ソフトウェア工学　136－5
（２００２．３．７）

シナリオの変換・統合とルールによる検証

張　紅輝，大西　淳

立命館大学 大学院理工学研究科 総合理工学専攻

本研究はシナリオ記述言語によってシナリオが書かれている事を前提とし，あるビューポイントから書かれたシナリオを別のビューポイントから書かれたシナリオに変換する手法，幾つかのビューポイントから書かれたシナリオ群を特定のビューポイントから書かれたシナリオに統合する手法，さらにルールを別に記述する事によってシナリオの時系列の矛盾や抜けを検証する手法について紹介する．これらの手法を用いてシナリオの抜けを検出し，より完成度の高いシナリオを作成し，ルールによってシナリオの妥当性確認を可能にしている．共通問題「国際会議のプログラム委員長の業務」にこれらの手法を適用した結果を述べる．

Translation and Integration Method of Scenarios and Scenario Verification by Rules

Honghui Zhang, Atsushi Ohnishi

Integrated Science and Engineering, Graduate School of Science and Engineering,
Ritsumeikan University

The authors have developed a language for describing scenarios with which simple action traces are embellished to include typed frames based on a simple case grammar of actions. With the preconceived notion of scenarios are written in the scenario description language, a method is proposed for translating a scenario written from a certain viewpoint into another scenario with a different viewpoint, and for integrating multiple scenarios described from different viewpoints into a more complete scenario with a designated viewpoint. Further, the authors have proposed a method to detect the inconsistency and lack of a scenario's events based on rules written with a rule description language. We illustrate our methods with the common problem "requirements of jobs of international conference program chair".

1　はじめに

シナリオはソフトウェアの開発において重要な役割を果たしている，それは主にソフトウェア振舞いの定義とユーザによる妥当性確認においてよく用いられる．特に，シナリオを用いたシステム開発支援のツールと手法の構築に関する様々な研究が行われてきた．具体的には，環境とシステムのインターフェイスを表すことによるソフトウェア振舞いの抽出と仕様化 [5]，システムのプロトタイプとの組合せによる要求抽出 [6]，シナリオ分析による目標抽出 [9] などを対象にしている．一方，ソフトウェア開発の現場で，ワークフローの分析，抽象モデルの構築の支援道具としてシナリオが利用される場合も多い [1]．欧州の CREWS プロジェクトチームは４つの欧州各国で 15 の異なるプロジェクトをシナリオに基づく要求工学の実践について調査しており，現地でのインタビューを行い，作成された文書や利用した環境やツールについて調べた．そしてその結果，シナリオの意図や利用は予測していた以上に幅広く異なっているという結論が得られた [1]．

以上をまとめると，システムが何をどうするかというソフトウェア工学の視点で，シナリオを作成し分析することにより，構築されるシステムの品質向上が期待される．従って，ソフトウェア工学分野でシナリオをさらに有効利用するために，それ自体が作成物であ

表 1: アクションフレームの例

アクション	格フレーム (* は任意格)	イベントの型
渡す，入れる，出力する，受け取る，入力される，入力する，渡る，渡される，配布する，受理する，通知する，送付する，出す，返す，伝える，等	動作主，源泉，目標，道具 *	送付

るシナリオを開発し維持していくための，より優れた管理やツールによる支援が必要である．CREWS プロジェクトはマルチメディア記録システムを用いた要求分析作業支援，自然言語理解による要求獲得支援などを対象にしている．初期の CREWS プロジェクトでは，シナリオを分類するためのフレームワークを定義した．ここでは，形式，内容，目的，ライフサイクルという４つの視点からシナリオを分類し，これを基準にしてシナリオに基づく各種の手法の比較を試みている [1]．

現在，シナリオに基づいた設計，開発においては，まだ解決しなければならない課題が数多くある．例えば，要求工学での主作業はユーザや顧客の要求を分析し，それを満足するようにシステム動作仕様を定義することである．そのため，ユーザの視点に重点を置いたシナリオを利用することが多い．シナリオの内容は書かれた時のユーザの視点によって変化するので，さまざまなユーザの視点ごとにシナリオを用意するのが難しく，またシナリオが妥当かどうかを見極めるのが困難である．すなわち，シナリオに基づくソフトウェア工学において,「どうすればシナリオ作成作業を軽減することができるのか？シナリオの質を向上させられるのか？」ということは重要な課題である．本稿では上記の課題の実現を目的としている．

2 アクションフレームとシナリオ記述言語

シナリオはユーザが目標を達成するために行う行動とそこから得られるイベントを時系列に沿って記述したものである．シナリオは，イベントを表すイベント文と，イベント間の時間的順序を表す制御文から構成される．

2.1 イベントの記述

我々は格文法 [2] に基づいた要求フレームモデル，要求言語，及びその処理系を開発してきた [3][4]．要求フレームモデル [4] は要求仕様の枠組を与えるものである．シナリオ中のイベントはアクションを含んでいる，これらのアクションも格文法に基づいて，その意味を格構造として定義することができる．

我々は要求フレームモデルに基づいたアクションフレームを提案している．シナリオ記述言語はこのアクションフレームに基づいている．アクションフレームではアクションを１２種のイベント型（送付，依頼，制御，AND 木，OR 木，検索，追加，更新，削除，ファイル処理，生成，チェック）のいずれかに分類する，また９個の格（動作主格，源泉格，目標格，道具格，目的格，検索キー格，操作格，基準格，行為格）を定義している．この中で，基準格はチェック型イベントのチェック基準オブジェクトを示す，また行為格は依頼型イベントの依頼される行為オブジェクトを示す．本稿でアクションフレームの全般を示すことは省略する，本稿でよく使うデータフローに関するアクションを表１に示す．

アクションフレームを使ってシナリオのイベントを記述する．各アクションは固有の格構造を持っている [4]．例えば「送付する」のアクションは必須格として「動作主格」,「源泉格」,「目標格」を持ち，任意格として「道具格」を持つ．もしシナリオ記述者がアクションフレームに用意されていないアクションを使いたい場合は，記述者に新しいアクションとそのイベント型の対応や格フレームを定義させることによって記述できるようになる．

またシナリオ中のイベント文は，アクションフレームに基づいて，イベントの型と格構造を表す内部表現に変換することによって解析される．例えば，イベント文「A さんは論文を郵便で学会に送付する」は,「送付する」というアクションが「送付」のイベント型に当たる，また送付対象となる「動作主格」，送付元となる「源泉格」，送付先となる「目標格」，送付に用いられる「道具格」の４つの格を持つものとして，格助詞を手がかりに解析される．結果として表２のような内部表現に変換される．

動詞を一つしか含まないような単文を内部表現に変換することによって必須格の欠落の検出ができる．また，格構造と文脈情報による欠落した格や代名詞の補

表 2: 内部表現の例

イベントの型	動作主	源泉	目標	道具
送付	論文	A さん	学会	郵便

完ができる．一方，このような単文の羅列は読みにくく，また書きにくいため，重文や複文（連体修飾節・主語節・対立節に限定している）で書かれた日本語の動詞を一つしか含まないような単文に分割してから，対応する内部表現に基づいて解析する [3]．これにより，シナリオを読みやすく，書きやすくしている．

2.2 イベント間の時間的順序

イベント間の時間的な順序は

- 順接（sequence）
- 選択（selection）
- 繰り返し（iteration）
- 並行 AND 分岐（and-fork）
- 並行 OR 分岐（or-fork）
- 並行 XOR 分岐（xor-fork）
- 並行同期終了（join）

を想定し，これらをシナリオ中に明示する必要がある．

イベント間の時間的順序は殆んど順接であるため，順接制御文は特には明示しない．並行するイベントでは optional な（生起してもしなくても，どちらでもよい）イベントを明示できる．例えば,「CFP を学会誌等への掲載を依頼する」と「CFP を学会員に配布する」のように，２つのイベントが共に起こる場合は，下記のように明示する．

AND-fork
・CFP を学会誌等への掲載を依頼する．
・CFP を学会員に配布する．
AND-join

また,「投稿者が e-mail アドレスをもつならば，受理通知を e-mail で出す，そうでなければ郵送で出す」のように，条件分岐は，下記のように明示する．

if(投稿者が e-mail アドレスをもつ)
then・投稿者に受理通知を e-mail で出す.
else・投稿者に受理通知を郵便で出す.
fi

このように，イベント間の７種類の時間的順序を明示できる．

今までは，曖昧性と非形式性のために，シナリオの分析と検証は困難であったが，シナリオ記述言語を使ってイベントとその間の時間的な順序を記述したシナリオの意味を正確に，一意に把握できるようになったことで，分析と検証の正確性を保証できる．

3 異なるビューポイント間のシナリオの変換と統合

3.1 シナリオの例：「国際会議のプログラム委員長の業務」

シナリオはユーザを中心とした記述である．従ってシナリオを書くことで，ユーザの視点をシステム設計に導入することができる．ここで特定のユーザから見た視点をビューポイントと言う．本稿では要求工学 WG での共通問題 [7][8]「国際会議のプログラム委員長の業務」を例として使い，シナリオ記述言語に基づいてシナリオを記述する．

3.1.1 プログラム委員長のビューポイントからのシナリオ

プログラム委員長のビューポイントから見たシナリオを下記に示す．C1），C2），…はプログラム委員長（program committee **C**hair）のビューポイントから見たシナリオのイベントの順番を表す．

C1）重要な日付からスケジュールを決める．

C2）スケジュール等により CFP を作成し，
AND-fork
・CFP を学会誌等への掲載を依頼し，
・CFP を学会員に配布する．
AND-join

C3）研究者からプログラム委員を選び，プログラム委員候補に委員の担当を依頼する．

C4）投稿論文に番号を割り当て，論文リストに投稿論文を登録する．

C5）if(投稿者が e-mail アドレスをもつ)
then・投稿者に受理通知を e-mail で出す.
else・投稿者に受理通知を郵便で出す.
fi

C6) XOR-fork

・論文リストを e-mail でプログラム委員に配布する.

・論文リストを FTP でプログラム委員に配布する.

XOR-join

C7) プログラム委員から担当希望を受け取り，委員の希望をもとに担当委員を決める.

C8) 担当委員に論文の査読を依頼し，担当委員に論文，査読用紙を送付する.

C9) 担当委員から査読結果を回収，査読結果により点数を集計する.

C10) …

3.1.2 プログラム委員のビューポイントからのシナリオ

プログラム委員のビューポイントから見たシナリオを下記に示す. M1)，M2)，…はプログラム委員（program committee **M**ember）のビューポイントから見たシナリオのイベントの順番を表す.

M1) プログラム委員長より研究者から選ばれ，プログラム委員長より委員の担当を依頼される.

M2) プログラム委員長に承諾の旨を返し，プログラム委員長に名前，所属等の情報を通知する.

M3) XOR-fork

・プログラム委員長より論文リストを e-mail で受理する.

・プログラム委員長より論文リストを FTP で受理する.

XOR-join

M4) プログラム委員長に担当希望を伝える．プログラム委員長によって委員の希望をもとに担当委員が決まる.

M5) プログラム委員長より論文の査読を依頼され，プログラム委員長から論文，査読用紙を受理する.

M6) 基準により論文を査読する.

M7) プログラム委員長に査読結果を通知する.

M8) 合計点をもとに論文採否結果をプログラム委員会で決定する.

M9) …

表 3: イベントの内部表現

イベントの型	動作主	源泉	目標	道具
送付	担当希望	プログラム委員	プログラム委員長	—

3.2 異なるビューポイント間のシナリオの変換

3．1に示したように，プログラム委員長の業務のシナリオはプログラム委員長のビューポイントからとプログラム委員のビューポイントからの両方の記述ができる．例えば,

(a) 「プログラム委員長がプログラム委員より担当希望を受け取る（プログラム委員長のビューポイント)」と

(b) 「プログラム委員がプログラム委員長に担当希望を伝える（プログラム委員のビューポイント)」

は同じ意味のイベントである．アクションフレームによると,「受け取る」アクションと「伝える」アクションは同じイベント型，同じ格構造を持っている．この二つのイベントを内部表現に変換すると，両方共表 3 のような内部表現に変換される．この内部表現を見ると，イベント型，格構造と格の内容だけを表現していて，特定のビューポイントを持っていない．しかし，このビューポイントに依存しない内部表現から「源泉格」に該当するビューポイントから見たイベントに変換すると (b) の文になる，または「目標格」に該当するビューポイントから見たイベントに変換すると (a) の文になる.

従って，あるビューポイントから書かれたイベントをビューポイントに依存しない内部表現に変換できる，またビューポイントに依存しない内部表現から特定のビューポイントから書かれたイベントにも変換できる．この変換規則によって，先の例の二つイベントはお互いに変換することが実現できる．例えば，まずイベント (a) を表 3 の内部表現に変換して，それから表 3 の内部表現からプログラム委員を主語にした文に変換すると，イベント (b) が得られる．また逆も可能である．ビューポイントが変わっても，イベントの時間的な順序は変化しないと考えられるので，イベント文のビューポイントを変換することにより，シナリオのビューポイントを変換できる.

この手法を共通問題の例に適用することによって，プログラム委員長のビューポイントから見たシナリオからプログラム委員のビューポイントのシナリオをある程度生成することができ，またその逆にもできる．従っ

て，３．１．１のシナリオからプログラム委員を含んだイベントを切り出すことによって下記に示すようにプログラム委員のビューポイントのシナリオを生成できる．CM1），CM2），…はプログラム委員長（program committee **C**hair）のビューポイントから見たシナリオから変換したプログラム委員（program committee **M**ember）のビューポイントのシナリオのイベントの順番を表す．

CM1）プログラム委員長より研究者から選ばれ，プログラム委員長より委員の担当を依頼される．

CM2）XOR-fork
・プログラム委員長より論文リストを e-mail で受理する．
・プログラム委員長より論文リストを FTP で受理する．
XOR-join

CM3）プログラム委員長に担当希望を伝える．プログラム委員長によって委員の希望をもとに担当委員が決まる．

CM4）プログラム委員長より論文の査読を依頼され，プログラム委員長から論文，査読用紙を受理する．

CM5）プログラム委員長に査読結果を通知する．

CM6）…

このシナリオをプログラム委員に妥当性確認してもらうこと，またはこのシナリオと３．１．２のシナリオを計算機によって比較することで，プログラム委員のビューポイントから見たシナリオのイベント M2，M6，M8 はプログラム委員長のビューポイントから見たシナリオから抜けているといったことが判明した．従って，ビューポイントを変えたシナリオを生成することにより，抜けたイベントも指摘できる．

3.3 異なるビューポイント間のシナリオの統合手法

アクションフレームに基づいたイベントの内部表現はビューポイントに依存していないので，図１のように異なるビューポイントからのシナリオを統合して，新しいシナリオを得ることができる．

３．１の二つのシナリオを統合し，下記のプログラム委員長のビューポイントからの新しいシナリオが生成できる．

C1）重要な日付からスケジュールを決める．

C2）スケジュール等により CFP を作成し，
AND-fork
・CFP を学会誌等への掲載を依頼し，
・CFP を学会員に配布する．
AND-join

C3,M1）研究者からプログラム委員を選び，プログラム委員候補に委員の担当を依頼する．

M2）プログラム委員から承諾の旨を受け，プログラム委員から名前，所属等の情報を受理する．

C4）投稿論文に番号を割り当て，論文リストに投稿論文を登録する．

C5）if(投稿者が e-mail アドレスをもつ)
then・投稿者に受理通知を e-mail で出す．
else・投稿者に受理通知を郵便で出す．
fi

C6,M3）XOR-fork
・論文リストを e-mail でプログラム委員に配布する．
・論文リストを FTP でプログラム委員に配布する．
XOR-join

C7,M4）プログラム委員から担当希望を受け取り，委員の希望をもとに担当委員を決める．

C8,M5）担当委員に論文の査読を依頼し，担当委員に論文，査読用紙を送付する．

M6）担当委員は基準により論文を査読する．

C9,M7）担当委員から査読結果を回収，査読結果により点数を集計する．

M8）合計点をもとに論文採否結果をプログラム委員会で決定する．

C10）…

この新しいシナリオは３．１のプログラム委員長のビューポイントからのシナリオより，抜けていたイベント M2，M6，M8 を補完した．

3.4 シナリオの変換と統合手法の評価

以上に示したように，シナリオの変換と統合手法による共通問題「国際会議のプログラム委員長の業務」に適用した結果として，まずプログラム委員長のビューポイントから見たシナリオからプログラム委員のビューポイントのシナリオを生成できた，また生成されたシ

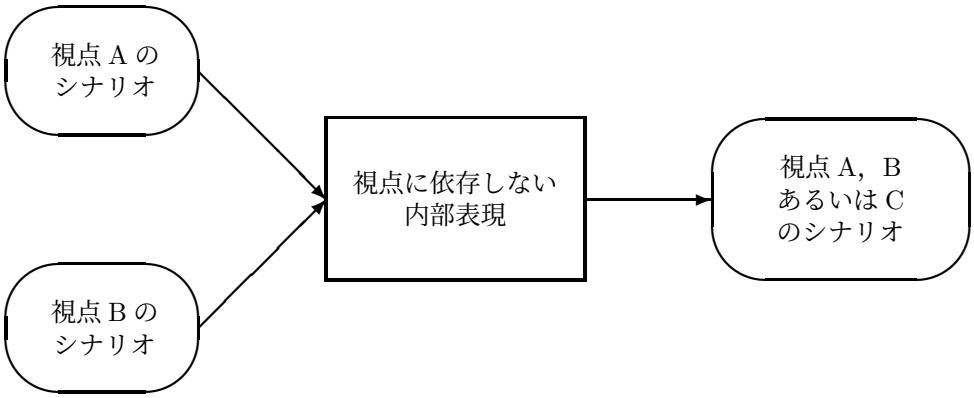

図 1: シナリオの統合

ナリオをプログラム委員に妥当性確認してもらうことで，プログラム委員長のシナリオの抜けを検出することができた．そしてプログラム委員長と委員のそれぞれのシナリオを合成し，そこからプログラム委員長のビューポイントより完成度の高いシナリオを生成した．

ユーザの視点をシステム設計に取り入れるため，異なるビューポイントのシナリオを作成するのは必要である．しかし，これまでのシナリオ作成は，異なるビューポイントの重複したシナリオを作成する手間が必要となっている，またシナリオ中のイベント間の矛盾や想定していなかったイベントの抜けも生じるなどの問題が存在している．これらの問題を解決することを目的にしているシナリオの変換と統合手法により，異なるビューポイントのシナリオの重複部分の作成工数を減らし，イベントの矛盾や抜けをある程度埋めることができると考えている．

4 ルールによるシナリオの検証

シナリオ中のイベント間の論理，時間関係をルールとして定義することによってシナリオの時系列の矛盾や抜けが検出できる．ルールはイベント間の関係を記述したものである．アクションフレームに定義しているイベントの型を利用して，シナリオ中のイベントとルール記述中のイベントとの対応を自動的にとることができる．

4.1 ルールの記述

ルールの記述文法はルールがどのような形をしているか（ルールの構文），そしてそれがなにを意味するのか（ルールの意味）を記述することによって定義される．ルールを表現するには，疑似言語を採用した．また付録に表しているように BNF 記法を用いてルールの構文を定義した．

共通問題の例で，プログラム委員長の業務は論文が中心になっている．それで,「論文受理」,「論文査読」,「論文採否決定」は必須な業務である，またこれらの業務が「CFP 配布」の後に発生するという一般的な常識がある．これを一つのルールとして考えると，下のルールを記述できる．

ルールの例 1
「CFP を送付するが起こる後に
プログラム委員長へ論文を送付する and
論文をチェックする and
論文採否結果を生成するが起こる」

ルールはシナリオ記述者が与える．先のルールは共通問題に特化したものである，しかも世の中の一般常識をルール化しておくことも考えられる．例えば，共通問題のようなインタラクティブシステムの一般化ルールとして，

ルールの例 2
「A から依頼するが起こる後に
A へ返事を送付するが起こる
if A へ返事を送付するが起こらない then
A から督促状を送付する or
A から依頼するが起こる」

を定義することができる．問題領域によってルールは異なるので，ドメインごとに常識を一般的なルールとして予め用意することにより，ルールの再利用ができ，シナリオ記述者がルールを定義することの面倒さも軽減できると考えている．

4.2 ルールによるシナリオの検証

ルールをあるシナリオが満たすかどうか調べることによって妥当性確認をすることが可能になる．イベン

表 4: 「CFP を送付する」の内部表現

イベントの型	動作主	源泉	目標	道具
送付	CFP	△△	○○	××

表 5: 「プログラム委員長へ論文を送付する」の内部表現

イベントの型	動作主	源泉	目標	道具
送付	論文	△△	プログラム委員長	××

トの内部表現による，シナリオ中のイベントとルール記述中のイベントとの対応をとることができる．また，シナリオ中でイベント間の時間的順序を表す制御文とルール記述中の時間助詞（例えば，後に，直後に，等）による，時間的な順序の対応をとることもできる．４．１で記述したルールの例１による，３．３で統合したプログラム委員長のビューポイントからのシナリオを検証すると，

（1）「CFP を送付する」の対応内部表現を表４に表す．シナリオ中の並行イベント
AND-fork
・学会誌等への掲載を依頼し，
・配布する．
AND-join
の「CFP を 配布する」はこの内部表現に対応する，この後に，

（2）「プログラム委員長へ論文を送付する」の対応内部表現を表５に表す．シナリオの中でこの内部表現に対応するイベントがない．

（3）「論文をチェックする」の対応内部表現を表６に表す．シナリオ中のイベント「担当委員は基準により論文を査読する」はこの内部表現に対応する．

（4）「論文採否結果を生成する」の対応内部表現を表７に表す．シナリオ中のイベント「合計点をもとに論文採否結果をプログラム委員会で決定する」はこの内部表現に対応する．

表 6: 「論文をチェックする」の内部表現

イベントの型	動作主	源泉	基準
チェック	△△	論文	○○

表 7: 「論文採否結果を生成する」の内部表現

イベントの型	動作主	源泉	目標
生成	△△	○○	論文採否結果

イベント「CFP を 配布する」の後に,「プログラム委員長へ論文を送付する」というイベントがシナリオから抜けていることが判明した.

4.3 ルールによるシナリオ検証手法の評価

以上に示したように，３．３で統合したプログラム委員長のビューポイントのシナリオをルールに照らし合わせることにより，シナリオの時系列の矛盾や抜けを検出する，そしてこれらを修正，補完してユーザの要求と整合性の高いシナリオを作成する事ができる．利用者はルールによってシナリオの妥当性検証が可能になり，より完成度の高いシナリオを作ることができ，機能や処理の抜けの検出もできる．すなわち，ルールによるシナリオ検証手法を利用してシナリオの品質を向上させることができる．

しかし，ルールによる検証については，まだ幾つかの問題点が存在している，例えば，ルール「CFP の送付が起こる後に…」のようなイベント間の時間関係は検証できるが,「4 週間内に査読する」のような定量的な時間に関する検証ができない；イベントの順序やイベントの存在は検証できるが,「全ての論文は査読される」,「全ての著者に受理通知を送る」といった量的な表現を含むルールの検証が難しい．またシナリオ中のループに対する検証も困難である．

5 終わりに

本稿で提案したシナリオ記述言語，シナリオの変換と統合手法，及びルールによるシナリオの検証手法はシナリオを用いてソフトウェア開発を進める分野で利用できる．これらの手法を利用して，異なるビューポイントからのシナリオの作成，矛盾や抜けの発見，シナリオの妥当性の検証などを支援できる．共通問題「国際会議のプログラム委員長の業務」に適用した結果によるこれらの手法の有効性を証明できる．今後の課題として，まずルール定義と検証の問題点を解決して，それからシナリオ記述言語とルール記述言語の処理系の開発と評価を行いたいと考えている．

謝辞　本研究に協力してくれた，大学院生の池中慎人，浜岡洋一，藤本宏の諸君に感謝します.

参考文献

[1] K.Weidenhaupt,K.Pohl,M.Jarke,and P.Haumer.: Scenario Usage in System Development:A report on Current Practice, *Proc. of ICRE Third International Conference on Requirements Engineering*, pp.222 (1998).

[2] Fillmore, C.J.: The Case for Case, in Bach and Harms (Eds.), Universals in Linguistic Theory.

[3] 大西　淳：要求定義のためのコミュニケーションモデル，情報処理学会論文誌 33 巻 8 号，pp.1064-1071 (1992).

[4] Ohnishi, A.: Software Requirements Specification Database Based　on Requirements Frame Model, *Proc. of the IEEE second International Conference on Requirements Engineering (ICRE'96)*, pp.221-228 (1996).

[5] J.C.S.P.Leite, G.Rossi, F.Balaguer, V.Maiorana, G.Kaplan, G.Hadad, and A.Oloveros.: Enhancing a requirements Baseline with Scenarios, *Proc. of the 3rd IEEE International Symposium on Requirements Engineering*, pp.44-53 (1997).

[6] A.Sutcliffe.: A Technique Combination Approach to Requirements Engineering, *Proc. of the 3rd IEEE International Symposium on Requirements Engineering*, pp.65-73 (1997).

[7] 大西　淳: 要求工学ワーキンググループ活動報告, 情報処理学会研究報告, Vol.2001, No.31, 2001-SE-130, ソフトウェア工学 130-18, pp.127-134 (2001).

[8] ACM Conference Committee job description, Conference Manual, Section 6.1.1, http://www.acm.org/sig_volunteer_info/conference_manual/6-1-1PC.HTM.

[9] C.Potts.: ScenIC:A Strategy for Inquiry-Driven Requirements Determination, *Proc. of the 4th IEEE International Symposium on Requirements Engineering*, pp.58-65 (1999).

付録　ルールの構文

```
＜ルール＞ ::= ＜イベント状態＞
             |＜ルール＞＜時間＞＜ルール＞
             |if　＜イベント状態＞
               then ＜ルール＞
             |if　＜イベント状態＞
               then ＜ルール＞
               else　＜ルール＞
             |＜集合＞＜助詞＞＜数＞
               {＜数＞} ＜状態＞
             |＜ルール＞ {, ＜ルール＞}
             |＜ルール＞　condition
               #＜条件＞ {, ＜条件＞} #
＜イベント状態＞ ::= ＜イベント状態１＞|
                     ＜イベント状態２＞
＜イベント状態１＞ ::= ＜イベント＞
                       ＜助詞＞＜状態＞
＜イベント状態２＞ ::= ＜イベント組合せ＞
                       ＜助詞＞＜状態＞
＜イベント組合せ＞ ::= ＜イベント＞
                   {＜論理演算子＞＜イベント＞}
＜集合＞ ::= ＜イベント＞ {, ＜イベント＞}
＜イベント＞ ::= {＜認定格＞＜助詞＞}
                 ＜イベント型＞＜能動受動＞
＜認定格＞ ::= ＜通常識別子＞
＜条件＞ ::= ＜格＞{＜比較演算子＞}{＜格＞}
＜イベント型＞ ::= 送付｜依頼｜制御｜AND 木｜
                   OR 木｜検索｜追加｜更新｜
                   削除｜ファイル処理｜
                   生成｜チェック｜
                   任意イベント｜
                   あるイベント
＜格＞ ::= 動作主｜源泉｜目標｜道具｜目的｜
           検索キー｜操作｜基準｜行為｜
           任意格｜ある格
＜助詞＞ ::= は|が|も|に|へ|の|あるいは|
             で|を|と|や|から|にの|まで|
              への|からの|までの|による|
              によって|により|および|または|
＜能動受動＞ ::= する｜される
＜状態＞ ::= 存在する｜存在しない｜
             起こる｜起こらない
＜時間＞ ::= 直後に｜後に｜直前に｜前に
＜論理演算子＞ ::= and｜or
＜比較演算子＞ ::= <>｜＝
＜数＞ ::= 0|1|2|3|4|5|6|7|8|9
＜通常識別子＞ ::=（英字，仮名，漢字，
                   下線で始まる英仮名漢字下線数字列）
```