社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
KBSE2008-24 (2008-11)

組込みシステム非正常系分析のための IFD と分析マトリクスを統合した定式化

井上 富雄† 三瀬 敏朗‡ 新屋敷 泰史‡ 橋本 正明†
片峯 恵一† 鵜林 尚靖† 中谷 多哉子§

†九州工業大学 〒820-8502 福岡県飯塚市川津 680 番 4
‡パナソニック電工株式会社 〒571-8686 大阪府門真市門真 1048 番地
§筑波大学 〒305-8571 茨城県つくば市天王台 1-1-1
E-mail: †tomio@minnie.ai.kyutech.ac.jp, hasimoto@ai.kyutech.ac.jp

あらまし 組込みシステムは障害などの非正常系について，IFD (Information Flow Diagram)を用いた静的な側面と，分析マトリクスを用いた動的な側面から，捉える事が出来る．そこで，IFD と分析マトリクスの関係を厳密に示すため，両者の定式化を図り考察する．また，分析マトリクスにおける抽象化のメカニズムを定式化によって探り，両者を用いた分析を助ける．本稿では，両者を定式化した定義とその有効性について論ずる．

キーワード 組込みシステム，非正常系，情報フロー・ダイアグラム，分析マトリクス，定式化

Formalization for Integrating Information Flow Diagram and Analysis Matrix to Analyze Unexpected Obstacles in Embedded Systems

Tomio INOUE† Toshiro MISE‡ Yasufumi SHINYASHIKI‡ Masaaki HASHIMOTO†
Keiichi KATAMINE† Naoyasu UBAYASHI† Takako NAKATANI§

†Kyushu Institute of Technology 680-4 Kawadu, Iizuka-shi, Fukuoka, 820-8502 Japan
‡Panasonic Electric Works Co., Ltd. 1048 Kadoma, Kadoma-shi, Osaka, 571-8686 Japan
§University of Tsukuba 1-1-1 Tennodai, Tsukuba-shi, Ibaraki, 305-8571 Japan
E-mail: †tomio@minnie.ai.kyutech.ac.jp, hasimoto@ai.kyutech.ac.jp

Abstract We can treat embedded systems as two aspects about unexpected obstacles such as the failure. One is a static side using IFD(Information Flow Diagram). Another is a dynamic side using analysis matrix. Therefore I formalize IFD and the analysis matrix to show the relations between both exactly. In addition, I investigate mechanism of the abstraction in the analysis matrix by formalization and help with the analysis that I used both for. In this paper, I discuss it about a definition and the effectiveness that formalized both.

Keyword Embedded System, Unexpected Obstacles, Information Flow Diagram, Analysis Matrix, Formalization

1. はじめに

現在，我々の身の回りにはソフトウェアを組込んだ製品（以下，製品）があふれており，生活に欠かすことのできないものとなってきている．これらは使用にあたって特別なトレーニングを要するものではないため，ユーザが正規の使用法に依らず，製品を動作させることがある．また，製品は長期間にわたって使用されるため，軽微な故障が発生しても安全かつ安心して使用できることが期待されている．そのため，誤使用や経年劣化などを製品開発時に予期し，設計しておく必要がある．しかし，製品はそれ自体の機能的価値だけでなく，付加価値を加えた高機能化が現在では必要とされている．また，ユーザニーズの急速な変化に対応するべく，短期間で開発しなければならないといった現状がある．その結果，高機能化に伴う多くの複合的な事象を限られた時間内で想定する必要がある．

それにもかかわらず，複合的な事象を想定しうる経験豊富な技術者（以下，熟練技術者）は少ない．また，組込みソフトウェア産業に関わる人材も不足している．その結果，システム開発工程の上流段階における不具合の混入やそれに起因する手戻りが発生している[1]．

そこで，熟練技術者の指導の下，経験の浅い技術者（以下，未熟練技術者）とともに製品の複合的な事象を分析する手法である改ESIM（Enhanced Embedded

Systems Improving Method) [2]が提案された．この手法では，情報フロー・ダイアグラム（以下，IFD（Information Flow Diagram)）[3][4]と分析マトリクス[5]を併用して対象製品を分析する．両者は対象こそ同一であるが，その捉え方が異なるため，どのような関係にあるのか明確な定義を補う必要がある．

本研究の目的は，この両者に現れる概念を定式化し，どのような相互関係があるのかを論理記述を用いて裏付ける．また，その相互関係により得られた概念のつながりを改 ESIM 上に展開することによって，同手法の一助とする．さらに分析時の推論を定式化することによって，推論の自動化，ツールの開発に寄与する．

本稿の構成は，第 2 章において本稿と関係のある研究について整理し，研究の要件ならびに本稿の目的について述べる．第 3 章では IFD 及び分析マトリクスの定式化について述べ，第 4 章においてその結果を考察する．第 5 章において本稿をまとめる．

2. 研究の要件と関連研究

関連知識及び研究の要件について述べる．

2.1. 組込みシステムと非正常系

組込みシステムは，それ本来の機能を実現する正常仕様と例外を処理する異常仕様に分けることができる．異常仕様である例外処理の部分は，プログラム全体の約 7 割にもなる．これらの仕様はシステム開発工程のアーキテクチャ設計までに開発者によって分析され，決められる．先の研究より，このアーキテクチャ設計段階において判明している仕様を正常系，判明していないものを非正常系と呼んでいる．

正常系現象とはシステムの持つ本来の機能やその例外処理などの振る舞いを指し，ソフトウェア使用マニュアルに記載されているものである．非正常系現象とは例外処理が複合的に発生する想定外の振る舞いである．この非正常系現象が他に波及する一連のつながりを非正常系シナリオと呼んでいる．このうち障害として顕在化するものを障害シナリオと呼んでいる．

改 ESIM において，熟練技術者が分析マトリクスによって想定及び発見した非正常系現象を未熟練技術者が IFD 上に展開する．その現象から非正常系シナリオを発見し，その障害可能性を探る．この際，新たに非正常系現象が発見されることがある．これを繰り返すことによって設計品質を高めている．

2.2. 情報フロー・ダイアグラム

組込みシステムを静的にモデリングする手法としてIFDがある．IFDはデバイス・ダイアグラム（DD）とプロセス・ダイアグラム(PD)から構成されている．デバイス・ダイアグラムは，アーキテクチャ設計時に確定したシステム中のデバイスまたはオブジェクトとその間を流れるキャリアをモデリングする．プロセス・ダイアグラムは，アクティビティとその通信間の情報であるフロー情報をモデリングする．IFDの例を図 1 に示す．

IFD はデバイスやプロセスを障害原因とするシナリオの作成，可視化に用いられている．そのため，分析に必要な構成情報の多さと原因から障害までのフローの追いやすさから，未熟練者によって使用される．また，非正常系分析時に正常系の IFD が必要になるため，正常系のシナリオをそれ以前に記述しておかなければならない．

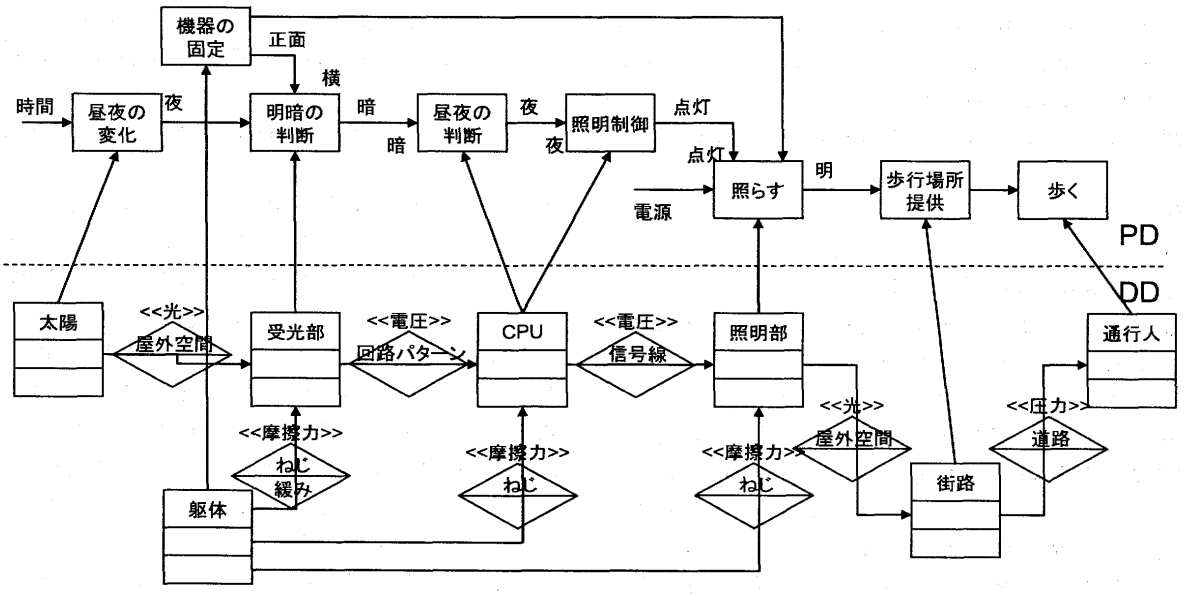

図 1. 情報フロー・ダイアグラムの例

状態 / イベント	ポットが傾いている.	温度上昇からの水量推定と水位センサが一致しない.	蓋近くまで水位がきているがマイコンは, 水位はそれほど高くないと判断している.
ヒーターが作動する.	温度上昇からの水量推定と水位センサが一致しない.		
ポットが傾く(水位が蓋近くまで来る).		蓋近くまで水位がきているがマイコンは, 水位はそれほど高くないと判断している.	
蓋が開く.			蓋から熱湯がこぼれる(可能性がある).

図 2. 分析マトリクスの例

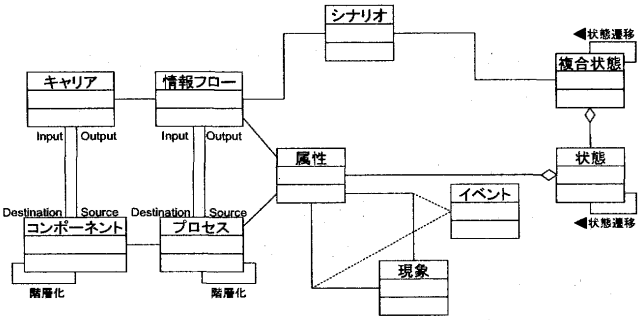

図 3. 概念モデル

2.3. 分析マトリクス

分析マトリクスは組込みシステムの非正常系を動的に分析するツールである. 表の各行の先頭にイベントを書き, 各列の始めに状態を記述する. 各状態においてイベントを受け取った時, システムがどのような振る舞いをするのか想定し, 現象をその交点に記述する. そしてこの現象が状態かイベントかを判断して表を拡張していく. 分析マトリクスの例を図 2に示す.

分析マトリクスは非正常系現象の想定及び発見に特化しているため, 構成情報の一部が省略されていることが多い. また, 経験的知識を要求される面もあり, 熟練技術者によって使用される. 分析の開始に当たって, 正常系シナリオを記述した IFD から状態, イベントを分析マトリクス上に転記する必要がある.

分析マトリクスにおいて, 3 つの抽象化が行われている. 1 つ目は, 構成要素の抽象化である. これは複数のデバイスを 1 つのコンポーネントとして取り扱うことである. 2 つ目は, 状態の抽象化である. これは状態を 1 つの複合状態として取り扱うことである. 3 つ目はフロー系列の抽象化である. 離れたデバイスとプロセスの間で小さな非正常系が積み重なることにより顕在化してくるものを表す際に出現する抽象化である. つまり, 非正常系現象の始点のデバイスから顕在化したプロセスまでの流れを省略し 1 つにまとめて取り扱うことである.

2.4. 概念モデル

組込みシステムが包含している概念をIFDと分析マトリクスから抽出し, その関係を示したものが概念モデルである[6]. この概念モデルでは, IFDからコンポーネントとキャリア, プロセス, 情報フロー, 属性をその要素として取り出している. また, 分析マトリクスからは状態と複合状態, イベントを取り出している. さらに, 両者共通の要素としてシナリオを持つ. 概念モデルを図 3に示す.

本研究ではこの概念モデルに沿って概念整理と論理的裏づけを与える. 要素間の関係についても, 同様に取り扱う.

2.5. 研究の要件

本研究の要件は, 改 ESIM において提案された分析の流れ及び概念モデルによって示された各要素とその関係を定式化することにある. また, 分析マトリクスにある抽象化のメカニズムについて探ることである.

本研究は, 非正常系の分析を構造と操作, 制約条件とに分けて定式化する. まず, 本稿は IFD 及び分析マトリクスに存在する概念を定式化し, 両者の相互関係を明確にする. これによって, 組込みシステム非正常系分析の構造が定義される.

3. 定式化

本研究において定式化した IFD 及び分析マトリクスついて述べる.

3.1. 情報フロー・ダイアグラムの定式化

まず, デバイス・ダイアグラムについて定式化する. デバイス及びキャリアは次のように定義することができる.

まず Device の集合, 略して D は,

$$D = \{ d_i \mid 1 \le i \le n \}$$

のことである.

そして Carrier の集合, 略して C は,

$$C = \{ (d_i, d_j) \mid d_i, d_j \in D\ (1 \le i, j \le n) \}$$

のことである. ここで,

1. D は, Device の集合

である.

最後に Device Diagram, 略して DD とは,

$$DD = (D, C)$$

のことである.

続いて, それぞれの属性を定義する. なお, デバイスが持つ故障パターンと呼ばれる情報もデバイスの属性の 1 つとして取り扱う.

Device d_i の属性の集合, 略して DA_i は,

$$DA_i = \{ DA_{ij} \mid DA_{ij}(d_i) = v_{ij},\ d_i \in D\ (1 \le i \le n),\ v_{ij} \in V_{ij}\ (1 \le j \le m) \}$$

のことである. ここで,

1. D は, Device の集合

2. V_{ij}は，値の集合
3. 属性DA_{ij}は，Dを変域とし，V_{ij}を値域とする関数

である．

Carrier c_iの属性の集合，略してCA_iは，

$$CA_i = \{ CA_{ij} \mid CA_{ij}(c_i)=v_{ij},\ c_i \in C\ (1 \le i \le n),\ v_{ij} \in V_{ij}\ (1 \le j \le m) \}$$

のことである．ここで，

1. Cは，Carrierの集合
2. V_{ij}は，値の集合
3. 属性CA_{ij}は，Cを変域とし，V_{ij}を値域とする関数

である．

次に，プロセス・ダイアグラムについて定式化する．プロセス及び情報フロー（Input Output Information Flow及び Control Information Flow）は次のように定義することができる．

まずProcessの集合，略してPは，

$$P = \{ p_i \mid 1 \le i \le n \}$$

のことである．

そしてInput Output Information Flowおよび Control Information Flowの集合，略してIOIFおよびCIFは，

$$IOIF = \{ (p_i, p_j) \mid p_i, p_j \in P\ (1 \le i, j \le n) \}$$

および

$$CIF = \{ (p_i, p_j) \mid p_i, p_j \in P\ (1 \le i, j \le n) \}$$

のことである．ここで，

1. Pは，Processの集合

である．

最後にProcess Diagram，略してPDとは，

$$PD = (P, IOIF, CIF)$$

のことである．

続いて，それぞれの属性を定義する．

Process p_iの属性の集合，略してPA_iは，

$$PA_i = \{ PA_{ij} \mid PA_{ij}(p_i)=v_{ij},\ p_i \in P\ (1 \le i \le n),\ v_{ij} \in V_{ij}\ (1 \le j \le m) \}$$

のことである．ここで，

1. Pは，Processの集合
2. V_{ij}は，値の集合
3. 属性PA_{ij}は，Pを変域とし，V_{ij}を値域とする関数

である．

Input Output Information Flow または Control Information Flowをi_iとする．そのとき，i_iの属性の集合，略してIA_iは，

$$IA_i = \{ IA_{ij} \mid IA_{ij}(i_i)=v_{ij},\ i_i \in IOIF(1 \le i \le n) \text{ または } i_i \in CIF\ (1 \le i \le n),\ v_{ij} \in V_{ij}\ (1 \le j \le m) \}$$

のことである．ここで，

1. IOIFは，Input Output Information Flowの集合
2. CIFは，Control Information Flowの集合
3. V_{ij}は，値の集合
4. 属性IA_{ij}は，IOIFまたはCIFを変域とし，V_{ij}を値域とする関数

である．

DeviceとProcessを関連付けるMechanismについて定義する．

$$PM = \{ (d_i, p_j) \mid d_i \in D\ (1 \le i \le m),\ p_j \in P\ (1 \le j \le n) \}$$

のことである．ここで，

1. Dは，Deviceの集合
2. Pは，Processの集合

である．また，

3. Device d_iは，Process p_jの Process Mechanism である

という．

他のDDとPDの関係づけるMechanismについてはスペースの関係上，ここで述べない．他のMechanismはPM同様DD，PDそれぞれ1つずつを2項組みとした定義で与えられる．また，Mechanismにも属性が定義でき，DDとPDの属性の変換を担っている．IFDはDD，PDおよびMechanismの三要素から構成されている．

IFD上では，状態及びイベントを明確に記述していないが，その概念を属性から形成することができる．状態をもつことができる箇所はデバイスとキャリア，プロセスである．ここで，情報フローは通信の内容だけを取り扱い，状態はプロセスが担うものとする．次に，IFD上における状態を定義する．

Device d_iの状態，略してDS_iは，

$$DS_i = DA_1 \times DA_2 \times \cdots \times DA_n$$ （×は直積を表す）

のことである．

Carrier c_iの状態，略してCS_iは，

$$CS_i = CA_1 \times CA_2 \times \cdots \times CA_n$$ （×は直積を表す）

のことである．

Process p_iの状態，略してPS_iは，

$$PS_i = PA_1 \times PA_2 \times \cdots \times PA_n$$ （×は直積を表す）

のことである．

Device Diagramの状態，略してDDSは，

$$DDS = DS_1 \times DS_2 \times \cdots \times DS_m \times CS_1 \times CS_2 \times \cdots \times CS_n$$ （×は直積を表す）

のことである．

Process Diagramの状態，略してPDSは，

$$PDS = PS_1 \times PS_2 \times \cdots \times PS_m$$ （×は直積を表す）

のことである．

イベントはキャリアまたは情報フローの属性値を受け取ったデバイスまたはプロセスが，それを解釈することによるものである．次に，IFD上におけるイベントを定義する．

Device d_q に作用するイベント，略して DE_q は，

$$DE_q = \{ v_{xi} \mid CA_{xi}(c_x)=v_{xi},\ c_x=(d_p,d_q),\ d_p,d_q \in D(1\le p,q\le l),\ c_x \in C(1\le x\le m),\ v_{xi} \in V_{xi}(1\le i\le n) \}$$

のことである．ここで，

1. D は，Device の集合
2. C は，Carrier の集合
3. V_{ij} は，値の集合
4. 属性 CA_{xi} は，C を変域とし，V_{ij} を値域とする関数

である．

Process p_q に作用するイベント，略して PE_q は，

$$PE_q = \{ v_{xi} \mid IA_{xi}(i_x)=v_{xi},\ i_x=(p_p,p_q),\ p_p,p_q, \in P(1\le p,q\le l),\ i_x \in IOIF(1\le x\le m) \text{または } i_x \in CIF(1\le x\le m),\ v_{xi} \in V_{xi}(1\le i\le n) \}$$

のことである．ここで，

1. P は，Process の集合
2. IOIF は，Input Output Information Flow の集合
3. CIF は，Control Information Flow の集合
4. V_{ij} は，値の集合
5. 属性 IA_{ij} は，IOIF または CIF を変域とし，V_{ij} を値域とする関数

である．

Device Diagram のイベント，略して DDE は，

$$DDE = DE_1 \times DE_2 \times \cdots \times DE_m \quad (\times \text{は直積を表す})$$

のことである．

Process Diagram のイベント，略して PDE は，

$$PDE = PE_1 \times PE_2 \times \cdots \times PE_m \quad (\times \text{は直積を表す})$$

のことである．

以上が IFD における概念の定式化である．

3.2. 分析マトリクスの定式化

分析マトリクスの定式化を開始する以前に抑えておかなければならないことがある．分析マトリクスにはシステムの構成要素の情報が記述されていない．その代わりに IFD から構成要素の情報を取得している．そのため，足りない概念は IFD の概念を用いて表すこととする．

分析マトリクスは IFD と違いシステムを動的な視点から捉えたものであるため，その変化を関数によって定義する．

ある属性の属性値を変化させる関数 CAF_i を定義する．

$$CAF_i = \{ F \mid F(v_{ij})=v_{ik},\ DA_{ij}(x_i)=v_{ij},\ x_i \in D\ (1\le i\le n) \text{ または } x_i \in C\ (1\le i\le n) \text{ または } x_i \in P\ (1\le i\le n) \text{ または } x_i \in IOIF\ (1\le i\le n) \text{ または } x_i \in CIF\ (1\le i\le n),\ v_{ij},\ v_{ik} \in V_{ij}\ (1\le j,k\le m) \}$$

ここで，

1. D は，Device の集合
2. C は，Carrier の集合
3. P は，Process の集合
4. IOIF は，Input Output Information Flow の集合
5. CIF は，Control Information Flow の集合
6. V_{ij} は，値の集合

である．

ある属性群から属性を抜き出す関数 SAF_i を定義する．

$$SAF_i = \{ F \mid F(x_i)=Ass,\ Ass\subseteq DA_i \text{ または } Ass\subseteq CA_i \text{ または } Ass\subseteq PA_i \text{ または } Ass\subseteq IA_i,\ x_i \in D\ (1\le i\le n) \text{ または } x_i \in C\ (1\le i\le n) \text{ または } x_i \in P\ (1\le i\le n) \text{ または } x_i \in IOIF\ (1\le i\le n) \text{ または } x_i \in CIF\ (1\le i\le n) \}$$

ここで，

1. D は，Device の集合
2. C は，Carrier の集合
3. P は，Process の集合
4. IOIF は，Input Output Information Flow の集合
5. CIF は，Control Information Flow の集合
6. DA_i は，Device の属性の集合
7. CA_i は，Carrier の属性の集合
8. PA_i は，Process の属性の集合
9. IA_i は，IOIF または CIF の属性の集合

である．

まず，最初にすべての現象及び状態，イベントを定義する．この際，現象は IFD のすべての属性における変化を意味するものとする．よって IFD の状態とイベントを用いて定義する．また，この現象のうちデバイスやプロセスの属性変化を状態とし，キャリアや情報フローの属性変化をイベントとして取り扱う．その後，分析マトリクス上に現れる現象や状態，イベントとして扱える形に構成する．

System 及び Sub system 上のすべての現象 SP は，

$$SP = DDS \cup PDS \cup DDE \cup PDE$$

である．

System 及び Sub system 上のすべての状態 SS は，

$$SS = DDS \cup PDS \cup SAF(DDS) \cup SAF(PDS)$$

である．

System 及び Sub system 上のすべての状態 SE は，

$$SE = DDE \cup PDE \cup CAF(DDS) \cup CAF(PDS)$$

である．

先に述べたように分析マトリクスにおいて 3 つの抽象化がなされている．上記の SP 及び SS，SE にその抽象化を適用することにより，分析マトリクスの現象や状態，イベントとして定義する．そのため，3 つの抽象化について定義する．

構成要素を抽象化する関数 ADF は，

$$ADF = \{ F \mid F(D)=Dss,\ Dss\subseteq D \}$$

によって定義される．ここで．

1. Dは，Deviceの集合

である．

状態を抽象化する関数ASFは，

$$ASF = \{ F \mid F(P)=Pss, Pss \subseteq P \}$$

によって定義される．ここで．

1. Pは，Processの集合

である．

フロー系列を抽象化する関数AFFは，

$$AFF = \{ (d_i, p_j) \mid (d_i, p_j) \not\in PM, d_i \in D\ (1 \le i \le m), p_j \in P\ (1 \le j \le n) \}$$

によって定義される．ここで．

1. Dは，Deviceの集合
2. Pは，Processの集合

である．

以上の概念を利用して分析マトリクスの現象及び状態，イベントを定義する．

Analysis Matrixの現象，略してAMPは，

$$AMP = ADF(SP) \cup ASF(SP) \cup AAF(SP)$$

である．

Analysis Matrixの状態，略してAMSは，

$$AMS = ADF(SS) \cup ASF(SS) \cup AAF(SS)$$

である．

Analysis Matrixのイベント，略してAMEは，

$$AME = ADF(SE) \cup ASF(SE) \cup AAF(SE)$$

である．

以上が分析マトリクスにおける概念の定式化である．

4. 考察

IFDと分析マトリクスの概念を関数と論理式によって表した．これにより，IFDと分析マトリクスのそれぞれの概念整理とその関係を明確にすることができた．特に分析マトリクス上における現象の概念は，IFDには明確に出現しておらず，その関係がよくつかめていなかった．しかし，現象及び状態，イベントすべてがIFD上の属性の変化を用いて定義できると示すことができた．これにより，熟練技術者によって使われてきた分析マトリクスを未熟練技術者にも使うことが可能になるための橋渡しの第一歩になった．

しかし，概念構造のみが明確になっただけで，分析を始める起点やその方法はまだ整理ができていない．分析を開始する際，IFDにおいてプロセスや情報フローに着目して非正常系を抽出する．この着目という操作は品質特性から導出可能であると仮定している．また着目した箇所から分析するための操作を考えなくてはならない．

さらに，本稿において定式化した概念が実システムに対して記述可能かどうか調査する必要がある．そして，概念モデルにおける各要素間の関係定義についてこの結果と整合性をとる必要もある．

今後の課題として，今回定義した概念を概念モデルと不整合のないように概念モデルについて考証を広げる必要がある．また，この概念構造における正常系及び非正常系の制約条件について整理し，分析操作も同じく定式化する必要がある．これらにより，未熟練技術者も熟練技術者のように分析マトリクスを用いて非正常系を分析することが可能になると考えられる．

5. おわりに

本稿では，IFD及び分析マトリクスの定式化を行い，その考察を行った．まだ，概念の定式化の段階であるため，未熟練技術者の手助けとなるものではないが，未熟練技術者のための非正常系分析に対して大きな進展となることを期待している．

今後は熟練技術者の非正常系分析時の思考法をこの概念と定性推論を使って定式化していく必要がある．

文　献

[1] 経済産業省 “2007年版組込みソフトウェア産業実態報告書”，経済産業省，2007

[2] Y.Shinyashiki, T.Mise, M.Hashimoto, K.Katamine, N.Ubayashi, and T.Nakatani. “Enhancing the ESIM(Embedded Systems Improving Method) by Combining Information Diagram with Analysis Matrix for Efficient Analysis of Unexpected Obstacles in Embedded Software” In The Proceedings of the 14th ASIA-Pacific Software Engineering Coference, pages 326-333, December 2007.

[3] H.Kamtani, Y.Shinyashiki, T.Mise, M.Hashimoto, N.Ubayashi, K.Katamine, T.Nakatani, ”Information Flow Diagram and Analysis Method for Unexpected Obstacle Specifcation of Embedded Software”, Proc of the Knowledge-Based Software Engineering (JCKBSE 2006), pp115-124, 2006

[4] 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”，情報処理学会誌，pp2894-2903，2007

[5] T.Mise, Y.Shinyashiki, T.Nakatani, N.Ubayashi, K.Katamine, and M.Hashimoto, “A Method for Extracting Unexpected Scenarios of Embedded Systems”, Proc of the Knowledge-Based Software Engineering (JCKBSE 2006), pp41-50, 2006

[6] K.Katamine, Y.Shinyashiki, T.Mise, M.Hashimoto, N.Ubayashi, and T.Nakatani. “A Conceptual Model for Analysis Method of Extracting Unexpected Obstacles of Embedded Systems”, Proc of the Knowledge-Based Software Engineering (JCKBSE 2008), pp22-31, 2008