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
KBSE2012-52 (2012-11)

ディペンダビリティケースへの責任属性の導入法の検討

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 ディペンダビリティケースがシステムの安全性や説明責任を保証する方法として注目されている．しかし，これまでのディペンダビリティケースでは担当者の責任分担を明示的に扱う方法が明確ではなかった．このため，本報告では，ディペンダビリティケースを構成するノード集合に対して責任属性を持たせることにより，担当者の役割を明確化する手法を提案する．

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，責任，説明責任，アクタ，組織構造

A Consideration on Introducing Responsibility Attributes to 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 achieve accountability based on dependability cases are not sufficient. In this paper, problems and issues to achieve accountability based on dependability cases are clarified. Then a method to allocate responsibility for dependability cases is proposed and explained with an example.*

Keyword Dependability case, Assurance Case, Responsibility, Accountability, Actor, Organizational structure

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[1][2][3][4][5][6][7]．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[1][2].

筆者らが参加しているDEOSプロジェクト[8][9][10]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[11]. 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．

DEOSプロジェクトでは，説明責任をD-Caseに基づいて遂行することを目標の一つとしている．説明責任を遂行するためには，組織の責任分担構造を反映したディペンダビリティケースが必要となる．しかし，これまでのディペンダビリティケースでは，責任主体を明示できないので，主張と主体との関係がわからないという問題があった．このため，システム開発プロセスやシステム開発文書に即して，責任の遂行主体を具体的にディペンダビリティケース上で明示する表記方法と，説明責任の遂行手順を具体化する必要がある．

本稿では，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を説明責任遂行に適用する上での課題と基本的な考え方を明らかにする．

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．

以下では，まず第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に責任属性を導入する上での課題を明らかにする．第4節では，責任属性を

導入したディペンダビリティケースを記述するための基本的な考え方を提案し比較する．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[7].

ディペンダビリティケースでは，図1に示したように，主張（Claim），説明（Strategy, 戦略），前提（Context, コンテクスト），証拠（Evidence, 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．なお，本稿で「戦略（Strategy)」に対する訳語として「説明」を用いたのは，説明責任を遂行するために，主張の階層関係の理由をStrategyが「説明」しているからである．

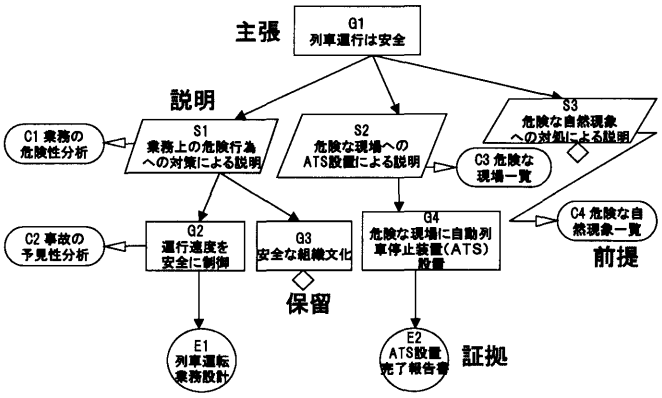

図１ ディペンダビリティケースの例

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][12]，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[13][14]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[15]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[16].

UMLのステートチャートでは，アクタ（主体）ごとにスイムレーンを用いてアクタが責任を持つ機能を定義することができる．KAOSではエージェントが責任を持つ要求を明確にするために，エージェントと要求に対するノードを責任関係によって図式上で接続する方法を提案している．[17]．またSommervilleらは，エージェントが与えられた責任の遂行に失敗するという責任障害(responsibility failure)について議論している．そこで責任を次のように定義している[18].

[定義]責任

組織的，社会的および文化基準についての適合性に従って，与えられた状態を達成し、維持し、遂行するために、ある代理人が負う義務を責任という．

責任では，複数の属性集合について記述する．ワークフローモデルのプロセスごとに責任を持つエージェントを割り当てることにより責任属性を記述することを提案している．責任属性では，ゴールや，遂行されるコンテクスト，仮説を定義する事前，事後条件などが記述される．

KAOSやi*などのゴール指向手法では，システムの高水準ゴールをエージェントに関連付け，ゴール階層によって上位ゴールを下位ゴールが達成することを記述できる．このようなゴール階層によって，下位ゴールの障害によって上位ゴールの障害が発生するかどうかなどを分析できる．しかし，Sommerbilleらは，医療サービスなどの遂行では，必ずしも明示されていない医療倫理などのある一定範囲の制約を明らかにすることが重要になるので，ゴール分析だけではなく，責任属性を分析するための責任モデル(Responsibility model)が必要になると指摘している．

筆者らは DEOS プロジェクトの一環として，D-Case

作成手法[19][20][21][22]，スーパーコンピュータの運用手順[23][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][26]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．さらに，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[27][28][29]．

筆者らは，主体間でディペンダビリティゴールが依存する関係を表現するために，d*フレームワークを提案している[10][30]．しかし，責任属性を対象にした具体的な手法については具体化していなかった．

また概念文書，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[31]．またアシュアランスケースの研修コースも提供されている[32][33][34].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，社会技術システムとして責任主体をディペンダビリティケースに対応づける具体的な手順を明確にする必要がある．

この点で，現状のディペンダビリティケース作成手法を用いた説明責任の遂行を支援する手法は，明確になっていないという問題がある．

2.2 研究の位置付け

上述したことから，本研究では，組織構造を用いて，ディペンダビリティケースに対する責任主体の明確化する手法を具体化することとした．この理由は、上述したように，ディペンダビリティケースでは責任属性を表現できないためである．

さらに，責任属性が付与されたディペンダビリティケースを用いて説明責任を組織的に遂行する方法を確立する．

3 責任属性導入法の検討

3.1 責任属性導入上の課題

ディペンダビリティケースに責任属性を導入する場合,次のような基本的な課題を解決する必要がある．

① 組織構造との対応
② 主体間の責任委譲関係の扱い
③ 責任委譲関係定義法
④ 責任委譲関係の妥当性評価法
⑤ 責任属性の表現方法
⑥ 説明責任遂行方法

以下ではこれらについて述べる．

3.2 組織構造との対応

サービスを提供する組織には，階層構造だけでなく，組織横断的にサービスの安全性を統括する横断的な構造がある．

たとえば，図1の鉄道運行サービスに対する組織構造は，図2のようになる．図2では，二重線で組織を構成する部門を表現している．矢線によって接続された部門間に組織上の関係があることを示すことができる．すなわち，矢線の開始点に対応する部門が，終了点に対応する部門に対して課した責任の遂行を管理することを示している．

鉄道サービスを統括する運行本部があり，この配下に，運輸部門と施設部門，安全統括部門がある．また安全統括部門は倫理部門を持ち，運輸部門と施設部門に対して安全性を統括する．

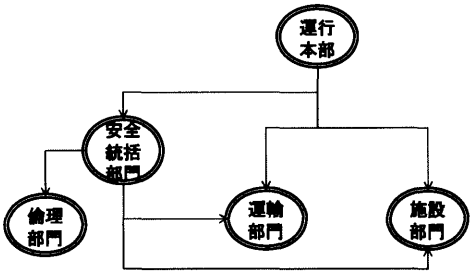

図2　組織構造の例

したがって，ディペンダビリティケースでも，組織が持つ階層構造と横断構造に対する責任属性間の関係を表現できる必要がある．この2つの関係を用いて説明責任の遂行をディペンダビリティケースで支援できる．

3.2 主体間の責任委譲関係の扱い

責任委譲関係はどのようにして定義されるだろうか？明示亭に定義するか，それとも暗黙に定義するか？責任委譲もディペンダブルであるためには，命じていな主張として責任委譲を定義して，その根拠を示す必要がある．

たとえば，上述した組織図における責任委譲関係では，図3に示したような責任委譲についての主張を明示的に定義できる．

なお，安全統括部門から施設部門に責任委譲関係がないのは，図1では施設部門の主張には安全統括部門に対応する主張がないためである．

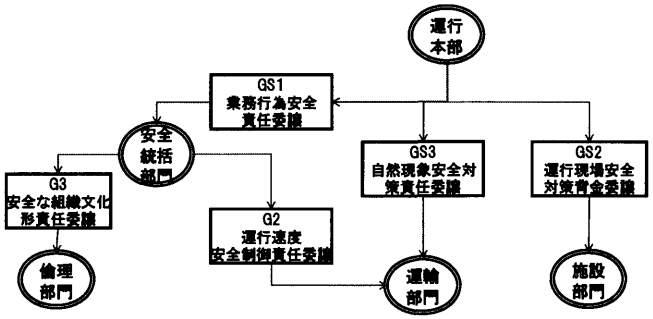

図 3　部門間の責任委譲関係の例

3.3 責任委譲関係の定義法

主体間で責任が委譲されることを明示するためには，責任が委譲されることについての主張と，責任の委譲元と委譲先の主体を定義する必要がある．また，責任委譲についての主張に対する証拠が必要になる．したがって，図 4 に示すような，責任委譲主張に対するディペンダビリティケースが必要になる．

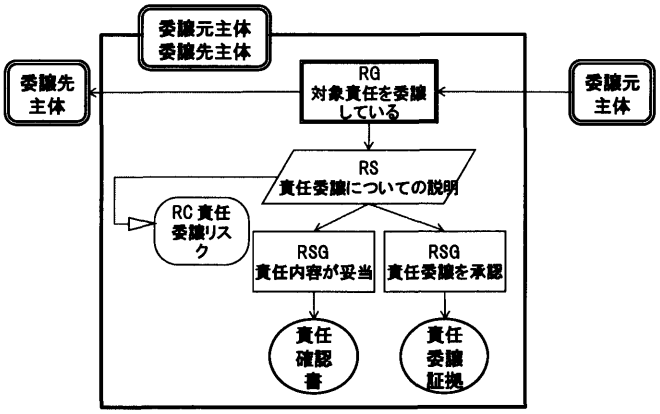

図 4　責任委譲主張に対するディペンダビリティケースの例

3.4 責任委譲関係の妥当性

ディペンダビリティケースに導入された責任の委譲関係が組織における対応する主体間の組織的関係と一貫している必要がある．たとえば，責任の委譲関係が組織の階層関係や横断的な統制関係を逸脱することは，現実には許されないことである．

したがって，責任属性が導入されたディペンダビリティケースに対して，責任委譲関係が組織構造に対して一貫していることを確認する必要がある．

3.5 責任属性の導入方法

ディペンダビリティケースに責任属性を導入する方法として，次の 2 つがある．

[方法 1] 名称拡張方式

ディペンダビリティケースの図式要素の名前に責任主体名を付与することにより責任属性を明示する方法

[方法 2] 図式拡張方式

ディペンダビリティケースの新たな図式要素として責任主体を追加して責任属性を明示する方法

名称拡張方式では，従来のディペンダビリティケースの図式表現を変更する必要がないので導入が容易である．しかし，責任主体ごとに主張や証拠をまとめて把握することが困難である．

これに対して，図式拡張方式では，従来のディペンダビリティケースの表記法を拡張する必要がある．しかし，責任主体ごとに対応する主張や証跡をまとめることができるだけでなく，責任主体間の関係を理解しやすいという特徴がある．

第 4 節で，図 1 の例を対象として，責任属性を導入することにより，両手法を具体的に比較する．

3.6 説明責任遂行方法

ディペンダビリティケースに責任属性を導入することにより，ディペンダビリティケースの主張，前提，説明，証拠に対して，責任主体を明示できる．

すなわち，責任属性が導入されたディペンダビリティケース D では，D を構成する要素としての主張，前提，説明，証拠に対して必ず責任主体が定義されている．したがって，障害が発生した要素が特定できれば，その要素について責任主体を特定できる．その要素が満足すべき上位の主張をディペンダビリティケースの構造を探索して，主体内の最上位の主張を決定できる．次いで責任委譲関係を探索することにより，影響を受ける責任委譲元のディペンダビリティケースの要素を特定できる．この手順を繰り返すことにより障害についての関係者としてのすべての責任主体と責任対象を明らかにできる．この内容に基づいて関係者が説明責任を遂行できる．

4. 責任属性表現方法の比較

4.1　名称拡張方式

図 1 のディペンダビリティケースに対して名称拡張方式で責任属性を明記した例を図 3 に示した．ただし，責任の委譲関係については記述していない．名称拡張方式では責任委譲関係については，別のディペンダビリティケースを作成することで対処できると考えられる．

この記述から，図式要素ごとに責任主体を明示できることが分かる．しかし，組織間の関係を直接理解するは容易ではない．このため，この図式記述から組織間の関係を図式要素間の接続関係に基づいて抽出する必

要がある．

説明責任遂行では，図式要素ごとに対応する，責任主体を特定できることから，階層関係を上位に探索することで影響を受ける責任主体の集合を決定して，ディペンダビリティの障害に対する説明責任の遂行を支援できる．

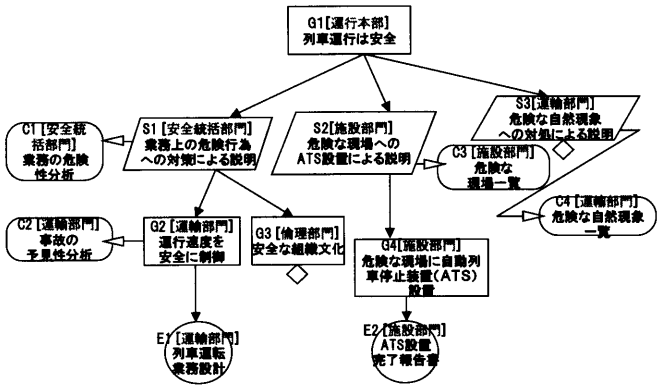

図 5　名称拡張方式に基づくディペンダビリティケース

4.2　図式拡張方式

図 1 のディペンダビリティケースに対して図式拡張方式で責任属性を明記した例を図 6 に示した．この図では責任の委譲関係についても記述している．図式拡張方式では責任委譲関係についても，同じディペンダビリティケースの図の上で，別のディペンダビリティケースを作成することなく記述できることが分かる．

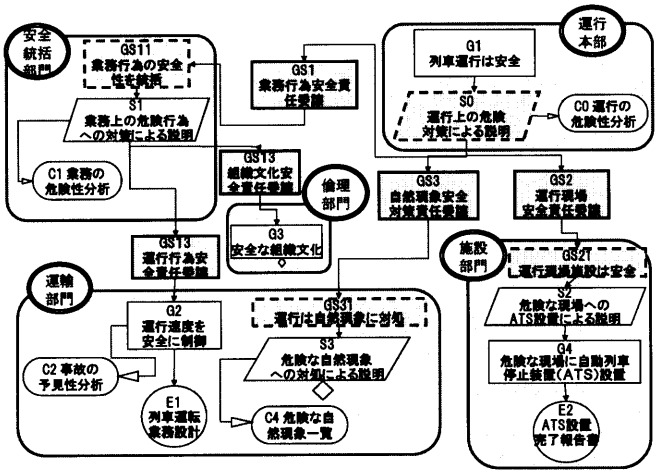

図 6　図式拡張方式に基づくディペンダビリティケース

ただし，責任の委譲元主体と委譲先主体との共通の主張である，責任委譲についての主張を分解していない．ここで，この責任委譲についての主体が委譲元と委譲先両方の主体についてのディペンダビリティケースであることを示す領域線を省略している．この理由は図が煩雑になるためである．

また，図 6 では，責任主体内に，点線を用いて記述された主張と説明を追加している．これらの主張と説明は図 1 にはないが，責任委譲関係を明確にするために追加した要素である．たとえば，S0 は，運行本部が主張 G1「列車運行安全」を，安全統括部門，運輸部門，施設部門に責任を委譲するために，GS1「業務行為安全責任委譲」，GS2「自然現象安全対策責任委譲」，GS3「運行現場安全責任委譲」に分解するための説明である．また GS 1 1「業務行為の安全性を統括」は，運行本部から安全統括部門に委譲された責任に対する安全統括部門内の主張である．

5　考察

5.1 有効性

ディペンダビリティケースに責任属性を導入する方法を具体化することにより説明責任を遂行できることを確認した．しかし，責任属性の導入方法について導入効率や説明責任遂行効率について有効性を評価する必要がある．また，提案した 2 種類の責任属性の導入法について有効性を定量的に比較評価する必要がある．

5.2　適用性

提案したディペンダビリティケースへの責任属性導入方法を他のシステム事例に適用することにより，適用性を評価していく必要がある．また，UML のシーケンス図や責任モデルなどとの連携可能性についても評価する必要がある．

5.3 定式化

主張の集合：P，前提の集合：C，説明の集合：S，証拠の集合：E，最上位の主張 P0，未定義の主張と説明の集合 U，接続関係 R に基づいて，ディペンダビリティケースを G＝<P,C,S,E, R, P0, U>で定式化できる．このとき，責任主体集合 A と，A に関する責任委譲関係は，A と，A の要素関係の対に対して G の要素を対応付けることによって定式化できる．

6 おわりに

本稿では，ディペンダビリティケースに責任属性を導入する上での課題を明らかにするとともに，具体的な責任属性の導入手法として名称拡張方式と図式拡張方式を提案し，各手法を比較した．

今後は本稿で提案したディペンダビリティケースへの責任属性導入法の記述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ディペンダビリティケースに基づく説明責任遂行を支援するための手法を

具体化する必要がある．説明責任をより良く遂行するためには，説明責任遂行を前提にしたディペンダビリティケースの作成方法が必要になる．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[8][9].

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] Jackson, D. et al, Software for dependable systems–sufficient evidence?, NATIONAL RESEARCH COUNCIL, 2008
[8] DEOSプロジェクト, http://www.crest-os.jst.go.jp
[9]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*
[10] Mario Tokoro eds., Open Systems Dependability, Dependability Engineering for Ever-Changing Systems, CRC Press, 2012
[11] D-Case エディタ, http://www.dependable-os.net/tech/D-CaseEditor/
[12] 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.
[13] 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
[14] 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.
[15] European Organisation for the Safety of Air Navigation, Safety Case Development Manual, 2nd ed., EUROCONTROL, Oct. 2006.
[16] 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.
[17] Axel van Lamsweerde and Emmanuel Letier, Integrating Obstacles in Goal-Driven Requirements Engineering, 1998
[18] Sommerville, I., Lock, R., Storer, T., Dobson, J.E., 2009. Deriving information requirements from responsibility models. In: Proceedings CAiSE 2009: 21st International Conference on Advanced Information Systems Engineering.Springer, London, UK, pp. 515–529
[19]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！～，ダイテックホールディング，2012，ISBN 978-4-86293-079-8
[20] 山本修一郎，松野裕，ディペンダビリティケース作成法に関する一考察，KBSE研究会，IEICE-112, vol. IEICE-SS-164, No. IEICE-KBSE-165, pp.61-66, 2012
[21] 松野 裕，高井利憲，ヴァイセ パテゥー，山本修一郎，アシュアランスケース構築法の提案，KBSE 研究会，2012
[22] Vaise Patu, Yutaka Matsuno, Shuichiro Yamamoto, Application of D-Case to the usage flow diagram scenario of the Distributed E-Learning System called KISSEL in Asian Pacific Universities, KBSE 研究会，2012
[23] 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，信学技報，vol. 112, no. 165, KBSE2012-18, pp. 37-42 2012
[24] 高間翔太，松野裕，山本修一郎，ディペンダビリティ・コンテクストの推定手法の提案，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，信学技報，vol. 112, no. 165, KBSE2012-36, pp. 145-150 2012
[26] 徳野達也，松野 裕，山本修一郎，TOGAF NEXT に対する ADM プロセステンプレートの提案，KBSE 研究会，2012
[27] 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，信学技報，vol. 112, no. 165, KBSE2012-20, pp. 49-54, 2012
[28] Vaise Patu, Yutaka Matsuno, Shuichiro Yamamoto, Research framework for dependability science based on assurance cases, IEICE Tech. Rep., vol. 112, no. 165, KBSE2012-21, pp. 55-59, July 2012
[29] 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，IEICE-112, vol. IEICE-SS-164, No. IEICE-KBSE-165, pp. 151-156,2012
[30] Shuichiro Yamamoto, Yutaka Matsuno, d* framework: Inter-Dependency Model for Dependability, DSN 2012
[31] Ewen Denney and Ganesh Pai, Ibrahim Habli, Perspectives on Software Safety Case Development for Unmanned Aircraft, DSN2012
[32] Adelard,www.adelard.com/asce/index.html
[33] AAMI, Safety Assurance Cases for Medical Devices, http://www.aami.org/meetings/courses/safety.html
[34] Dependability computing, assurance case course, http://www.dependablecomputing.com/courses.html