

## 活動記述に基づく活動完了基準作成法の提案

山本 修一郎

名古屋大学情報連携統括本部 〒464-8601 愛知県名古屋市千種区不老町

E-mail: syamamoto@acm.org

あらまし 工程活動の完了を確認するためには、工程内の活動について、作成された成果物と活動が完了基準を満たすことを確認する必要がある。本稿では、活動記述に基づいて、活動内容が必要な基準を充足することを確認することにより、活動の完了を保証する方法を提案する。さらに、提案手法を用いて、IT サービス継続性活動について、「ITIL サービスデザインプロセス 4.5 節 IT サービス継続性管理」に基づき、65 項目からなる活動完了基準を作成した結果について述べる。

キーワード 活動記述, 活動完了基準, 活動確認項目, 保証ケース

## A Task Completion Criteria Design Method based on the Activity Description

Shuichiro YAMAMOTO

Strategy Office, Information and Communication Headquarters, Nagoya University Furo-cho, Chikusa-ku, Nagoya,  
Aichi, 464-8601 Japan

E-mail: syamamoto@acm.org

**Abstract** To assure the completion of activities, it is necessary to develop check criteria for activities such as checklists. In this paper, the method is proposed to assure the completeness of activities by developing criteria constitutes check items for completing activities. We also explain that the method is effectively applied to develop the activity completion criteria. It includes 65 check list items for IT Service continuity activities based on the description of IT Service Design Process document.

**Keyword** Activity description, Task completion Criteria, Checklist, Assurance case

### 1. はじめに

ソフトウェア開発ならびに運用などの活動が完了したことを確認するためには、これらの活動が完了基準を満たすことを確認する必要がある。

保証ケース(Assurance case)では、立証したい主張を証拠に基づいてステークホルダ間で議論することによって、主張についての合意を形成できる[4]-[30]。また、テスト結果を証拠としてシステムの安全性を確認するために、保証ケースが用いられてきた。筆者らは、テストの十分性自体を保証ケースによって確認する方法を提案している。また、保証ケースを用いて工程完了基準を確認する手法を提案した。しかし、開発活動や運用活動は文書で記述されており、これらの文書に基づいて活動の完了基準を作成する手法は、これまで明確ではなかった。

本稿では、文書による活動記述を用いて活動の完了を保証するための確認項目を作成しておき、この確認項目の達成を確認することで、活動の完了を保証する方法を提案する。

以下では、2 節で関連研究について述べる。3 節で活動記述に基づく確認項目からなる活動完了基準の作成法と、確認項目の達成を確認することによる活動完了確認手法を提案する。4 節で提案手法の具体例を説明する。5 節で考察を述べ、最後に 6 節でまとめと今後の課題を明らかにする。

### 2. 関連研究

以下では、保証ケースと工程の完了基準についての研究動向を紹介する。

#### 2.1. 保証ケース

保証ケースの構成要素は、主張 (claim), 説明 (strategy), コンテキスト(context), 証拠 (evidence),

未展開である。構成要素間の関係は、主張、説明、証拠を関連付ける矢線と、主張と説明をコンテキストと関連付ける白抜きの矢線の2種類である。

【主張】システムが達成すべき性質を示す矩形。下位主張や戦略に分解される

【説明】主張の達成を導くために必要となる論証を示す平行四辺形。下位主張や下位説明に分解される。

【コンテキスト】主張や説明が必要となる理由としての外部情報を示す楕円。

【証拠】主張や説明が達成できることを示す証拠

【未展開】まだ具体化できていない主張や説明を示すひし形

保証ケースの例を図1に示す。この図では「システムがディペンダブルである」という主張 G1 を前提条件に基づいて S1 で3つの下位の主張 G2「前提条件の下でシステムを正常に運用可能」、G3「発生した想定内の逸脱に対してシステムを継続運用可能」、G4「発生した想定外の逸脱に対してシステムを復旧可能」に分解することによって説明している。G2,G3,G4 の主張が成立する根拠として、それぞれ3つの証拠 E1「前提条件に基づくシステム開発完了報告」、E2「前提条件に基づく想定リスク対策実施完了報告」、E3「想定外の逸脱に対する運用プロセス定義完了報告」が提示されている。

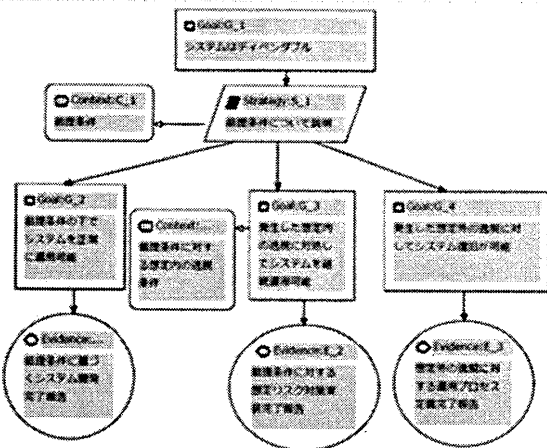


図1 保証ケースの例

ISO/IEC 15026 では、保証ケースの構造と内容に対する最低限の要求を規定している。保証ケースには、システムや製品の性質に対する主張 (claim)、主張に対する系統的な議論(argumentation)、この議論を裏付ける証拠(evidence)と明示的な前提(explicit assumption)が含まれる。ISO/IEC 15026 では、保証ケースが持つべき構造と内容を規定対象としており、保証ケースの品質については規定していない。また保証ケース (Assurance case) については、ISO/IEC 15026[31]やOMG の ARgumentation Metamodel(ARM)[32] と Structured Assurance Case Metamodel (SACM) [33]などで標準化がすすめられている。ISO/IEC 15026 では、対象範囲、適合性、利用法、保証ケースの構造と内容、

適用成果物などについて規定している。

最近では ISO26262 で、自動車分野における機能安全を保証するために、開発対象だけでなく開発プロセスの安全性についても保証ケースの適用が推奨されている。しかし、工程完了基準に対して保証ケースをどのように適用すべきかについての具体的な言及はない。

## 2.2. 工程完了基準

サービスの提供判断条件を明確にして十分なリスク対策を施しておかないと、サービス開始後に様々な問題が発生する可能性がある。このため、筆者らは、保証ケースを用いたサービス開始判断方法を提案した[2]。この手法では、サービスの提供判断のための保証項目を挙げるために、以下のようなプロセスを明らかにした。まず、現状(As-Is)の項目をリストアップする。すでに存在する作業項目や成果物及び成果物の目次を参考にする。続いて、あるべき姿(To-Be)の項目もリストアップする。さらに、フェーズの前後関係から、整合性が取れていないと考えられるものを新たにリストアップする必要があることを明らかにした。たとえば、運用フェーズで「監視対象の監視を実行している」という保証ケースがある場合、企画フェーズに「必要な監視対象を定義している」がなければならない。

なお、この手法は、itSMF が策定している IT サービスのストラテジ、デザイン、トランジション、オペレーション、継続的改善に関する包括的なガイドラインである ITIL[3]に基づいている。

## 3. 活動記述に基づく完了基準確認項目の抽出

以下では、活動記述を分析することにより、活動の完了を保証するための確認項目を抽出する手順について述べる。

### 3.1. 活動と完了基準の関係

活動は複数の階層構造から構成されている。たとえば、ITILでは、①サービス活動サイクル、②活動工程、③活動、④活動段階、⑤要素活動という5段階で活動が記述されている。したがって、活動の完了基準を要素活動に対して定義すれば、対象とする活動を構成するすべての下位の活動の完了基準を要素活動の完了基準を合成することによって構成できることになる。

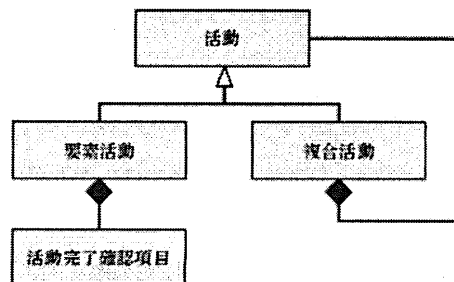


図2 活動のメタモデル

この考察から、活動のメタモデルを図2に示すように定義できる。すなわち、活動には要素活動と複合活動がある。要素活動には、複数の活動完了基準が対応する。複合活動には、複数の活動が含まれる。

### 3.2. 動確認項目の作成手順

活動確認項目の作成手順は、以下のようになる。

**【手順1】活動構造の分析**

活動には、複合活動と要素活動があるため、複合活動の階層構成を明らかにした後に、要素活動を明らかにする必要がある。これによって、活動全体の階層的な構造を明らかにすることができる。

**【手順2】要素活動記述の抽出**

要素活動が識別できると、対応する要素活動の記述を抽出する。

**【手順3】要素活動の状態分析**

要素活動記述に基づいて、要素活動における状態を分析することにより、要素活動を開始する契機とそれに基づく開始状態から、要素活動が完了した終了状態に至るまでの重要な状態を明らかにする。

**【手順4】要素活動状態に対する確認項目の作成**

要素活動の各状態に到達するための条件を、要素活動の完了基準としての確認項目として作成する。この理由は、もし要素活動が完了したとすると、要素活動に対応するすべての状態が完了しているはずだからである。

**【手順5】活動全体に対する確認項目の作成**

要素活動ごとに作成した確認項目を活動構造に基づいて統合する。

この活動の完了基準が達成されたことを確認するための活動確認項目表の構成例は、表1のようになる。表1では、上位活動に対して複数の要素活動があり、要素活動ごとに複数の確認項目がある場合の構成例を示している。

表1 活動確認項目表の構成例

上位活動	要素活動	確認項目
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>
		<input type="checkbox"/>

## 4. 具体例

以下では、上述した活動完了基準を定義する活動確認項目表の作成手順を ITIL「サービスデザイン」における「IT サービス継続性管理」に対する活動記述の例

[3]を用いて、手順ごとに説明する。

### 4.1. 活動構造の分析

ITIL に対する活動サイクルを分析した結果を表2に示す。

次に、ITIL「サービスデザイン」における「IT サービス継続性管理」に対する活動を分析すると、表3に示すように、開始段階、要求と戦略段階、導入段階、継続的な運用段階に分類されており、活動段階ごとに要素活動が記述されていることが分かる。

### 4.2. 要素活動記述の抽出

開始段階における「方針の設定」の記述を抽出すると、以下のようになる。

**「■方針の設定**

これは、事業継続性に関する、あるいはその影響を受ける組織のすべてのメンバが、ITSCM に従い、ITSCM を支持する責任を認識するよう、できるかぎり迅速に確立され伝えられるべきである。最小限、方針では管理の意図と達成目標を設定するべきである」

表2 ITIL 活動サイクル

活動サイクル	活動
サービスストラテジ	①市場の定義、②提供内容の開発、③戦略的資産の開発、④実行の準備、⑤サービスポートフォリオ管理、⑥需要管理、⑦財務管理
サービスデザイン	①サービス設計、②サービスカタログ管理、③サービスレベル管理、④キャパシティ管理、⑤可用性管理、⑥IT サービス継続性管理、⑦情報セキュリティ管理、⑧サプライヤ管理
サービスランジョン	①移行計画立案、②変更管理、③サービス資産管理・構成管理、④知識管理、⑤リリース管理・展開管理、⑥サービス妥当性確認・テスト、⑦評価
サービスオペレーション	①イベント管理、②インシデント管理、③要求実現、④問題管理、⑤アクセス管理
継続的サービス改善	①改善プロセス、②サービス報告、③サービス測定

表3 ITサービス継続性管理活動

段階	要素活動	
開始	①方針の設定、②委任事項・適用範囲の規定、③リソースの割り当て、④組織・統制構造の定義、⑤プロジェクトと品質計画の合意	
要求と戦略	要件定義	①ビジネス影響分析、②リスク分析
	戦略策定	①リスク対応手段、②復旧手段選択
導入	①サービス継続性計画策定、②災害対策組織、③計画のテスト	
継続的な運用	①教育トレーニング、②レビュー、③テスト、④変更管理、⑤発動（最終テスト）	

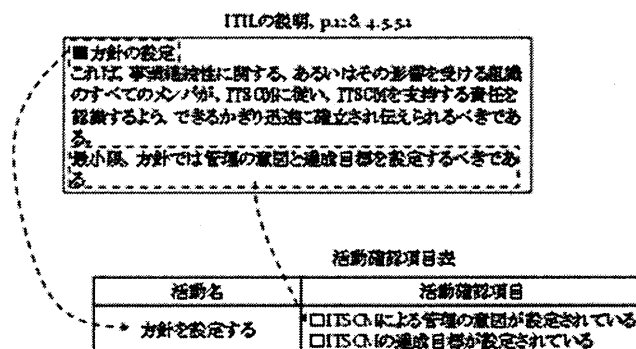


図2 要素活動に基づく活動確認項目の抽出例

表4 開始段階に対する活動確認項目表の例

開始段階の活動	活動確認項目
方針を設定する (2)	<input type="checkbox"/> ITSCMによる管理の意図が設定されている <input type="checkbox"/> ITSCMの達成目標が設定されている
委任事項・適用範囲の規定 (4)	<input type="checkbox"/> 適用範囲が定義されている <input type="checkbox"/> 組織内の全スタッフの責任が定義されている <input type="checkbox"/> 責任に応じたタスクが記載されている <input type="checkbox"/> コンプライアンスの課題を考慮している
リソースの割り当て (2)	<input type="checkbox"/> 適切にコストと人材のリソースが配分されている <input type="checkbox"/> 組織によるプロセス進行の独立性を考慮している
プロジェクトの組織・統制構造の定義 (3)	<input type="checkbox"/> 組織が定義されている <input type="checkbox"/> 統制構造が定義されている <input type="checkbox"/> 組織構造の複雑化を低減するために、PMBOKやPRINCEを考慮している
プロジェクト品質計画の合意 (3)	<input type="checkbox"/> 品質計画が定義されている <input type="checkbox"/> 品質計画では成果に対する許容可能な品質レベルが定義されている <input type="checkbox"/> リソース要件の下で、成果物の品質計画についてすべての当事者が合意している

表5 確認項目数

段階	確認項目数
開始	14
要求と戦略	14
導入	24
継続的な運用	13
合計	65

### 4.3. 要素活動の状態分析

抽出した「方針の設定」の記述は、2つの文で構成されている。第1文は概要の説明である。第2文には、この活動において実施すべき事柄が述べられている。したがって、第2文から、以下の2つの状態が「方針の設定」活動にあることが分かる。

- [状態1]ITSCMによる管理の意図が設定されている
- [状態2]ITSCMの達成目標が設定されている

この2つの状態が達成されれば、第1文で述べられている「方針の設定」活動が完了できると考えられる。

### 4.4. 要素活動状態に対する確認項目の作成

4.3で示した手順で抽出された状態を要素活動「方針の設定」の確認項目に転記することにより、活動確認項目表を作成する。

上述した要素活動記述に基づく活動確認項目の抽出例を図2に示す。

### 4.5. 活動全体に対する確認項目の作成

上述の手順をすべての要素活動に対して実施することにより、活動全体に対する確認項目表を作成する。表4では、開始段階に対する活動確認項目表の例を示している。ここで、カッコ内の数字は確認項目の個数である。また、ITSCMの確認項目数の合計と段階ごとの内訳を表5に示す。

## 5. 考察

以下では、上述した活動完了基準の作成法について考察する。

### 5.1. 活動確認項目表

活動確認項目表を用いることにより、活動完了についての基準を、活動状態の観点から網羅的に抽出できる。これによって、活動の完了条件が明確化できる。

活動確認項目表により、従来の活動記述では文書中で明示的に触れられていなかった状態確認項目を系統的に抽出できることを明らかにした。一方、提案した手法では、文書記述から活動状態を抽出する作業が属人的になることになり、重要な活動状態を適切に選択できるかという課題がある。このため、文書から状態についての記述を識別するための観点を整理しておくなどの配慮が必要になる。

### 5.2. 活動完了基準

前節で示したように、本提案によって、自然言語で記述された要求仕様に基づいて、活動確認項目を網羅的に抽出できることから活動の完了基準の充分性について、ステークホルダ間で合意形成できることは明らかである。

本手法では、活動について記述された文書に基づいて段階的に活動確認項目を作成できるだけでなく、抽出した確認項目が必要な理由を明確に説明できる。たとえば、確認項目が、前提とした文書のどこに記述されていたかを追跡できる。したがって、選択した活動確認項目が十分であることを客観的に説明できる。

また、自然言語による活動記述から段階的に活動完了の確認項目を抽出できるので、実際のプロジェクトに適用することは容易であると考えられる。したがって、今後、本手法を実際のプロジェクト活動に適用する研究についても進めていく予定である。

### 5.3. 適用範囲

本稿で対象とした事例は広く普及している ITIL 文書であり、本手法が他の開発・運用標準の記述にも適用できることは明らかである。

### 5.4. 限界

本稿では、提案手法が ITIL 活動の一部に対して適用できることを示した。しかし、どのような能力の要員がどれくらいの工数で活動完了基準を作成できるかなどの生産性や品質に関する有効性については、定量的に評価していない。このため、本手法の有効性について定量的な効果を明らかにしていく必要がある。

## 6. まとめと今後の課題

本稿では、運用活動を対象として活動記述による活動完了基準を作成する手法について提案した。適用対象とした事例は 1 件だけであるが、手法としての一般性は高いので、他の事例についても適用できると考えている。

また、本稿の内容は手法の実行可能性を評価するための試行評価の段階にとどまっているため、定量評価までは至っていない。今後、実際のソフトウェア開発・運用工程を対象とする評価実験を進める予定である。

## 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 [34][35][36].

## 文 献

- [1] 山本修一郎, 保証ケース手法に基づくテスト充分性に関する合意形成手法の提案、人工知能学会、KSN 研究会、2013
- [2] 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, KBSE, 知能ソフトウェア工学 111(489), 7-12, 2012-03-08
- [3] itSMF Japan, ITILV3 ファウンデーションハンドブック日本語版, 2009
- [4] Kelly, T. P, A Six-Step Method for the Development of Goal Structures, York Software Engineering, 1997
- [5] T. Kelly. "Arguing Safety, a Systematic Approach to Managing Safety Cases". PhD Thesis, Department of Computer Science, University of York, 1998
- [6] 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.
- [7] I. Bate, T. Kelly, Architectural considerations in the certification of modular systems, Reliability Engineering and System Safety 81, pp.303-324,2003
- [8] 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
- [9] 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.
- [10] Jackson, D. et al, Software for dependable systems-sufficient evidence?, NATIONAL RESEARCH COUNCIL, 2008
- [11] 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
- [12] 松野裕, 高井利憲, ヴァイセ パテウー, 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
- [13] 松野裕, 山本修一郎, ユースケース分析に基づくディペンダビリティケース作成法の提案, KBSE 研究会, IEICE-112, vol. IEICE-KBSE-419, KBSE2012-61, pp.19-24
- [14] 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
- [15] 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
- [16] 高間翔太, 松野裕, 山本修一郎, ディペンダビリティ・コンテキストの推定手法の提案, KBSE 研究会, 信学技報, vol. 112, no. 314, KBSE2012-42, pp. 25-30, 2012
- [17] 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
- [18] 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, 信学技報, vol. 112, no. 314, KBSE2012-55, pp. 103-108, 2012
- [19] 山本修一郎, 松野裕, ディペンダビリティケースへの責任属性導入法の検討, KBSE 研究会, 信学技報, vol. 112, no. 314, KBSE2012-52, pp. 85-90, 2012
- [20] 松野裕, ヴァイセ バトゥ, 山本修一郎, アシユ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
- [21] 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
- [22] 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, 信学技報 vol.112, No.165, pp.151-156, 2012
- [23] Shuichiro Yamamoto, Yutaka Matsuno, d\* framework: Inter-Dependency Model for Dependability, DSN 2012
- [24] Robin Bloomfield and Peter Bishop, Safety and Assurance Cases: Past, Present and Possible Future – an Adelard Perspective, 2010 [4] 山本修一郎, 松野裕, ディペンダビリティケース分解パターンについての考察, KBSE 研究会, 2013.3.15
- [25] 山本修一郎, 松野裕, ディペンダビリティケース分解パターンについての考察, KBSE 研究会, 信学技報, vol. 112, no. 496, KBSE2012-80, pp. 67-72, 2013
- [26] 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
- [27] 松野裕, 山本修一郎, 実践 D-Case, ～ディペンダビリティ・ケースを活用しよう!～, ダイテックホールディング, 2013, ISBN 978-4-86293-091-0
- [28] D-Case 実証評価研究会, <http://dcase.jp/>
- [29] D-Case エディタ, <http://www.dependable-os.net/tech/D-CaseEditor/>
- [30] 松野裕, 山本修一郎, アシユアランスケースツールへのプログラミング言語技術の適用, KBSE 研究会, 信学技報, vol. 112, no. 496, KBSE2012-81, pp. 73-78, 2013
- [31] ISO/IEC 15026, Systems and software engineering -- Systems and software assurance -- Part 2: Assurance case, 2011
- [32] OMG, ARgumentation Metamodel (ARM), <http://www.omg.org/spec/ARM/Current>
- [33] Structured Assurance Case Metamodel (SACM), <http://www.omg.org/spec/SACM/>
- [34] DEOS プロジェクト, <http://www.crest-os.jst.go.jp>  
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](http://www.dependable-os.net/ja/topics/file/White_Paper_V3.0J.pdf)
- [35] Mario Tokoro eds., Open Systems Dependability, Dependability Engineering for Ever-Changing Systems, CRC Press, 2012