社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－SE－145　(16)
2004／8／20

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

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****

*　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　　**　松下電工株式会社／九州工業大学
***　九州工業大学　　****　九州工業大学／エス・ラグーン

組込みソフト分野では、専任オペレータや安定した環境とは限らない状況での運用を要求される。組込みシステムの信頼性や安全性を確保するために、通常運用で想定から抜け落ちしやすい状況である非正常系に焦点を絞り、仕様分析段階で非正常系要件の抽出を行い、システムの設計要件を明確にする方法について検討を行った。このため、非正常系の分析や要因等の体系化を行い、システムにおける非正常系の連鎖から障害に至る挙動の抽出を行うために、状態とイベントのマトリクスを用いて分析を行う方法を検討し試行した。その結果、基本的な有効性を確認し、今後の課題を検討した。

Exception Analysis Matrix for Embedded Systems Software Specification

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

*　Matsushita Electric WorksSystem Solutions Co.,Ltd / Kyusyu Institute of Technology
**　Matsushita Electric Works, Co.,Ltd / Kyusyu Institute of Technology
***　Kyusyu Institute of Technology　　****　Kyusyu Institute of Technology / S-Lagoon

Embedded systems are in need of the consideration about exceptions because of the environment and requirements for running. However, in practice, the consideration about exceptions often slipped out of specifications. Thus, it is an important problem about the cost and quality of embedded systems.
Therefore, we have been studying about the methodology to analyze exceptions with exception knowledge systematization and exception state/event analysis matrix.
In this paper, we describe the analysis matrix, its application example, and future studies.

１．　はじめに

近年マイクロコンピュータを応用した組込みソフトウェアが身の回りの様々な商品に応用されている。同時に、特にインターネット対応家電製品、ＬＡＮに対応した車載システムや設備コントロールシステムが増加するなど、組込みソフトウェアの大規模化、複雑化が進んでいる。

経済産業省２００４年版組込みソフトウェア産業実態調査報告書において、要求仕様の問題と品質の問題が大きく取り上げられているが、組込みソフトウェアの８０%は、エラー処理に対する機能と言われており、組込みシステムの運用環境に対して、通常ではない運用状態への配慮が必要であり、仕様設定時に想定から抜け落ちやすい。

到来するユビキタス社会において、組込みソフトウェアは生活のありとあらゆる場所で動作し、色々なシステムが有機的に結合してくる事が予想される。このような状況においては、システムの一部に障害が発生した場合、その影響の内容と範囲が想定できない可能性がある。

このように、信頼性や安全性への要求が高まる中、抜け洩れのない仕様を出せるしくみの確立が早急に必要となる。以上のような背景から、我々は、組込みソフトウェアの特徴である想定されない状況（以降、非正常系と呼ぶ）における要件定義を、システム分析段階で抽出し、仕様に盛り込みが行えることを目的としている。

システム分析段階で非正常系仕様を配慮する効果は次のとおりである。

－　仕様分析段階で非正常系仕様を盛り込むことで理想的なアーキテクチャ設計が可能となり、システム評価段階で重要な非正常系の品質要件が発見されるなどの設計手戻りによるソフトウェア構造の劣化を防ぐことができる。
－　重要な障害可能性要件に対し、その対応がソフトウェアで行えなえずハードウェアの仕様変更を伴い、開発が大きく遅延する場合があるため、早期の分析が必要である。
－　非正常系に関するソフトウェア機能要件が大きいた

め、非正常系を網羅しておくことにより、開発工数を正確に見積もることができる。

－　非正常系は試験が困難であり、設計段階から試験の項目抽出と試験の方法を明確にし、試験を可能にするためのツールの準備や擬似試験を行う為の機能を設計仕様に組み込む必要がある。

現状、非正常系の分析は、ＳＥの個人ノウハウに依存し、設計レビューが行いにくく属人化しているため、過去に発生した類似の非正常系検討不足によるトラブルが発生している。優秀なＳＥは分析や設計を複雑にせず、未経験なシステムであっても障害に対して強い設計を行える。これは、経験をうまく水平展開し、過去の障害ノウハウと自分の分析手順を組み合わせて非正常系に配慮しているためである。今後、品質を確保できる組込みソフトウェア要員を育成していくためにも、非正常系を体系的に考慮し、分析が非実用的な複雑さにならないようにし、ＳＥに依存している品質組込み技術を工学的なアプローチに変換することが必要である。我々は、組込みソフトウェアの非正常系仕様抽出を研究し、本稿では、２章に非正常系の要件と特徴、３章に非正常系の分析段階での抽出手法、４章で事例検証を行い、５章で今後の研究課題を明確にした。

２　非正常系の要件と特徴

２．１　組込みシステムでの非正常系の要件

組込みシステムが、特に仕様や品質の課題を重要視される原因は、下記の運用環境と期待される運用要件にある。

－　利用者がコンピュータ専門家でないため、再起動が絶対許されない。また、誤操作、イタズラなどの状況を把握し、対応する必要がある。

－　設備系システムは２４時間稼動を要求され、部分障害が発生しても可能な限り機能を有効にする必要がある。

－　設備系システムは、制御の振る舞いにより人命に影響が出る場合があり、徹底した安全性が要求される。

－　ハードウェアコストをより低価格に押さえる為、ハードウェアの性能や信頼性を最低限に抑え、ソフトウェアでそれをカバーし、商品の運用に対応した性能や信頼性の確保が必要となる。

これらから、組込み系では仕様要件が、顧客からの情報に付け加え、システムのおかれている環境と運用される具体的な状況、さらに機構、ハードウェア、施工などが総合的に含まれており、分析を困難としている。

要求要件は、機能要件と非機能要件からなり、機能要件は、要求者が想定している運用状況に直接必要な要件である。一方、非機能要件は、性能や信頼性など機能要件を確実にかつ安定的に持続させる為に必要な間接的な要件である。組込み系では、非正常系への配慮が特に重要であり、この場合の非機能要件が非正常系要件となる。そのような要件は、過去の実施経験を持たずに完全な分析を行うことは困難であるが、過去の現象や経験を水平展開し、適切な分析手法へ取り込むことにより、たとえ未経験のシステムであっても事前に非正常系を予測でき、品質を高めることが可能となる。

また、非正常系として大きく二つの状況を定義する。ひとつは、一時的な好ましくない状況で、過負荷や一時的な輻輳など時間経過などにより回復しうるものであり準正常系と呼ぶ。もう一つは、ハードウェアの故障など、回復不可能な好ましくない状況からなるもので異常系と呼ぶ。回復が可能か不可能かにより対応要件も大きく変わる為、意識して扱う必要がある。また性能の劣化もそれ自体が好ましくない動作であり、それのためにあらたな非正常な状態を発生させる要因となるため、非正常系として扱う。

非正常系の要件は、ソフトウェア品質要件のＩＳＯ９１２６分類として、機能性のひとつである安全性、信頼性の分類中の誤り許容性と可用性、効率性の一部の要件に該当し要求された信頼性要件（機能維持要件）とパフォーマンス要件（性能維持要件）を満たす為に、先に述べた組込みソフトウェアの特徴である免責事項を除くすべての環境や使用状態のなかで最大限の上記品質項目を保持することにある。

２．２　非正常系の要因

非正常系を引き起こす要因は、外部あるいは内部の発生事象であり、非正常系の要因としては、直接のシステムだけでなく、システムを取り巻く環境も考える必要がる。その要因の対応を的確に行うことにより非正常系の状態にならないか、あるいはその影響を最小に食い止めることができる。しかし、避けることができず発生した非正常状態は、それが要因となり新たな非正常状態を引き起こすため、非正常状態の発生は、非正常要因となりうる。これらの挙動を分析することにより、非正常系の特性が把握でき、それによる障害影響を最小にすることができる。そのためには、まず外部に起因する非正常系の一次要因の抽出が必要である。一例として下記項目がある。

環境　　水、油、雨、水滴・・
　　高温、低音、急激な温度変化
　　明るい、暗い、日光、熱線、反射・・
　　振動、騒音、声、・・
　　遮蔽物、穴、壁・・・
　　煙、蒸気、
　　高周波電波などのノイズ
　　坂、重力・・・
機構　　故障、外れ、割れ・・・・
　　磨り減りなどの劣化、汚れ・・
　　誤動作、・・
　　位置などのタイミング・・
　　スリップ、クラッシュ、オーバーラン・・
　　挟まり、負荷
　　過負荷、軽負荷、逆負荷
　　整備不良（プリンタ紙切れ・・・）・・・

回路　　回路部品故障（メモリ、ＬＳＩ・・）・・
　　回路部品劣化、製造ロットバラツキ
　　配線短絡、切断、接触不良、コネクタ外れ・
　　ノイズや振動による誤動作
　　リセット
　　停電、部分停電、電圧不安定、瞬停・・・
施工　　異機種や異バージョン接続
　　重複アドレス
　　取り付け不良
　　誤結線、構成設定ミス
運用　　誤操作、イタズラ
　　中途半端の操作放置
　　装着ミス、使い方不良、手順ミス
　　誤カード挿入
　　誤運用、目的外運用
処理　　ソフトウェアデッドロックなど機能の停止
　　処理の問題によるパフォーマンスの劣化
　　データの破壊
　　リソースがすべて使用中

２．３　非正常系のソフトウェア処理視点の要因

過去の経験がない新たなシステムの非正常系要因を抽出するためには、抽象化した視点からの分析が必要であり、２．２の非正常系要因から発生する現象を、ソフトウェア処理視点から見た非正常系の要因として下記に抽出した。

・情報の意味　大きくずれた値／小さくずれた値
　　状況から矛盾した値
・情報の量　　多すぎる／少なすぎる
　　長すぎる／短い
・情報の時間　タイミングが早い／遅い
　　同期している／同期がはずれている
　　長すぎる／短すぎる
・情報の構成　書式に一致しない
　　未定義のコード
　　時間的順序構成が一致しない
・情報の対象　本来の対象ではない対象からの情報
・情報の状態　喪失・不定
　　固定している
　　不安定
　　発振

２．４　非正常系の連鎖

非正常系は、一次要因から連鎖的に非正常状態が生じる状況を考える必要がある。具体的な事例を示す。

・　設備の状況を常時診断し自動でプリンタに印字する装置で、長期間紙切れを放置し、印字すべきデータが蓄積され、使用メモリが飽和しシステム動作がすべて停止した。
・　通信で接続された多くの端末を定期的にポーリングすることにより状態を監視し、情報を通知する装置で、一部の配線が断線したため、応答のない複数端末に対するエラー処理に通信時間が常時余分に発生し、接続している端末に対して、ポーリング最大遅延時間をオーバーし、情報抜けが発生した。
・　複数のＣＰＵが連動するシステムにおいて、瞬時停電が発生し、瞬時停電を検出し停電リセットしたＣＰＵと、電源容量範囲内で瞬時停電しなかったＣＰＵが発生し、それらのＣＰＵ間の連動情報が不一致となり、連結した動作で誤動作が発生した。

以上の事例で、何らかの外的非正常イベントが発生し、正常、非正常なイベントと状態が関係しあって障害にいたっている。これらを整理すると、通常の状態で、非正常イベント発生時に非正常事象が生じる場合、非正常状態で、通常のイベント発生時に非正常事象が生じる場合、非正常状態で、非正常イベント発生時に非正常事象が生じる３つの場合があり、さらにその非正常事象は別の非正常現象の要因となる場合がある。このことから、非正常系の分析には、正常なイベント、正常な状態、非正常なイベント、非正常な状態を組み合わせて分析する必要がある。たとえば、携帯電話が鳴ること、車を運転していることはそれぞれ問題ないが、前の車が急停車するという非正常状態で、運転中で携帯電話が鳴ったときに、気をそらしてしまい、対応が遅れ事故が発生する。これを運転前に予測することにより、運転中は、携帯電話の音を鳴らさないという対応ができる。

２．５　タイミングに関する非正常系の要因

非正常分析としてタイミングによる問題も重要な課題としてとして取り上げなければならない。タイミングに関しては、障害要因を分類していき、競合や衝突、リソース不足、状態等の遷移中や処理中による不安定期間に大きく分類できた。すべてのタイミングを網羅的に分析するのは膨大な検討が発生するため分析段階では現実的ではない。上記３つを静的な状態として捉え、非正常状態として状態の一つに定義することにより、タイミングに関する詳細分析を省くことができるとして検討した。

３　非正常系の仕様分析段階での抽出方法

３．１非正常系の抽出手法の要件

システムは、避けられない非正常系要因を基点として連鎖的に非正常系が発生し、障害に至る。これらは常に発生するものとは限らず、たまたまあるタイミングで非正常系が発生した場合のみ障害が生じる場合や、ある非正常系の発生中に別の非正常系が発生した場合に重大な障害が起こるなどが考えられ、特に複合的な非正常系に対する検討に抜け落ちがあり、重要な障害が発生する要因となってい

る。非正常系への対応は、一次的な非正常系で対応できれば連鎖を考える必要がないが、連鎖した結果でしか対応できない場合も多い。
事例：
煙感知器による火災判断の場合、火災の煙以外にタバコの煙による誤作動の可能性がある。煙センサからの入力としてはフィルタリングできないが、アプリケーションレベルでは、たとえば、部屋の人感センサが煙発生以前から点灯している、部屋の電気鍵が開けられている、部屋の照明が点灯している、火災の熱検知器は作動していないことから煙による火災検知の誤報をフィルタリングできる。
このため、組み込みシステムの非正常系として、連鎖していく非正常系障害が発生するような複合事象を扱える設計段階での分析手法が必要となる。

３．２非正常系に対する従来技術

要求仕様定義段階で非正常系である障害可能性の予測分析を行う手法として従来から、ＦＴＡやＦＭＥＡ手法が用いられている。

ＦＴＡ：(Fault Tree Analysis)
最終障害系を定義し、原因系をツリー状に分析していく手法。

ＦＭＥＡ：(Failure Model and Effect Analysis)
すべての部品をベースにそれらの故障モードを調べ、それらがどのように影響を及ぼしていくのかボトムアップ的に連鎖影響を分析していく手法。

しかしこれらの手法による連鎖の抽出については、分析者のスキルに依存する。ＦＴＡでは、重要な障害結果を軸に分析を行うため、障害が当初想定つかないものは障害予測できない。また、ＦＭＥＡでは、個別の障害から連鎖していく分析を行うが、どのような連鎖をあるかについて分析者のノウハウに依存し、支援が行えない。

３．３　非正常系抽出手法の検討

非正常系は、連鎖した非正常系の分析をしていく必要があり、複合的な要因を抽出していく必要がある。状態遷移表では、状態とイベントによりソフトウェアの挙動を設計していくことができ、システムの動きを網羅的にとらえることが出来る。非正常系の抽出においてもイベントと状態のマトリクスによる分析が可能であると考えた。
最初に仕様分析段階で抽出できる正常イベントと正常状態を抽出する。非正常系は、外部からの何らかのイベントにより生じるため、外部の非正常要因を検討することにより、一次的な非正常イベントを抽出する。これにより、状態とイベント（正常、一次非正常）の表を作成する。この表のマトリクス部分に注目していくと連鎖的に発生する非正常系が抽出できる。この非正常系は、回避処理により影響を受けないようにできる場合とできない場合があるが、できない場合は、二次的非正常系の状態やイベントとしてあらたに発生させる。これをマトリクス表の状態やイベントの項目に追加し表を展開する。これを続けていった場合、ある障害可能性対し、回避処理により防ぐことができるか、障害を受け入れざるを得ない状態になり、これ以上新たな非正常系が発生しなくなる時点がくる。すべてのマトリクスの交点から新たな非正常系が抽出できなくなるまで行う。これにより、非正常系の抽出とその対応を分析することができる。状態遷移表が、ソフトウェア設計を目的としているのに対し、この状態とイベントの分析マトリクスは、仕様分析段階で、非正常系の分析に用いるため、下記のように留意点を検討した。

・　設計段階ではなく、分析段階での非正常系抽出を目的とし、状態遷移表のような状態とイベントの網羅性を追及しない。
・　非正常系抽出を目的とするため、状態やイベントは、ソフトウェアの挙動だけでなく、非正常系の要因となりうる環境・機構や構造・回路・処理を含めて扱う必要がある。
・　この表のマトリクスの升目には、処理の記述は行わず、非正常系要件抽出の整理を目的とした記述を行う。該当記述内容がない場合は記述を行わない。

３．４　非正常系抽出手法による記述

図１に非正常マトリクスの作成手順の概要を示す。
①　一次非正常イベントの作成
システムの構成要件と機能、運用状態を分析することにより、正常イベントの抽出を行った後、一次非正常イベントの抽出を行う。外部からの一次非正常系イベントの抽出に対して、非正常系の要因から抽出する方法と、ＣＰＵから見た非正常系の項目と照らし合わせて抽出する方法を検討した。非正常系の要因としては、２．２で述べた非正常系の要因である環境、機構、回路、施工、運用、処理といった要因リストとシステム構成のマトリクスから一次非正常イベントを抽出する。また、ソフトウェア処理からの要因項目としても、同様にシステム構成要素と要因とのマトリクスから抽出する。
②初期の非正常マトリクスの作成
①で得られた正常イベント、一時非正常イベントと正常状態項目から初期の非正常マトリクス表を作成する。
③新たな非正常系の抽出
各イベントと状態の交点での挙動を検討し、新たな非正常系の発生可能性を検討する。新たな非正常系が発生しないものについては、網掛け等により検討済みのマークを行い、処理は記述しない。
④抽出した非正常系の分析
非正常系が発生する可能性がある場合、その検出方法と回避方法の検討を行う。そのために新たな処理機能を追加する必要があり、その処理によりあらたな状態が発生する場合がある。また、発生する可能性のある非正常系を回避できない場合、新たな非正常系状態と非正常系イベントが

発生する。状態とイベントの交点には、下記を記入する。
－　発生する非正常系。
－　非正常系のすべてあるいは一部の回避が可能であり、処理することが妥当な場合には、その検出方法。
－　検出した非正常系の回避方法

⑤非正常系イベントと状態の展開

④で抽出できた非正常系に対して、処理により回避処理が行え、外部に影響を全く発生させない場合は、新たな、非正常イベント、非正常状態は発生しない。回避処理を行っても非正常の影響を与える場合は、新たな状態あるいはイベントとし、マトリクスに追加し拡張する。これにより、新たに拡張した非正常系の状態とイベントは、正常系や、既に抽出した他の非正常系との相互作用により、さらに次の非正常系が発生する可能性を分析する。

非正常処理機能を考える場合、その発生方法を理解し、それに応じた対応を取らなければならないため、連鎖している状況をマトリクスから逆にトレースできることが必要である。⑤で展開した状態やイベントに、どのイベントと状態により発生したかを連鎖コードとして、イベントあるいは状態項目の欄に記入する。

⑥非正常マトリクスの完成

すべての交点から新たな非正常系の抽出がないか、あるいは、対応できないか対応する必要がなく、運用等でカバーでき、これ以上表が展開できなくなるまで分析を行う。

⑦非正常要求仕様要件表の作成

表に記述された、非正常系と検出方法、対応方法を表に記述し、非正常系処理要件としてまとめる。

３．５　大規模システムでの対応

構成要素が増えてきた場合、マトリクスが膨大になり、分析が非現実的になる。この場合、できる限る構成ブロック単位単位で分析を行い、各ブロックで回避できない非正常系と各ブロックの外部から認識できる状態とイベントを構成要素として同様の分析を行うことにより、現実的な範囲の分析を行う。

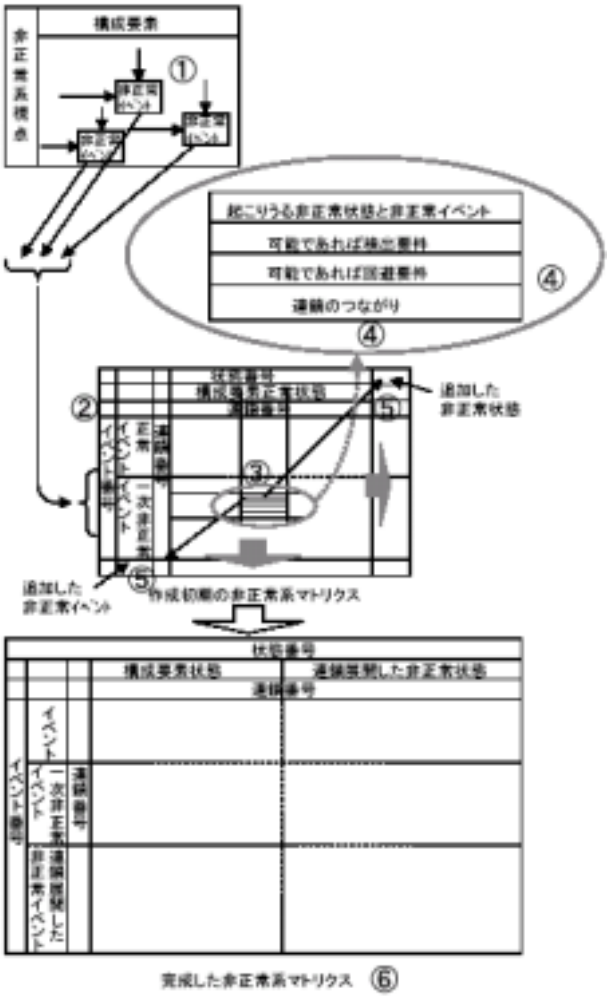

図　1　非正常マトリクスの作成

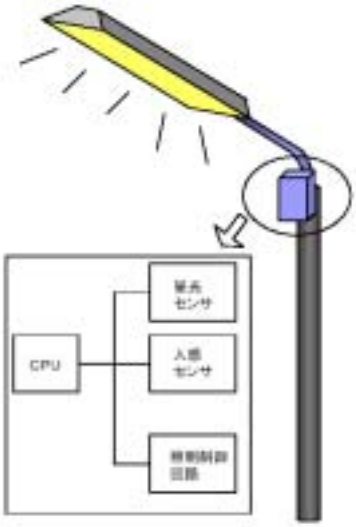

図2　道路灯の事例

4　事例による検証

4．1　事例の構成

夜間に人が近づいてきたときに点灯する道路灯を事例に検討を行った。

事例の構成
ＣＰＵ：低コスト化のためバックアップメモリなし。
昼光検出センサ:ハードウェアのアナログ／デジタル変換により光の量を数値として読み出す。
人感センサ:ハードウェアで一定量以上の大きさの移動物体の最短距離をアナログ／デジタル変換で数値として読み出す。
照明制御:ラッチングリレーによりＩ／ＯポートよりＯＮ，ＯＦＦをパルス出力する。
タイマ:ＣＰＵへの定期割り込みをカウントし内部タイマとして用いる。

4．2　事例のマトリクス記述

事例システムに対し、昼光センサ部と人感センサ部の分析を別に行い、その後道路灯全体の分析を行った。ここでは、昼光センサブロックの分析を取り上げて説明する。図３に昼光センサの分析結果を示す。

４．２．１初期状態のマトリクス

昼光センサの構成であるケース、センサ回路、処理機能に対して、２．２，２．３の要因と照らし合わせながら、初期状態の一次非正常イベントとして、早朝・夕方、街路樹の陰、汚れ・経年劣化、車のヘッドライト、ノイズ、停電・リセット、昼光センサの故障を抽出した。

図３は、状態コードとしてA，B、で表し、イベントコードとして１、２、で表す。以下の説明ではマトリクスの交点を、たとえばＡ５のように示す。状態とイベントの各下には連鎖コードとして、どのマトリクスの交点から新たな状態やイベントとして展開したかを記している。各交点には３行あり、上から発生しうる非正常系、検出方法、対策方法として記した。実際には詳しく記すが表の見易さのために簡略化している。

図３の状態としてAからＥ、イベントとして１から９のマトリクスの部分が、初期状態の昼光センサ処理に関するマトリクスとして作成した部分となる。

4．2．2　マトリクスの展開

Ｂ７の電気的ノイズにより、昼光センサ回路に誤動作を与え、アナログ／デジタル変換ミスにより入力値が一時的に変化してしまう可能性がある。また、Ｄ４の太陽光が街路樹の影になる場合、木の葉の風による揺らぎや枝、幹など遮り方を配慮する必要がある。さらにＥ６の車のヘッドライトを受光した場合、センサ入力値が一時的に変化する。昼光センサ回路の故障によりセンサの値が発振した場合も考慮する必要があり、それらの要因による照明ちらつき防止のため、平均値処理によるノイズ除去を追加する必要があり、その処理をＦとして付け加える。またＦ４、Ｇ４，Ｆ６、Ｇ６のように回避処理からではすべてのセンサ値の読み間違えを回避することは不可能であり、センサ値の誤入力イベントととして１２を加える。

Ｄ３、Ｅ３の早朝・夕方では、昼光センサが昼と夜の判断付近でセンサＯＮ，ＯＦＦの判断変化を繰り返し、照明がちらつく可能性があるため、充分照度が低下して昼光センサをＯＦＦに、充分照度が高くなってからＯＮ処理を行う必要がある。このため、その間の照度での処理をＧに展開する。

Ａ５、Ｂ５のケースの汚れやセンサ回路の経年劣化から、センサ入力値が低下していくことになり、非正常系イベントとして１１を付け加える。

Ｄ１１では、昼光センサが劣化した場合、昼の判断処理が経年的にずれてくるため、一日の最大値と最小値を計算し、それから昼と夜の判定レベルを判定し、動的に判断レベルを変化させることにより劣化や汚れに強いシステムとなるため、この処理をＨに追加する。また、センサ入力値の絶対値が小さくなっており問題が発生する可能性があるため、この状態をＩに追加する。

Ｂ９の昼光センサ回路故障時では、あり得ない値になるか、値が固定になるか、発振する。発振に対する対応は、Ｆ、その他の故障判断処理を追加してＪとし、昼光センサ故障判断したというイベントを１３に追加する。また、これらの判断ができない場合は、センサ値を読み間違えるため、１２にも展開する。

Ｄ１２、Ｅ１２では、ノイズ等の除去ができなかったセンサ入力エラーのため、誤動作する可能性があるが、最悪でも短時間に照明がちらつくことがないように変化判断後一定時間は判断を変えない処理を付け加え、Ｋに展開する。

Ｊ４では、センサ故障判断中に街路樹の木の葉の揺れで太陽光が遮られることが同時に発生した場合を考える必要があり、実験などにより適切なアルゴリズムを考える必要がある。

Ｉ・４では、昼光センサが汚れてセンサの絶対値が低下している状態で、街路樹による太陽光の遮りは全体レベルが低下している為、通常は問題ないが、受光部が平面でなく突起しているなどむらのある汚れ方をしている場合、その方向によって影響を受けてしまう可能性がある。これについては、受光部がむらのある汚れをしないような構造にしなければならない。

Ｈ８の瞬時停電があった場合、前日までの平均値処理がリセットされるため、経年劣化が進んでいる場合、昼夜の判断レベルが一日ずれてしまう。また、Ｇ８の停電リセット状態で、Ｆの昼夜判定レベルの中間にあった場合、判定ができないため、不定とする。Ｃ１３の昼光センサ故障認識状態で、初期化処理が発生した場合、故障認識がリセットされ、センサ判断を誤る可能性があるため、初期処理時にセンサ判断処理を加えることにより誤動作を回避する。

以上の処理を検討し、図３の昼光センサ非正常マトリクスを作成した。

４．３　非正常系の要求仕様要件

図３より昼光センサの非正常系仕様要求要件を整理した。

下記に非正常要求仕様要件表を示す。

①早朝・夕方の照明のちらつきを防止する為に、センサＯＮの判定レベルとＯＦＦの判定レベルに充分幅を持たせ、一旦昼と判断した場合、十分暗くならないと夜と判断しないようにする。

②街路樹などの陰、電気的ノイズ、ヘッドライト、センサ故障時の発振に、できる限り誤動作しないセンサ平均化処理を組み込む。

③センサ受光部の汚れ、センサ回路の経年劣化に対応するため、一日の変化を記録し、昼と夜の判断レベルを変化させる。停電・リセット時にはデータが消える為、規定値に戻し、一日は補正が効かない仕様とする。

④初期化処理時と常時定期的にセンサ故障判定を行う。センサ故障時は、センサ値を不定値として扱う。

⑤平均値処理時には、太陽光の街路樹による遮りが発生するが、昼用の判定レベルには問題ないよう配慮する。

⑥センサ判断は環境により確実ではないため、最悪照明のちらつきが発生しないようにセンサ判断変化後一定時間は再判断しない。

⑦センサ故障判断処理については、街路樹の陰など、値が連続的に大きく変化する環境もあるため、実験しアルゴリズムを定める。

以上の仕様に対し、図３が詳細な状況説明資料として用いることができるため、きめ細かいソフトウェア処理が可能となる。また、システム評価段階で、非正常系仕様抽出の資料として、非正常系の試験を抜けなく実施し確認することが可能となる。

４．４　全体分析

これらの分析から、昼光センサの非正常処理が明確になった。昼光センサを部品として見たシステム全体の分析としては、昼光センサがＯＮである場合、ＯＦＦである場合、不定である場合の３状態のみを意識するだけでよい。これは、電源立ち上がり時やセンサ故障時はすべて処理で不定状態に扱われ、ノイズや環境からの入力に対しては、処理によりちらつき防止処理が働き、外部からは意識しなくても良いためである。以上と同様に、人感センサでの分析を行うことにより、センサＯＮ，センサＯＦＦ、不定の判断だけで扱うことができる。この結果、システム全体の分析としては、各センサからの出力は、センサＯＮ，センサＯＦＦ、不定（初期処理中、故障中）の状態だけになり、道路灯としては、それらマトリクスにより仕様を定義できる。

５　おわりに

要求仕様定義段階での非正常系の抽出方法として、連鎖的な複合状態の分析が行える状態遷移表に着目し、適用を行う為の手法について検討し試行を行った。この結果、分析手法として用いることができることが確かめられた。今後の課題として、多くの事例に適用し手法として確立していく必要がある。検討していく事項として、一次の非正常イベントの抽出に関して、外部的な非正常系の一次要因、ソフトウェア処理からの非正常要因と照らし合わせながら検討を行い、有効であることは確認できたが、今後、非正常系の更なる分析により、非正常要因の体系化を行い、確実に一次非正常系を網羅できる手法として確立していく必要がある。さらに、非正常系の連鎖に関して、今回の試行は、センサが２つで制御が１つの簡単な事例であったが、規模が大きくなると、マトリクスが膨大になり、現実的に使えなくなる可能性がある。このためには、非正常系の連鎖に関してさらに分析を進め、連鎖の分類を体系的に整理し、一時要因とシステム構成が定まった時点で、自動的な作表支援ができるしくみの検討、展開の抜けがないことを確認できる手法の検討に発展させていく必要がある。これらの検討により、ＳＥの個人能力に依存しない、組み込み系の品質設計手法を確立していく。

参　考　文　献

1）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

2)鈴木順二郎、牧野鉄治、石坂茂樹：FMEA,FTA実施法

3)北川賢司：FMEA/FTAの導入法

4)日本技術士会：リスク分析工学

5)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

6)Karl E.Wiegers　Software　Requirement　Second edition

状態コード	A	B	C	D	E	F	G	H	I	J	K
イベント「状態」	昼光センサーケース	昼光センサ面路	初期化処理	昼の処理中	夜の処理中	ノイズ除去のための平均値処理	昼の判定レベルと夜の判定レベルの差の範囲内	1日の平均値から昼夜の検出レベルを計算	経年劣化により変化レベル減少	センサ故障判断処理（範囲外、無変化、値の発振検出）	昼、夜の判断変更後、一定時間の間、判断固定
イベント 遷移コード						B7 B9 D4 D5	D3 E3	A5、B5	A5 B5	B9	F4
1 昼	—	—		—	—	—	—	—	—	—	—
			センサ値は不定								
2 夜	—	—		—	—	—	—	—	—	—	—
			同 上								
3 夕方・早朝		—		昼夜判断レベル付近で頻繁判断を実行	昼夜判断レベル付近で頻繁判断を実行	—		—	判断レベル差が減少、誤動作	—	—
			同 上	判定レベルの差を充分持たせ、回避	判定レベルの差を充分持たせ、回避						
4 太陽光が窓や街路樹などの陰になる		—		木の葉や小枝、幹などの影になり値が安定しない。		木の葉の揺らぎレベル以外は除去が不可	日没の暗さでは問題ないが、環境により影響がある	日中が環境により平均値が大きく異なる。	全体にレベルが下がり問題なし。部分的に汚れないこと。	木の葉の影が風で揺らぎ、センサ異常と見なす可能性	—
			同 上	ノイズ除去を行う必要がある			安定時間タイマにより判断を固定	問題ないような判断値にする		異常と見なさないような判断処理を追加	
5 汚れや経年劣化	汚れでセンサ値低下	アナログ値経年変化									
6 ヘッドライト		—		問題なし	長時間照射された場合、昼と誤判断	長時間照射された場合、誤判断	長時間照射された場合、誤判断	—	全部分的に汚れないこと。	—	—
			センサ値は不定		時間的変化を検出 平均値処理	センサ設置位置注意					
7 ノイズ		面値の取得									
		ノイズ除去平均値処理	同 上								
8 停電、リセット発生後				問題なし	問題なし	初期化後平均値算出までは未定	初期化直後は判断できない	1日の平均計算ができない	初期設定の判断値に戻る	故障情報がリセットされ、判断を誤る	タイマセットのため遷移すると動作が不動作
			同 上			初期化後平均値算出まで不定とする	初期化後平均値算出まで不定とする	初期値に戻す。		初期化処理時に正常か判定	
9 光センサ面路故障		範囲外の値、無変化、値の発振検出									
		範囲チェックと変化の状態で判断									
		センサ故障判断処理	同 上								
11 光センサ値が時間的に変化し狙い値となる A5、B5				昼判断の可能性		—	—			—	—
			同 上	1日の平均値処理を行い判断する							
12 センサ値を誘み間違える場合がある F4、G4、F8、G8、D9				昼判断の可能性 時間的変化を検出	夜判断の可能性 時間的変化を検出		—	—	—	—	—
			同 上	充分ちらつかないレベルまで平均値処理	充分ちらつかないレベルまで平均値処理						
13 センサ故障判断処理 D9				判断を不定にする	判断を不定にする	処理しない	処理しない	処理しない	処理しない		処理しない
			初期故障診断								

図3　光センサ非正常系マトリクス