一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
KBSE2012-79(2013-3)

ディペンダビリティケース用語構成規則の適用評価

松村　昌典† 　松野　　裕†† 　山本修一郎††

† 名古屋大学 工学部 〒464-8601 愛知県名古屋市千種区不老町
†† 名古屋大学 情報連携統括本部 情報戦略室 〒464-8601 愛知県名古屋市千種区不老町
E-mail: †matsumura.masanori@e.mbox.nagoya-u.ac.jp, ††{matsu,yamamotosui}@icts.nagoya-u.ac.jp

あらまし　ディペンダビリティケースで使用されている用語を統一的に構成する規則の有用性を評価するために実施した適用評価例とその考察について述べる．

キーワード　ディペンダビリティ，ディペンダビリティケース，用語関係図

A proposal on a rule to construct a word of Dependability Case

Masanori MATSUMURA†, Yutaka MATSUNO††, and Shuichiro YAMAMOTO††

† School of Engineering Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601 Japan
†† Strategy Office, Information and Communications Headquarters Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601 Japan
E-mail: †matsumura.masanori@e.mbox.nagoya-u.ac.jp, ††{matsu,yamamotosui}@icts.nagoya-u.ac.jp

Abstract　We reports examples of application evaluation and the discussion to evaluate usability of the rules to consistently identify words used in dependability cases.

Key words　Dependability, Dependability Case, Word Relationship Diagram

1. はじめに

システム開発者は，システムの開発時や運用時に，常に正しいサービスを提供するように努めるべきであり，もし障害が発生した場合はシステム利用者を満足するように対応する必要がある．しかし，システムがディペンダブル（可用性，信頼性，安全性，一貫性，保守性を満足していること）でないと，開発時や運用時に未知の障害が発生した場合にシステム利用者へ説明が不十分になり，システム開発者としての責任を果たせないことがある．このことから，システムのディペンダビリティを保証し説明責任を果たすことは，システム開発者にとって重要である．この手法の一つに，ディペンダビリティケース [3] [2]（Dependability case，アシュアランケース [1] とも呼ぶ）がある．

しかし，ディペンダビリティケースは，記述される人，物，活動等に対する関係性が明確ではないという問題がある．用語間の関係はディペンダビリティケースには記述しないが，用語間の関係を知らなければ，ディペンダビリティケースを記述は困難である．ディペンダビリティケースに記述される用語や用語間の関係を明確化するために，用語関係図 [4] [5]（Word Relationship Diagram）を用いてディペンダビリティ用語辞書を構築する．前回の発表 [5] では，ディペンダビリティケースから用語関係図への生成規則（D2W），用語関係図からディペンダビリティケースへの生成規則（W2D）を提案したが，十分な評価を行うまでは至らなかった．本稿では，ディペンダビリティケースで使用されている用語を統一的に構成する D2W 生成規則，W2D 生成規則の有用性を評価するために実施した適用評価例とその考察について述べる．

本稿の構成は次のとおりである．2 章では，用語関係図を説明する．3 章では，前回の発表で提案した生成規則の問題点を述べる．4 章では，3 章で挙げた問題の解決方法と修正した生成規則を説明する．5 章では，4 章で説明した生成規則を適用して結果を述べる．6 章では，本稿についての考察をと課題を記述し，最後にまとめを述べる．

2. 用語関係図

ディペンダビリティケースでは，記述される人，物，活動等に対する用語や用語間の関係を明確に記述しない．そこで，人，物，活動等を要素としてその要素間の関係をまとめた図である，用語関係図を作成する．ディペンダビリティケースの用語や用語間の関係を明確にすることができる．その結果，ディペンダビリティケースの記述や理解の助けとなる．

用語関係図の構成要素は表1で記述する3つである．人，物，活動等の書かれたノードに対し，ノード間に主要な関係があれば実線矢印を書き，その矢印に関係名を書く．各ノードには矢印がいくつか接続することができる．

表1　用語関係図要素の記述法

構成要素	用語関係図での記述法	文中内における記述法
ノード	X	ノード< X >
関係名	関係名	関係名【X】
矢印	→	矢印

3. 生成規則の問題点

前回の発表[5]では，ディペンダビリティケースから用語関係図への生成規則（D2W），用語関係図からディペンダビリティケースへの生成規則（W2D）を提案したが，以下のような課題と問題点がある．

3.1 D2W生成規則の様々なディペンダビリティケースノード文への未対応

ディペンダビリティケースに記述しているゴール文は，「システムはディペンダブルである.」のように安全性等を具体化するような記述（type P，ポジティブ性文章）や，「ハザードについて対策している.」のように危険性等を具体化するような記述（type N，ネガティブ性文章）と分類できる．しかしディペンダビリティケースでは，ゴール「自動列車運行装置（ATS）が設置されている.」のように，安全性，危険性ではなく，対策等を具体化して記述する場合がある．このようなゴール文に対する規則が存在していない．

また，ディペンダビリティケースに記述しているストラテジ文は，「想定リスクに基づき議論する.」のように分析結果に基づき議論を展開する記述や「プロセス毎に分けて確認する.」のように予め定義された要素に分解を展開する記述と分類できる．しかし，このようなストラテジ文に対するそれぞれの規則が存在していない．

3.2 D2W生成規則，W2D生成規則適用の不一致性

W2D生成規則を適用して，用語関係図から生成したディペンダビリティケースと元のディペンダビリティケースを比較することを行った．比較の結果，元のディペンダビリティケース内に，コンテクストを接続していないストラテジが存在することによって，生成したディペンダビリティケースが不適切な形になり，用語関係図内の一部のノードと不適切な関係名やノードが接続することが起きた．この原因について述べる．

D2W生成規則では，あるストラテジにコンテクストが接続されていなければ，用語関係図もコンテクストの用語に相当するノードと関係名は存在しない．つまり，ストラテジの分解理由が記述されていないということになり，その部分の用語や用語間の関係が曖昧になる．例えば，ストラテジ「プロセスを分解して議論する.」にコンテクスト「プロセス定義表」が存在しない場合を考える．この時，ストラテジに接続している下位ゴールが「メッセージを適切に受け取れる.」だと，＜プロセス＞と＜メッセージ受信＞がどのような関係であるかは，分野知識がないと理解が難しい．

3.3 D2W生成規則のパターン分類方法

前回の発表[5]では，ディペンダビリティケースに記述しているノードを，D2W生成規則のパターンとして，各部位へ分類した．しかし，分類方法は未定義であり，どのように各ノード部位に分類するか，複数のパターンに合致する部位ではどちらのパターンとして分類するのかといった問題が生じた．

4. 適用規則

3章では，生成規則の問題点を示した．D2W生成規則，W2D生成規則を修正した．4章では，問題点を解決するために生成規則を修正した．

4.1 規則の修正箇所

3章で挙げた3つの問題を以下の方法で解決する．

4.1.1 D2W生成規則の様々なディペンダビリティケースノード文への対応

3.1.1で挙げた問題を解決するために，ディペンダビリティケースに記述しているゴール文を以下の3つのタイプに分類し，各規則に従って用語関係図を生成する．

- 安全性を具体化できる内容（type P）
- 危険性を具体化できる内容（type N）
- 対策を具体化できる内容（type C）

また，3.1.3で挙げた問題を解決するために，ディペンダビリティケースに記述しているストラテジ文を以下の2つのタイプに分類し，各規則に従って用語関係図を作成する．

- 分析結果に基づき議論を展開する内容
- 予め定義された要素に分解を展開する内容

4.1.2 D2W生成規則における空ノード追加

また，3.1.2で挙げた問題を解決するために，用語関係図に空ノードを生成する規則を導入する．コンテクスト等が無い場合，該当箇所に空ノードを作成しておき，用語関係図を生成していく．最後に周りのノードや関係名から，空ノードに入るべき用語を推測し，用語や用語間の関係を明確に記述する．

4.1.3 D2W生成規則のパターン分類方法の定義

ディペンダビリティケースに記述しているノードのパターンを，各部位へ以下のように分類する．これらに分類したのち，それぞれの生成規則を適用する．

パターン1　議論展開部パターン

ストラテジ文「システムはディペンダブルである.」のように，分析結果に基づき議論を展開する内容で記述されている．1つのゴールと下位に接続しているストラテジ（存在すれば，ゴールとストラテジに接続しているコンテクスト）の部分

パターン2　要素分解部パターン

ストラテジ文「システムの構成要素について分ける.」のように，予め定義された要素に分解を展開する内容で記述されている．1つのゴールと下位に接続しているストラテジ（存在すれば，ゴールとストラテジに接続しているコンテクスト），ストラテジの下位に接続している0個以上のゴールの部分

パターン3　証拠部パターン

1つのゴールと下位に接続しているエビデンス（存在すれば，

ゴールに接続しているコンテクスト）の部分

パターン4　存在部パターン

ゴール文が「～は存在する」,「～は確認できる」のように，ゴール文の用語が存在する内容で記述されている．該当するゴールと上位に接続しているストラテジ (存在すれば，ゴールに接続しているコンテクスト) の部分

4.2 生成規則

4.2.1 D2W 生成規則

- 議論展開部パターン

(1) ゴールの用語「G」を抽出し，ノード＜ G ＞とする．ゴール文が Type P であれば，(2.1) へ進む．ゴール文が Type N であれば，(2.2) へ進む．ゴール文が Type C であれば，(2.3) へ進む

(2.1) ノード＜ G ＞からノード＜分析＞を関係名【関係対象】で接続する．また，ノード＜危険な G ＞を関係名【状態制約】で接続する．(3) へ進む．

(2.2) 用語「G」から形容部分を取り除いたノード＜ G' ＞を生成し，ノード＜ G' ＞からノード＜ G ＞を関係名【状態制約】で接続する．ノード＜ G' ＞からノード＜分析＞を関係名【関係対象】で接続する．(3) へ進む．

(2.3) 空ノード＜ CM ＞からノード＜ G ＞を関係名【対策】で接続する．(2.1) へ進む．

(3) ゴールのコンテクストが存在すれば，コンテクストの用語「G-C」を抽出し，ノード＜ G-C ＞とする．ノード＜ G-C ＞からノード＜ G ＞を関係名【前提条件】で接続する．

(4) ストラテジにコンテクストが存在すれば，コンテクストの用語「S-C」を抽出し，ノード＜ S-C ＞とする．ストラテジにコンテクストが存在しなければ，空ノード＜ S-C ＞を作成する．ノード＜分析＞から，ノード＜ C ＞を関係名【関係結果】で接続する．また，ノード＜ X ＞から，ノード＜ S-C ＞を関係名【一覧化（変換)】で接続する．

(5) 空ノードがあれば，用語関係図のノードと関係名からノードを推測し，記述する．

(6) ノード＜ S-C ＞から関係名【具体化】を接続する．

- 要素分解部パターン

(1) ストラテジを矢印で指しているゴール（以下，上位ゴール）の用語「parentG」を抽出し，ノード＜ parentG ＞とする．

(2) ノード＜ parentG ＞からノード＜ parentG 構成要素＞を関係名【分解】で接続する．

(3) 上位ゴールに構成要素が記述されているコンテクストが接続されていれば，コンテクストの用語「G-C」を抽出し，ノード＜ G-C ＞とする．ノード＜ G-C ＞からノード＜ parentG ＞を関係名【前提条件】で接続する．上位ゴールのコンテクストが存在しなければ，ノード＜構成要素一覧＞からノード＜ parentG ＞を関係名【前提条件】で接続する．

(4) ストラテジから矢印で指されているすべてのゴール（以下，下位ゴール）の用語「childG」を抽出し，それぞれノード＜ childG ＞とする．

(5) ノード＜ parentG 構成要素＞からすべてのノード＜ childG ＞を関係名【具体化】で接続する．

- 証拠部パターン

(1) ゴールの用語「G」を抽出し，ノード＜ G ＞とする．ゴール文が Type P であれば，(2.1) へ進む．ゴール文が Type N であれば，(2.2) へ進む．ゴール文が Type C であれば，(2.3) へ進む

(2.1) ノード＜ G ＞からノード＜分析＞を関係名【関係対象】で接続する．また，ノード＜危険な G ＞を関係名【状態制約】で接続する．(3) へ進む．

(2.2) 用語「G」から形容部分を取り除いたノード＜ G' ＞を生成し，ノード＜ G' ＞からノード＜ G ＞を関係名【状態制約】で接続する．ノード＜ G' ＞からノード＜分析＞を関係名【関係対象】で接続する．(3) へ進む．

(2.3) 空ノード＜ CM ＞からノード＜ G ＞を関係名【対策】で接続する．(2.1) へ進む．

(3) ゴールにコンテクストが存在すれば，(4) へ進む．そうでなければ，(5) へ進む

(4) コンテクストの用語「C」を抽出して，ノード＜ C ＞とする．ノード＜ G ＞（または，ノード＜危険な G ＞）からノード＜分析＞を関係名【関係対象】で接続し，ノード＜分析＞からノード＜ C ＞を関係名【関係結果】で接続した後，ノード＜ G ＞からノード＜ C ＞を関係名【一覧化（変換)】で接続する．(5) へ進む．

(5) ノード＜ C ＞（ノード＜ C ＞が存在しなければ，ノード＜ G ＞）から，ノード＜ G ＞の対策を空ノード＜ A ＞とし，関係名【対策】で接続する．

(6) ゴールに接続しているエビデンスの用語「E」を抽出する．空ノード＜ A ＞からノード＜ E ＞を関係名【証拠】で接続する．

(7) 空ノードがあれば，用語関係図のノードと関係名からノードを推測し，記述する．

- 存在部パターン

(1) 存在部パターン中のすべてのゴールから用語「X」を抽出し，ノード＜ X ＞とする．

(2) ストラテジの用語「S」を抽出し，ノード＜ S ＞とする．

(3) ノード＜ X ＞から矢印を示していき，ノード＜ S ＞までたどりつくまで (3.1) を行う．

(3.1) あるノード＜ A ＞に対し，その関係のあるノード＜ B ＞を推測し，その用語とノード A を適当な関係名で接続する．

4.2.2 W2D 生成規則

(1) 用語関係図からトップゴールを定める．トップノードを用いて，「X はディペンダブルである.」という命題文に書き換え，ディペンダビリティケースのトップゴールとする．

(2) 用語関係図から関係名【一覧化（変換)】,【分解】を分割探索する．関係名【一覧化（変換)】を発見した場合 (3.1)，関係名【分解】を発見した場合 (3.2) へ進む．

(3.1) 関係名【一覧化（変換)】の矢印が示しているノード＜ Y ＞を用いて，「Y に基づき，分けて議論する.」

というストラテジを一番新しく作成したディペンダビリティケースの上位ゴールに接続する．そのストラテジにコンテクスト「Y」を接続する．もし，ノード＜Y＞がない，もしくはノード＜Y＞が示しているノードがなければ，代わりに一番新しく作成した上位ゴールにアンデベロップドを接続する．(4) へ進む．

(3.2) 関係名【分解】の矢印が示しているノード＜Y＞を用いて，「Y に基づき，分けて議論する．」というストラテジを一番新しく作成したディペンダビリティケースの上位ゴールに接続する．(4) へ進む．

(4) ノード＜Y＞が示している 1 つ以上のノード＜Z＞を用いて，「Z について対策している．」という命題文に書き換え，ディペンダビリティケースのサブゴールとする．もし，ノード＜Y＞が示しているノードがなければ，代わりにアンデベロップドを接続する．

(5) ノード＜Y＞を示している矢印があれば，それらすべてのノード＜A＞において，「A において適切である（存在する）．」というサブゴールをさらに付加する．もし，あるノード＜A＞から，探索した際に通った矢印でない矢印が出ていれば，(1) へ進む．それ以外はアンデベロップドを接続して，(6) へ進む．

(6) 関係名【対策】を発見するまで (2) - (5) を繰り返す．

(7) 関係名【対策】を発見した場合はさらに探索していき，関係名【証拠】の矢印が示しているノード＜E＞を用いて，「E」というエビデンスを最近作成したディペンダビリティケースの上位ゴールに接続する．

5. 評価検証

5.1 方　　法

4 章で示した W2D 生成規則，D2W 生成規則を評価するために，以下の手順で生成規則の適用を行う．

(1) ディペンダビリティケースを W2D 生成規則を適用し，用語関係図を作成する．

(2) 作成した用語関係図を W2D 生成規則を適用し，ディペンダビリティケースを作成する．

(3) 元のディペンダビリティケースと作成したディペンダビリティケースを以下の点で比較する．

【1】 全体の形

【2】 ノードの内容

5.2 対　　象

対象は，以下の 2 つのディペンダビリティケース（図 1，図 2）である．

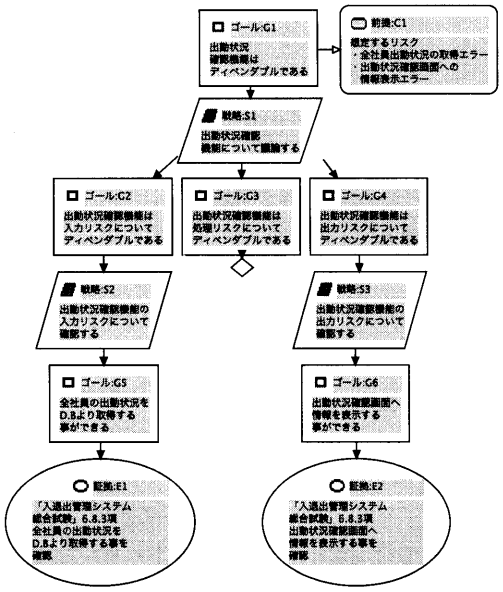


図 1　対象 1:入退出管理システム「出勤状況確認機能」

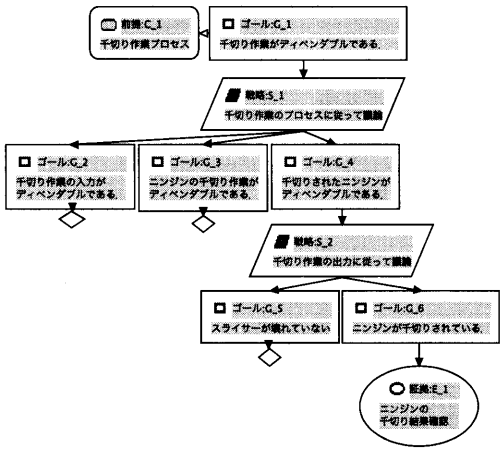


図 2　対象 2:ニンジンの千切りプロセス

5.3 結　　果

図 1，付図 3 と，図 2，付図 4 のディペンダビリティケースを比較した．結果は以下の通りである．

【1】 全体の形

適用前のディペンダビリティケースより，適用後のディペンダビリティケースの方がコンテクストノード数は増加した．適用前に記述しているストラテジ部分にコンテクトに 1 つずつ接続するようになった．

【2】 ノードの内容

適用前のノード内容より，適用後のノード内容が曖昧に記述されるようになった．内容が曖昧になったノードはゴール，ストラテジである．

また，ゴールノードとストラテジノードの内容が一部変化した．表 2 のように，対象 1 での内容が変化したゴールノードを挙げている．

表 2　対象 1：ゴールノード内容に対する適用前後の比較

適用前	適用後
出勤状況確認機能は入力リスクについてディペンダブルである.	入力リスクについて対策している.
出勤状況確認機能は処理リスクについてディペンダブルである.	処理リスクについて対策している.
出勤状況確認機能は出力リスクについてディペンダブルである.	出力リスクについて対策している.
会社員の出勤状況を D.B より取得する事ができる.	出勤状況データ取得失敗について対策している.
出勤状況確認画面へ情報を表示する事ができる.	出力状況確認画面への情報表示について対策している.

6. 考　　察

評価検証を行った結果，ディペンダビリティに記述している用語や用語関係を明確にすることで，ディペンダビリティケースに記述していないノードを発見することができ，用語関係図を用いることは，ディペンダビリティケースの整形に有用である．一方，用語関係図から生成したディペンダビリティケースノードの内容が曖昧になるため，ノードの内容が適当であるか確認することが必要となる．これに基づき，考察と今後の課題を述べる．

6.1　課題 1:ディペンダビリティケースノード記述の自由度

ディペンダビリティケースはゴールとストラテジのノードにはいろいろな形で記述することが可能である．例えばゴールノードは,「システムはディペンダブルである.」といったディペンダブルであるかという記述,「システムリスクについて対策している.」といったリスク対策ができているかという記述,「サーバーシステムのバックアップを行っている.」といったリスク対策を明記している記述など，命題文であれば自由に記述可能である．しかし，このようにゴール内容の文構造を統一しなければ，ディペンダブルを確認できているか分かりづらい．

ここで用語関係図を用いて，ディペンダビリティケースに記述している用語や用語間の関係を明確にし，記述内容の意味を保ったまま文を変換できる．例えば，表 2 のように，すべてゴール内容を「リスクに対策している.」といった変換が可能となる．

現段階では，ゴールノード内容の変換パターンがどの程度存在するのかが不明であるため，今後も評価検証を行っていく必要がある．

6.2　課題 2:D2W 生成規則でのディペンダビリティケースノードから用語切出し規則の提案

D2W 生成規則では，ディペンダビリティケースのゴールノード，ストラテジノードから重要な用語を一つ切り出して，用語関係図のノードにしていた．現段階ではその切出し方法は「原則として，文中に記述されている名詞」としていた．しかしこの規則のみで切出しを行うと，一部のディペンダビリティケースから用語関係図を生成できない場合があった．そこでその原因を明確にし，文から適切な用語を切出す規則を提案する必要がある．

6.3　課題 3:用語関係図の関係名【状態制約】の種類

用語関係図の関係名【状態制約】では，ある対象に対して分析を行い，そこでノード＜○○な対象＞という用語を生成している．しかし，分析の種類（妥当性分析や安全性分析など）によって，生成するノード（妥当なシステム，危険なシステムなど）は変化する．ここで，生成するノードがどのような種類があるのかを調べる必要がある．

6.4　課題 4:規模が十分大きいシステムへの生成規則適用

生成規則を適用した対象が規模の小さいシステムであり，十分大きいシステムに対して，生成規則を適用した結果がどのようになるのか不明である．このことから十分大きいシステムでの生成規則の適用を行う必要がある．

6.5　課題 5:生成規則における自動生成可能箇所の分析

現在，用語関係図・ディペンダビリティケース間の自動生成ツールの作成を目指しているが，ディペンダビリティケースのゴールノード内容のタイプ判断等が自動化できるか不明である．どの部分を手動で入力し，どの部分が自動生成できるかの分析を行う必要がある．

7. おわりに

本稿では，前回の発表で提案したディペンダビリティケースから用語関係図への生成規則（D2W）と，用語関係図からディペンダビリティケースへの生成規則（W2D）の適用評価を行った．これによりディペンダビリティケースの整形や，ディペンダビリティケースに記述されていないノードの追加に有用であること言える．また，以下の課題があることが分かった．

- ディペンダビリティケースノード記述の自由度
- D2W 生成規則でのディペンダビリティケースノードから用語切出し規則の提案
- 用語関係図の関係名【状態制約】の種類
- 規模が十分大きいシステムへの生成規則適用
- 生成規則における自動生成可能箇所の分析

謝　　辞

本研究は JST-CREST「実用化を目指した組み込みシステム用ディペンダブル・オペレーティングシステム」研究領域（DEOS プロジェクト）の支援を受けたものである．

文　　献

[1] Peter Bishop, Robin Bloomfield, Sofia Guerra, The future of goal-based assurance cases, DSN, 2004.

[2] Tim Kelly and Rob Weaver. The goal structuring notation - a safety argument notation. In Proc. of the Dependable Systems and Networks 2004, Workshop on Assurance Cases, 2004.

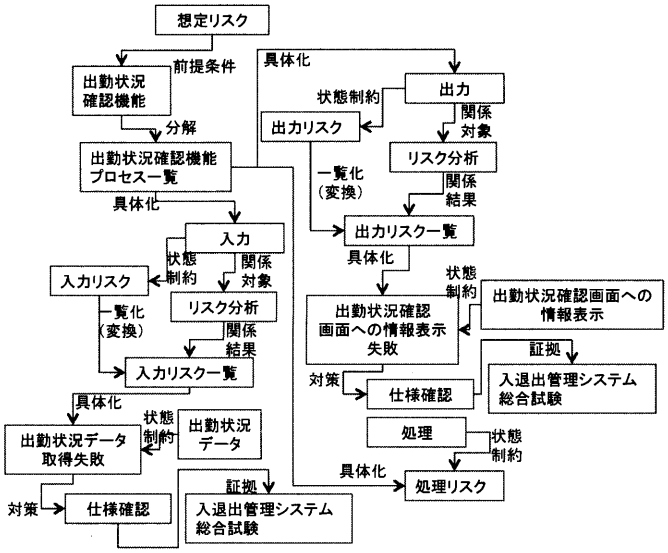
[3] 松野裕, 高井利憲, 山本修一郎. D-Case 入門 ～ディペンダビリティ・ケースを書いてみよう！～. 株式会社ダイテックホールディング, 2012. ISBN: 978-4-86293-079-8.

[4] 松村昌典, 松野 裕, 山本修一郎, ディペンダビリティ用語辞書構築方法の提案, 信学技報, vol. 112, no. 314, KBSE2012-57, pp. 115-120, 2012 年 11 月.

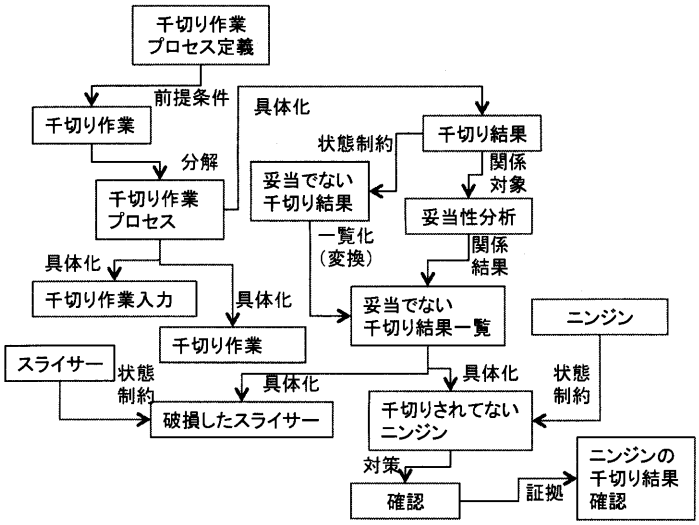
[5] 松村昌典, 松野 裕, 山本修一郎, ディペンダビリティケース用語構成規則の提案, 信学技報, vol. 112, no. 419, KBSE2012-63, pp. 29-34, 2013 年 1 月.

[6] DEOS プロジェクト http://www.crest-os.jst.go.jp

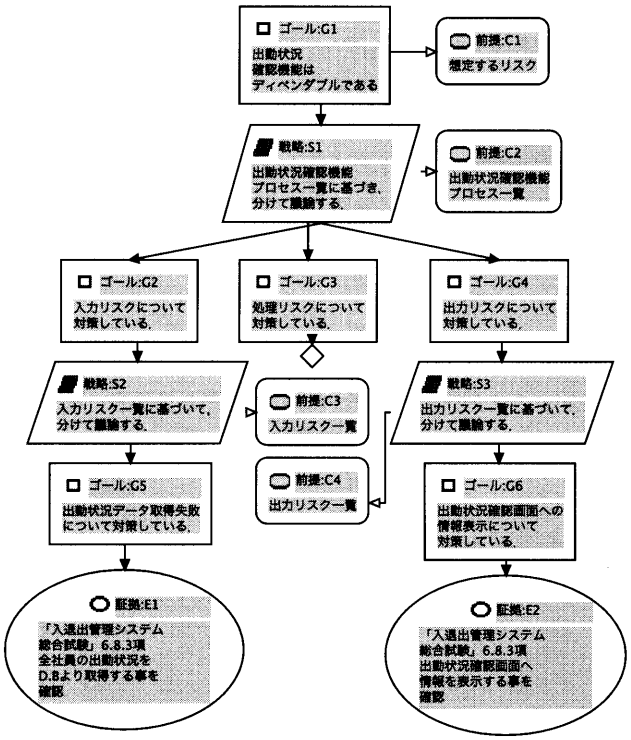
[7] D-Case Editor
http://www.dependable-os.net/tech/D-CaseEditor/
[8] 山本修一郎，要求工学基礎知識，名古屋大学情報連携統括本部情報戦略室，2012



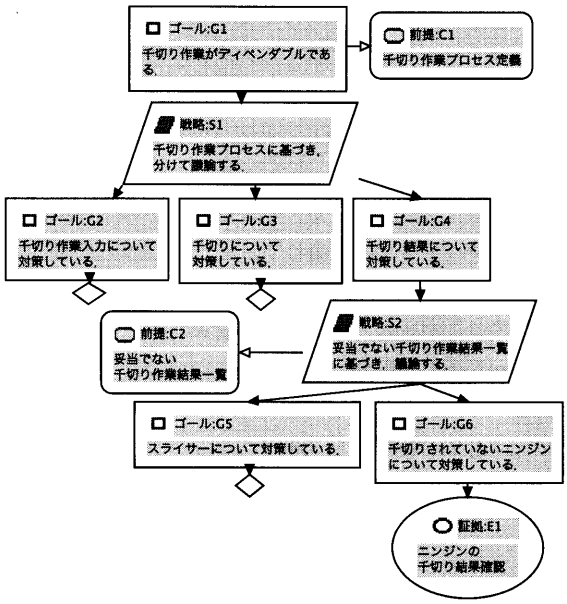
付図 1 D2W 生成規則を適用生成した対象 1 の用語関係図



付図 2 D2W 生成規則を適用生成した対象 2 の用語関係図



付図 3 W2D 生成規則を適用生成した対象 1 のディペンダビリティケース



付図 4 W2D 生成規則を適用生成した対象 2 のディペンダビリティケース