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
SS2012-20 KBSE2012-22 (2012-7)

# ディペンダビリティケース作成法に関する一考察

山本　修一郎　松野　裕

名古屋大学 情報連携統括本部 情報戦略室
〒464-8601 名古屋市千種区不老町

E-mail:　syamamoto@ acm.org

**あらまし**　システムの安全性や可用性を保証するために，ディペンダビリティケースが注目されている．しかし，ディペンダビリティケースの作成法についての明確な手順や記述上のガイドラインについては必ずしも明確ではなかった．

本稿では，まずディペンダビリティケース作成上の課題を明らかにする．次いで課題に対する取り組み方針について述べる

**キーワード**　ディペンダビリティケース，アシュアランスケース，リスク分析，機能要求，非機能要求，ライフサイクルプロセス，システム開発文書

# A Consideration on Developing Dependability Case

Shuichiro Yamamoto and Yutaka Matsuno

Nagoya University, Strategy Office, Information and Communications Headquarters
Furo-cho. Chikusa-ku, Nagoya 464-8601 Japan
E-mail:　syamamoto@ acm.org

***Abstract*** *Although dependability case is attracted to assure system safety and availability, methods and guidelines how to develop dependability cases. In this paper, problems and issues to develop dependability cases are clarified. Then practical methods to describe dependability cases are discussed.*

**Keyword** Dependability case, Assurance Case, Risk analysis, Functional requirements specification, Non functional requirements, Lifecycle process, System development documents

## 1 はじめに

システムの安全性を確認するために，安全性ケース(Safety case)，アシュアランスケース（Assurance case，保証ケース）やディペンダビリティケース（Dependability case）が注目されている．このためGSN(Goal Structuring Notation)を用いてこれらを記述する方法が提案されている．DEOSプロジェクトの一環としてディペンダビリティケースの作成を支援するためにD-Caseエディタが開発されている．D-Caseエディタでは，GSNに基づいてディペンダビリティケースを記述できる．しかし，実際のシステム開発プロジェクトの担当者によるディペンダビリティケースの作成を容易化するためには，エディタを提供するだけでは不十分である．システム開発プロセスやシステム開発文書に即して，より具体的なディペンダビリティケース作成手順を提供する必要がある．

このためDEOSプロジェクトでは，ディペンダビリティケースを作成する上での実用上の

課題を明らかにするために，実験的なシステム開発への D-Case エディタの試行適用を進めている．本稿では，このようなディペンダビリティケースを適用する上での課題と基本的な方法を明らかにする．

なお本稿では，安全性ケースやアシュアランスケース，ディペンダビリティケースを総称してディペンダビリティケースという用語を用いる．

本稿の以降の部分の構成は次のようになる．

第２節で本研究の背景を示す．第３節でディペンダビリティケースを作成する上での課題を明らかにする．第４節では，ディペンダビリティケースを作成するための基本的な考え方を提案する．第５節ではまとめと今後の課題を明らかにする．

## 2 研究の背景

### 2.1 研究動向

重要システムの実行中に優先順位の高い要求を満足することを確認するために，ディペンダビリティケースが必要とされている[12].

ディペンダビリティケースでは，図1に示したように，ゴール（主張），戦略，前提（コンテクスト），根拠（証跡，ソリューション）によって，システムのディペンダビリティに関する議論を構造化して確認することができる．

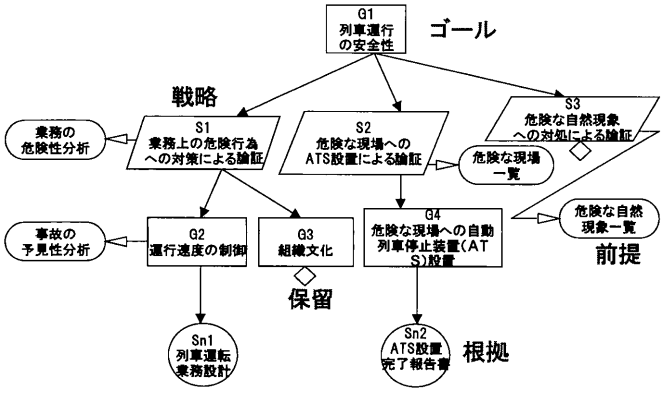


図１ ディペンダビリティケースの例

このため，ディペンダビリティケースの関連研究として，安全性ケースやアシュアランスケースについて以下のような手法が研究開発されている．

GSN を作成するために，①ゴールを識別する，②ゴールを記述するための基礎を定義する，③ゴールを満足させるための戦略を識別する，④戦略を記述するための基礎を定義する，⑤戦略を吟味する，⑥基本的な証跡を識別するという6段階の手法を Kelly が提案している[1][2].

システム障害に至る可能性のある潜在的なソフトウェアの故障モードを識別し，故障モードについての証跡を提示するという証跡に基づくソフトウェア安全性プロセス（evidence based software safety process）の必要性が指摘されている [3].

安全性ケースを再利用するための安全性ケースパターンや，安全性ケースをモジュール化できるモジュラー安全性ケースが提案されている[4][5].

従来のガイドワードを拡張した逸脱分析を用いてディペンダビリティケースを作成する手法[6][9], シナリオを用いてソフトウェア設計時にディペンダビリティ要求を満たすような代替案を選択する手法[7][8]が提案されている．

European Organisation for the Safety of Air Navigation 制定している安全性ケース開発マニュアル[10]では，安全性ケースのコンテクストを定義することが重要であると指摘している．また安全性ケースをレビュするためのチェックリストを提案している．

複数のシステムから構成されるシステム（System of Systems）の開発過程で，システム分析，ゴール要求抽出，代替設計案の識別，矛盾点の解消からなるディペンダビリティケースを構造化して作成する手法が提案されている[11].

ディペンダビリティケースを作成する際に必要となる議論分解の観点として，システム構成や機能構成，属性構成などが整理されている[13].

コンテクストモデル，フィーチャモデル，構造モデル，ユースケースモデル，振舞いモデル，故障管理振舞いモデルに基づいて安全性ケースを作成する手法が提案されている[14]．また医療機器分野におけるアシュアランスケースのレビュ手法が提案されている[15].

筆者らが参加しているDEOSプロジェクトではディペンダビリティケースに基づいてオープンシステムの障害対応サイクルと変化対応サイクルを支援する研究を推進している[16][28]．このためディペンダビリティケースを編集できるD-Caseエディタが開発された[17].

最近になって，概念文書，設計書，運用手順書，準備ハザードリストに基づいて安全性ケースを作成する手法が提案されている[18].

アシュアランスケースの研修コースも提供

されている[19][20][21].

これらの手法では，多様な適用分野や開発工程を対象としてディペンダビリティケースの作成法が個別的に提案されている．しかし，ディペンダビリティケースを用いて実際のシステムがディペンダブルであることを確認するためには，システム開発プロセスや工程生産物との具体的な対応関係や利用手順が一貫した形で明確になっている必要がある．

この点で，現状のディペンダビリティケース作成手法はまだ多くの改善すべき課題が残っている．

### 2.2 研究の位置付け

上述したことから，本研究では，システム開発運用工程生産物を用いたディペンダビリティケースの作成手順の具体化を進めることとした．この理由は、上述したように，ディペンダビリティケースの作成手順が，断片的な取り組みにとどまっており、システム開発プロセスや工程生産物を利用する系統的な手順が明確になっていなかったからである．

また故障解析（FTA）や FMEA 分析，Hazard 分析などのリスク分析技術がある．前述したように逸脱分析を用いてディペンダビリティケースを作成する手法[6][9]も提案されている．しかし，開発運用プロセス全体を通じたディペンダビリティケースとリスク分析の具体的な適用関係は必ずしも明確になってはいない．

したがってサービスに対する統一的な分析手法として確立されてはいないという問題があった．

また，ディペンダビリティケースが前提とする「証跡に基づく論理的な議論による妥当性の論証」という考え方は，日本の文化にはなじまないところがある．この理由は，このような論証の基礎的な訓練が日本では不足しているからである．これは，これまで数年にわたって GSN を技術者に教育してきた筆者らの経験に基づく仮説である．逆にいえば，論証についての知識とスキルがあれば，ディペンダビリティケースを作成することは難しいことではないともいえる．

このためディペンダビリティケースを開発者が作成しようとすると，なにを論証すべきかどのように論証すべきか，なにが証跡なのか分からず初心者が躓くことになる．たとえば，Kelly による 6 段階作成法の最初でゴールを識別するところから悩むことになる．

したがってディペンダビリティケースを記述するために GSN の構文を教えただけでは，論証に慣れていない日本の現場における技術者がディペンダビリティケースを作成することは困難である．このため，日本では欧米で開発されてきたディペンダビリティケースの作成手法よりも具体的な手法が必要になる[27].

## 3 ディペンダビリティケース作成上の課題

### 3.1 基本的な課題

ディペンダビリティケースを作成するためには，主張（ゴール），戦略，コンテクスト，証跡とそれらの関係を記述する必要がある．ディペンダビリティケースを作成する際には，まずディペンダビリティについてシステムが満たすべき主張を列挙する必要がある．このために，戦略ノードを用いて主張を下位主張に分解することになる．

この場合，次のような基本的な疑問が生じることが多い[27].

① 主張として何をどう書くのか
② 戦略に何を書くのか
③ 戦略で分解する幅をどこまで広げるのか
④ コンテクストに何を書けばいいのか
⑤ 証跡に何を書けばいいのか
⑥ 木構造をどこまで深くするのか
⑦ コンテクストと証跡の関係をどのように分析すればいいのか

これらの疑問に答えるためには，適用対象分野を限定することにより，分野に即したディペンダビリティケースの階層構造と構成要素に記述すべき内容を予め規定しておく方法が有効である[27].

しかし，対象分野が限定できない場合には，より一般的な方法が必要となる．たとえば開発文書や運用保守文書などの既存文書に基づいて，ディペンダビリティケースを作成する方法が考えられる．このような既存文書に基づく方法の利点としては，指定された文書の構造や内容によって作成すべきディペンダビリティケースの構造と構成要素に記述すべき事柄を明確化できることである．

たとえば，既存文書の章節を証跡として用いる方法が提案されている[10]．またリスク分析で作成される故障木や FMEA，HAZOP などの文書がコンテクストや証跡で用いられている．

### 3.2 議論構造と制御構造の混同

ディペンダビリティケース構文の理解不足から，初心者が作成したディペンダビリティケースでは以下のような誤りが多い．

① 戦略をゴールと取り違えている
② 主張として書くべき内容が命題になっておらず実行文や機能文になっている．
③ 戦略を判断分岐だと誤解する
④ 議論ではなく機能の実行順序で分解している

このような取り違えは，GSN 構文がフローチャートに似ていると考えるからであると思われる．たとえば主張が矩形であることから実行文に対応させ，戦略が平行四辺形であることから判断文に対応させていると思われる．さらに GSN では矢印が上から下に向かうので，フローチャートの実行制御の流れと混同するのであろう．GSN の構造を既知であるフローチャートの構造で理解してしまうことからくる誤解である．

とくに，戦略を用いるときに，場合分け，判断，代替案についての論証と，これらを区別できないことが多い．このため，これらに対する議論を例示して教育しておく必要がある．

### 3.3 記述範囲の制御

安全性ケース開発マニュアル[10]で指摘されているように，ディペンダビリティケースで確認する境界範囲を定義することが重要である．このような境界範囲を決めないと，どこまでも際限なく確認することになる．また確認範囲の十分性を保証できない．たとえば，図 1 で示したディペンダビリティケースでは列車運行の安全性を業務上の危険行為，危険な現場，危険な自然現象についての対策を確認している．しかし，列車自体の整備や整備業務についての危険性についての対策までは確認していない．したがってディペンダビリティケースの記述では，予め記述して確認する範囲についての合意が必要である．そうしないと，上述したように確認範囲の十分性を保証できなくなるからである．

### 3.4 分解の観点の多様性

主張を戦略によって分解する際の観点や，分解の順序についての指針が明確に整理されていなかったために，どのように論証を展開すべきかが分からないという問題があった．このため，前述したように議論分解の観点として，システム構成(Architecture),機能構成 (functional), 属性構成(set of attributes), 帰納分解(infinite set), 要求やリスクの完全集合(complete),単調分解(monotonic),具体化(concretion)が示されている[13]．ここで帰納分解は帰納法による主張の分解である．単調分解は従来システムを将来システムに改善するためだけの分解である．具体化は曖昧な主張を具体的な主張に分解する．

しかし，Bloomfield の観点にはプロセスや条件による分解が考慮されていない．

たとえば，条件判断に対する論証では図 2 に示すような戦略によって主張を分解することができる．ここでは，条件が定義されていること，条件が成立する場合と成立しない場合のいずれの場合でも，ディペンダブルであることを証跡によって確認している．

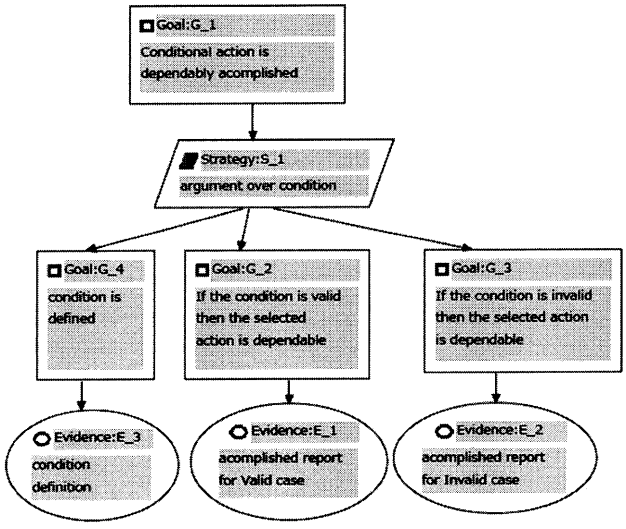


図 2 条件分解戦略

### 3.5 システム開発運用文書との関係

ディペンダビリティケースとシステム開発運用文書との関係では，文書とコンテクストや証跡との対応付けが具体的ではないため，複数の文書と複数のディペンダビリティケースが集合レベルで対応しているとされているだけであった．このため，要素単位での具体的な関係が不明確だった．したがってシステム開発運用文書にある貴重な情報が十分活用できないという問題があった．

もし，既存の開発情報を活用できればディペンダビリティケース作成効率と品質を向上できる．

一方，この過程でディペンダビリティケースを適切に作成できなければ開発情報の品質が

低いことになるので，システムや開発運用文書の品質もまた向上できない．システムに対するディペンダビリティを確認するためには図3に示すように，開発運用文書とディペンダビリティケースを適切に連携させた手法を開発することが望ましい．このような手法を用いることによりディペンダビリティケースの作成を容易化できるだけでなくシステムと開発運用文書の品質を向上できる．

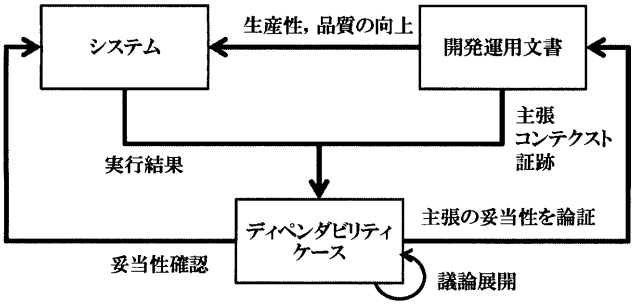


図 3 システム，開発運用文書とディペンダビリティケースの関係

## 4 プロセスの妥当性確認のためのディペンダビリティケース作成法

一般的なプロセスを定義する場合，少なくとも目的，手順，入力，出力が必要である．

したがって，プロセスの妥当性を確認するためのディペンダビリティケースを次のようにして作成できる（図4）．

①目的に基づいて主張を記述

②戦略により，目的を達成するための手順ごとに議論

③コンテクストに，入力情報を記述

④証跡として，このプロセスの出力の確認結果を記述

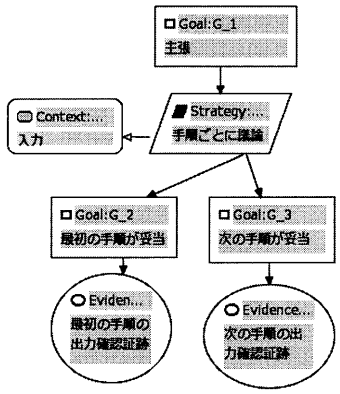


図 4 プロセス定義に基づくディペンダビリティケース

また先行プロセスの出力としての証跡が，後続プロセスの入力となる場合，先行プロセスの証跡が後続プロセスのコンテクストになるという関係がある．しかし，現在のディペンダビリティケースでは，このような証跡とコンテクストの依存関係を明示的に記述する手段はない．このため，それぞれの名前によって依存関係を記録することになる．

なお，この手法は，多様なプロセス定義に対して適用できる一般性の高い手法である．たとえば，構造化分析手法におけるデータフロー図のプロセス，ワークフローの活動，BPMN などのビジネスプロセス，UML のユースケース図，アクティビティ図，状態遷移図，SysML などのアーキテクチャ記述，フローチャートやコードなどにも適用できる．

## 5 おわりに

本稿では，ディペンダビリティケースを作成する上での研究動向と適用上の課題を明らかにした．次いでプロセス定義に基づくディペンダビリティケース作成法を提案した．

現在，本手法の有効性を明らかにするため，スーパーコンピュータの運用手順[24]ならびに，エンタープライズアーキテクチャ開発手法（TOGAF ADM）[25]に対するディペンダビリティケース作成実験を進めている．また，ディペンダビリティケース作成知識を体系化するための調査も進めている[22][23]．

今後は本稿で提案したディペンダビリティケース作成法の適用実験を通じて手法として洗練していく．また前述したように各種のモデル表現や HAZOP などのリスク分析表などからディペンダビリティケースを作成するための手法を開発している．これらについても今後引き続き報告していくとともに，一連の研究成果を踏まえて，ディペンダビリティケース作成ガイドラインとしてまとめる予定である．

## 謝辞

本研究は CREST「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」研究領域（DEOS プロジェクト）の支援を受けたものである[28]．

## 参考文献

[1] Kelly, T. P, A Six-Step Method for the Development of Goal Structures, York Software Engineering, 1997

[2] T. Kelly. “Arguing Safety, a Systematic Approach to Managing Safety Cases”. PhD Thesis, Department of Computer Science, University of York, 1998

[3] J. A. McDermid. Software safety: where's the

evidence? In SCS '01: Proceedings of the Sixth Australian workshop on Safety critical systems and software, pages 1-6, Darlinghurst, Australia, Australia, 2001. Australian Computer Society, Inc.
[4] I. Bate, T. Kelly, Architectural considerations in the certification of modular systems, Reliability Engineering and System Safety 81, pp.303–324,2003
[5] Tim Kelly and Rob Weaver, The Goal Structuring Notation – A Safety Argument Notation, Proceedings of the Dependable Systems and Networks 2004 Workshop on Assurance Cases, July 2004
[6] Despotou G., Kelly T., Extending the Concept of Safety Cases to Address Dependability. In proceedings of the 22nd International System Safety Conference (ISSC), Providence, RI USA, proceedings by System Safety Society 2004.
[7] G. Despotou, T. Kelly. "Using Scenarios to Identify and Trade-off Dependability Objectives in Design. Proceedings of the 23rd International System Safety Conference (ISSC), 2005
[8] T. Kelly. Using software architecture techniques to support the modular certification of safety-critical systems. In Proceedings of the 11th Australian Workshop on Safety-Related Programmable Systems, August 2005.
[9] G. Despotou, T. Kelly, EXTENDING SAFETY DEVIATION ANALYSIS TECHNIQUES TO ELICIT FLEXIBLE DEPENDABILITY REQUIREMENTS, In proceedings of the 1st IET (Former IEE) International Conference on System **Safety** Engineering (ICSS), London, UK, June 2006.
[10] European Organisation for the Safety of Air Navigation, Safety Case Development Manual, 2nd ed., EUROCONTROL, Oct. 2006.
[11] G. Despotou, T. Kelly, Design and Development of Dependability Case Architecture during System Development, . In proceedings of the 25th International System Safety Conference (ISSC), Baltimore, MD USA. Proceedings by the System Safety Society, 2007
[12] Jackson, D. et al, Software for dependable systems–sufficient evidence?, NATIONAL RESEARCH COUNCIL, 2008
[13] R. Bloomfield and P. Bishop, "Safety and assurance cases: Past, present and possible future – an Adelard perspective," in Proc. 18th Safety-Critical Sys. Symp., Feb. 2010.
[14] Ibrahim Habli, Ireri Ibarra, Roger Rivett, Tim Kelly, Model-Based Assurance for Justifying Automotive Functional Safety, 10AE-0181, 2010
[15] Oleg Sokolsky, Insup Lee, Mats Heimdahl, Challenges in the Regulatory Approval of Medical Cyber-Physical Systems, In Proceedings of the International Conference on Embedded Software (EMSOFT 2011). Taipei, Taiwan, October 2011.
[16] DEOS プロジェクト, 2011 科学技術振興機構 White Paper DEOS-FY2011-WP-03J, *www.dependable-os.net/ja/topics/file/**White_Paper**_V3.0J.pdf*
[17] D-Case エディタ, http://www.dependable-os.net/tech/D-CaseEditor/
[18] Ewen Denney and Ganesh Pai, Ibrahim Habli, Perspectives on Software Safety Case Development for Unmanned Aircraft, DSN2012
[19] Adelard,**www.adelard**.com/asce/index.html
[20] AAMI, Safety Assurance Cases for Medical Devices, http://www.aami.org/meetings/courses/safety.html
[21] Dependability computing, assurance case course, http://www.dependablecomputing.com/courses.html
[22] 松野裕, ヴァイセ バトゥ, 山本修一郎, アシュアランスケースへの構造化文書の適用に関する調査, KBSE研究会, 2012
[23] Vaise Patu, Yutaka Matsuno, Shuichiro Yamamoto, Research framework for dependability science based on assurance cases, KBSE研究会, 2012
[24] 高間翔太, 松野裕, 山本修一郎, スーパーコンピュータ運用手順に対するディペンダビリティの確認手法の提案, KBSE 研究会, 2012
[25] 徳野達也, 松野裕, 山本修一郎, エンタープライズアーキテクチャ開発プロセスに対するディペンダビリティケース作成法の提案, KBSE研究会, 2012
[26] 猿渡卓也, 松野裕, 星野隆, 山本修一郎, Modular GSNの定式化, KBSE研究会, 2012
[27] 小林茂憲, 山本修一郎, 保証ケースを用いたサービス提供判断方法の提案, 信学技報, vol. 111, no. 489, KBSE2011-70, pp. 7-12, 2012年3月.
[28] DEOSプロジェクト, http://www.crest-os.jst.go.jp