D-Case

ディペンダビリティ合意形成のための手法とツール

DEOS プロジェクト*
倉光研究チーム　松野・恩田・山本研究グループ

平成 25 年 5 月 1 日

DEOS-FY2013-DC-02J

* 「実用化を目指した組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」（DEOS プロジェクト）は科学技術振興機構（JST）の戦略的創造研究推進事業 CREST の研究領域のひとつです。

1　はじめに

DEOS プロジェクトで研究開発中の D-Case 手法 [7, 17, 12, 10] とツール [9, 2]、および D-Case の応用例をご紹介いたします。

D-Case とは、システムのディペンダビリティをシステムに関わる人たち（ステークホルダ）が共有し互いに分かり合い、そのディペンダビリティを社会の人々にわかってもらい、説明責任を果たすための手法とツールです。図 1 の DEOS プロセスの中で D-Script と一緒にオープンシステム・ディペンダビリティを実現するために研究開発を行なっています。

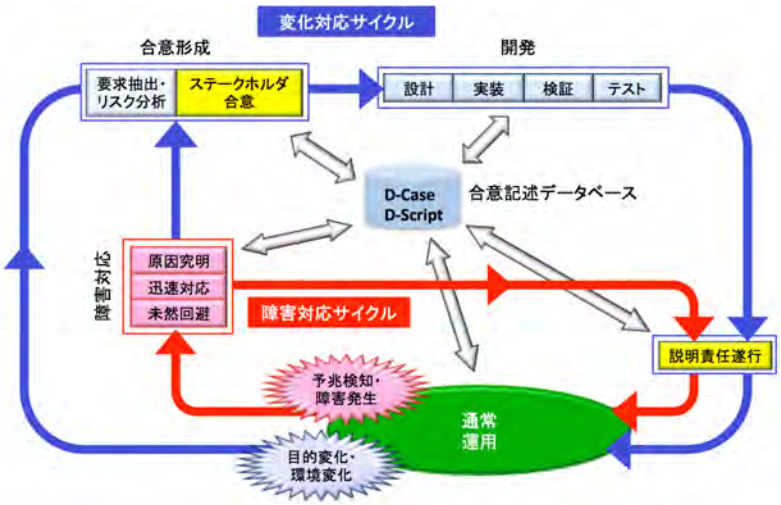


図 1　DEOS プロセス

D-Case は、欧米で近年高い安全性が要求されるシステムの開発運用において、提出が義務付けられるまでに普及している安全性ケース (Safety Case) [3] をもとに研究開発が始まりました。安全性ケースとは、テスト結果や検証結果をエビデンスとしてそれらを根拠にシステムの安全性を議論し、システム認証者や利用者などに保証する、あるいは確信させる (assure) ためのドキュメントです。近年、安全性だけでなくセキュリティやディペンダビリティなども対象として使われ始めています。その場合、セキュリティケースや、ディペンダビリティケースと呼ばれます。これらを総称してアシュアランスケースと呼ばれます。安全性ケースにおいて用いられるエビデンスを元にした議論、保証という考え方は、オープンシステムのディペンダビリティを達成するために重要であると考え、安全性ケースをオープンシステム・ディペンダビリティの考え方で発展させ、D-Case が生まれました。現在の安全性ケースでは、システム供給者、第 3 者コンサルティグ会社、システム利用者（国防省など）の間のコミュニーケーションなどに使われますが、主には認証のために提出されるドキュメントとして使われてきました。D-Script と一緒になって、開発・運用を通して、多くのステークホルダやシステムと連携してオープンシステム・ディペンダビリティ [14] の達成に貢献するためには、多くのチャレンジがあります。D-Script や D-RE などとの連携のための基礎研

究 ([11, 15, 8] など) と開発に加えて、特に、企業の方との議論や記述実験を通して、以下の 3 点が実用化のために重要であると考えました。

1. 一般の企業の方にとってわかりやすい入門書や講習の開発
   安全性ケースはこれまで高い安全性が求められるシステムに対して、高度な専門知識を持つコンサルタントなどによって書かれてきました。そのため、安全性ケースのガイドブックなどは、安全性分析など高い専門知識を前提とされたものがあるだけでした。オープンシステムのディペンダビリティは、一般の企業の多くの方が参加されなければ達成できません。そのためには、わかりやすい入門書や講習が必要であると考えました。
2. 一般の企業の方にとって使いやすい、ニーズに即したツールの開発
   安全性ケースが普及し始めてまだ日が浅いこともあってか、ツールはイギリス Adelard 社の ASCE ツール[*1]などいくつかあるだけです。また ASCE ツールなどは、主に認証ドキュメントを作成するためのツールであり、他の開発ツールとの連携が容易ではなく、企業の方の実際のニーズに即したツールにはまだなっていません。D-Script や D-RE との連携に加えて、企業の方が使いやすい、ニーズに即したツールが必要であると考えました。
3. 記述、応用例の充実
   安全性ケースの問題の一つは、企業の重要な情報を含むことから、なかなか実際の例が表に出てこない点があります。しかしそれでは一般の、特に日本企業の方に具体的なイメージを持っていただくことは困難です。わかりやすく、具体的な記述例や応用例が必要であると考えました。

D-Case チームは、これまでの基礎研究・開発に加えて、上記 3 点に着眼し、広く企業の方に使っていただけるための活動を本格化しています。その一環として、2012 年 9 月 14 日に、名古屋で第 1 回 D-Case 実証評価研究会を開催しました。30 名以上の企業の方にご参加いただき、活発な議論ができました (図 2 がその様子です)。ご興味のある方は、ぜひご参加いただけたら大変幸いです。D-Case のホームページ[*2]をぜひ御覧ください。

2 節では D-Case について、その必要性、記述ルールと記述ステップ、簡単な記述例、さらに D-Case10 箇条、3 節では D-Case ツールをご紹介します。4 節では ET ロボコン南関東大会で最優秀賞 (エクセレント・モデル) を受賞した D-Case の応用事例をご紹介します。

2　D-Case について

D-Case はディペンダビリティ合意形成の議論のための手法およびツールです。開発、運用を通じて活用されます。ライフサイクルの各フェーズで生成されるドキュメントを元にして、GSN(Goal Structuring Notation) [6] と呼ばれる安全性ケースの表記法の一種を拡張した表記法によって基本構

[*1] http://www.adelard.com/asce/choosing-asce/index.html
[*2] http://www.dcase.jp

図 2　D-Case 実証評価研究会の様子

造を記述します。図 3 に D-Case の例を示します。

2.1　D-Case の必要性

ディペンダビリティとは，「アベイラビリティ (可用性) 性能及びこれに影響を与える要因，すなわち信頼性性能，保全性性能及び保全支援能力を記述するために用いられる包括的な用語である」と，JIS Z8115 (2000) で定義されています。また、Avizienis らはシステムのディペンダビリティを以下の 5 特性で定義しています [1]。

- 可用性: 正しいサービスを提供できること
- 信頼性: 正しいサービスを持続できること
- 安全性: ユーザと環境に破滅的な事態を生じさせないこと
- 一貫性: 不適切な変更がないこと
- 保守性: 修理・修正できる能力

Jackson は，ディパンダブル・システムのソフトウェアには，明示的主張 (Explicit claims)，証拠 (Evidence)，知識 (Expertise) の 3 つが必要であると指摘しています [5]．この理由は，システムがディペンダブルであるというためには具体的な性質を明示的に主張することと，ディペンダビリティの主張を支持する証拠を提示する必要があるからです。システムが実行条件下で重要なディペンダビリティ要求を満足することを示すために、ディペンダビリティケース (D-Case) が必要とな

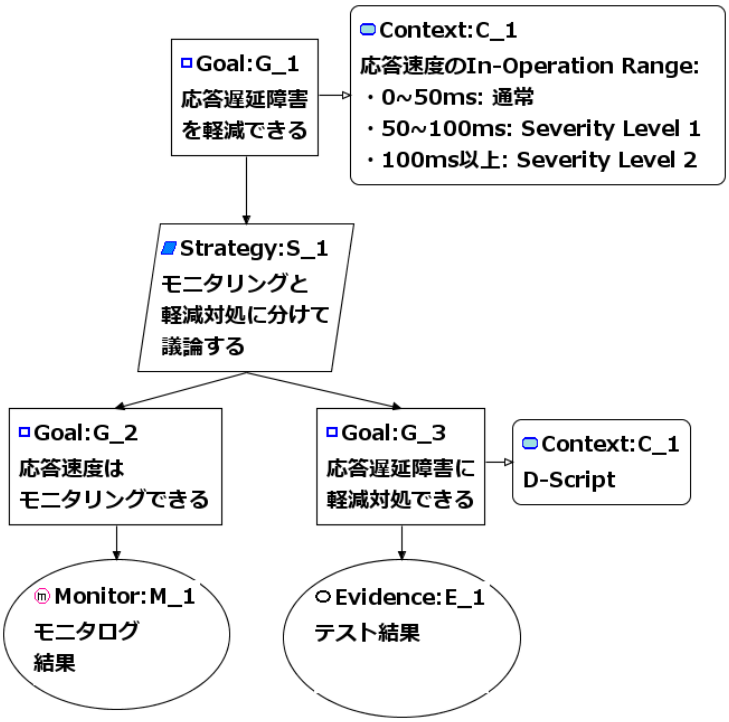


図 3　D-Case の例

ります。

D-Case の必要性を規格からの要請、用途、期待効果の観点から説明します。

1. 規格からの要請

   ISO 26262 は自動車の電気/電子に関する機能安全についての国際規格です。Part10 [4] では，機能安全についてのガイドラインが説明されています。このガイドラインの 5.3 節「安全性ケースについて理解する」で，安全性ケースの表記法として，GSN と CAE(Claims-Argument- Evidence)[*3]が紹介されています。このガイドラインでは，開発対象システムとしてのプロダクトについてだけではなく，システム開発やアセスメントのプロセスについても安全性ケースが必要だと指摘しています。

2. 用途

   環境と相互作用するシステムや製品が持つ不確実性やリスクに対してシステムや製品が望ましい性質を持ち，危険な状況に陥らないことを確認できます。また，システムや製品の開発プロセスにおける計画や，生産物，人間活動，意思決定などに対する合意形成の結果を記録します。主張に含まれる不確実性を関係者が許容できるかどうかを D-Case の作成を通じて議論します。したがって，D-Case を作成した結果にも，主張が持つ性質の影響度とその不

[*3] http://www.adelard.com/asce/choosing-asce/cae.html

確実性を反映することになります。

3. 期待効果

D-Case の効果を列挙すると次のようになります。

- 主張するサービス水準を提供できることを示すエビデンスを提供できます
- システム異常の検出と修正を早期化できます
- 開発・運用プロセスと生産物のリスクに対する客観的な管理を推進できます
- システムのディペンダビリティの影響評価を早期化できます
- システム要求の充足性に対して，客観的なエビデンスに基づく確認プロセスを提供できます
- システム開発・運用プロセスを統合的に確認することによりプロセス改善を推進できます

システム開発・運用プロセスに対する D-Case の構成例を図 4 に示します。D-Case を用いたディ

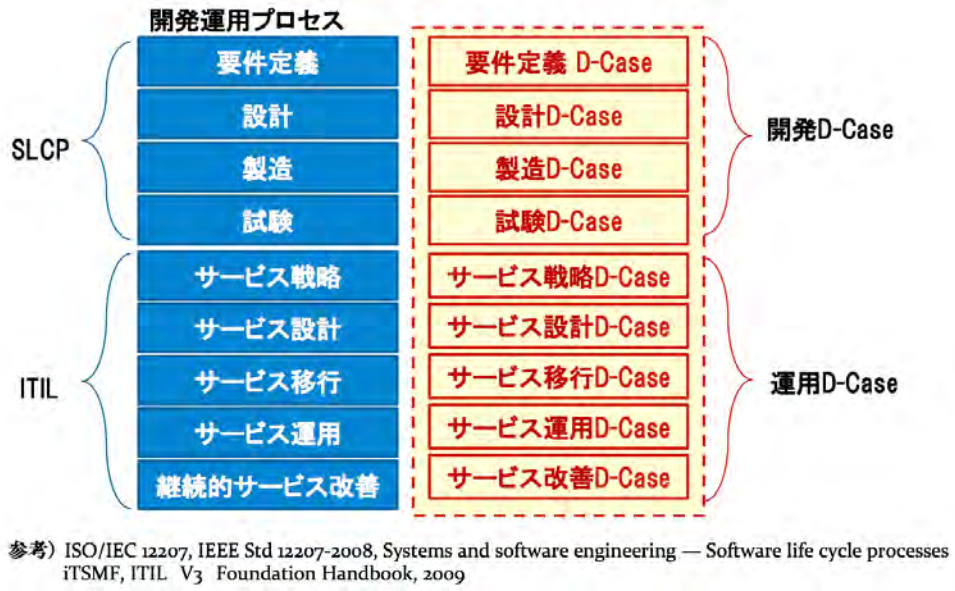


図 4　システム開発・運用プロセスに対する D-Case の構成例

ペンダブルな開発・運用プロセスと，現状の開発・運用プロセスを比較すると，図 5 のようになります。現状でも開発運用文書を用いてシステム開発や運用を効率化しています。しかしこれらの開発運用文書では，ディペンダビリティについての主張や，主張が成立することを示す明示的な証拠がありません。このため，システム障害やシステム改善を実施する上での活動が妥当であることを示すことが困難です。これに対して D-Case を用いた開発運用プロセスでは，システムのディペンダビリティに対する主張，前提，証拠が明示的に記録されているので，主張が成立することを客観的に論証できます。このため，システム障害やシステム改善の際に D-Case を活用することができ，迅速な障害対応やシステム改善の妥当性を容易に確認できます。

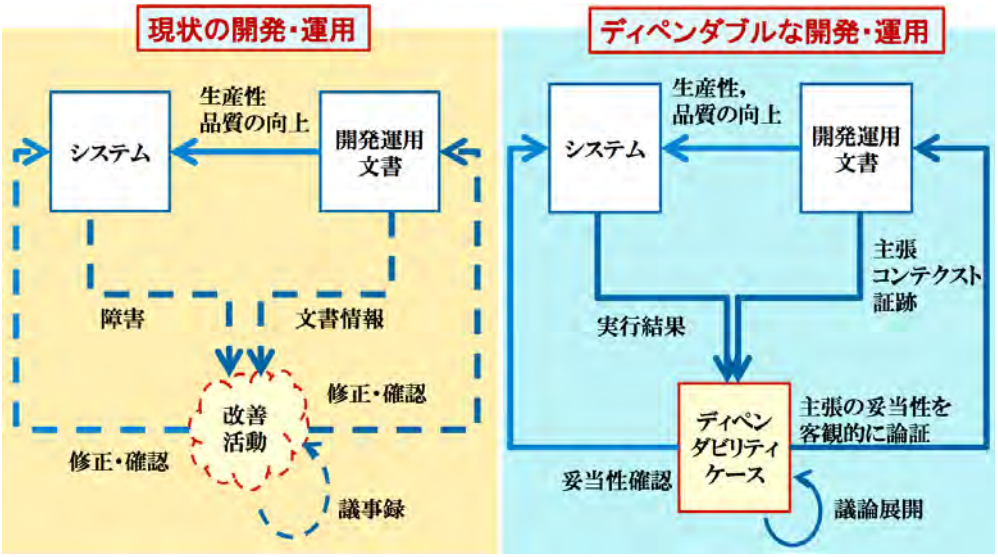


図 5　開発・運用プロセスの比較

2.2　記述ルール

D-Case は主に以下のノードにより構成されます (図 6)。このうち、モニタと外部接続は D-Case の GSN からの拡張です。

1. ゴール (Goal)
   対象システムに対して、議論すべき命題です。例えば「システムはディペンダブルである」とか「システムは適切な安全性をみたす」などです。
2. 戦略 (Strategy)
   ゴールが満たされることを、サブゴールに分割して詳細化するときの議論の仕方です。例えば、「システムは安全である」というゴールに対して、現時点で識別されているハザードに対処できていることによって議論したいとき、戦略ノードとして「識別されたハザードごとに場合分け」を用いると、例えばひとつのサブゴールは「システムはハザード X に対処できる」となります。
3. 前提 (Context)
   ゴールや戦略を議論するとき、その前提となる情報です。例えば、運用環境や、システムのスコープ、あるいは「識別されたハザードのリスト」などです。
4. エビデンス (Evidence)
   詳細化されたゴールを最終的に保証するものです。例えばテストや形式手法による検証結果などです。
5. 未達成 (Undeveloped)

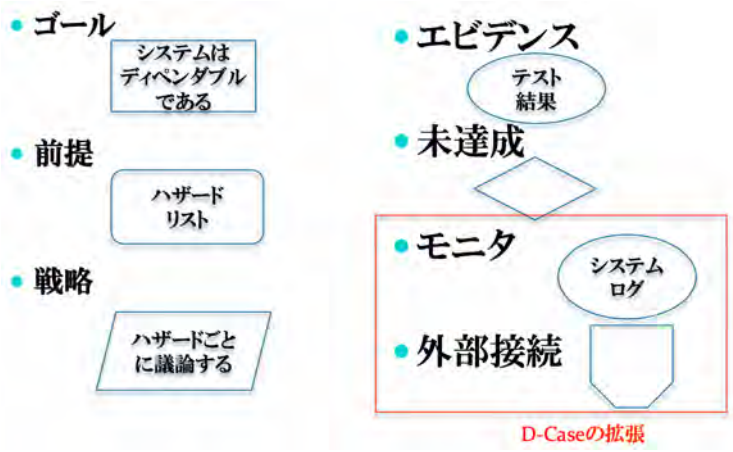


図 6　D-Case の主なノード

ゴールを保証するための十分な議論もしくはエビデンスがないことを表します。

6. モニタ (Monitor)
   運用中のシステムより取得可能なエビデンスです。たとえばインターネットのアクセスログなどです。D-Case ツールはシステムと連携し、モニタリングを用いながらディペンダビリティの保証を支援します。
7. 外部接続 (External)
   他のシステムの D-Case へのリンクです。システムは単一でディペンダビリティを保つことはなく、助けあいながら保ちます。

ノードのつなげ方、矢印は以下のとおりです (図 7)。

1. ゴールは戦略を通して分解されます。
2. D-Case の葉は、エビデンス、モニタ、外部接続、未達成のいずれかです。
3. 前提は、ゴール、もしくは戦略につなげられます。
4. 矢印は 2 種類あります。
   - ゴールから戦略、戦略からゴール、ゴールからエビデンス、モニタ、外部接続、未達成
   - ゴールから前提、戦略から前提 (白抜きの矢印)

2.3　記述ステップ

一般の企業の方が使える D-Case の記述ステップを目指して開発中です [12]。詳しくは、D-Case 入門 [17] を御覧ください。まだ完成していませんが、学生さんなどへの模擬講習などによって改善し、企業の方に使っていただけるようなステップにする予定です。

D-Case 記述ステップ

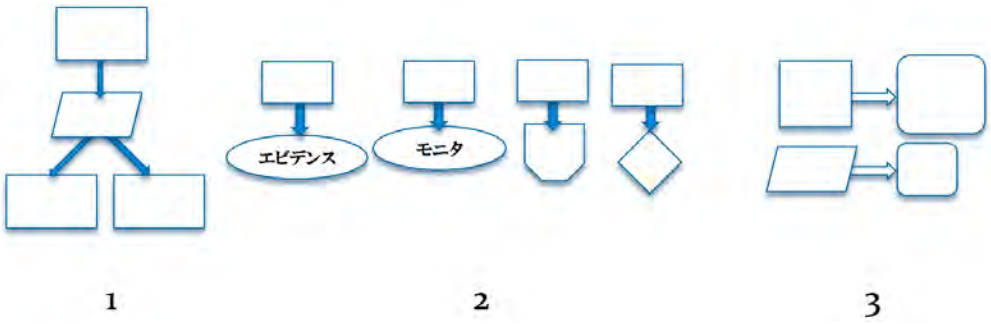


図 7　ノードのつなげ方

1. システムライフサイクルを整理し、それぞれのフェーズの入力、出力ドキュメントをまとめる
2. 入力、出力ドキュメントを分類する
3. トップゴールを置く:「システムはディペンダブルである」
4. ディペンダビリティ要求、環境情報、語彙定義を前提としてトップゴールにおく
5. 大まかに D-Case の構造を考える
6. 必要なドキュメントを前提として置く
7. ドキュメントから D-Case のサブツリーを作る
8. サブツリーができていない部分を典型的な議論構造を使って作る
9. 上記を必要なだけ繰り返す

2.4　記述例

D-Case 記述ステップを使った例をご紹介します。DEOS では ET で毎年デモを行なっています。

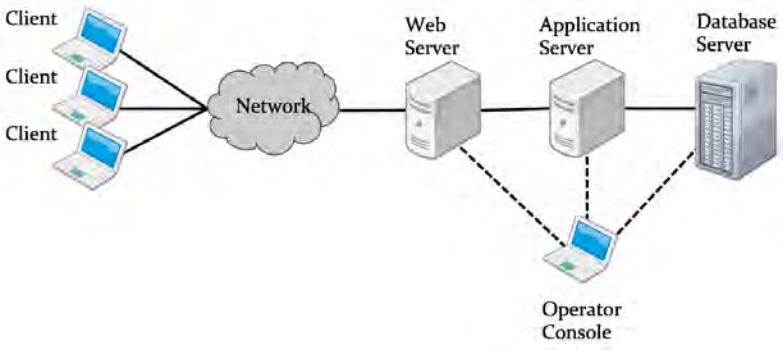
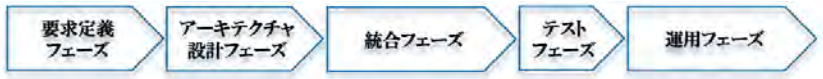


図 8　ET2011 での DEOS ウエブサーバデモシステム

図 8 は 2011 年に発表したウエブサーバシステムです。このシステムの D-Case を書いてみましょう。DEOS センターが行った記述実験を元にしています。

1. システムライフサイクルを整理し、それぞれのフェーズの入力、出力ドキュメントをまとめる
   このウエブサーバシステムは、既存のサーバ PC を統合して開発しました。そのライフサイクル、入出力ドキュメントは図 9 であったとします。ただしデモシステムなので、いくつかのドキュメントには実際には存在しないものもあります。ここでは特に運用ワークフロー定義文書に注目します。



フェーズ	入力	出力
要求定義フェーズ	ユーザインタビュー文書	要求定義文書、SLA文書
アーキテクチャ設計フェーズ	要求定義文書、SLA文書	アーキテクチャ設計文書、運用ワークフロー定義文書、リスク分析文書
統合フェーズ	アーキテクチャ設計文書、運用ワークフロー定義文書、リスク分析文書, サーバ仕様書	統合されたプログラムコード
テストフェーズ	統合されたプログラムコード	テスト結果
運用フェーズ	運用ワークフロー定義文書	運用ログ、システムログ

図 9　ウエブサーバのライフサイクルと各フェーズのドキュメント

2. 入力、出力ドキュメントを分類する
   各フェーズの入出力ドキュメントを、ディペンダビリティの観点から整理して分類します。D-Case を書くときの入力となるものです。
3. トップゴールを置く:「システムはディペンダブルである」
   D-Case のトップゴールはシステムのディペンダビリティに関する命題です。まず「システムはディペンダブルである」という決まり文句をおいてください。そして対象システムに応じて、変更、詳細化してください。この例では、「ウエブサーバシステムは十分に SLA(Service Level Agreement) を満たす」としました。
4. ディペンダビリティ要求、環境情報、語彙定義を前提としてトップゴールにおく
   トップゴールを議論するために必要になる前提情報を前提としておきます。どういったディペンダビリティ要求なのか、どのような環境運用されるのか、トップゴールを理解するために必要な語彙定義などを置きます。特にシステムのスコープを明確にします。この例では、図 10 のような前提をおきました。ここではディペンダビリティ要求は SLA(Service Level Agreement) に相当します。
5. 大まかに D-Case の構造を考える
   トップゴールから一歩一歩ゴールを作っていくのは時として大変なことが多いです。まず大まかな議論構造を考えましょう。この例では以下の様にして決めました。
   「サーバ PC の中身は、開発はしていないので、詳細はわからない。最近の PC 技術は十分に成熟しているので、特にこのような小さなシステムでは、PC 内部の欠陥による障

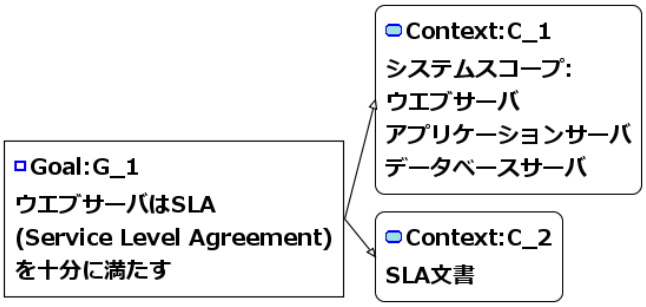


図 10　トップゴールと前提

害はそれほど考えなくてよいだろう。むしろ最近では、サーバ運用上のミスによって重大な情報損失事故などが起こっている。ワークフローにそって、きちんと起こりうるリスクに対処できるか、議論することが重要なのではないか。その上で、障害即応、変化対応議論をしよう」

図 11 が上記の議論からでてきたおおまかな構造です。

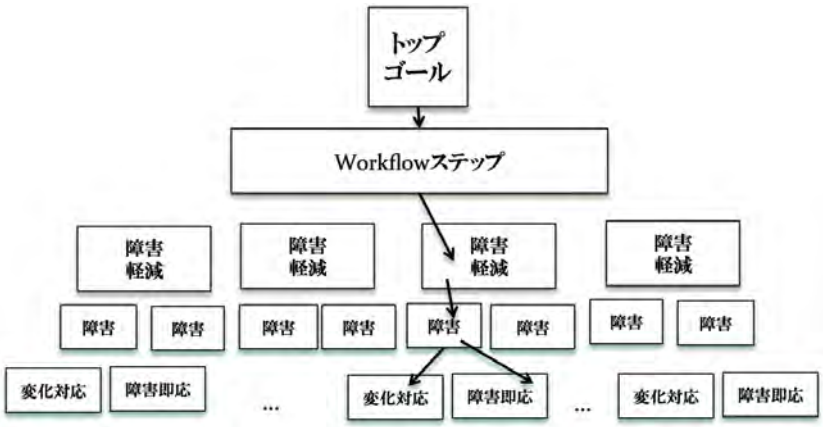


図 11　ウエブサーバシステムの D-Case の大まかな構造

6. 必要なドキュメントを前提として置く
   運用ワークフローにそって議論するためには、ワークフローを定義しているドキュメントが必要です。ワークフローは、ユーザログイン, ショッピングカート処理, クレジットカード認証, 終了処理, 配達, クレーム処理の 6 ステップからなり、それぞれのステップがさらにサブステップに分かれていると定義されていました。
7. ドキュメントから D-Case のサブツリーを作る
   参照するドキュメントの構造から、自動的にゴールをサブゴールに展開することができます。この例の D-Case のトップゴールは、ワークフロー定義書にしたがってステップごとに、図 12 のように展開できます (最初と最後のステップのみ展開)。

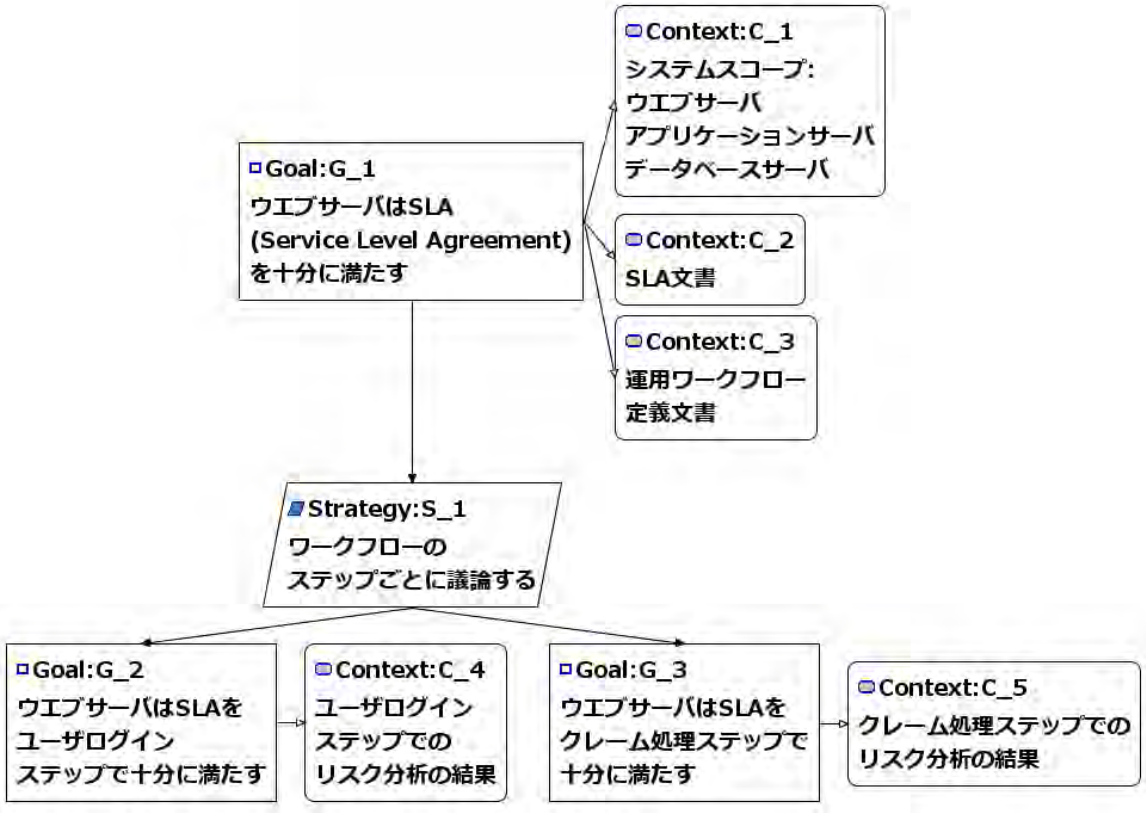


図 12　ウエブサーバシステムの D-Case トップレベル

このようにして D-Case を書いていきます。

2.5　D-Case10 箇条

D-Case で大切なことを 10 箇条としてまとめています。

D-Case 10 箇条

1. ゴールは命題の形をしていなくてはいけない
   基本の一つです。「システムは安全である」など、はい、いいえで答えられることをゴールに書きます。
2. トップゴールに、システムのスコープを前提としておく
   システムのディペンダビリティを議論するときは、まずそのシステムがどの範囲にあるのか、明確にする必要があります。そうでないと、何について議論しているのかわからなくなったり、議論が発散したりします。トップゴールの前提に、明確にシステムのスコープを書きましょう。
3. ライフサイクル、各フェーズの入出力ドキュメントを整理する

D-Case は、システムのライフサイクルで作られるドキュメントをもとに作られます。これらを整理してから、D-Case を書きはじめましょう。

4. 前提は必要な場所のみにできるだけおく
   前提には、議論をするときの前提となるドキュメントが置かれます。議論の前に置いてあればどこでもいいのですが、なるべく必要な場所だけにおきましょう。そうしないと、前提の影響が及ぶ範囲が広くなりすぎ、議論の構造が見えにくくなります。
5. D-Case の構造を大まかに把握する
   D-Case は時として大きくなり、関係するドキュメントも多くなります。おおまかな構造を把握しながら、議論をしたほうがいいです。木だけに注目せず、D-Case 全体の森を意識しましょう。
6. 典型的な議論構造を用いる
   他の人に一所懸命に説明しても、なかなか伝わらないことがあります。そういった場合は、自分勝手なやり方で議論をしている場合があると思います。D-Case も同じで、できるだけ、誰にもわかりやすい、当たり前な議論の仕方を心がけましょう。
7. 他のシステムとの関係を考えながら記述する
   一つのシステムは、単独で存在することはありません。必ず他のシステムと関係します。システムのスコープの外にある、他のシステムを常に意識しながら書きましょう。
8. 常に想定外なことを考えながら記述する
   D-Case に書かれている議論が、その外にあるスコープを前提にしたものであることを常に認識したほうがいいです。オープンな世界では常にスコープから外れた想定外のことが起こります。記述するということはなにかを想定することになりますが、常にその外にあることを意識して書きましょう。
9. 完全な D-Case は存在しない。常によりよい D-Case を目指す
   システムのディペンダビリティを完全に正しく記述することは不可能です。正しさ自体、人によって違うし、時間と状況に応じて変わっていきます。スコープを完全に正しく設定することも、前提を完全に正しく設定することも、すべてのリスクを尽くして議論することも不可能です。不完全さを認めた上で、よりよい、わかりやすい D-Case を書くことを心がけましょう。
10. ほかにもいろいろ大切なことがあります。みなさんのお考えをぜひお教えください。

3　D-Case ツール

D-Case 普及促進のため支援ツール、D-Case Editor [9] を開発しています。D-Case Editor は D-Case/GSN を記述・編集するためのグラフィカルエディタです。本報告書の D-Case の図も D-Case Editor を使って描かれています。D-Case Editor には以下の機能があります。

- D-Case/GSN(Goal Structuring Notation) のサポート

D-Case のフォーマットに加え、OMG ARM など標準的な Assurance Case フォーマットのファイルを読み込み、作成可能です。

- D-Case/GSN パターンライブラリ
  再利用可能な D-Case テンプレートのライブラリ
- 基礎的な変数型チェック
  D-Case の記述に型つきの変数を用いることができます。型チェック機能により内容の妥当性・安全性をチェックできます。
- DEOS プロジェクト成果物との連携
  D-RE（DEOS ランタイムアーキテクチャ）との連携など

図 13 にスクリーンショットを示します。

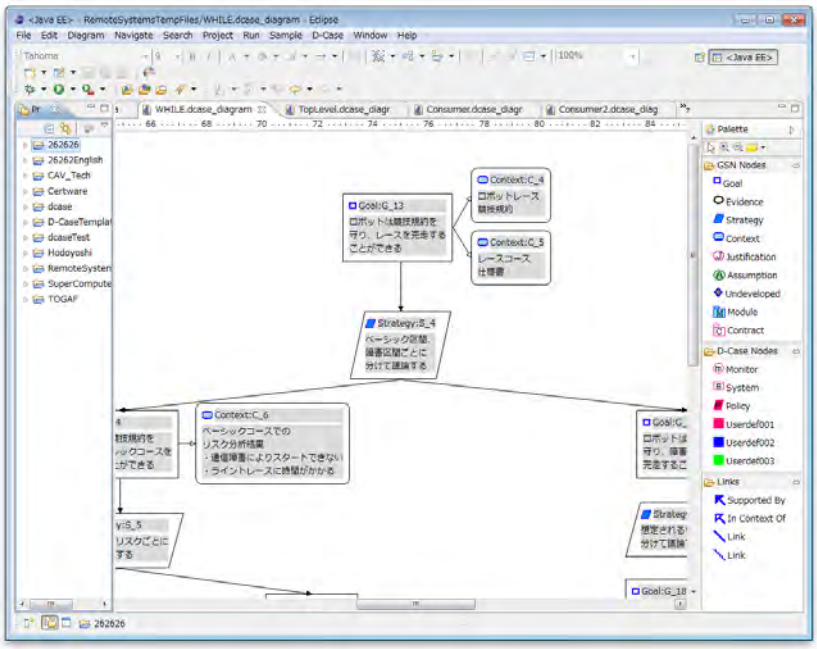


図 13　D-Case Editor のスクリーンショット

D-Case Editor は Eclipse プラグインとして実現されており、Eclipse 上で動作します[*4]。

上述のとおり、これまでは必要となる D-Case 作成支援ツールの基本機能セットを整備してきました。 一方、D-Case を用いてディペンダビリティ保証議論を構築し、ステークホルダ間の合意形成を支援するためには、システムライフサイクルを通じて作成される文書（各種仕様書、評価レポート、会議議事録等）とそれらと D-Case との紐づけが重要になります。そこで恩田グループでは今年度から、「エビデンス文書の管理支援と実証実験」として D-Case および D-Case に紐づけられた文書の適正な管理手法についての研究に取り組んでいます。図 14 に概念図を示します。

[*4] http://www.dependable-os.net/tech/D-CaseEditor/D-Case_Editor_J.html

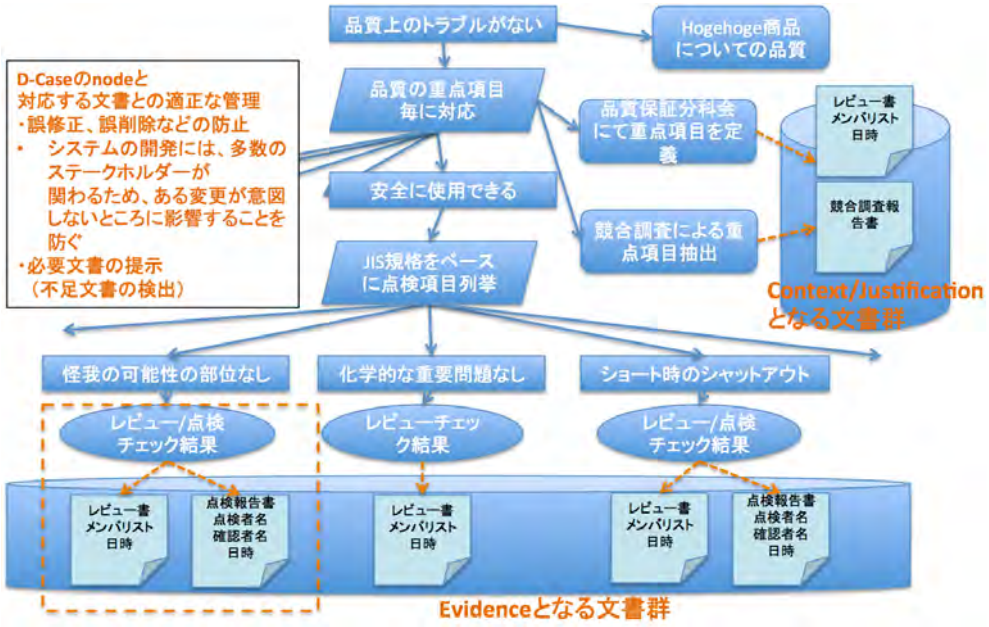


図 14　エビデンス文書の管理支援と実証実験の概念図

D-Case ツールにはこの他に、ベンチマークツール DS-Bench/D-Cloud [2][*5] との連携機能や、汎用的なモニタリング機能拡張 [18][*6]、そして Web Browser 上で D-Case を作成・編集できるツールである D-Case Weaver[*7] があります。

4　D-Case 応用事例

D-Case チームではこれまで様々な企業や大学の方と、D-Case の記述実験を行ってきました。例えば、自動車会社の方との、エンスト問題を対象とした記述実験や、航空宇宙工学を専攻する学生さんと、超小型人工衛星 [13] を題材に一緒に考えてきました。現在は加賀美チームの日本未来館の巡回ロボットの D-Case 記述などに協力しています。今後より多くの共同実験を行う予定です。恩田グループでは ET ロボコンや社内の製品を題材とした記述実験 [16] を行なっています。以下では 2012 年度の ET ロボコンを題材にした記述実験をご紹介します。

ET ロボコンとは、一般社団法人 組み込みシステム技術協会が主催する組み込みソフトウェア開発分野における若年層および初級エンジニアへの分析・設計モデリングの教育機会を提供するイベントです。本イベントは、決められた走行体（LEGO Mindstorms®）で指定コースを自律走行させた走行結果と、UML などで記述された走行競技システムの分析、ソフトウェア設計モデル内容の評価結果の総合結果で競います。本事例では、このイベントの競技に使うロボットの機能を実現

[*5] http://www.dependable-os.net/tech/DSBenchDCloud/index_J.html
[*6] http://web.sfc.keio.ac.jp/~jin/dm/D-Case_EXP_JA/D-Case_EXP.html
[*7] http://www.dependable-os.net/tech/DCaseWeaver/index_J.html

する過程で、以下に示す手順で D-Case 記述を行いました。

1. 要求分析：
   今回、開発の対象となるロボットは、具体的な要求者が存在しません。そのため、競技参加メンバー自身が要求者の立場となり、競技規約や走行コースを参考にして、どのようなロボットを構築して欲しいかをマインドマップを用いて要求を抽出し、その詳細化を行いました。
2. 要件定義：
   次に、ロボットに必要な機能を洗い出すため、詳細化された要求を機能/非機能要求に分類しました。そして、機能要求については、UML のユースケース図などを用いて具体的な機能要件を定義し、非機能要求については、D-Case を用いて具体的な非機能要件を定義しました。
3. D-Case による非機能要件の定義：
   D-Case を記述するに当たり、まず、「時間をかけずに総合優勝」をトップゴールとしました。これは、短納期で総合優勝できるロボットの実現を意味しています。そして、このトップゴールを「目標要素ごとに検討」というストラテジに基づいて、「総合優勝」と「効率的な開発」というサブゴールに分け、ロボットに対する要求と短納期を実現するプロセスに対する要求を別々に検討するようにしました。

次に、「効率的な開発」をどのようにすれば実現できるかの効率化の方針をサブゴールとして定義し、さらに、その方針を実現する手段をサブゴールとして定義しました。そして、最終的に定義された実現手段によって開発作業が行われたことの証拠をエビデンスとして定義しました。さらに「総合優勝」を目指したロボットの実現するために必要なサブゴールとして、「短い時間で走行」、「リタイアしない」、「ボーナスステージで最高タイムを獲得」という要求を設定しました。「短い時間で走行」については、それを実現する方針とその実現方法によってゴールをブレークダウンし、実際にその手段が実現されていることを確認した結果をエビデンスノードとして定義しました。また、「リタイアしない」については、まず、リタイアするリスクを分析した結果を用いて、そのリスク対策としてどのような手段で対応するのかをサブゴールに定義しました。そして、その対策がリスクに対して十分であることの証拠をエビデンスとして定義しました。ET ロボコンでは、走行コースがベーシックステージとボーナスステージに分かれています。ボーナスステージでは、階段やシーソーといった障害物が走行路にある難所と呼ばれる箇所が存在しました。これらを走行規約で決められた通りに通過することで、ボーナスタイムが付与されます。そのため、「ボーナスステージで最高タイムを獲得」については、この難所ごとにリスクを分析し、そのリスク対策をサブゴールとして定義しました。そして、リスク対策が十分であることのエビデンスを定義しました。図 15 は、実際に設計モデルとして ET ロボコン南関東大会に提出した資料の D-Case です。D-Case の上位は、要求分析結果を受けているため、類似していることがわかります。また、上位の要求に対して、どのような考え方で必要な機能を抽出し、エビデンスによって実現方法の検証結果までを、容易に辿れるようになっています。

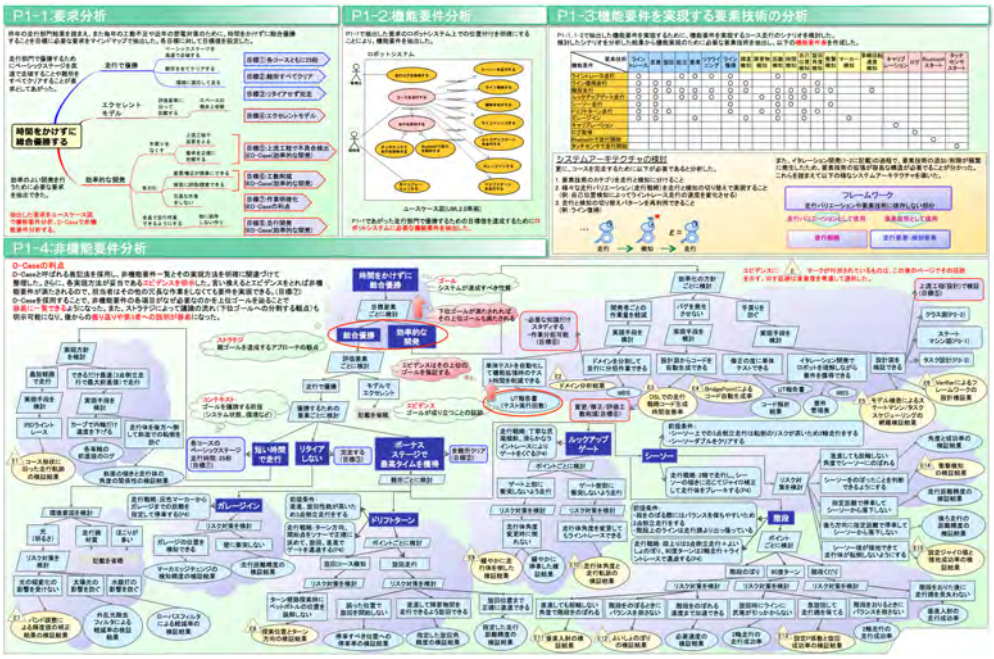


図 15　実際に記述した D-Case

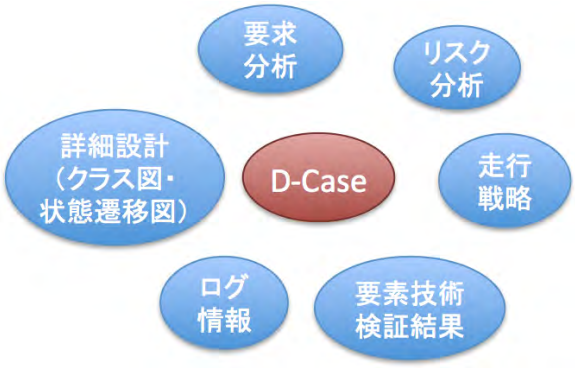


図 16　D-Case で関係が明確になった情報

本記述事例を通して、D-Case を記述するメリットとして、非機能要件の議論の流れが容易に一覧可能になったことが上げられます。図 16 は本記述事例において、D-Case を介して関係が明確になった情報です。D-Case を記述することで、これまでは要素技術がなぜ必要となったのか、何を検証しなければならないのかという点が、プロジェクトの進行やプロジェクトに関わる人が増えるにつれて、曖昧になっていましたが、それを担当者レベルでも容易に確認できるようになりました。また、第 3 者への説明や、後から振り返る際にも D-Case で一覧できるため、容易にできるようになりました。2012 年 9 月 29 日に ET ロボコン南関東大会が神奈川工科大学で開催され、D-Case を記述した設計モデルは最優秀賞（エクセレント・モデル）を獲得しました。さらに 2012

年 11 月の ET2012 で行われた本選でも最優秀賞を受賞しました。大会審査委員長からは、「エビデンスを伴った非機能要件分析は、D-Case を利用し、要素技術との対応が分かりやすい。」と評価を受けました。D-Case は初めて見る人にも理解しやすく伝えることができる手法であるといえます。

5 まとめ

ディペンダビリティ合意形成の手法とツールである D-Case についてご紹介しました。D-Case の研究開発は 2010 年 4 月より本格的に始まり、D-Case 実証評価研究会などで、企業の方との共同研究やコミュニーケーションの機会が増えてきました。我々の研究開発が、企業のみなさんが現在お困りになっていることに少しでも役立てるよう、今後も研究開発を行なっていきます。ご興味をお持ちいただけたら、ぜひ下記にご連絡ください。

- 松野グループ 電気通信大学 松野裕 (matsuno@is.uec.ac.jp)
- 恩田グループ 富士ゼロックス 恩田昌徳 (Masanori.Onda@fujixerox.co.jp)
- 山本グループ 名古屋大学 山本修一郎 (yamamotosui@icts.nagoya-u.ac.jp)

参考文献

[1] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, January 2004.

[2] Hajime Fujita, Yutaka Matsuno, Toshihiro Hanawa, Mitsuhisa Sato, Shinpei Kato, and Yutaka Ishikawa. DS-Bench toolset: Tools for dependability benchmarking with simulation and assurance. In *Proc. IEEE DSN 2012*, 2012. 8pages.

[3] J.R. Inge. The safety case, its development and use in the United Kingdom. In *Proc. of ISSC25*, 2007.

[4] ISO. ISO 26262 road vehicle - functional safety -, part 10: Guideline on ISO 26262, 2012.

[5] Daniel Jackson, Martyn Thomas, and Lynette I. Millett. *Software for Dependable Systems Sufficient Evidence?* The National Academies Press, Washington D.C., 2007.

[6] 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.

[7] Yutaka Matsuno, Jin Nakazawa, Makoto Takeyama, Midori Sugaya, and Yutaka Ishikawa. Toward a language for communication among stakeholders. In *Proc. of the 16th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'10)*, pages 93–100, 2010.

[8] Yutaka Matsuno and Kenji Taguchi. Parameterised argument structure for GSN patterns. In *Proc. IEEE 11th International Conference on Quality Software (QSIC 2011)*, pages 96–101, 2011.

[9] Yutaka Matsuno, Hiroki Takamura, and Yutaka Ishikawa. A dependability case editor with pattern

library. In *Procs. IEEE 12th International Symposium on High-Assurance Systems Engineering (HASE)*, pages 170–171, 2010.

[10] Yutaka Matsuno and Shuichiro Yamamoto. Consensus building and in-operation assurance for service dependability. In *Proc. of CD-ARES, LNCS 7465*, pages 639–653. Springer, 2012.

[11] Yutaka Matsuno and Shuichiro Yamamoto. Toward dynamic assurance cases. In *Proc. JCKBSE 2012*, pages 154–160. IOS Press, 2012.

[12] Yutaka Matsuno and Shuichiro Yamamoto. A new method for writing assurance cases. *International Journal of Secure Software Engineering (IJSSE)*, Special Issue on Cybersecurity Scientific Validation, January 2013. Accepted for Publication.

[13] Kohei Tanaka, Yutaka Matsuno, Yoshihiro Nakabo, Seiko Shirasaka, and Shinichi Nakasuka. Toward strategic development of hodoyoshi microsatellite using assurance cases. In *Proc. of International Astronautical Federation (IAC2012)*, 2012.

[14] Mario Tokoro, editor. *Open Systems Dependability: Dependability Engineering for Ever-Changing Systems*. CRC Press, 2012.

[15] Shuichiro Yamamoto and Yutaka Matsuno. A review method based on a matrix interpretation of GSN. In *Proc. JCKBSE 2012*, pages 36–42. IOS Press, 2012.

[16] 伊東敦、松野裕. ET ロボコンを対象としたドメインからの D-Case による保証議論の構築. In **ソフトウエアシンポジウム** *2012* **予稿集**, 2012.

[17] 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.

[18] 中澤 仁、 松野 裕、 徳田 英幸. D-Case を用いたユビキタス・センサ・ネットワーク管理ツール. **電子情報通信学会論文誌（和文** *B*）**ユビキタス・センサネットワークを支えるシステム開発論文特集**, J95-B(11), 11 2012.

DEOS プロジェクト

Homepage *http : //www.dependable − os.net*
Tel 03 − 3526 − 6724
E-mail *center@dependable − os.net*