

## システムダイナミクスによる IT サービスプロセスの評価

山本 修一郎

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** IT サービスの継続性を保証するために、ディペンダビリティケースを用いた手法が提案されている。しかし、ディペンダビリティケースを用いて開発されたシステム事例がないために、ディペンダビリティケースの有効性についての客観的な効果が良く分からないという問題があった。本稿では、この問題に対処するために、ディペンダビリティケースの有効性を定性的に確認するためにシステムダイナミクスを用いたシミュレーション手法を提案するとともに、シミュレーション評価結果に基づいてディペンダビリティケースの有効性を議論する。

**キーワード** ディペンダビリティケース, サービス継続性, システムダイナミクス, シミュレーション

## Evaluation of IT Service Process with System Dynamics

Shuichiro Yamamoto

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

E-mail: syamamoto@acm.org

**Abstract** Several methods on dependability case are proposed to assure IT service continuity. As these methods have few industrial applications, the effectiveness of dependability case is still unclear. In this paper, a simulation modeling method is proposed to evaluate the effectiveness of IT service process using dependability case. We also discuss the effectiveness of dependability case based on simulation results.

**Keyword** Dependability case, Service continuity, System dynamics, Simulation

## 1 はじめに

筆者は 2011 年 6 月から DEOS プロジェクトに参画して、要求変化に対応するためのディペンダビリティエンジニアリング手法について研究を進めている。

本プロジェクトから筆者に解決を要請された課題の一つが、ディペンダビリティを保証するための要求マネジメント手法を開発することであった。

本稿では、まず 2 節で DEOS プロセスとは何かについて述べ、次いで 3 節では DEOS プロセスにおける要求マネジメントモデルを提案する。提案した要求マネジメントモデルに基づいて作成したシステムダイナミクスモデルを 4 節で述べる。5 節でシステムダイナミクスモデルを用いた定性的なシミュレーション結果を示す。6 節で考察、7 節で関連研究について述べる。最後に、まとめと今後の課題を明らかにする。

## 2 DEOS プロセス

科学技術振興機構の戦略的創造研究推進事業 CREST における DEOS( Dependability Engineering for Open Systems) プロジェクトでは、「変化しつづける目的や環境の中でシステムを適切に対応させ、継続的にユーザが求めるサービスを提供することができるシステムの構築法を開発すること」を目標としている。このため、これまで機能、構造、境界が固定的なシステムを対象に考えられてきたディペンダビリティ技術だけでは不十分であることから、新たな構築法の基礎となる概念として、「オープンシステムディペンダビリティ」

の考え方を提案している。この考え方に基づいて DEOS プロセスと呼ばれる反復のプロセスの研究を進めている。図 1 に示すように DEOS プロセスは目的環境変化対応サイクルと障害対応サイクルからなる [1]。目的環境変化対応サイクルでは、要求マネジメントにより、要求抽出とリスクマネジメントを実施するとともに、ステークホルダと合意する。このとき、D-Case と呼ばれる保証ケースと、D-Script と呼ばれる実行可能シナリオを作成する。D-Case と D-Script を合意記述データベースに格納しておき、開発工程と運用工程で利用する。保証ケースによってシステムのディペンダビリティの保証方法についてステークホルダと合意することができる。運用時に発生する逸脱事象に対しても D-Script で記述された実行可能シナリオによる迅速対応についても事前に合意することができる。

照)。

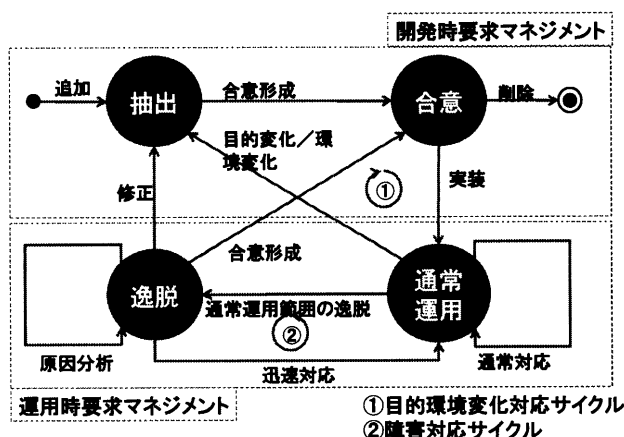


図 2 要求状態管理モデル

まず、要求がステークホルダから抽出される。抽出された要求が互いに矛盾する可能性があるため、合意形成によってステークホルダ間で要求について合意する必要がある。合意された要求は実装されると通常運用される。目的や環境が変化すると、通常運用されていた要求が陳腐化して新しい要求が抽出されることになる。これは変化対応サイクルである。対応する通常運用範囲からの逸脱が発生すると、要求が逸脱状態に遷移する。迅速対応が可能なら、通常運用状態に復帰する。これは障害対応サイクルである。もし逸脱状態の要求に対してサービス継続シナリオが有効に機能しない場合、これらの要求を変更して新たな要求を抽出する必要がある。また逸脱の原因が要求ではなく、実装に問題があった場合には、実装をやり直すために、要求の合意状態に遷移する。抽出状態と合意状態の要求は開発時に管理される。これに対して通常運用状態と逸脱状態の要求は、運用時に管理される。システム状態は要求状態の集合で表現される。したがって、図 2 の要求状態は個々の要求ごとに管理されることになる。

時間軸に沿って要求状態が変化する様子を図 3 に例示する。

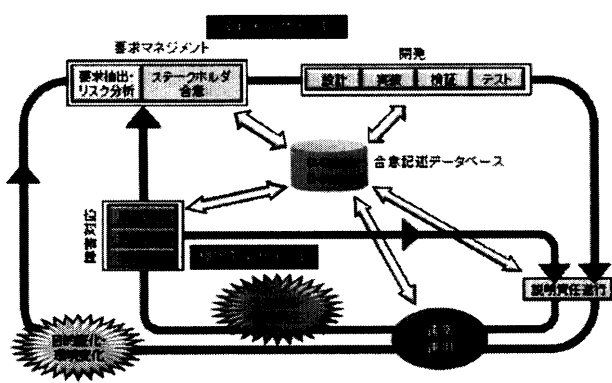


図 1 DEOS プロセス

障害対応サイクルでは、上述したように通常運用時に発生する障害について予め合意した迅速対応によって障害から回復できる。また予めリスク分析されている障害発生の予兆についても検知できるので未然に対応できる。このように迅速対応した結果について、合意記述データベースと実行記録の証跡に基づいて説明責任を遂行できる。また、合意記述データベースに格納した保証ケースによって障害原因の究明を支援することができる。

なお、要求マネジメントにおいて作成した合意記述データベースを利用することでシステム開発の信頼性を向上できる。また、通常は開発工程の中に要求工程を含めることが多いが、DEOS プロセスでは、運用と要求を一貫して対応づけるためにあえて、開発工程の外に配置していることを注意しておく。

以下では、DEOS プロセスの有効性を評価するために、要求に着目した要求状態管理モデルを提案する。

### 3 DEOS 要求状態管理モデル

要求マネジメントでは、上述した DEOS プロセスの目的環境変化対応サイクルと障害時の迅速対応サイクルを管理できるように、要求の状態を管理する。この場合、要求状態には、抽出、合意、通常運用、通常運用範囲からの逸脱という 4 種類の状態がある(図 2 参

		要求発展 →				
時間		R1	R2	R3	R4	R5
T1	抽出	抽出	—	—	—	—
T2	合意	合意	抽出	—	—	—
T3	運用	運用	合意	抽出	抽出	抽出
T4	運用	逸脱	合意	合意	合意	合意
T5	運用	運用	運用	運用	運用	運用
T6	運用	運用	運用	運用	運用	逸脱
T7	運用	逸脱	運用	運用	運用	運用
T8	運用	合意	運用	逸脱	逸脱	逸脱
T9	運用	運用	運用	合意	抽出	抽出
T10	運用	運用	運用	—	合意	合意

図 3 要求状態の変化例

たとえば、時刻 T1 で要求 R1,R2 が抽出されている。

時刻 T2 では、R1 と R2 が合意され、新たに R3 が抽出されている。時刻 T3 で、R1 と R2 が運用状態に移行し、R3 が合意状態、R4 と R5 が抽出されている。時刻 T4 で要求 R2 に逸脱が発生し、T5 で運用状態に回復している。これは迅速対応の結果である。また時刻 T7 で再度逸脱が発生すると、T8 では合意状態に遷移している。これは逸脱した原因が実装に合ったため合意状態に戻って要求を再度実装しなおすことになったためである。時刻 T10 では要求 R4 が削除されている。

#### 4 システムダイナミクスによる要求管理モデル

基本的には、図 2 に示した要求状態管理モデルに基づいて、各要求状態をストック、要求状態間の遷移をフローに対応づけることで、システムダイナミクスモデルを作成することができる。

システムダイナミクスモデルでは、ストック、フロー、パラメータを定義する。ストックには、要求状態、成果物、要求モデル、関係者（ステークホルダ）が対応する。フローには、状態遷移が対応する。パラメータには、関係者の能力と目的と環境の変化率が対応する。

ストックに対応する要求状態として、抽出済み要求、合意済要求、実装済要求、運用中要求、継続中要求、不変の要求、逸脱した要求、逸脱し続ける要求を定義した。図 2 で定義した抽出、合意、通常運用、逸脱という 4 個の要求状態には、抽出済み要求、合意済要求、運用中要求、逸脱した要求が対応している。実装済要求、継続中要求、不変の要求、逸脱し続ける要求を追加した理由は、次のとおりである。すなわち、要求に対して実装活動を考慮したこと、運用中の要求が逸脱する要求と逸脱しない要求に分かれること、継続中の要求が目的環境変化によって新しい要求が抽出される場合と、不変のままの要求に分かれるためである。

成果物ストックとして、D-Case、コード、運用手順書、説明責任遂行を定義した。説明責任遂行をストックとした理由は説明責任遂行結果としての事実を計上するためである。

関係者（ステークホルダ）ストックとして、サービス提供者、スポンサ、D-Case 専門家、開発者、エンドユーザ、運用者を定義した。

要求フローとして、要求の抽出、要求の非合意、要求の合意、要求の実装、要求の運用、要求の逸脱、要求の継続、要求の維持、逸脱からの回復、逸脱した要求の放置、逸脱した要求の抽出、再実装についての合意、目的・環境変化に基づいて要求抽出を定義した。図 2 の状態遷移には、追加、合意形成、削除、実装、通常対応、目的変化/環境変化、通常運用範囲の逸脱、迅速対応、修正、合意形成、原因分析がある（11 個）。要求の実装、要求の継続、要求の維持、逸脱した要求の放置を追加しているのはストックで要求状態を追加したことに対応している。

要求以外のフローとして、実装漏れ、運用設計漏れ、D-Case の作成、開発、運用の設計、逸脱の原因分析、要求の目的・環境変化分析を定義した。実装漏れと運用設計漏れに対するフローはそれぞれ、開発と運用の設計で漏れが生じることを表現したためである。

D-Case の作成フローによって、保証ケースを作成している。ここで、D-Case の作成には漏れが生じないと仮定した。逸脱の原因分析フローと、要求の目的・環境変化分析フローによって説明責任遂行ストックが作成されるとした。

能力パラメータとして、要求委員会、D-Case 専門家能力、開発者能力、運用設計能力、説明責任遂行能力を定義した。また、比率パラメータとして、目的変化率、環境変化率、エンドユーザによる運用されている要求の利用率、サービス継続シナリオ充足率を定義した。

これらのパラメータを変化させることで、ストック量がどのように変化するかを定性的に観察することができる。

これらをまとめると、表 1 のようになる。この表の例では、ストック数 18、フロー数 22、パラメータ数(8)となっている。

表 1 DEOS システムダイナミクスモデルの構成要素

構成要素	内容	
ストック	要求状態	抽出済み、合意済、実装済、運用中、継続中、不変の要求、逸脱した要求、逸脱し続ける要求
	成果物	D-Case、コード、運用手順書、説明責任遂行
	関係者	サービス提供者、スポンサ、D-Case 専門家、開発者、エンドユーザ、運用者
フロー	要求	要求の抽出、要求の非合意、要求の合意、要求の実装、要求の運用、要求の逸脱、要求の継続、要求の維持、逸脱からの回復、逸脱した要求の放置、逸脱した要求の抽出、再実装についての合意、目的・環境変化に基づいた要求抽出、要求の継続、要求の維持
	その他	実装漏れ、運用設計漏れ、D-Case の作成、開発、運用の設計、逸脱の原因分析、要求の目的・環境変化分析
パラメータ	能力	要求委員会、D-Case 専門家能力、開発者能力、運用設計能力、説明責任遂行能力
	比率	目的変化率、環境変化率、利用率、サービス継続性シナリオ充足率

構成要素間の関係には、パラメータからフローへの関係、ストックからフローへの関係、ストックからパラメータへの関係がある。

パラメータからフローへの関係には、D-Case 検討能力から D-Case の作成フローへの関係や、要求委員会の能力から要求の抽出フローへの関係などがある。

ストックからフローへの関係には、抽出済み要求ストックから要求の合意フローへの関係や D-Case から

合意済み要求への関係などがある。

ストックからパラメータへの関係には、サービス提供者やスポンサから要求委員会の能力への関係がある。

上述した DEOS システムダイナミクスモデルを iThink™ で記述した結果を付図 1 に示す。

## 5 シミュレーション評価

### 5.1 要求状態の変化傾向

運用までの要求の状態について、D-Case 検討能力が低い場合と高い場合を比較すると、図 4、図 5 に示すようになる。

D-Case 検討能力が低い場合、D-Case の蓄積も低調であり、合意済要求と実装済要求の増加は低調である。運用中の要求は継続的に増加している。

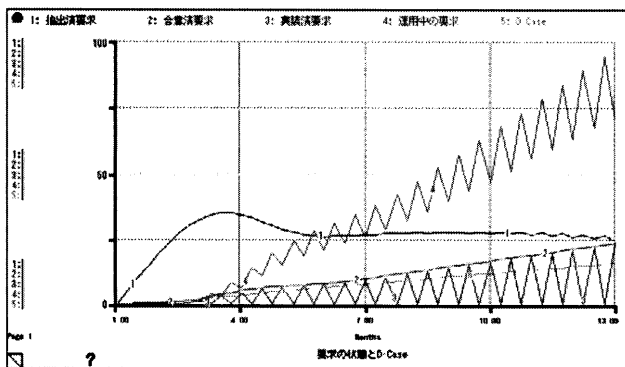


図 4 D-Case 作成能力が低い場合

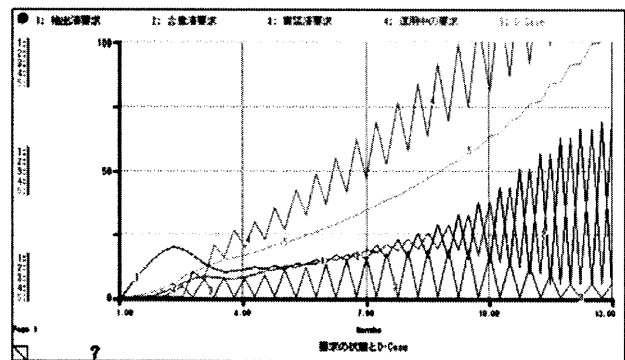


図 5 D-Case 作成能力が高い場合

D-Case 検討能力が高い場合、D-Case の蓄積が増加し、要求の抽出、合意済の要求が増加する。運用中の要求の増加率も早く高くなる。また D-Case の増加により、実装済要求が運用中の要求に効率的に移行している。この理由は DEOS 要求マネジメントプロセスで要求の状態が効率的に移行しているためであると考えられる。

### 5.2 逸脱要求

要求の逸脱に関する状態について、D-Case 検討能力が低い場合と高い場合を比較すると、図 6、図 7 に示すようになる。

D-Case 検討能力が低い場合、D-Case の増加量が低いため、逸脱し続ける要求と逸脱した要求の放置が増加する。また逸脱からの回復が低調である。

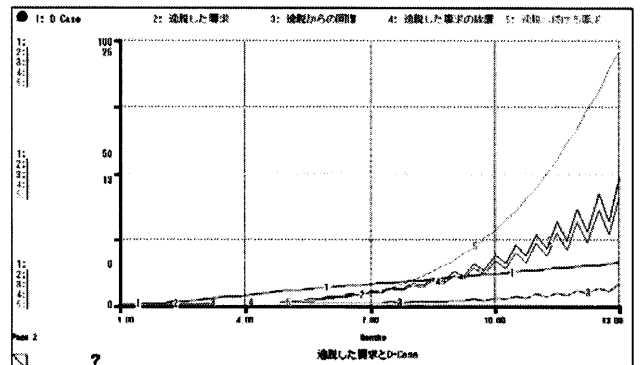


図 6 D-Case 作成能力が低い場合

D-Case 検討能力が高い場合、D-Case の順調な増加により、7 単位期間を過ぎると、逸脱した要求の放置がなくなる。また、逸脱からの回復が 10 単位期間を過ぎると収束する。7 単位期間以降に、逸脱した要求の放置がなくなっている。D-Case の記述量が増大するので逸脱した要求量が増大する一方、逸脱からの回復も迅速になっている。

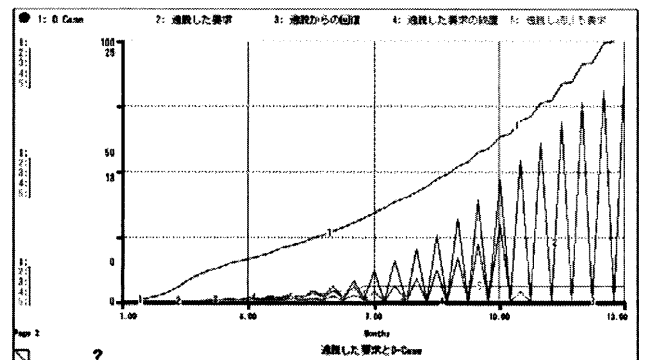


図 7 D-Case 作成能力が高い場合

図 7 で、逸脱からの回復が振動しているのは、逸脱に対して再度合意形成するための時間が必要になるためである。

## 6 考察

### 6.1 有効性

DEOS プロジェクトでは次の 3 条件を満足するように要求の変更管理方式を提案している。

(条件 1) 要求変更に対するディペンダビリティを保証できること

(条件 2) 要求変更に対する合意形成を容易化できること

(条件 3) 変更要求に対する説明責任の遂行を容易化できること

上述したように、D-Case によってこれらの条件を満足することを定性的に確認できた。

## 6.2 シミュレーションの効果

本稿で提案した DEOS プロセスの評価分析手法では、上述したようにシステムダイナミクス手法を利用している。この理由は、DEOS プロセスを適用したシステムが現存しないためである。

## 6.3 ソフトウェア開発文書の有効性評価

本稿で提案した手法によって、要求マネジメントプロセスにおいて、保証ケースというソフトウェア開発文書の効果を評価することができるようになった。

## 6.4 限界

提案方法は定量的なデータがないため、定性的な評価しかできていない。しかし、複雑な開発プロセスをシミュレーションによって定式化できたので、要求の観点から見たソフトウェアライフサイクルの全体の挙動を可視化できるようになった点が重要である。

## 7 関連研究

### 7.1 SD による開発プロセスの評価

これまでに、ソフトウェア開発におけるプロジェクト管理に対して多数の SD(System Dynamics)モデルが開発されてきた[2]。これらの研究では、計画、要員計画などの管理面と、設計、製造、レビュー、テストなどの生産活動面を統合している。

互いに関連するアジャイルプラクティスとして、顧客参画、変更管理、アジャイル計画制御、リファクタリングと設計品質を SD モデルでシミュレーションすることにより、アジャイル開発プロセスが分析されている[3]。

SD モデルを用いてインスペクションが開発サイクルを通して経費、計画、品質に及ぼす効果をシミュレーションする手法が提案されている[4]。またコードインスペクションプロセスに対して離散イベントモデルでシミュレーションする方法を提案している。活動計画、準備会議、準備、インスペクション会議、やり直し、フォローアップに対する SD モデルを評価している[5]。

要求の合意形成を支援する EasyWinWin プロセスを SD モデルでシミュレーションする方法が提案されている[6]。

ソフトウェア開発のリスク管理についてのシミュレーションモデルが研究されている[7]。またソフトウェア発展に対して SD モデルを適用する研究がある[8]。

### 7.2 要求変更管理プロセス

従来の要求工学の教科書[9][10][11][12][13][14][15]で述べられている変更管理プロセスでは要求変更活動を、準備、体制構築、実施、完了に分類している。

REOS モデル[16]では、①要求の誤りが R に、②利害関係者の知識の変化、③技術・計画・経費、④優先順位の変更が E に、⑥組織変更が O に、⑤システム環境の変化が S に、それぞれ対応する。

### 7.3 保証ケース

保証ケース (Assurance Case) については GSN( Goal

Structuring Notation)[17]に基づく手法が提案されている[18]。D-Case は GSN を拡張して実行時に証跡の妥当性を監視できるようにしている[19]。

## 7.4 新しいシステム工学

最近の新しいシステム工学でも 2 重ループに基づく複雑なシステムの発展モデルが提案されている。これらの現代システム工学では外部ループによって問題空間と解空間におけるオープン性、適応性、相互作用性を持つ機能的物理的なサブシステムを定義することができる[20]。また Enterprise Systems Engineering (ESE) プロセスでは内部ループに対する入力と制約を外部ループで提供している[21]。

## 8 まとめと今後の課題

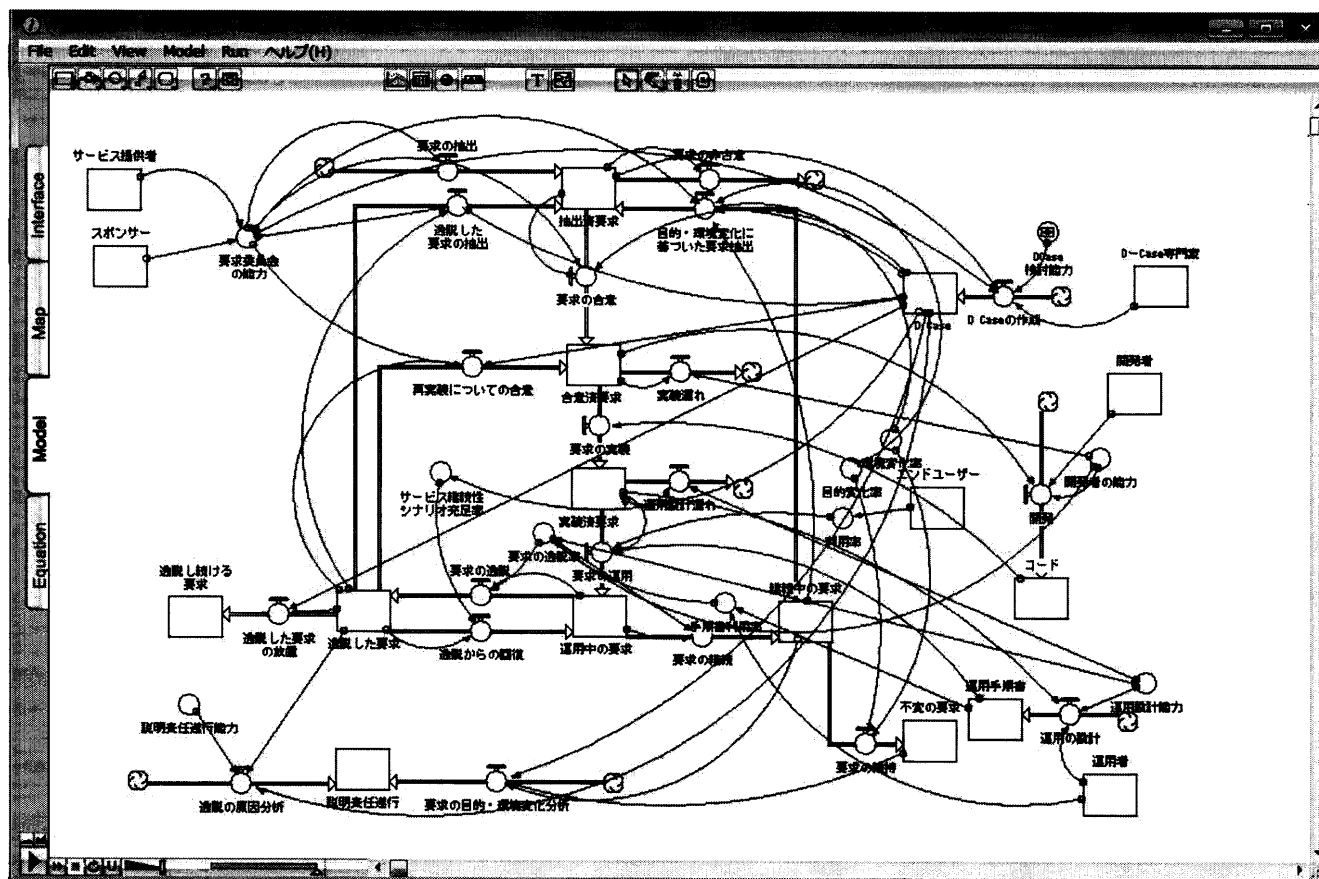
本稿では DEOS プロセスをシステムダイナミクスでモデル化することにより、保証ケースの効果を定性的に評価した事例を紹介した。今後、提案したプロセス評価手法の適用評価実験を予定している。この実験により、プロセス評価方式の妥当性を定量的に確認することが今後の課題である。

また、システム障害は開発中ではなく、運用中に発生する。このため運用プロセスに対するシステムダイナミクスを用いたシミュレーション評価も重要な今後の課題である。運用プロセスの評価では、運用活動の定義技法[22]などを用いることにより、運用プロセスを定義することで運用プロセスのシミュレーションを容易化できると期待している。

## 参考文献

- [1] 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)
- [2] ABDEL-HAMID, T. K. ANDMADNICK, S. E. 1991. Software Project Dynamics: An Integrated Approach. Prentice Hall, Englewood Cliffs, NJ.
- [3] Lan Cao, Balasubramaniam Ramesh, Tarek Abdel-Hamid, Modeling dynamics in agile software development, ACM Transactions on Management Information Systems (2010) Volume: 1, Issue: 1, Publisher: ACM, Pages: 5:1-5:26
- [4] Madachy R.J. System Dynamics Modeling of an Inspection-Based Process. Proc. ICSE 18, IEEE, 376-386, 1996
- [5] Ulrike Becker-Kornstaedt and Holger Neu, Learning and Understanding a Software Process through simulation of its Underlying Model, Advances in Learning Software Organizations, Lecture Notes in Computer Science, 2003, Volume 2640/2003, 81-93.
- [6] STALLINGER, F. AND Grünbacher, P., System dynamics modeling and simulation of collaborative requirements engineering. Journal of Systems and Software, Volume 59, Number 3, pp. 311-321(11), 2001.
- [7] Dapeng Liu, Qing Wang, Junchao Xiao, The Role of

- Software Process Simulation Modeling in Software Risk Management: a Systematic Review, pp. 302-311, ESEM '09: Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement
- [8] Lehman MM and Wernick P, System Dynamics Models of Software Evolution Processes, Proc. Int. Workshp. on the Principles of Software Evolution, ICSE'98, Kyoto, Japan, April 20-21, 1998
- [9] 山本修一郎, 要求定義・要求仕様書の作り方～要求を可視化するための～, ソフト・リサーチ・センター, 2006
- [10] Lamsweerde, A., Requirements Engineering- From System Goals to UML Models to Software Specifications, John Wiley & Sons Inc., 2009
- [11] Breian Berenbach, Daniel J., Paulish, Juergen Kazmeier, and Arnold Dudorfeer, Software & Systems Requirements Engineering In Practice, McGraw Hill, 2009.
- [12] M. Elizabeth Hull, Ken Jackson, and Jeremy J. Dick, Requirements Engineering, Springer, 2002
- [13] Gerald Kotonya, and Ian Sommerville, Requirements Engineering- Process and Techniques, John Wiley and Sons, 2002
- [14] Alan M. Davis, Just Enough Requirements Management- Where Software Development Meets Marketing, Dorset House Publishing, 2005
- [15] Dean Leffingwel and Don Widrig, Managing Software Requirements A Unified Approach, Addison-Wesley Professional, 2000
- [16] 山本修一郎, 要求変化プロセスのモデル化についての考察, KBSE 研究会, 2011
- [17] 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
- [18] T. Scott Ankrum, Alfred H. Kromholtz, Structured Assurance Cases: Three Common Standards, IEEE International Symposium on High - Assurance Systems Engineering, 2005
- [19] D-Case Editor A Typed Assurance Case Editor, <http://www.il.is.s.u-tokyo.ac.jp/deos/dcase/>
- [20] Hitchins, Derek, K., Systems Engineering- A 21st Century Systems Methodology, John Wiley & Sons, Ltd., 2007.
- [21] Rebovich, G., and Brian, W., Enterprise Systems Engineering - Advances in Theory and Practice, CRC Press, 2011.
- [22] 山本修一郎, IT 運用知の社会的獲得手法の構築, 日本情報経営学会誌, Vol.31, No.3, pp.41-51, 2011



付図1 DEOSプロセスに対するシステムダイナミクスモデル