2005－SE－147（11）
2005／3／18

組込みソフト非正常系における基礎モデル及び仕様分析手法の提案

三瀬　敏朗† 新屋敷　泰史†† 橋本　正明††† 鵜林　尚靖††† 片峯　恵一††† 中谷　多哉子††††

近年、商品組込み系ソフトウェア（以下、組込みソフトと呼ぶ）の大規模化と複雑化に伴った開発手法の整備が必要とされている。組込みソフトの要求分析工程での課題は、非正常系仕様の設定に関する方法論が不十分な事である。そのため我々は、組込みソフト分野における非正常系仕様の設定技術について研究している。我々はこれまでの研究で、非正常系の概念モデルと、状態及びイベントのマトリクスを用いた非正常系の分析手法を検討した。本稿では、これらの基礎となる非正常系の基礎モデルを提案し、基礎モデルの点からマトリクスを用いた非正常系の分析手法について検討する。

A Basic Model and Specification Analysis Method for Embedded Software Exceptions

Toshiro Mise† Yasufumi Shinyashiki†† Masaaki Hashimoto†††
Naoyasu Ubayashi††† Keiichi Katamine ††† Takako Nakatani††††

Abstract

Recently, embedded software needs the improvement of development methodology because of increasing scale and complexity. One of the important problems in requirement analysis of embedded software is lack of the methodology that handles exceptions. Therefore, we have been studying an exception handling methodology in embedded software. We have already proposed a conceptual model of exceptions and an analysis matrix. In this paper, we descibe a basic model of these and discuss the analysis matrix from the viewpoint of the model.

1　はじめに

各種の設備や機器に組み込まれて、制御、監視、通信、表示などを行う組込みシステムは、規模の増大化や複雑化が進み、優秀な組込みソフトウェア技術者が不足している状況にある。一方、このような組込みシステム開発において、現在のソフトウェア工学による支援が充分行えていないのが現状である。

組込みシステムは、社会インフラとして取り込まれ、情報処理システムとして意識されず、不特定多数により不特定環境の中で動作する必要がある。このため、組込みシステムはそれが置かれている環境や運用を制限することが困難であり、このような状況を受け入れながら、システムとしての品質を確保しなければならない。このため、顧客分析が重要とされる業務系のシステムに対し、組込みシステムでは、それが置かれる環境や運用に対する分析重要となる。

†松下電工システムソリューション株式会社／九州工業大学
Matsushita Electric WorksSystem Solutions Co.,Ltd / Kyusyu Institute of Technology
††松下電工株式会社／九州工業大学　Matsushita Electric Works, Co.,Ltd / Kyusyu Institute of Technology
†††九州工業大学 Kyusyu Institute of Technology

さらに組込みシステムでは、ハードウェア部品の品質・量が変動費として商品コストに直接影響を与える為、できる限りソフトウェアで仕様を実現する事が望まれる。このため、商品の品質確保がソフトウェアの品質に依存する部分が大きい。

このような状況下において、組込みシステムにおけるソフトウェアの開発の現場では、機能要件に関しては充分な検討が行われている一方、信頼性、耐久性、安全性、弊害抑止性については、ハードウェアとの関係や処理機能と密接な関連を持つため、充分な要求機能分析が行えていないのが現状である。結果としてこれらの要件に対する仕様が欠如することにより、評価工程に至るまでその欠落が見落とされたままとなる。たとえ設計・実装段階で欠落を発見し定義できたとしても開発計画の破綻や基本設計の崩壊が発生する。

組込みシステム開発における課題として仕様変更が多いことが従来から指摘されているが、その原因は商品競争による仕様変更以上にこの信頼性などの要件が欠落していたことによる仕様変更が多い。

近年ではアジャイル開発などの手法が提唱されているが、組込みシステム分野においては信頼性などの分析を形式的に行う手法を用いなければ真にシステムに要求される要件を抽出する事は困難である。

そのため我々は、組込みソフト分野における信頼性などの分析を形式的に行う事を目的として、非正常系仕様

の設定技術について研究している。これまでの研究において我々は、非正常系の概念モデルと、状態及びイベントのマトリクスを用いた非正常系の分析手法を検討した。

本稿では、これらの基礎となる非正常系の基礎モデルを提案し、基礎モデルの点からマトリクスを用いた非正常系の分析手法について検討する。

2 非正常系の定義

基礎モデルの提示に先立ち、ここで本研究における非正常系の概念について定義する。

まず議論の対象として、状態を持ち、イベントによって状態変化と出力を行う一般的なシステム（以下システム）を想定する。システムにおいて、状態及びイベントを、システム設計者が想定しているものと、想定していないものに区別する。前者の状態及びイベントを正常系、後者を非正常系と定義する。更に非正常系を、準正常系と異常系と定義する。準正常系とは一時的な非正常系であり、正常系へ移行する可能性があるもの、異常系はシステム単体では正常系へ移行する可能性がないものと定義する。

この定義によれば、システム開発の過程において当初想定していなかった状態やイベント、即ち非正常系も、考慮されシステムの機能として組み入れられることで、正常系として扱われる事を表している。図1に、システム開発の要求分析段階において、考慮から抜け落ち易いと思われるイベントを、非正常系として分類した例を示す。

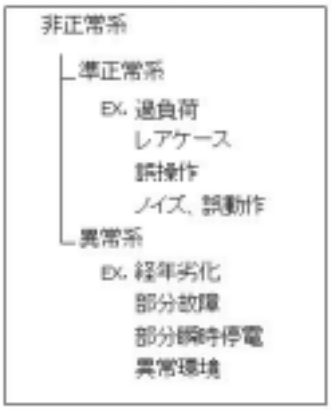

図1 非正常系の構成

当初考慮されていなかった状態やイベントを考慮する事で、システムの要件は増加するが、一般にそれはコスト要件の増加にもつながるため、最終的には考慮される状態やイベントであっても、システム自身に組み込めない非正常系が発生する。これがシステムにおける制約条件である。

制約条件が発生した時点でシステム自身に組み込まれた要件について、開発者はそれらの要件を満足させる事が求められる。一方、システム自身に組み込まれていない、この時点における非正常系に対しては、システムの外部に対する影響を最小化する事が求められる。制約条件の発生とその時点での要件について図2に示す。

図2 非正常系の変化

3. 組込み系の基礎モデル

組込みシステムにおける要件の分析手法を検討するために、組込みシステムのベースとなるモデルと、その特徴を明確にする必要がある。組込みシステムは、それが使用される環境の影響を受けると同時に、環境へ影響を与える。組込みシステムに対して影響を与える物としては、想定している利用者だけでなく、例えば処理する信号を乱すノイズなどもある。さらにノイズにはいくつもの種類があるが、これはノイズ源の特徴に依存する。よって、組込みシステムに影響を与えるノイズの特徴を定義するためには、そのノイズ源の定義と特徴の分析が必要となる。このような例を想定すると、組込みシステムにおける基礎モデルが備える特徴が明確になる。

まず、組込みシステム基礎モデルが記述する世界は、開発対象である組込みシステムとそれが置かれる環境内の主体をオブジェクトと見なしたオブジェクト群で構成される閉世界である。環境内の主体としては、組込みシステムに直接または間接的に影響を与える可能性があるもの全てのものを扱う必要がある。

次に、基礎モデルにおける各オブジェクトは状態を持ち、オブジェクト間で相互に通信(情報のやり取り)を行う。

組込みシステム自身、システムが直接対象とするオブジェクト、そのオブジェクトあるいはそのオブジェクトに影響を与えるオブジェクト、影響を与えるオブジェクトに影響

を与えるオブジェクトで構成され、それらは関係を持たないオブジェクトからは切り離され閉世界で構成される。オブジェクト間は、通信として定義する。各オブジェクトは、主体性を持ち、それぞれ状態をもつ。(図3)。

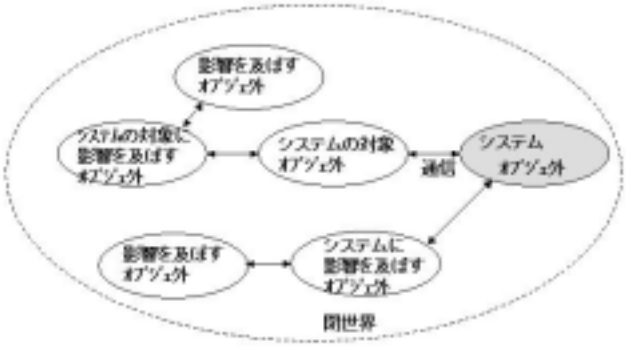

図3 非正常系の基本モデル

我々は、従来、非正常系の静的側面としては、それぞれの特性と関係を分析する必要があり、非正常系の概念モデルとして研究を行ってきた[2]。また、各オブジェクトの状態と通信であるイベントの時間的な動作に着目しそれらの組合せによる障害へのプロセスを中心として動的な側面として分析マトリクスを提案してきた[1]。すなわち、非正常系の特徴を研究していきながら、静的に分析した概念モデルを研究し、また動的に分析する分析マトリクスを提案してきた。本稿では、基礎モデルをベースとして、分析マトリクスを中心に一連の分析手法として提案する。

4. 非正常系分析手法の提案

前章で述べた基礎モデルとの関連を考慮して、非正常系分析マトリクスを用いた分析手法について再度検討を行う。検討事例としては、組込みソフトウェア管理者・技術者育成研究会(SESSAME: Society of Embedded Software Skill Acquisition for Managers and Engineers)が提供している組込みシステム教育教材である、話題沸騰ポットGOMA-1015型 要求仕様書を用いる。

本題材を要求者から開発当初に提示された要求仕様として扱い、要求仕様分析者の視点から非正常系の機能要件を抽出することを試みる。本検討における制約条件は明確でないが、年配者やポットについてあまり詳しくないユーザーを含む広範な利用者が使用するものと想定し、信頼性、耐久性、安全性、弊害抑止性における十分な配慮が要求され、一方で使用するハードウェアも自己異常検知機能などを持たない安価なハードウェアを用いて実現する事が要求されているものと仮定する。

[話題沸騰ポットの仕様抜粋]

図4にハードウェア構成を示す。

高温保温、節約保温、ミルク保温の3つのモードを持ち、保温設定ボタンで設定する。沸騰ボタンを押すとポット内の水を沸騰させ、カルキ抜きを行った後、保温になる。給湯ボタンにより、モータを制御し給湯する。蓋が空いているときは、ヒータは制御しない。

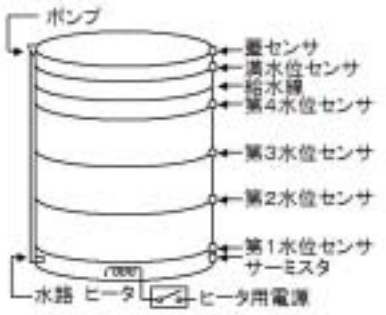

図4 話題沸騰ポットの構成

4.1 提案分析手法の概要

基礎モデルで述べたようにシステム仕様を分析するためには、環境を含むシステムを取り巻く世界の分析が必要となる。たとえば、ハードウェアからはインタフェース回路を通じてセンサの数値が読み取れるが、その値が大き過ぎるあるいは小さ過ぎるといった瞬時的な非正常系は読み出せるが、それが時間的な経緯を含めた信号波形として、その数値の傾向が正常であるかは環境の特性を含めて分析しなければならない。また、そのような信号がそのときのシステムの運用状態において適切かといった非正常系は運用状態を含めた分析を行わなければならない。したがってシステムの分析対象の閉世界の構成には、図5で示すように、環境、ハードウェア構成、運用が含まれる。

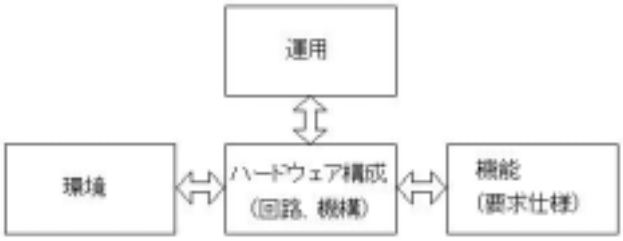

図5 分析すべきシステムの構成

以上を踏まえて、分析フローを図6に示す。

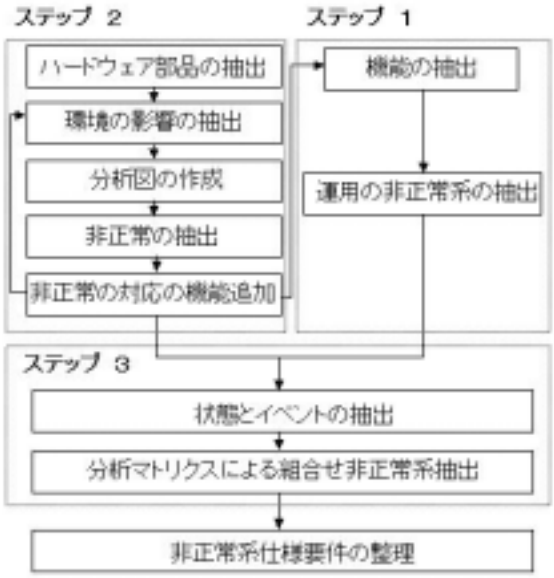

図6 分析フロー図

4．2 ステップ1（運用非正常分析）

システムに影響を与える、想定されていない動作の可能性を抽出する。動作の主体は開発対象システムの外部にあるオブジェクトである。本検討においてはポットのユーザーである人間を主な対象とする。この場合、不慣れ、イタズラ、悪意を持った運用によるシステムに対する影響を分析する必要がある。ここでもステップ2と同様、影響に対してポット自身で対応する要件が追加され、これらの要件に対する実現手段が検討される。

運用においては、一般的な5W1Hを考慮した分析によって検討を実施した。

Who： 不特定多数のユーザのため、高齢者や若年者、システムの熟練者や未熟練者などの特性の抽出

例 高齢者で操作する力が弱い、操作を忘れる
若年者でいたずらや届かない所に手が届く

When： 主体者の状態（時間やタイミングは、マトリクスで抽出）

例 手が濡れた状態で操作する

Why： 影響を与える人あるいは、他のシステムは、意図を持ってシステムに対して影響を及ぼす為、その意図の状態があり、それに対しての運用が行われるため、意図によりシステムの状態や動作が非正常系になりうる。

例 ポットをミルク保温に設定している場合、沸騰直後の熱湯を給湯することが危険になる場合がある。

What： 想定していない対象物あるいは、対象物の状態により影響を受ける。

Where： システムの設置や運用している場所により影響を受ける。また、施工のミスなどにも影響を受ける。

How： 想定していない手順や強さ、間違った操作

基本動作の抽出を行い、文の要素から非正常系を抽出する。図8に例を示す。

	Who	When	Why	What	Where	How
ポットを置く						ポットを傾ける ポットを倒す
水を入れる。		沸騰中に	お湯の増加量する	お湯 熱湯 水 塩水 入れ過ぎる		水を入れ忘れる 少しずつ入れる 一気に入れる 水を足す
コンセントを接続する	コンセントが抜ける（停電）	長時間 短時間				
給湯ボタンを押す。		沸騰中に 設定温度 沸騰中に	洗うために	水なしで 水抜き		給湯を止める 何度も押し続ける

図8 運用非正常系分析例

4．3 ステップ2（ハードウェア環境分析）

対象とするシステムの構成部品、それに影響を与える環境の主体の抽出、それらの状態の抽出を行い影響の特性を分析する。

1） 分析段階で抽出できるハードウェア（回路、機構）の部品をできる限り細かく抽出する。

2） 各構成部品が取り得る状態を定義する。このとき、非正常状態として、各ハードウェアの経年劣化や故障、

製造のバラツキ、施工時の不良などがありうる場合は状態として定義する必要がある。これら劣化や故障に対する検出機能をシステムの機能として追加する。但し、これらの機能は、実現可能であるか、あるいは機能が必要であるかは、実現の為のコストと品質レベルのバランス判断で省略されることになるが、最初の段階では出来るだけ抽出する。

3) 各部品の機能をベースとして関係する環境の抽出を行う。環境にはそれぞれ環境の主体、その環境の主体が取り得る状態を明記する。また、その環境の主体に影響を与え、環境の特徴を分析する為に必要な環境があれば同様に記述を行う。

本事例における、温度検出に関する部分分析例を図7に示す。サーミスタ(温度センサ)は、水を対象として温度を計測する。サーミスタの状態としては、サーミスタの検知能力(センサ周辺ハードウェアの劣化も含む)の劣化、異常な値を出力する故障、不安定な値を出力する故障、値が固定してしまう故障が考えられる。あらかじめハードウェアとして故障するパターンが決まっている場合は、これらの不要なものは省略できる。また、サーミスタを含む温度検出回路に電波ノイズなどが混入し、センサ値が一時的に誤った値を出力することが考えられる。これらの状態やイベントが考慮されることから、蓋の故障、ヒータ制御の故障、水位センサの故障、温度センサの故障を検出する要件の追加を行う。

この要件は、できるだけポット自身によって達成される事が望ましいので、開発者はここで、これらの新たな要件に対する実現手段を検討する事になる。実現手段が考案された場合には、これをポットに組込む事で、この要件は正常系となる。実現手段が考案されない場合、あるいはコスト要件などによって実現できないと判断された場合には、その要件が境界条件として扱われる事になる。この場合、ポット自身では解決できないため、ユーザー自身の注意などによって解決を図ることになる。本検討においては、蓋センサの故障は、蓋が空いている場合と閉まっている場合で保温状態での温度下降速度、ヒータ制御中での温度上昇速度が異なる事を利用すれば検出可能である、と開発者が判断したため、蓋センサの故障検知機能をポットの機能として追加した。同様に、ヒータ、水位センサ、サーミスタの故障についてもそれぞれ検知手法が考案されたため、それぞれをポットの機能として追加した。ただし、これらの実現手法による機能の達成が最適なものであるかについての判断は、分析終了時点まで保留される。

これらの故障検知機能の達成手段を実現するためには、サーミスタで検知される現在の温度以外に時間的な温度変化の情報を取得することが必要になる。そのことから今度は、温度変化に影響する要素を分析に加える必要がある。例としては水の量、水の質(酒などの水以外を含む)も分析に加える必要がある。また、環境として、大気の状態として温度、湿度、気圧も影響する。また、蓋の開閉、ヒータ制御も分析対象となる。

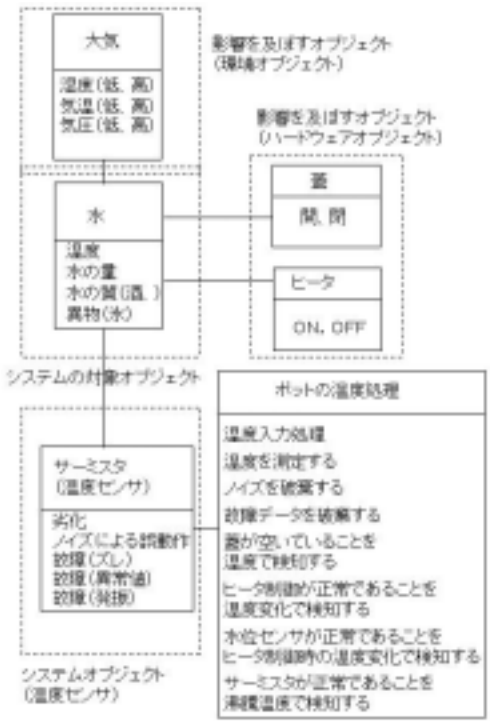

図7 ハードウェア、環境分析例

4.4 ステップ3(非正常分析マトリクス)

ステップ1とステップ2によりシステムを取り巻く環境と運用の主体と特性の抽出を行ってきた。これをベースに状態とイベントの抽出を行う必要がある。

なお、システムで想定されている、されていないに関わらず、本来システム化によって達成しようとしていた目的が達成できない状態を、ここでは障害と呼ぶ。

4.4.1 状態の抽出

本分析は、組込みシステムを含む関係する環境全体の分析を行っているため、状態は閉世界全体の状態を取り扱う。このため、組込みシステム外部の状態を含む。分析段階での組込みシステムから外の環境の状態は、設計段階では、組込みシステムの内部処理によりシステム外部の情報を取り込み処理するため、システム内部に取り込まれるが、状態として必ずしも表されない。また、分析段階では、非正常系の抽出を目的とするため、その視点から影響のない閉世界のオブジェクトの1つあるいは複数のも

つ状態を集合させ抽象化して扱うことにより分析の複雑化を防ぐ。設計段階では、システム内部の動作を記述するために、これら集合させ抽象化していた状態はそれぞれ分離され設計に組み込まれる。このため、分析段階での状態は、組込みシステムの設計段階での内部状態に引き継がれるものとそうでないものがある。

特に障害に至るプロセスに注目し、障害に近づく付近に焦点をあてた非正常系の状態の事例を示す。

・基本的な制約値の限界付近の状態

時間、位置、強度、温度、速さなど制約値近傍を非正常系状態として表し、制約の範囲内は集約して扱う。

・不安定な状態

制御直後の出力と実際の状態あるいは、制御結果の確認が取れるまでの状態不定状態、状態遷移中の過渡期の不安定状態、電源投入直後のハードウェアの不定状態

・未定義状態

全ての情報が入手できていない電源ON時の初期化処理中の状態

・過負荷の状態

CPUに対する負荷や通信の輻輳などのCPUリソースが限界に近い状態

・制御が正しくおこなえていない状態

制御出力の実際との不一致、制御に対する妨害の発生、実際の状態と監視との不一致

4．4．2 イベントの抽出

組込みシステムを含む関係する環境全体の分析を行っているため、イベントも状態と同様に閉世界全体の状態を取り扱う。

イベントの種類としては、下記がある。

運用イベント:

4．3ステップ1で抽出する運用の操作から発生する論理的な動作

擬似イベント:

前述で状態を定義したが、その状態が遷移する時のふるまいをマトリクスで評価するために擬似イベントを定義する。設計では、状態監視機能が、定義された範囲を超えた場合にイベントを発生させる。

タイムオーバイベント:

一定時間を測定し、状態を遷移するときに発生するイベント

本分析では、環境の状態も含めた世界の分析を行っているため、上記の擬似イベントが多く、状態の抽出で述べたように、システム分析段階として非正常系の状態に焦点を当て、非正常系に影響のないものは、マクロ的な複数の状態を組合せ定義し、逆に非正常系の分析に必要なものは、ミクロ的な細部の状態を定義しているのに連動し、複数の組み合わせた抽象的なイベントと詳細なイベントが存在する。

4．5 分析マトリクスによる分析

分析マトリクスによる分析手順については、[1]で提案を行った。分析マトリクスでは、動的な非正常の組合せを扱う為、4．4で抽出した状態とイベントをベースに分析を行う。状態とイベントの組合せにより新たな非正常系が検出できた場合は、新たな状態やイベントとしてマトリクスを増やしていく為、非正常系に特化した複雑な事象は、複合した状態とイベントの集合として抽象化した状態あるいはイベント名によって扱うことができ、ミクロ的な事象の分析が可能となる。図14に部分的な分析マトリクス例を示す。

図9 サーミスタ周辺の分析マトリクス例

分析マトリクスでは、システムに影響する環境全体を取り扱うため、分析が複雑になる。このため、できるだけ小さい単位で分析して組み合わせていくことにより分析を複雑にせず抜けのない分析が可能となる。

1) 情報の送受が出来るだけ少なくかつ小さな単位で基礎モデルで示したオブジェクトを分割する。

2) この単位で、ステップ1から3までを行う。この場合、分割した単位外から発生するイベントと分割単位を分析するために影響を受ける単位外の状態は、分析で扱う必要がある。分析中で分割範囲内の内部状態や内部イベントが発生するが、障害回避処理を組み込むことにより分割単位以外に影響を与えるイベントと状態は整理される。

3) 同様に、他の分割単位ごとの分析を行っていく。この結果として外部に影響を与える新たなイベントや状態が発生した場合は、それを加えて関係する分割単位の分析を繰り返す。

4) 全体として分割単位間に関わる非正常系の組合せを

検証するために、複数の分割単位の外部に与える状態とイベントを基に分析マトリクスを作成する。同様に階層的に繰り返し、全体の分析を行う。この流れを図11に示す。

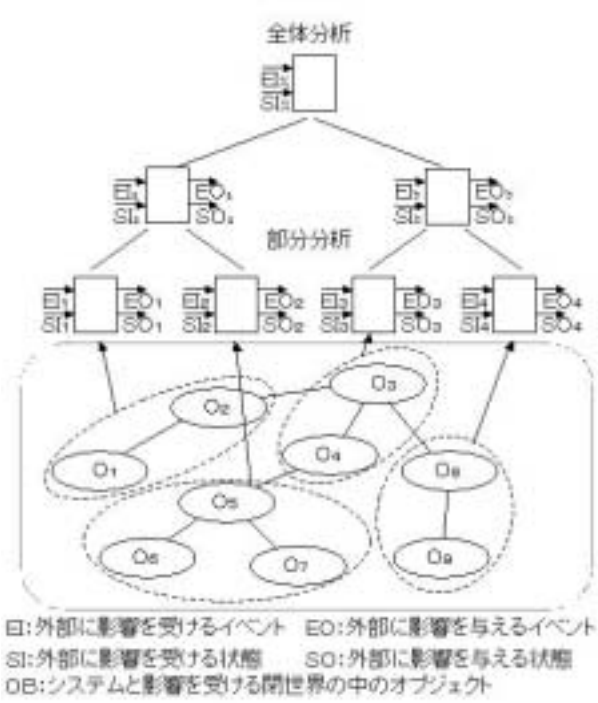

図10 部分分析から全体分析への展開

話題沸騰ポット事例の非正常処理として考えられるものとして下記が考えられる。

・下2つの水位センサOFFで上のセンサがONのとき、上のセンサを故障判断し、無効にする。それが満水センサの場合、蓋センサが正常で、蓋が閉められている場合のみ、温度上昇で満水を検知し停止する。

・上2つのセンサがONで下のセンサがOFFのとき、下のセンサを故障判断し無効にする。それが最低水位センサの場合、ヒータ制御時の温度上昇で空を判断しヒータを停止する。

・水面が揺れている場合があるため、水位の判断は、充分時間を取った平均値で処理をする。

・沸騰時の温度が通常気圧の沸騰温度から大幅にずれた場合、サーミスタ異常として保温能力がないと判断し停止する。

・蓋センサ故障で蓋が空いた状態の場合、ヒータON時放熱が気温と水温の差で放熱することから高温時の温度上昇の変化により蓋空き検出する（実験が必要）。ただし、途中水温の変化（水追加時など）の場合には、判断できない。

・保温のメモリの値が無効値の場合、デフォルトとして、高温保温に再設定する。

・ミルク保温設定で、沸騰後の冷却ができていない間の給湯は、安全のためブザーで通知する。

・一定時間以上ボタンを押さないと判断しない（ノイズ防止）。

・一定時間以上ボタンを離さないとボタンを押した判断をしない（チャタリング防止）。

・2つのボタンの同時押しは判断しない（操作ミス防止）。

・長時間ボタンが押され続けている場合は、ボタンを故障と判断し、以後OFFになるまでボタンを無効とし、上記2つのボタン同時押しの判断から切り離す。

5．考察

5．1 分析のレビューに向いた表記形式

一般的に非正常系仕様は、設計段階まで明確にならない場合が多く、その仕様が必要かつ充分であるかどうかのレビューが行いにくい。非正常分析マトリクスでは、状態とイベントの項目のレビューにより、経験的検知からの運用場面検討の抜け漏れが発見しやすい。

5．2 分析と設計の分離

非正常系の詳細な検討は、設計段階で具体的なコンポーネントやソフトウェアの内部構造が決まらないと分析は行えない。しかし、非正常系をどのように扱うのかは要求要件であり、設計やり直しをなくす為に分析段階で抽出しておきたい。組込み系システムが分析と設計を明確に分けにくいため、仕様変更が多いなどの組込み系に特徴的な問題の原因ともなっている。非正常系の仕様分析の経験と過去の事例データを蓄積することにより、非正常系分析マトリクス中に抽象的な非正常を定義することが可能となってくる。これにより、分析段階では、抽象的な非正常系だけを扱い、非正常系に対する運用処理を検討する。設計段階では、経験により、抽象化された非正常系の範囲に入ることが明確であるため、設計段階でハードウェアの特性やソフトウェア内部構造を基に抽象化された非正常系に合わせ込むことができる。これにより、分析段階と設計段階の分離が可能となり、組込み特有の課題が軽減出来る。

6．まとめと今後の課題

本稿では、非正常分析マトリクスを中心としながら、それに至るためのシステム外部環境の定義から非正常系の

抽出までの検討を行った。基礎モデルをベースとしてシステムに影響を与えるオブジェクトの主体とその状態を明確に定義しながら、非正常系の抽出を行うことにより、抜けのない分析が可能になる。また、分析を分割することにより、非正常系を扱うことによる分析の非現実的な複雑化に至らないことも確認できた。

今後の課題として下記2つを考えていきたい。

1) 分析マトリクスによる分析支援ツールの開発

分析手法として事例を試行してきたが、第三者が容易に適用できること、レベルが保てることを考えた場合早期に知識データベースとによる支援等を考えていく必要がある。また、一般的に使えるようにするために、UMLツールとの連携も考える。

2) 分析段階では、環境を含めた系を分析するが、設計段階では組込みシステム内部を分析する。基本的な設計の考え方は継承できると考えるが、分析から設計に効率よく引き継げる為のしくみを考える。

参 考 文 献

[1]三瀬敏朗, 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”, 情報処理学会ソフトウェア工学研究会研究報告 No.145-16,2004

[2]新屋敷泰史, 三瀬敏朗, 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”, 情報処理学会ソフトウェア工学研究会研究報告 No.145-15, 2004

[3] 組込みソフトウェア管理者・技術者育成研究会（SESSAME）セミナー資料話題沸騰ポット要求仕様書

[4] 経済産業省 商務情報政策局：2004年版組込みソフトウェア産業実態調査

[5] S.シュレィアー, S.Jメラー著本位田真一, 伊藤潔監訳：続オブジェクト指向システム分析 啓学出版 1992

[6] 鈴木順二郎, 牧野鉄治, 石坂茂樹：FMEA,FTA実施法 日科技連出版社 1990

[7] 北川賢司：FMEA/FTAの導入法 総合技術センター 1984

[8] 日本技術士会：リスク分析工学 丸善 2004

[9] 渡辺政彦：拡張階層化状態遷移表設計手法Ver2.0 Embedded SEのための設計手法 キャッツ 1998

[10] Karl E.Wiegers Software Requirement Second edition

[11]渡辺政彦ほか：UML 動的モデルによる組込み開発 オーム社 2003

[12]渡辺博之ほか：組込み UML e UML によるオブジェクト指向組込みシステム開発 翔泳社 2002

[13]中谷多哉子ほか：ウィンターワークショップin神戸報告 情報処理学会ソフトウェア工学研究報告 No142-2003

[14]細川卓誠ほか：組込みソフトウェア開発におけるユースケース分析・設計方式の提案 情報処理学会ソフトウェア工学研究報告 No140-2003

[15]青木利晃：組込みシステムの動向 情報処理学会ソフトウェア工学研究報告 No137-2002

[16] F.Kasati and G.Cugola, “Error Handling in Process Support Systems”, Advance in Exception Handling Techniques, LNCS 2002, pp.251-270, 1998