す．キャリアと属性と故障パターンのそれぞれの名前は，菱形と重ね合わせて記述する．

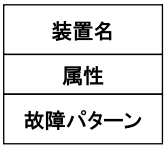

図 1 装置
Fig. 1 Device.

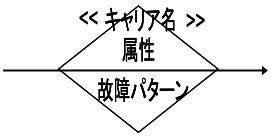

図 2 キャリア
Fig. 2 Carrier.

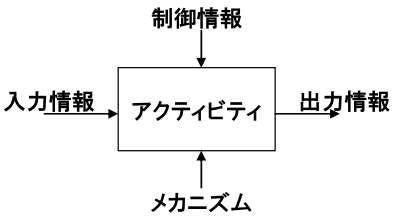

図 3 IDEF0 のプロセス
Fig. 3 Process of IDEF0.

3.2 プロセス・ダイアグラム（PD）

PD には，プロセスの機能を静的に表すための手法 IDEF0 を適用する．その手法においてはプロセスをアクティビティ単位に分割して，そのアクティビティを，**図 3** に示すように四角形を用いて図示し，その中にアクティビティの名前を記述する．そのアクティビティには，入力情報と，出力情報，制御情報，メカニズムのそれぞれを，必要に応じて矢印によって指定し，その名前も記述する．アクティビティは，それらの情報の流れによって，相互に接続する．アクティビティは入力情報をメカニズムによって処理し，出力情報を作り出す．制御情報は，その処理の条件を表す．

PD は，DD に示した各装置が処理するアクティビティと，そのアクティビティ相互の情報の流れを表す．

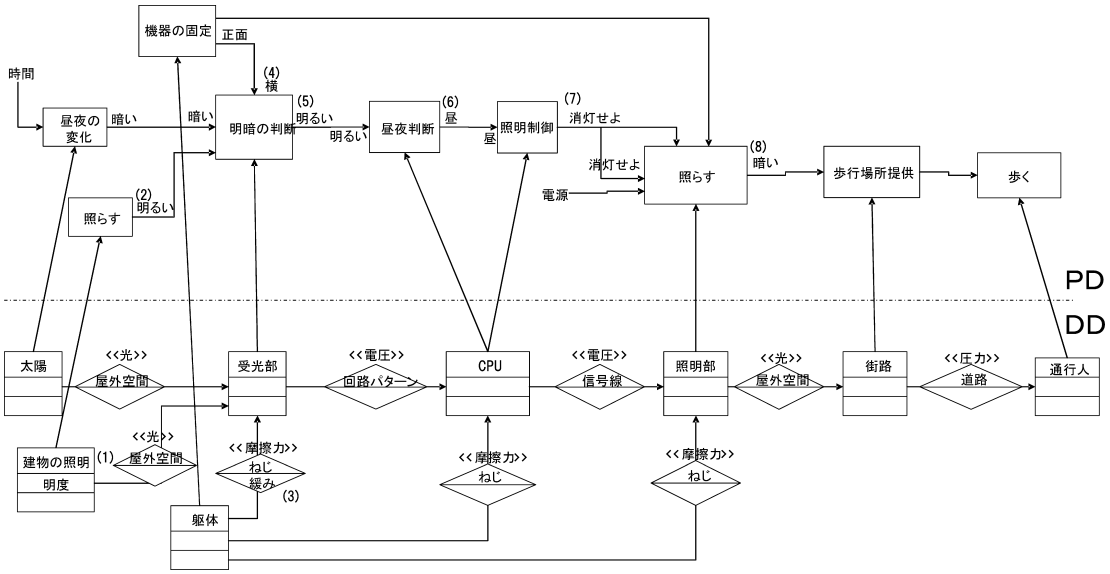

図 4 情報フロー・ダイアグラムの例
Fig. 4 Example of Information Flow Diagram.

DD と PD の関係は，装置と，その装置が処理するアクティビティを，メカニズムの矢印によって接続して表す．ところで，IDEF0 は階層的に詳細化して，記述することができる．その詳細化の階層に対応して，DD においても装置をサブシステム等にまとめて階層的に図示することができる．

IFD には組み込みシステムのほか，利用者や動作環境も含めている．そのため，IFD は閉世界になっており，IFD の外部と情報の授受はない．なお，IFD において情報はキャリアと明確に区別するので，「情報フロー」の用語を用いる．正常系の IFD の例を**図 4** に示す．

4. 非正常系の要求分析手法

本章では，IFD を用いて，FTA と FMEA と HAZOP の考え方を統合した非正常系の分析手法を提案する．

4.1 分析の前提条件

分析対象システムの開発が，以下に述べる状況に達したとき，本手法を適用できるものとする．最初に，要求分析工程において，システムの要求仕様として，システムが提供すべき機能と，安全基準としてシステムが回避すべき障害現象が与えられている．ここで，障害現象とは，ユーザに害を及ぼす，システムおよび動作環境の振舞いを指す．また，例外現象とは，システムおよび動作環境が，システムの要求仕様を達成するのに必要な振舞いから逸脱することを指す．なお，例外現象を引き起こす条件が，例外条件である．次に，システムの要求仕様を受けた設計工程では，システムのハードウェア構成図と，ハードウェアの故障について記述した設計ドキュメントが存在する．ソフトウェア設計については，ソフトウェアのモジュール説明書が存在する．このように，正常系の設計が進んだ後，非正常系の分析に着手する．

4.2 分析の手順

非正常系の要求分析の手順を以下に述べる．

（1）正常系 IFD の記述

本手法の適用に先立ち，設計内容に基づいて，正常系の IFD を記述する．

（2）回避すべき障害現象の追記

正常系の IFD に対して，回避すべき障害現象を追記する．具体的には，各障害現象の発生をプロセスに追記し，その障害によって周辺に及ぼす悪影響も，そのプロセスが出力する情報フローの障害現象と想定して PD に追記する．このステップは，障害シナリオの終点に着目したものである．

また，PD 上の各プロセスの間の情報フローに対して，ガイドワードを適用することによって，情報フローが例外になる現象を想定する．この例外現象も PD に記述する．**表 1** に，ガイドワードの例を示す．このステップは，障害シナリオの中間点に着目したものである．

（3）IFD の構造による例外条件の検討

上記（2）項に述べたそれぞれの例外現象に対して，

表 1 ガイドワードの例
Table 1 Examples of Guide-Words.

影響種別	ガイドワード
挙動	停止，不安定，固定，…
負荷	過大，過少，…
意味	範囲外，未定義，…
手順	順序違い，タイミング違い，…
頻度	継続的，反復的，一時的，…

その例外条件を，IFD の構造を用いて検討する．PD 上において例外現象が想定された情報フローは，その情報を出力するプロセスに結び付いている．このプロセスを当該プロセスと呼ぶ．当該プロセスは，入力情報と制御情報，およびメカニズムとなる装置のみに結び付いている．そのため，当該プロセスから出力される情報の例外現象を起こす例外条件は，当該プロセスの入力情報と制御情報，およびメカニズムとなる装置の例外条件のいずれか，あるいはそれらの組合せのみによって表される．さらに，その入力情報と制御情報の例外条件は，入力情報と制御情報自身が情報フローパス上に存在するため，その情報の送信プロセスの例外条件か，その情報フローパス自体の例外条件のいずれか，あるいはそれらの組合せのみによって表される．一方，メカニズムとなる装置における例外条件も，装置の実現手段がハードウェアとソフトウェアからなるため，ハードウェアの例外条件か，ソフトウェアの例外条件のいずれか，あるいはそれらの組合せのみによって表される．そのため，例外現象を起こす例外条件は，以下のように分類できる．

1. 当該プロセスのメカニズムとなる装置の例外条件．これは以下の 2 つに分類される．
   (i) 装置の物理的な故障．これは部品の故障に関する設計ドキュメントから得られる．
   (ii) 装置の論理的な故障．これは装置のソフトウェア設計の不備を想定することに相当する．
2. 当該プロセスの入力情報と制御情報の例外条件．これは以下の 2 つに分類される．
   (i) 入力情報と制御情報の送信プロセスの例外条件．
   (ii) 入力情報と制御情報が伝達される情報フローパス自体の例外条件．これは，その情報フローパスの故障か，入力情報と制御情報を運ぶキャリアと同じキャリアを発信する，なりすまし装置からの情報受信のいずれかである．
3. 上記に述べた 2 つ以上の例外条件の複合

上記の分類に従って，着目している例外現象を起こす例外条件を検討する．その例外条件が存在する場合，これを採用する．その例外条件が存在しない場合，その例外現象は起こりえないので，検討は中止する．上記 2.，3. については，例外条件自身が情報フローの例外現象となっているので，その例外現象に対して，上記分類による検討を，再度繰り返し実施する．

上記のなりすましが発生しうると判断された場合，そのなりすまし装置とキャリアを DD に追記する．さらに，なりすまし装置の動作をプロセスとしてとらえ，そのプロセスを PD に追記する．また，そのプロセスから，なりすまし情報の送信先プロセスへ，情報フローを追記する．

本ステップにおいて，想定された例外現象から，故障，また，なりすましの装置に至る分析の中で判明した情報フローのつながりを障害シナリオ断片と呼び，IFD に記載しておく．なお，例外現象から，それが発生するための原因となる例外条件を検討するのは，因果関係を逆にたどるトップダウン分析である．

（4）障害シナリオの構築

前ステップ（3）において想定された故障を出発点として，因果関係に沿ってボトムアップに障害シナリオを構築する．具体的には，個々の故障がプロセスの出力情報に与える影響を想定する．次に，その出力情報が，PD 上の情報フローに沿って伝達される方向に情報を追跡し，伝達された情報が，障害を表す情報へ到達するまで追跡する．故障を出発点として，障害を表す情報まで到達する一連の情報フローのつながりが，1 つの障害シナリオとなる．そのため，このステップは，障害シナリオの始点に着目したものである．

障害シナリオの追跡において，故障のすべての組合せを考慮しなければならない．しかし，その組合せは情報フローの合流によって発生するので，障害シナリオの追跡は，以下の手順によって行う．

1. 追跡している情報を入力するプロセスに着目する．そのプロセスのすべての入力と制御の情報，および，メカニズムについて，その可能な例外現象のすべての組合せを考慮する．その組合せに対して，着目しているプロセスに，例外現象を持つ出力情報を出す可能性があれば，その出力情報に対して，追跡を繰り返す．
2. 上記 1. の追跡において，着目しているプロセスが，例外現象を持つ出力情報を出す可能性がなければ，その追跡を中止する．
3. 上記 1. の追跡において，着目しているプロセスとその出力情報が，前ステップ（3）におい

て述べた障害シナリオ断片の中に現れている場合，その断片を，それ以後の追跡において再利用する．

以上に述べた追跡を，すべての故障に対して行い，そのつど得られた障害シナリオが，本手法の成果物となる．なお，ステップ（3）において先に故障やなりすましの確実な存在を確認して，その後，ステップ（4）において障害シナリオを構築する．論理的には，その逆の手法，すなわち先に障害の可能性を確認する手法も考えられる．しかし，熟練技術者は実在性を重視するので，本論文に述べた手法を好む傾向が高い．

5. 適 用 事 例

前章に述べた非正常系の要求分析手法の適用事例を述べる．また，実際の製品を対象にしたIFDの記述実験も述べる．

5.1 事例の概要

事例として，光センサ付き街灯システムを用いる．本システムの要求仕様は，光センサによって外界の明るさを検知し，昼夜を判断して，その結果に応じて照明部の蛍光灯を点消灯させることである．**図 5** に本システムの外観を示す．また，本システムに対し，回避すべき障害として**表 2** に示した現象が示されているものとする．

5.2 事 例 分 析

事例の具体的な分析の手順を，以下に述べる．

（1）正常系 IFD の記述

事例の街灯システムの正常系IFDを，図4に示す．なお，同図には障害シナリオも記入している．

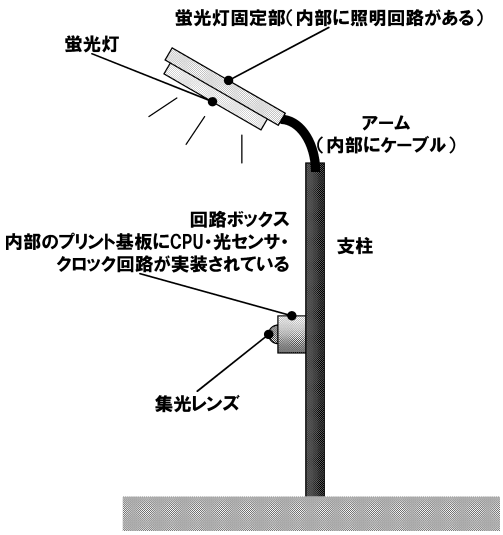

図 5 光センサ付き街灯システムの外観
Fig. 5 Outlook of lighting systems.

（2）回避すべき障害現象の追記

表2に示された障害をIFD上に追記する．検討した結果，燃焼については照明部の装置において発生する可能性があるので，IFD上の照明部とメカニズムによって関連付けて追記する．一方，感電については，本事例では発生する可能性がないので，無視する．燃焼について正常IFDに追記した部分図を**図 6** に示す．

また，PD上の各情報フローに対してガイドワードを適用し，例外現象を想定する．**表 3** に例外現象の一部を示す．

（3）IFD の構造による例外条件の検討

それぞれの例外現象についてトップダウンに分析して，例外条件の有無を検討する．一例として，照明部をメカニズムとするプロセス「照らす」の出力情報が，本来「明るい」ときに，「暗い」となる例外現象について，以下に検討する．なお，分析者が，分析の途中において想定できていなかった故障を，さらに分析手順を踏むことによって発見できる．その例を示すため，分析の途中において分析者がすでに想定できていた故

表 2 光センサ付き街灯システムが回避すべき障害
Table 2 Obstacles to be avoided in lighting systems.

品質項目	回避すべき障害	キャリア
安全性	燃焼	熱
	感電	電圧
	…	…

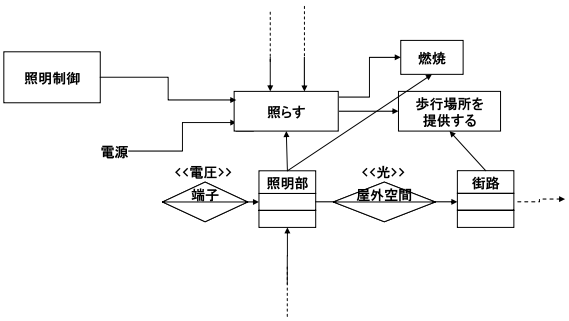

図 6 障害現象プロセス「燃焼」の追記
Fig. 6 Addition of process “burning”.

表 3 事例における例外現象の一部
Table 3 A part of exception phenomenon in this example.

発信元プロセス	ガイドワード	例外現象
明暗の判 断	停止	出力情報が発信されない．
	過少	入力情報「明るい」に対して，「暗い」が発信される．
	一時的	「暗い」の発信が続く中，ごく短時間「明るい」が発信される．

障と例外現象，および想定できていなかった故障を以下のように仮定する．

分析途中において想定できていた故障と例外現象：

- 照明部の破損や，汚損，経年劣化
- 信号線の断線による入力情報の遮断
- 電源の遮断と低下
- 前段プロセス「照明制御」の例外現象

まだ想定できていなかった故障：

- ねじの緩みによる照射方向の不正
- 信号線のノイズ重畳による入力情報の変化

上記の想定した 4 項目のうち,「前段プロセス『照明制御』の例外現象」については，4.2 節において述べた例外条件の分類に沿ってさらに分析する．

分析によって想定された故障の 1 つとして,「近隣の建物の照明が太陽になりすます」が想定されたとする．この故障は，照明が受光部に光を当てることによって，夜にもかかわらず「明るい」という情報を伝達する例外現象を示している．

これとは別に，躯体をメカニズムとしたプロセス「機器の固定」の出力であり，受光部にとっては制御情報となる「受光部に正面を向かせる」について，その例外現象「受光部に横を向かせる」を起こす条件として，受光部を躯体に固定しているネジの「緩み」を故障として想定する．

（4）障害シナリオの構築

想定した故障のうち,「近隣の建物が太陽になりすます」を出発点として，ボトムアップに障害シナリオの構築を開始する．PD において，建物の照明のプロセス「照らす」から出力される情報「明るい」を追跡する．PD の情報フローによれば，この情報は受光部のプロセス「明暗の判断」に到達する．ここで，そのプロセスの入力情報と制御情報とメカニズムの例外現象を組み合わせた場合，プロセス「明暗の判断」がどのような例外現象を持つ出力情報を出すかを検討する．

その組合せの中に,「明るい」という入力情報と,「受光部に横を向かせる」という制御情報の組合せが存在する．それぞれ単体では，プロセス「明暗の判断」の出力情報に対する影響はない．しかし，両者の組合せによって，建物からくる照明を効率良く受光する位置関係となり，結果として「暗い」と出力すべきところを「明るい」と出力する例外現象に至る可能性がある．

この例外現象によって変化した情報を追跡すると，最終的には夜であるにもかかわらず，照明部が点灯せず，プロセス「照らす」が街路をメカニズムとするプロセス「歩行場所を提供する」へ情報「暗い」を出力するという障害現象に至る障害シナリオを構築できる．図 4 に，この障害シナリオの追跡を，(1)，(2)，･･･，(8) 付きのコメントを用いて記入した．

5.3 IFD の記述実験

本論文に提案した分析手法を今後，本格的に実験するにあたり，その手法の効果的な運用体制を検討するため，IFD の記述実験を行った．この実験には，現実的な実験結果を得るため，実際のコンピュータ組み込み製品を適用した．その組み込みソフトウェアは，C プログラムのソースコードが約 50 K 行の規模であった．この実験においては，組み込みソフトウェアの熟練技術者 1 名の指導の下，学生 3 名が IFD を記述した．

実験の手順としては，学生が当製品の設計ドキュメントを参照して，正常系と非正常系の IFD をそれぞれ記述した．その記述において，PD と DD は 3 階層に詳細化した．DD には，当製品と動作環境を分けた第 1 階層と，当製品をサブシステムに分けた第 2 階層，さらにサブシステムをハードウェア・ブロックに分けた第 3 階層を設けた．また，PD にも，DD の各階層に記述されたシステムとサブシステムとハードウェア・ブロックのそれぞれについて，その機能を IDEF0 に従って記述した 3 階層を設けた．

正常系の IFD の記述において，組み込みソフトウェアの熟練技術者が学生を指導した内容は，DD を 3 階層に分けて記述する指針のみであった．3 学生がサブシステムに分かれて記述した IFD は相互に接続したが，その際の IFD 修正はわずかで済んだ．また，当製品開発時にすでに熟練技術者が指摘していた障害シナリオを，3 学生が正常系の IFD に書き加えることにより，非正常系の IFD を記述することもできた．その際，熟練技術者が学生を指導した内容は，障害シナリオの簡単な説明のみであった．なお，非正常系 IFD の階層的な詳細化において，装置の故障や情報フローの障害現象に着目した，正常系とは異なる視点の重要性も分かった．

6. 考　　察

本論文に提案した分析手法について，以下に考察する．5.3 節に述べた実験により，組み込みソフトウェアの熟練技術者の指導の下，未熟練の学生が，正常系 IFD の記述から手順を追うことによって，障害シナリオも含めた非正常系 IFD を記述できることが分かった．この実験結果より，IFD を用いた非正常系の要求分析手法の運用組織形態として，組み込みソフトウェアの少数の熟練技術者が，多数の未熟練技術者を指導して分析を進めるのが効果的と想定される．

ところで，筆者らは，本論文に述べた IFD による

要求分析手法のほかに，分析マトリクスによる分析手法[19]も研究している．後者は，状態遷移表に類似した分析マトリクスに，例外条件によって発生する状態やイベントを記入しながら，例外状態の遷移を追跡して，障害シナリオを構築する．その際，IFD の中の DD に類似したダイアグラムも併用する．もちろん，PD に記述される情報も，例外状態の遷移を考え出すのに必要である．しかし，その情報はドキュメント化されず，技術者の頭脳の中に存在するのみである．そのため，熟練が要求される．しかし，PD を記述するよりは分析マトリクスを記述する方が記述量は少ないので，作業効率が良い．したがって，分析マトリクスを用いた手法は，熟練技術者のみの組織形態によって，分析を効率的に進めるのに向いている．以上に述べたように，IFD と分析マトリクスによるそれぞれの手法は，運用組織形態の違いによって使い分けできる．

次に，既存の DFD（Data Flow Diagram）と IFD の関係を考察する．IFD の中の PD には IDEF0 を適用しているが，IDEF0 は，DFD におけるプロセスの入力情報を，入力情報と制御情報とメカニズムの 3 種に分けている．さらに，そのメカニズムを，DD に表示した装置とその接続によって，詳細に示すことができる．これらの IFD の特徴は，組み込みソフトウェアの非正常系の分析に有効である．次は，UML（Unified Modeling Language）と IFD の関係を考察する．その関係は，UML と DFD の関係と似ており，最初は IFD を適用して組み込みソフトウェアの非正常系を分析し，その分析結果に基づきソフトウェアを開発するのに，UML を適用する．そのため，IFD は主に，非正常系の分析に必要な組み込みシステムのハードウェアとソフトウェアのアーキテクチャを表している．一方，UML は，ソフトウェア開発に必要な詳細情報を，クラス図やコラボレーション図等によって表している．

次に，ガイドワード適用について考察する．ガイドワードは，種々の具体的な例外条件を抽象化することによって得られた条件である．そのため，抽象化の対象に含まれていないまったく未知の例外条件を，ガイドワードのみによって発見するのは困難である．これは，ガイドワード適用の限界である．そこで，本論文に提案した手法においては，ガイドワードの適用を分析の開始点として位置づけ，FTA や FMEA のコンセプトと統合することによって，分析漏れの防止を図っている．なお，本論文に述べたガイドワードは，PD に適用するため，情報に関する例外条件を対象にしている．表 1 の例に示した「影響種別」の網羅性は重要であり，今後もガイドワードを洗練していく予定である．

今後の研究課題として，IFD による手法の本格的な適用実験を行い，その手法を確立して，分析マトリクスによる手法と統合を図る．その統合した手法によって，少数の組み込みソフトウェア熟練技術者と，多数の未熟練技術者を組み合わせる組織形態も含めて，経済的かつ高品質な非正常系要求分析の方法を研究する．また，両手法の統合にあたり，状態と制約条件の概念を用いた定性推論を適用して，手法を形式化する．その形式を用いて，熟練技術者の非正常系知識の形式を探る．さらに，その形式に従って知識ベースを開発し，分析支援の CASE ツールと連動を図る．ところで，IFD は有向グラフの性質を持つので，たとえば，断線等による情報フローの欠損や，電磁誘導等による新たな情報フローの発生，情報フローのループによる正負フィードバック制御等[5]のグラフ解析についても研究を継続する．

7. ま と め

本論文において，組み込みソフトウェア非正常系の要求分析に適用するため，IFD とそれを用いた要求分析手法を提案した．組み込みソフトウェアの特徴を反映して，IFD には組み込みシステムのほかに利用者や動作環境も含めるとともに，情報処理プロセスのほかに装置も含めた．分析手法には，トップダウンに分析する FTA，および，ボトムアップに分析する FMEA と HAZOP のそれぞれの考え方を適用して，分析漏れの防止を図った．実際の製品事例を用いた IFD の記述実験において，組み込みソフトウェア熟練技術者の指導の下，未熟練者が正常系も非正常系も順調に記述することができた．

今後の研究課題として，IFD を用いた分析手法の適用実験を行って手法を確立した後，分析マトリクスを用いた手法と統合を図る．その統合によって，開発組織形態も考慮した経済的かつ高品質な非正常系要求分析の方法を研究する．また，非正常系の知識データベースと，IFD のグラフ解析も研究する予定である．

謝辞 本研究の実験にご協力いただいた，橋本研究室の田辺寛朗君，谷本真樹君，井上富雄君，久保純哉君に感謝いたします．

参 考 文 献

1) 三瀬敏郎，新屋敷泰史，橋本正明，鵜林尚靖，片峯恵一，中谷多哉子：組込みソフト非正常系における仕様分析手法の一提案，ソフトウェア工学の基礎 XII，日本ソフトウェア科学会 FOSE2005，

pp.227–235 (2005).
2) 鷲見 毅，平山雅之，鵜林尚靖：組込みシステムにおける外部環境の分析，情報処理学会ソフトウェア工学研究会報告書，Vol.2004-SE-146(5), pp.33–40 (2003).
3) 経済産業省商務情報政策局（編）：2004 年版組込みソフトウェア産業実態調査報告書，経済産業省 (2004).
4) Crook, R., Ince, D., Lin, L. and Nuseibeh, B.: Security Requirements Engineering: When Anti-Requirements Hit the Fan, *Proc. IEEE Joint International Requirement Engineering Conference, RE'02*, pp.203–205 (2002).
5) 畑中久典，新屋敷泰史，三瀬敏郎，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, pp.19–24, 電子情報通信学会 (2004).
6) 新屋敷泰史，三瀬敏郎，江浦洋平，畑中久典，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，情報処理学会組込みソフトウェアシンポジウム ESS2004 論文集，pp.8–11 (2004).
7) 三瀬敏郎，新屋敷泰史，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仕様抽出のための非正常系分析マトリクス，情報処理学会組込みソフトウェアシンポジウム ESS2004 論文集，pp.12–19 (2004).
8) Mise, T., Shinyashiki, Y., Hashimoto, M., Ubayashi, N., Katamine, K. and Nakatani, T.: An Analysis Method with Failure Scenario Matrix for Specifying Unexpected Obstacles in Embedded Systems, *Proc. 12th ASIA-Pacific Software Engineering Conference*, pp.447–454 (2005).
9) Kametani, H., Shinyashiki, Y., Mise, T., Hashimoto, M., Ubayashi, N., Katamine, K. and Nakatani, T.: Information Flow Diagram and Analysis Method for Unexpected Obstacle Specification of Embedded Software, *Proc. Knowledge-Based Software Engineering JCKBSE 2006*, pp.115–124 (2006).
10) Leveson, N.G.: Fault Tree Analysis, *Safeware: System Safety and Computers*, pp.317–326, Addison-Wesley (1995).
11) Leveson, N.G.: Failure Modes and Effects Analysis, *Safeware: System Safety and Computers*, pp.341–344, Addison-Wesley (1995).
12) Leveson, N.G.: Hazards and Operability Analysis, *Safeware: System Safety and Computers*, pp.335–341, Addison-Wesley (1995).
13) Bernus, P., Mertins, K. and Schmidt, G. (Eds.): *Handbook on Architectures of Information Systems*, Springer-Verlag (1998).
14) Pentti, H. and Atte, H.: Failure Mode and Effects Analysis of Software-based Automation Systems, *STUK-YTO-TR 190*, p.35 (2002).
15) Dehlinger, J. and Lutz, R.R.: Software Fault Tree Analysis for Product Lines, *Proc. 8th IEEE International Symposium on High Assurance Systems Engineering*, pp.12–21 (2004).
16) Redmill, F., Chudleigh, M. and Catmur, J.: *System Safety: Hazop and Software Hazop*, John Wiley & Sons Ltd. (1999).
17) Alexander, I.: Misuse cases, use cases with hostile intent, *IEEE Trans. Softw. Eng.*, Vol.20, No.1, pp.22–33 (2003).
18) Lamsweerde, A.V. and Letier, E.: Handling Obstacles in Goal-Oriented Requirements Engineering, *IEEE Trans. Softw. Eng., Special Issue on Exception Handling*, Vol.26, No.10, pp.978–1005 (2000).
19) Mise, T., Shinyashiki, Y., Nakatani, T., Ubayashi, N., Katamine, K. and Hashimoto, M.: A Method for Extracting Unexpected Scenarios of Enbedded Systems, *Proc. Knowledge-Based Software Engineering JCKBSE 2006*, pp.41–50 (2006).

(平成 19 年 1 月 9 日受付)
(平成 19 年 6 月 5 日採録)

新屋敷泰史（正会員）

1973 年生．1998 年九州工業大学大学院情報工学研究科博士前期課程修了．修士（情報工学）．現在，松下電工（株）に勤務する傍ら，九州工業大学大学院情報工学研究科博士後期課程に在籍．組み込みソフトウェアの開発および研究に従事．

三瀬　敏朗（正会員）

1977 年同志社大学工学部電子工学科卒業．現在，松下電工（株）に勤務する傍ら，九州工業大学大学院情報工学研究科博士後期課程に在籍．組み込みソフトウェアの開発および品質向上の研究に従事．電子情報通信学会会員．

橋本 正明（正会員）

1970 年九州大学大学院工学研究科修士課程修了．日本電信電話公社（現 NTT）研究所を経て，現在九州工業大学大学院情報工学研究科教授．ソフトウェア工学やプロジェクトマネジメントの研究に従事．工学博士．プロジェクトマネジメント学会，電子情報通信学会，人工知能学会，ソフトウェア科学会，IEEE 各会員．

片峯 恵一（正会員）

1992 年九州工業大学情報工学部機械システム工学科卒業．1994 年同大学大学院情報工学研究科情報システム専攻修士課程修了．1994 年九州工業大学情報工学部助手．現在，同大学大学院情報工学研究科助教．情報システムの仕様化，仕様記述言語，ソフトウェア開発支援環境等の研究開発に従事．博士（情報工学）．電子情報通信学会，プロジェクトマネジメント学会各会員．

鵜林 尚靖（正会員）

1960 年生．1982 年広島大学理学部数学科卒業．1999 年東京大学大学院総合文化研究科広域科学専攻広域システム科学系博士課程修了．博士（学術）．1982〜2003 年（株）東芝に勤務．2002〜2003 年芝浦工業大学システム工学部非常勤講師．2003 年九州工業大学情報工学部助教授，現在に至る．2003 年度本学会山下記念研究賞受賞．ソフトウェア工学，プログラミング言語モデルに興味を持つ．ACM，IEEE–CS，日本ソフトウェア科学会，電子情報通信学会各会員．

中谷多哉子（正会員）

東京大学大学院総合文化研究科広域科学専攻博士課程修了．日本電子計算（株），富士ゼロックス情報システム（株）を経て 1995 年よりエス・ラグーンを起業．2006 年より筑波大学大学院ビジネス科学研究科准教授．現職．オブジェクト指向分析手法，要求獲得手法，オブジェクト進化に関する研究に従事．博士（学術）．電子情報通信学会，ソフトウェア科学会，プロジェクトマネジメント学会，IEEE–CS，ACM 各会員．