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

2007－SE－155 (25)
2007／3／23

ユースケースによる安全性要求分析のための想定外シナリオ抽出法の提案

熊澤 努† 玉井 哲雄†
† 東京大学大学院総合文化研究科

近年のソフトウェアシステムの大規模化，複雑化により，システムの安全性の確保が困難になっている．この課題に応えるためには，要求分析段階での安全性の分析が重要である．そこで，本論文では，ユースケースをもとに，安全性分析に必要な安全性を脅かすシステム利用シナリオを，時相論理を用いて抽出する方法を提案する．また，その方法の有効性を確認するために行った事例研究につき報告する．

A Study on How to Elicit Anticipated Scenarios from Use Cases for Safety Requirements Analysis

Tsutomu Kumazawa† Tetsuo Tamai †
†Graduate School of Arts and Sciences, University of Tokyo

Recently, Increasing the sizes and complexities of software systems has made it difficult to maintain their safety. To solve the problem, it is important to analyze system safety during requirements analysis step. In this paper, we propose a method to elicit system use scenarios from use cases that threaten system safety, using temporal logic. We show the validity of our approach with a case study.

1. はじめに

近年，ソフトウェアが身近な存在として認知され，様々な製品やサービスに使われるにつれて，ソフトウェアシステムの大規模化，複雑化が進んできている[3),8)]．このような状況下で，ソフトウェアがユーザの要求を満たし，かつ，安全に動作することが求められている．そのため，開発者はソフトウェア開発の最初の段階からプログラムの実装に取り掛かるのではなく，まず，ユーザのソフトウェアに対する要求を定義，記述し，分析するという要求定義/分析工程を実施する．しかし，ユーザの要求は自然言語で記述されることが多いため，曖昧さや矛盾を含みやすい．また，ユーザがシステムに関する特定の状況しか考慮していないことも少なくなく，要求の網羅性が低くなりがちである．特に，ソフトウェアの安全性や信頼性などの非機能的な要求については，ユーザの意識が低くなりやすく，要求仕様から漏れてしまうことが多い．

本論文では，記述漏れを起こしがちな性質である安全性に焦点を当てる．安全性とは，事故や損害のないことである[11)]．安全性が十分に記述されないと，システム稼動時に障害や事故が発生して，企業や社会に大きな損害を与える可能性がある．1990年代の欧米の宇宙船舶事故の原因を調査した文献では，安全性分析が不十分だったことが事故要因の1つであったと述べている[12)]．したがって，要求分析時にシステムの安全性要求を記述，分析することが望まれる．本研究の目標は，機能要求分析時に用いられる自然言語で記述されたユースケースから，システムの安全性を脅かすシナリオを導出する手法を得ることである．このようなシナリオが得られれば，そこから安全性要求を導くことが可能になると考えられる．本論文では，ユースケースに対して時相論理を用いることで，システムの安全性を脅かす想定外シナリオを抽出する方法を提案する．

簡単な街灯システム構築事例を用いて提案法を説明する．この事例は三瀬らによる論文[16)]で扱われた．

本事例は，夜間に歩行者が近づいてきた際に点灯する街灯の開発である．日中には，歩行者が近くに来ても点灯しない．この街灯システムの，ハードウェア構成は，次のように基本的に決定されている．

- CPU：システムを制御する．
- 昼光検出センサ (昼光センサと呼ぶ)：光量を数値的に検出する．
- 人感センサ：一定の大きさ以上の移動物体とセン

表 1 本研究で扱うユースケース形式

構成要素	説明
ユースケース名	ユースケースを実行するアクタが達成したいと望む目標を表現した名称を記述する
アクタ	ユースケースに現れるアクタを記述する
システム	分析対象システムを記述する
オブジェクト	ユースケースで利用，操作される実体で，システムとアクタとで交換されるデータや情報も含む
事前条件	ユースケースの実行前に成り立つべき条件である
基本系列	ユースケースの目的が達成されるようなシステムとアクタとの相互作用を記述する
代替系列	基本系列の中で条件によって相互作用の内容が分岐する場合に分岐以降の相互作用を記述する
事後条件	ユースケースの実行後に真となる条件である

サとの最短距離を数値的に検出する．

• 他，照明制御回路，CPU への割り込み用タイマ．

昼光センサ値が一定値以下ならば夜間であるとすることで，現在の時間帯を判断する．また，歩行者が近くにいるかどうかは，人感センサにより検出する．本論文で扱うのは，安全性を脅かすシナリオの抽出である．

本論文の構成は次の通りである．2 節では，本論文で扱う背景知識について説明する．3 節で関連研究を概観する．4 節では，本論文で提案する想定外シナリオ抽出法を説明する．5 節では，街灯事例を用いて提案する方法の有効性を示す．最後に，6 節で本論文を総括し，今後の研究課題を議論する．

2. 背景知識

本節では，本論文で扱う背景知識を概説する．

2.1 ユースケース

ユースケースは，システムと利用者との間の相互作用を時系列に記述したものである．システムと対話する利用者や他のシステムの役割をアクタと呼ぶ．

ユースケースは，オブジェクト指向開発方法論の 1 つである OOSE において，方法論の主導的な役割を担う要求抽出手法と位置づけられた[6]．その後，ユースケースは，オブジェクト指向開発における統一的なモデル記法である統合モデル記述言語 UML(Unified Modeling Language)[7] の要素に組み込まれている．

ユースケースは主に自然言語の文章で記述されるが，その記法は定まっていない．本研究では，ユースケースの形式を**表 1** のように定める． 表 1 の代替系列には，ユースケースの事後条件が成立する相互作用を記述する．事後条件が成立しないシナリオは異なるユースケースとして記述する．

街灯の要求を記したユースケースを**図 1** に示す．

ユースケース名 街灯点灯
アクタ 歩行者
システム 街灯
オブジェクト 時間帯，ライト
事前条件 夜間である，街灯が消灯している，
　歩行者が街灯から遠い
事後条件 街灯が点灯している，歩行者が街灯に近い
　人感センサが歩行者が近いことを検出している
　昼光センサが夜間であることを検出している
基本系列
1.歩行者が街灯に近づく．
2.人感センサが歩行者を検出する．
3.街灯が点灯する．
代替系列
－

図 1 街灯点灯ユースケース

2.2 時相論理

時相論理は，古典的な論理学に時間の概念を追加したものであり，ある時点より未来や過去のことについての命題及び述語の真偽について述べる[13]．時相論理では，古典的論理の演算に加えて様相演算を用いる．時相論理は並行プロセスの仕様記述や検証に利用されていて，時間的な表現力に違いのある論理が複数研究されている[2]．本論文で用いる時相論理の意味論や推論システムには，Manna と Pnueli による体系[13] を採用する．本研究で用いる演算の記法を**表 2** にまとめる．ただし，表 2 で p，q は論理式であるとする．

表 2 論理記号表

論理記号	解釈
$p \wedge q$	論理積 (連言)
$p \vee q$	論理和 (選言)
$p \rightarrow q$	含意
$p \leftrightarrow q$	等価
$\neg p$	否定
時相論理演算子	
$\Box p$	"Henceforth(Always) p"
$\Diamond p$	"Eventually p"
$\bigcirc p$	"Next p"
$p\mathcal{U}q$	"p Until(Strong-until) q"
$p\mathcal{W}q$	"p Waiting-for(Unless/Weak-until) q"
$\ominus p$	"Previously p"
$\boxminus p$	"Has-always-been(always in the past) p"
$\blacklozenge p$	"Once p"
$p\mathcal{S}q$	"p Since q"
$p\mathcal{B}q$	"p Back-to q"

3. 関連研究

システムの安全性に関する関連研究は多くある．

ユースケースを用いた研究には，ミスユースケース[18] やアビューズケース[14] が提案されている．これらは，システムにとって望ましくないシナリオを，ユースケースと同様の記法を用いて記述したものである．

Uchitel らは，システムのコンポーネント同士のメッセージ通信を時系列に表示したメッセージ系列図を対

象とし，それを状態遷移プロセスとして表現することで，当初のシナリオにはないメッセージの組合わせを獲得する手法を提案している[19]．

Van Lamsweerde 等は，達成すべきではない目標を時相論理式で記述し，形式的詳細化パターンによって，その上位，下位の目標を系統的に導出する手法を提案している[9),10)]．

Jackson の問題フレーム[5)] を用いて，Haley 等は問題図から安全性要求を導出する手法を提案している[4)]．この手法では，開発対象の機械の持つ脆弱性を指摘することで，問題図を修正する手順を定める．

システム全体の障害分析法で，産業界でもよく知られた手法には，以下がある．

- 故障木分析 (FTA)：障害が発生する既知の事象を選び，その要因をトップダウン手法で求める．
- 故障モード影響分析 (FMEA)：システムの部品と故障モードを発見し，各故障モードが他の部品やシステム全体に与える影響を分析する．

また，近年提案されている障害分析法には，三瀬等による非正常シナリオがある[15),16)]．非正常シナリオは，システムの故障などの想定外のシナリオを記述したものである．この方法は，まず，システムを部品とそれらの間の通信に分解して，想定外現象を抽出する．そして，現象とシステム状態とで状態遷移表 (文献 16) では非正常系マトリクスと呼称) を作成して，想定外現象の原因や対策法を分析する．

4. 提案する想定外シナリオ抽出法

前節で述べた関連研究には，以下の一方または両方の問題点がある．

- 非形式的かつ発見的な手法のため，分析結果が分析担当者個人の能力に著しく依存する．
- 自然言語で記述されたユースケースを扱っていないので，手法の適用範囲に限界がある．

われわれは，上記の点を解決する手段として，自然言語で記述されたユースケースを時相論理表現し，それを使ってシステムの安全性を脅かすシナリオを抽出する手法を提案する．実施プロセスは以下の通りである．

(1) 分析対象のユースケースの整理を行う．
(2) ユースケースを分析規則により構造化する．
(3) 構造化したユースケースを論理式に変換する．
(4) 論理式で記述したユースケースから安全性を脅かす想定外シナリオを抽出する．

想定外シナリオの抽出は 2 つの戦略に基づく．最初の戦略 (戦略 1 と呼ぶ) は，ユースケースの事後条件が成立するために必要な暗黙の想定事項を用いて，安全性を脅かすシナリオを抽出する方法である．2 番目の戦略 (戦略 2 と呼ぶ) は，ユースケースに関わる実体であるアクタの自律的な活動に注目することで，新たなシナリオを発見する方法である．

4.1 分析対象ユースケースの整理

シナリオの記述法に関しては，Achour の文献[1)] に内容と記法についての詳細なガイドラインが提示されていて，本研究でも適用できる．特に，本研究では分析の便宜のため，以下の方法に従うこととした．

- 基本，代替系列は実行ステップごとに番号付け，1 つの文で記述する．
- 指示代名詞は，それが指示するものを明示する．
- 繰返し構造など，複雑な複合文を使用しない．
- 他のユースケースと <<include>> や <<extend>> の関連を持つ場合，必要に応じて両者を展開した完全なユースケース系列を新たに作成する．
- 代替系列が事後条件を満たさない場合，その系列を基本系列とするユースケースを作成する．

4.2 分析規則による構造化

自然言語記述のユースケースは曖昧さを含むことが多い．そこで，Rolland らが提案した，自然言語記述ユースケースから曖昧さのないユースケースを作成する手法[1),17)] を用いて，ユースケースの文を構造化する．

この研究では，構造化した意味パターンを用いて，自然言語の文が表す内容を一意に表現する．意味パターンは文や節の持つ等価な意味構造をテンプレートにしたもので，節意味パターンと文意パターンに分類される．自然言語から意味パターンのインスタンスへの変換規則を分析規則と呼ぶ．例えば，図 1 の基本系列ステップ 1 の記述「歩行者が街灯に近づく」は，アクタである歩行者の振舞いを述べた文であり，分析規則により次のように表現できる．

Action(近づく)[**Agent:**歩行者; **Object:**街灯;]

節意味パターンは，動詞の意味を動作と状態の 2 種類に分類したものである．動作は，実体の行為や実体間での通信を表し，状態は実体の持つ性質や実体同士の関係を表す．文意パターンは節や文の間の意味的な関係を表し，系列，制約，並行，繰返しがある．

分析規則に加えて，分類規則や完全性規則を用いる[17)]．分類規則は，指示語の指す対象を同定したり，条件文で判定条件を分類するの規則であり，完全性規則は，自然言語記述の曖昧さのために意味パターンのインスタンスで不明確な要素を補完する規則である．

図 1 の構造化を行った結果を，**図 2** に示す．

4.3 時相論理式への変換

以下の手順を実施して，構造化したユースケースを

ユースケース名 街灯点灯
アクタ 歩行者
システム 街灯
オブジェクト 時間帯，ライト
事前条件 **State[Object:**時間帯;**State:**夜間;],
State[Object:街灯;**State:**消灯;],
State[Object:歩行者;**State:**遠い;]
事後条件 **State[Object:**街灯;**State:**点灯;],
State[Object:歩行者;**State:**近い;]
State[Object:人感センサ;**State:**歩行者が近い;]
State[Object:昼光センサ;**State:**夜間;]
基本系列
1.**Action**(近づく)[**Agent:**歩行者;**Object:**街灯;]
2.**Action**(検出する)[**Agent:**人感センサ;**Object:**歩行者;]
3.**Action**(点灯する)[**Agent:**街灯;**Object:**ライト;]

図 2 街灯点灯ユースケースの分析結果

表 3 街灯事例で使用する対象定数

対象定数	説明
昼光センサ	昼光センサを表す定数
人感センサ	人感センサを表す定数
光量	光の量を表す定数
近い	歩行者の「街灯の近くにいる」という状態を表す
遠い	歩行者の「街灯の遠くにいる」という状態を表す
歩行者が近い	人感センサの「歩行者が街灯の近くにいる」という状態を表す
点灯	街灯の「点灯している」という状態を表す
消灯	街灯の「消灯している」という状態を表す
日中	時間帯の「日中である」という状態を表す
夜間	時間帯の「夜間である」という状態を表す

時相論理式に変換する．

(1) 扱う対象定数を決定する．

(2) 状態述語を定める．

(3) 論理式を作成する．

4.3.1 対象定数の定義

本研究では，ユースケース項目のシステム，アクタ，オブジェクトに属するものを対象ユースケースにおける対象定数と考える．また，必要があれば以降の分析で用いる対象定数を追加する．街灯点灯ユースケース記述から，街灯システム，歩行者，時間帯，ライトを本事例で用いる対象定数と定めた．さらに，**表 3** の対象定数を新たに導入し，今後用いることにする．

4.3.2 述語の定義と指示

ここでは，構造化ユースケースにおけるアクタやシステムの通信，動作とそれによって変化する状態を述語に変換する．システムやアクタの状態を表明した述語を状態述語という．ある動作が行われたことを表す述語を動作述語という．例えば，前述の意味パターンにおける状態パターンは，次の述語記号を用いて基本的な述語に変換する．

$State(agent, state) \approx agent$ の状態は $state$ である

ここで，$\approx$ は指示を表し，左辺の用語や述語を右辺の非形式的記述で意味づける[5),20)]．次に，以下の手順により，意味パターンの通信と動作が実行されたことを表現する述語を生成する．

(1) 構造化ユースケース内の通信，動作を抽出する．

(2) 各動作を表す述語 (または命題) を指示，定義することで，通信や動作を形式的に記述する．

各通信/動作の実行を主張する述語を**表 4** のように指示した．また，対象定数で具体化した述語の意味や述語同士の関係を**表 5** のように定めた．本研究では，述語を具体化した命題論理の範囲で論理を扱っているので，**表 6** に示す命題記号を導入し，命題時相論理として以降の分析を行うこととする．

表 4 街灯事例で使用する動作述語の指示

動作述語	指示
点灯する (a, o)	a は o を点灯するという動作を行う
消灯する (a, o)	a は o を消灯するという動作を行う
近づく (a, o)	a は o に近づくという動作を行う
検出する (a, o)	a は o を検出するという動作を行う

表 5 街灯事例で使用する対象定数

述語	意味や関連の説明
$State$(街灯, 点灯)	街灯が点灯している
$State$(時間帯, 夜間)	夜間である
$State$(歩行者, 近い)	歩行者が街灯の近くにいる
$State$(人感センサ, 歩行者が近い)	人感センサが歩行者が近くにいると判断している
$State$(昼光センサ, 夜間)	昼光センサが夜間であると判断している
$State$(街灯, 消灯)	$\neg State$(街灯, 点灯)
$State$(時間帯, 日中)	$\neg State$(時間帯, 夜間)
$State$(歩行者, 遠い)	$\neg State$(歩行者, 近い)

表 6 街灯事例で使用する述語の命題名

省略記号	定義
$Approach$	近づく (歩行者, 街灯)
$TurnOn$	点灯する (街灯, ライト)
$TurnOff$	消灯する (街灯, ライト)
$DetectPerson$	検出する (人感センサ, 歩行者)
$Light\_On$	$State$(街灯, 点灯)
At_Night	$State$(時間帯, 夜間)
$Near$	$State$(歩行者, 近い)
$Sensor\_Person$	$State$(人感センサ, 歩行者が近い)
$Sensor\_Night$	$State$(昼光センサ, 夜間)

4.3.3 ユースケース系列の論理表現

ユースケース S のもとで実行される実体の動作を，時相論理式で表現される性質である進行性で表現する．進行性とは，望ましいことが，プログラムやモデルの実行中にいつかは起こることを表明する性質である[13)]．

$$S \vdash action \Rightarrow \Diamond next\_action \quad (1)$$

ここで，$action$ と $next\_action$ は論理式である．直観

$$
\left[\begin{array}{l}
\textbf{ユースケース名}\ \text{街灯点灯} \\
\textbf{アクタ}\ \text{歩行者} \\
\textbf{システム}\ \text{街灯} \\
\textbf{オブジェクト}\ \text{時間帯，夜間，ライト} \\
\textbf{事前条件}\ At\_Night, \neg Light\_On, \neg Near \\
\textbf{事後条件}\ Light\_On, Near, Sensor\_Person, \\
\qquad Sensor\_Night \\
\textbf{基本系列} \\
\left[\begin{array}{l}
\Diamond Approach \wedge \\
(Approach \Rightarrow \Diamond DetectPerson) \wedge \\
(DetectPerson \Rightarrow \Diamond TurnOn)
\end{array}\right]
\end{array}\right]
$$

図 3 街灯点灯ユースケースの時相論理式変換結果

的には，式 (1) は，いつの時点でも *action* が成り立つならば，その後いずれ *next_action* が成り立つことを意味する．ただし，ユースケース系列における最初の動作は特別の場合と考え，(1) を弱めた以下の式を用いる．

$$\Diamond action \tag{2}$$

(1)，(2) によりユースケースにおけるシステムやアクタの動作の実行順序を記述できる．街灯点灯ユースケースを論理式に変換した結果を**図 3** に示す．

4.4 戦略 1：前提条件を用いた想定外シナリオ抽出

戦略 1 では，前項の論理式を用いて，ユースケースで暗黙の前提条件より想定外シナリオを抽出する．

戦略 1 は以下の想定に基づく．*PRE* を事前条件，*REQ* をユースケース系列とそれが表す要求，*DOM* を対象領域で成立する性質，そして *POST* を事後条件とすると，*PRE* と *REQ*，*DOM* が与えられれば，いずれ *POST* が成立することが論理的に示せるので，

$$PRE, REQ, DOM \vdash \Diamond POST$$

しかし，実際には，ユースケースの記述だけでは事後条件の証明が出来ないことが多い．本研究では，分析対象ユースケースに明記されていない，暗黙に想定されている条件が存在することがその理由と考え，事後条件の証明を完成させる暗黙の条件式を求める．暗黙の条件式には以下のようなものがある．

- *PRE*：分析対象ユースケースに明示されておらず，証明に必要な暗黙の事前条件である．
- *REQ*：ユースケース系列でのシステムに対する暗黙の要求である．ユースケース系列に暗黙の要求事項が存在する場合，開発者は当然の事柄として意識的に扱わないことが多い．この要求も，事後条件の証明過程で発見的に抽出する．
- *DOM*：対象領域において成立する性質であり，要求から仕様を得るのを支援する役割を果す[20]．本研究では，事前条件と系列から事後条件を成立させるのに必要な知識であると考える．

各式の否定を求めると，ユースケース系列において各前提が成立しない条件が得られる．求めた条件は，ユースケースに悪影響を与えたり，開発関係者が対象領域に対して想定していない条件である．得られた論理式が分析対象のユースケースにおいて成立する場合を考察することで，システムに有害な相互作用や見逃されていた系列を抽出できる．

以上より，戦略 1 の手順を以下のように定める．

(1) ユースケースの事前条件と各実行ステップを表す論理式から，事後条件を証明する．
(2) 証明できない場合，必要な事前条件や対象領域の性質を表す論理式を導入し，証明を完成させる．証明に必要なシステム要求も抽出する．
(3) 導入した各論理式の否定 (否定条件) を求める．
(4) ユースケースで各否定条件が成立する場合に起こる結果を考察し，新たなシナリオを抽出する．

4.4.1 暗黙の前提条件の抽出

街灯点灯ユースケースの場合も事後条件の証明ができない．そこで，各ハードウェア，歩行者，環境の性質を抽出した．以下にライトの性質の例を示す．

$$TurnOn \Rightarrow Light\_On \tag{3}$$

$$Light\_On \Rightarrow Light\_On\ \mathcal{W}\ TurnOff \tag{4}$$

式 (3) は，街灯が点灯動作を実行すれば，ライトが点灯状態になることを意味する．式 (4) は，ライトが点灯状態であるならば，ライトの消灯動作が行われない限り点灯状態であることを表す．

次に，街灯点灯ユースケースの事前条件を挙げた．

$$\Box At\_Night \tag{5}$$

式 (5) は，ユースケース実行時は常に夜であることを主張する．これらの領域の性質，事前条件，ユースケース系列を使って事後条件の証明を実施した．証明は紙面の都合上省略するが，その過程で街灯システムに関する以下のような要求を見出した．

$$\Box\neg TurnOff \tag{6}$$

式 (6) は，ライトの消灯動作が行われないことを表す．

4.4.2 否定条件と想定外シナリオの抽出

安全性に影響を与えるような想定外のシナリオを抽出するために，前節の性質や条件の否定を求める．前述のライトに関する前提条件の否定を示す．

$$\Diamond(TurnOn \wedge \neg Light\_On) \tag{7}$$

$$\Diamond(Light\_On \wedge (\neg TurnOff)\ \mathcal{U}\ (\neg Light\_On \wedge \neg TurnOff)) \tag{8}$$

式 (7) は，街灯が点灯動作を実行しても，ライトが消灯状態であることを意味する．式 (8) は，ライトが点灯状態であるが，ライトの消灯動作が行われる前に消灯状態になることを表す．これらの原因として，ライトが寿命であること，ライト本体の故障，および照明

制御回路の故障をシナリオとして抽出した (**表 7**).

次に，事前条件の否定条件を求めた．

$$\Diamond\neg At\_Night \tag{9}$$

式 (9) は，いずれ夜間でなくなることを主張する．夜が明ける場合が考えられる．

最後に，システムへの要求の否定条件を記す．

$$\Diamond TurnOff \tag{10}$$

式 (10) は，いずれ消灯動作が行われることを意味する．これは，システムに障害が発生する場合に相当する．

以上より抽出したシナリオを，表 7 に一覧で示す．

4.5 戦略 2：活動分析による想定外シナリオ抽出

戦略 2 は，戦略 1 では抽出が難しいアクタの活動に関するシナリオを抽出する．戦略 2 は発見的であり，分析対象ユースケースとは異なるシナリオを発見することを目的とし，以下のステップにより実施する．

(1) ユースケース系列の論理式から，各アクタが動作主となる通信/動作を表す式を取り出す．

(2) 式 (1), (2) により，取り出した論理式の時間的な関係を記述する．これは，ユースケース内での各実体の振る舞いの順序関係を論理式で表現することを意味する．

(3) 記述した論理式の否定を取り，ユースケースにあるアクタの動作/通信の実行順序関係が成立しない条件を導く．

(4) 否定された動作/通信の実行順序関係の代わりに，その実体がどのような活動を起こしうるかを考察する．そして，必要に応じて，新たに条件を追加して，考察した活動からシナリオを導く．

(3) で論理式の否定を取るのは，アクタの振る舞いが分析対象ユースケースの系列にない場合を考察するためである．この論理式の否定は，実体の振る舞いがユースケース系列の記述と一致しないことを表すので，新たなシナリオを求める手掛かりとなる．戦略 2 は発見的なので，このような手掛かりがあることは，シナリオ抽出において有効であると考えられる．以上のようにして得られたシナリオは，対象ユースケースとは異なる順序でアクタが振舞った場合や，未知の動作/通信を行った場合を記したものであり，新たなユースケースであると考えられる．

なぜ戦略 2 が必要なのだろうか．戦略 1 は暗黙の想定条件を抽出するが，ユースケースに記されている動作の実行順序関係は常に一定であると仮定している．しかし，動作の順序が変わったり，動作の抜けや挿入があることによって生ずる新たなシナリオの作成には対応できない．よって，想定外の活動の分析は戦略 1 では実施が困難なので，戦略 2 が必要である．

表 7 街灯点灯ユースケースの戦略 1 による新たなシナリオ抽出結果

シナリオ	発生ステップ	シナリオの説明
電球の寿命	3	電球の寿命が尽きているため，点灯しない
ライトの故障	3	ライトが故障のため，点灯しない
照明制御回路の故障	3	照明制御回路の故障のため，点灯しない
電球の寿命	3	電球の寿命が尽きているため，点灯状態になった後に消灯状態になる
ライトの故障	3	ライトが故障のため点灯状態になった後に消灯状態になる
照明制御回路の故障	3	照明制御回路の故障のため，点灯した後に消灯状態になる
人感センサの故障	2	センサが故障のため，センサが歩行者を検出しても，歩行者が近くにいると判断しない
人感センサの誤動作	2	センサの誤動作のため，センサが歩行者を検出しても，歩行者が近くにいると判断しない
人感センサの故障	2,3	センサが故障のため，歩行者が近くにいると判断した後で，歩行者が近くにいないと判断する．
人感センサの誤動作	2,3	センサの誤動作のため，歩行者が近くにいると判断した後で，歩行者が近くにいないと判断する．
昼光センサの故障	1,2,3	センサが故障のため，センサが夜間基準の光量を検出しても，夜間でないと判断する
昼光センサの誤動作	1,2,3	センサの誤動作のため，センサが夜間基準の光量を検出しても，夜間でないと判断する
昼光センサの故障	1,2,3	センサが故障のため，センサが夜間と判断した後で，いずれ夜間でないと判断する
昼光センサの誤動作	1,2,3	センサの誤動作のため，センサが夜間と判断した後で，いずれ夜間でないと判断する
外部照明の存在	1,2,3	外部照明の存在により，夜間に日中基準の光量が発生し，夜間でないと判断する
外部照明の存在	1,2,3	外部照明の存在により，夜間にセンサ仕様範囲をこえる光量が発生する
夜明け	1,2,3	夜明けに昼夜判断基準値付近で光量が安定しない
システム障害	1,2,3	システムに障害が発生し，街灯消灯動作を実行する
人感センサの故障	2,3	センサが故障のため，センサが歩行者を検出した後に，歩行者を検出しなくなる
人感センサの誤動作	2,3	センサの誤動作のため，センサが歩行者を検出した後に，歩行者を検出しなくなる

ユースケースに記されたシステムの振舞いは，システムに対する要求を表し，アクタの振舞いや環境に依存する．よって，戦略 2 では，システムの振舞いの分析は行わず，外部実体であるアクタのみを分析する．

4.5.1 実体ごとの活動の抽出とその論理式表現

歩行者の活動をユースケースと戦略 1 で導入した活動から取り出した．街灯点灯ユースケースの場合，歩行者の活動は近づくのみであり，それを論理表現した式は以下のように表される．

$$\Diamond Approach$$

表 8　街灯点灯ユースケースの戦略 2 による新たなシナリオ抽出結果

ユースケース名	発生箇所	シナリオの説明
歩行者が近づかない	ステップ 2	歩行者が街灯に近づかず，街灯が点灯しない．

表 9　表 8 のシナリオの戦略 1 によるシナリオ抽出結果

シナリオ	発生ステップ	シナリオの説明
ライトの故障	1	ライトが故障のため，点灯動作が実行されていないのに点灯状態になる
照明制御回路の故障	1	照明制御回路の故障のため，点灯動作が実行されていないのに点灯状態になる
人感センサの故障	1	センサが故障のため，歩行者が近くにいないのに，歩行者が近くにいると判断する．
人感センサの誤動作	1	センサの誤動作のため，歩行者が近くにいないのに，歩行者が近くにいると判断する
システム障害	1	システムに障害が発生し，人感センサが歩行者を検出していないのに，街灯点灯動作を実行する
人感センサの故障	事前条件	センサが故障のため，歩行者が近くにいると誤判断する
人感センサの誤動作	事前条件	センサの誤動作のため歩行者が近くにいると誤判断する
人感センサの故障	1	センサが故障のため，歩行者が近くにいないのに，歩行者の存在を検出する．
人感センサの誤動作	1	センサの誤動作のため，歩行者が近くにいないのに，歩行者の存在を検出する

4.5.2　論理式の否定

前項で求めた活動の否定を取り，ユースケースで想定されている活動を歩行者が行わない条件を求めた．歩行者の活動の否定は以下の通りである．

$$\Box\neg Approach$$

4.5.3　新たな活動の発見とシナリオ記述

街灯点灯ユースケースに対して抽出したシナリオを**表 8** に示す．また，求めたシナリオに戦略 1 を適用して分析した結果を**表 9** に示す．

5. 考　　察

ここでは，分析結果から提案法の有効性を考察する．われわれは，図 1 の他に，夜間に歩行者が街灯から立ち去る場合と，日中に歩行者が街灯に近づく場合のユースケースを作成し，提案法による分析を実施した．

昼光センサに関して，提案法による結果と，三瀬らの非正常系マトリクスによる結果[16] とを比較する．ただし，三瀬らはシステムの初期化や故障診断処理も分析対象としているが，具体的な内容が事例の記述にないため，ユースケースを作成することができない．そのため，これらの処理については考えないこととした．

夜間における街灯の分析結果に関して考察する．夜間に外界から照明が与えられることで，昼光センサが夜間でないと判断する場合に関しては，三瀬らはその要因例である車両のヘッドライトのみを発見的に分析している．表 7 から，提案法は，夜間に外界から照明が与えられる状況を論理により系統的に発見した．また，三瀬らは，センサの故障，経年劣化などの現象によりセンサが昼夜を誤判断する場合も分析している．表 7 から，提案法では，そのようなセンサの誤動作による誤判断シナリオを抽出できた．日中における街灯の分析に関しても同様に，三瀬らは異常を発生させる現象と要因例を抽出している．本手法の結果，彼らが分析した状況を論理により系統的に抽出できた．以上より，ユースケースで記述できる範囲ならば，提案法により三瀬らが分析した現象を網羅できた．われわれはユースケースのステップごとに安全性を脅かす状況の発生を検討し，ステップ単位で分析を実施した．

人感センサやライトの動作についても，表 7 からシステムの安全性を脅かすシナリオを抽出できた．

本事例では，分析対象ユースケースに戦略 2 を適用し，抽出したシナリオに対してさらに戦略 1 を適用した．その結果，表 9 に示すように，詳細な分析を実施できた．本手法は，戦略を繰り返し適用して分析ができることが分かる．

以上の考察から，提案法は以下の点で有効であると考えられる．

- 発生しうる障害や事故を系統的に分析できる．
- ユースケースのステップ単位の粒度で詳細にシナリオを抽出できる．
- 抽出したシナリオに対して，繰り返し適用して想定外シナリオを求めることができる．

6. おわりに

本論文でわれわれは，ソフトウェア開発における安全性要求分析のために，ユースケースから，以下の 2 つの戦略からなる想定外シナリオの抽出法を提案した．

(1)　戦略 1：事後条件を証明するために必要な暗黙の前提条件からシナリオを抽出する．

(2)　戦略 2：自律的な外部実体であるアクタの動作から新たなシナリオ抽出する．

本手法は，既存の手法に対して次のような利点があると考えられる．

- 自然言語で記述されたユースケースを対象としているので，手法の適用範囲が広い．
- 論理を用いることで，開発者個人の発見的能力への依存度が低い．

- 本来，システムの機能要求を記述するユースケースを，非機能要求記述への利用に拡張できる．

また，本手法による分析を事例を用いて実施し，既存手法の 1 つである文献 16) の成果と比較した．その結果，以下の点で，提案法の有効性を確認した．

- 障害や事故のシナリオを系統的に抽出できる．
- 分析粒度をステップレベルにすることができる．
- 抽出したシナリオに対して，繰り返し手法を適用して分析を実施できる．

一方，本研究には，多くの課題が残されている．

- 論理式への変換法に用いる Rolland らの意味パターン[17] の内，繰返しや並行などを扱っていない．全ての意味パターンについて論理表現できるようにする必要がある．本研究の戦略 1 では，まず，ユースケースの事後条件が証明可能であるために必要な条件を考えたが，必要な条件や性質を求め，適切な論理式で記述する手法が必要である．
- 本研究の手法から，想定外シナリオに対する対策を記述する手段が必要である．
- 戦略 1 で行う時相論理の体系を用いた証明の自動化が必要である．
- より実用的な事例へ適用し，成果を評価する必要がある．

謝辞 本研究を行うにあたって，有益な助言を下さった玉井研究室，増原研究室の皆様に感謝いたします．

参 考 文 献

1) C. Ben Achour: Guiding Scenario Authoring, *Proc. of the Eight European - Japanese Conf. on Information Modeling and Knowledge Bases*, pp. 181-200, 1998.
2) P. Bellini, R. Mattolini, and P. Nesi: Temporal Logics for Real-time System Specification, *ACM Comput. Surv. (CSUR)*, Vol. 32, No. 1, pp. 12-42, 2000.
3) 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: ソフトウェア開発データ白書 2006: IT 企業 1400 プロジェクトの定量データで示す開発の実態, 日経 BP 社, 2006.
4) C. B. Haley, R. C. Laney, and B. Nuseibeh: Deriving Security Requirements from Crosscutting Threat Descriptions, *Proc. of the Third Int. Conf. on Aspect-Oriented Software Development*, pp. 112-121, 2004.
5) M. Jackson: *Problem Frames: Analysing and Structuring Software Development Problems*, Addison-Wesley, 2001.
6) I. ヤコブソン，M. クリスターソン，P. ジョンソン，G. ウーバガード著，西岡利博，渡邊克宏，梶原清彦監訳: オブジェクト指向ソフトウェア工学 OOSE：use-case によるアプローチ, アジソン ウェスレイ・トッパン, 1995.
7) I. Jacobson, G. Booch, and J. Rumbauch: *The Unified Software Development Process*, Addison-Wesley, 1999.
8) 経済産業省商務情報政策局 情報政策ユニット情報処理振興課: 2006 年版組込みソフトウェア産業実態調査報告書 第 2 版, 2006.
9) A. van Lamsweerde and E. Letier: Handling Obstacles in Goal-Oriented Requirements Engineering, *IEEE Trans. on Software Eng.*, Vol. 26, No. 10, pp. 978-1005, 2000.
10) A. van Lamsweerde: Elaborating Security Requirements by Construction of Intentional Anti-Models, *Proc. of the 26th Int. Conf. on Software Eng.*, pp.148-157, 2004.
11) N. G. Leveson: *Safeware: System Safety and Computers*, Addison-Wesley, 1995.
12) N. G. Leveson: The Role of Software in Spacecraft Accidents, *AIAA J. of Spacecraft and Rockets*, Vol. 41, No. 4, pp.564-575, 2004.
13) Z. Manna and A. Pnueli: *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer-Verlag, 1992.
14) J. McDermott and C. Fox: Using Abuse Case Models for Security Requirements Analysis, *Proc. of the 15th Annual Computer Security Applications Conf.*, pp. 55-66, 1999.
15) T. Mise, Y. Shiniashiki, T. Nakatani, N. Ubayashi, K. Katamine, and M. Hashimoto: A Method for Extracting Unexpected Scenarios of Embedded Systems, *Proc. of the 7th Joint Conf. on Knowledge-Based Software Eng.*, pp. 41 - 50, 2006.
16) 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為の非正常系分析マトリクス, 情報処理学会研究報告「ソフトウェア工学」, No. 2004-SE-145, pp.113-119, 2004.
17) C. Rolland and C. Ben Achour: Guiding the Construction of Textual Use Case Specifications, *Data and Knowledge Eng. J.*, Vol. 25, No. 1-2, pp.125-160, 1998.
18) G. Sindre and A. L. Opdahl: Eliciting Security Requirements by Misuse Cases, *Proc. of the 37th Int. Conf. Technology of Object-Oriented Languages and Systems*, pp. 120-131, 2000.
19) S. Uchitel, J. Kramer, and J. Magee: Incremental Elaboration of Scenario-Based Specifications and Behavior Models Using Implied Scenarios, *ACM Trans. on Software Eng. and Methodology*, Vol. 13, No. 1, pp. 37-85, 2004.
20) P. Zave and M. Jackson: Four Dark Corners of Requirements Engineering, *ACM Trans. on Software Eng. and Methodology*, Vol. 6, No. 1, pp. 1-30, 1997.