

USDM の SysML への変換手順の提案

猿渡 卓也[†] 山本 修一郎[‡]

[†]株式会社 NTT データ 〒135-8671 東京都江東区豊洲 3-3-9

[†]名古屋大学大学院情報科学研究科 〒464-8601 愛知県名古屋市千種区不老町

[‡]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: [†] saruwatarit@nttdata.co.jp, [‡] yamamotosui@icts.nagoya-u.ac.jp

あらまし 組み込み系システムの開発では、USDM による要求仕様書が数多く作成されている。しかし、USDM による記述では、要求の抽出が不十分になることが考えられる。本稿では、USDM で書かれた要求仕様書を SysML に変換して分析する手順を提案する。提案した手順を適用し、USDM で書かれた要求仕様書を SysML に変換することにより、USDM では抽出できなかった要求が抽出できることを確かめた。

キーワード USDM, SysML, ゴール指向要求分析

A conversion method USDM to SysML

Takuya SARUWATARI[†] Shuichiro YAMAMOTO[‡]

[†]NTT DATA CORPORATION 3-3-9, Koto-Ku, Tokyo, 135-8671 Japan

[†]Strategy Office, Information and Communications Headquarters Nagoya University

Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601 Japan

[‡]Graduate School of Information Science Nagoya University Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601 Japan

E-mail: [†] saruwatarit@nttdata.co.jp, [‡] yamamotosui@icts.nagoya-u.ac.jp

Abstract Many requirement specifications by USDM are made in IT system development. However, by description by USDM, it is possible that extraction of a demand becomes insufficient. In this paper, a conversion method to SysML of USDM is proposed. It confirmed that the requirement which was not able to be extracted in USDM could be extracted by applying the proposed procedure and changing into SysML the demand specifications written by USDM.

Keyword USDM, SysML, GORE

1. はじめに

IT システム開発の上流工程において、要求仕様策定が十分に実施されない状況で開発に入った場合、設計工程・実装工程などに入ってから要求の追加が頻繁に実施される可能性が高くなる。この結果、手戻りにより開発コストが増加する。特に dependability (本稿では、dependability という言葉を“安全性”の意味で使用する。実際には、dependability には他の意味も含まれていると考えることができる) に関する要求が、開発の初期段階で抽出できなかった場合、手戻りによる開発コストの増加につながるケースが多くなると考えられる。なぜなら、dependability (安全性) に関する要求は、その要求の性質ゆえ、一般の機能要求と比較すると、強く実現が求められると思われるケースが多いからである。このことは、エンタープライズ系の IT システム、組み込み系の IT システム双方に言える。

最近の研究で、要求の抽出は、開発の初期段階の企画／要求定義などの上流工程だけでなく、製造工程など開発の途中段階でも抽出されることが体系的に整理されてきている。つまり、開発の上流工程で全ての要求を抽出／定義することは難しい、あるいは不可能である。しかし、手戻りのコストを考慮すると、なるべく上流工程で要求を抽出／定義するべきである。このような背景があり、近年要求の定義方法に関する研究が多く行われてきた[1]。ゴール指向要求分析は、要求をゴールと捉え、分析を行う手法である。代表的な具体的手法に、NFR[2]、KAOS[3]、i*Framework[4]と呼ばれるものがある。また、近年 UML を拡張して提案された SysML にも要求図が追加された[5]。この SysML が採用しているモデルも、階層構造を持ったモデルを採用しているという点で、ゴール指向要求分析の一種であると考えられる。

一方組み込み系システム開発の現場では、USDM[6]

を使った要求仕様記述が多く実施されている。USDMでは、要求を“上位要求”と“下位要求”にわけて定義する。この意味でゴール指向要求分析の一種と考えることができるが、深い階層構造を表現することが難しい。これは、要求抽出を実施する際の支障になることが予想された。すなわち、要求の抽出が不完全なものになってしまう可能性が考えられる。そこで、本稿ではUSDMの要求仕様記述をSysMLの記述に変換し、USDMでは抽出することが難しい要求を抽出する手順を提案した。また、提案した手法に従い、USDMで実際に記述された要求仕様をSysMLの記述に変換することで、その効果を確かめた。

2. USDM の SysML への変換方法

2.1. USDM における要求仕様の記述

USDMでは、要求を“上位要求”と“下位要求”にわけて記述する[6]。

“上位要求”では振る舞いを記述し、“id”が付与される。振る舞いとは、“イベント”から始まる“入力”、“変換”、“出力”で表現される“シナリオ”の性質をもつ一連の動作と定義されている。また、“上位要求”には“理由”と“説明”を補足情報として付与することができる。

“下位要求”は、“上位要求”に対して記述されるもので、“上位要求”に対する“手段の選択”、“入力手段”、“制約”等に関する仕様となる。“下位要求”は一つの“上位要求”に対して、複数記述することができる。“下位要求”についても、上位要求同様“id”が付与されており、補足情報として“理由”を付与できる。これらUSDMにおける要求の構成要素を表1に示す。

表 1 USDM における要求の構成要素

USDM における 要求の構成要素		構成要素の説明
上位 要求	Id	上位要求に付与される id
	振る舞い	上位要求を表現する一連の動作
	理由	上位要求の理由
	説明	上位要求に対する説明
下位 要求	Id	下位要求に付与される id
	仕様	上位要求に対する“手段の選択”、“入力手段”、“制約”等の仕様
	理由	下位要求の理由

USDMにおける要求間の関係には、上位要求と下位要求の関係が存在する。この関係を利用することで、階層的に要求を表現することができる。

2.2. SysML における要求の記述

SysMLでは要求をUMLクラスのステレオタイプ<<requirement>>として記述する[5]。SysMLにおける要求の記述は“要求の名前”と“要求の属性(property)”

からなる。“要求の属性”には要求の内容を記述する文章である“text属性”と、要求の識別番号を定義する“id属性”の2つがある。これらSysMLにおける要求の構成要素を表2に示す。

表 2 SysML における要求の構成要素

SysML における 要求の構成要素		構成要素の説明
要求の名前		要求を表現する名前
要求の属性	text属性	要求の内容を記述する文章
	id属性	要求の識別番号

SysMLにおける要求間の関係には、“階層関係”、“複製関係”、“派生関係”、“満足関係”、“検証関係”、“洗練関係”、“追跡関係”、“理由関係”の8つの関係がある。

2.3. USDM と SysML の対応関係

ここではUSDMとSysMLの記述の関係を論じる。

2.3.1. 要求の記述

USDMにおける要求は、“id”、“振る舞い”、“仕様”、“説明”で記述される。それに対して、SysMLでは、“id属性”と“要求の名前”と“text属性”で記述される。USDMにおける“id”は、SysMLにおける“id属性”に対応付けられる。また、USDMにおける“振る舞い”、“仕様”、“説明”は、SysMLにおける“要求の名前”と“text属性”に対応付けられる。

USDMでは、要求を“上位要求”と“下位要求”の2種類に分類している。それに対して、SysMLでは特に分類は無い。USDMでは、“上位要求”は“振る舞い”として記述され、“下位要求”は“仕様”として記述される。従って、要求の粒度をある程度揃えられる。SysMLでは、要求は“要求の名前”と要求の内容を説明する“text属性”となっており、自由に要求を記述できるが、要求の粒度を揃える仕組みは備わっていない。USDMの“理由”については後述する。

2.3.2. 要求の理由

USDMの要求仕様記述では、“上位要求”、“下位要求”とも要求の構成要素に“理由”を記述することができる。この“理由”は要求の記述を補強しており、要求の目的となる場合がある。つまり、“理由”と“要求”の間には目的と手段の関係が成り立つ場合があり、その場合、要求の“理由”には、別の要求が記載されていると考えることができる。このように、“理由”に記載されている事項が別の要求であった場合、SysMLでは、別の要求として“id属性”、“要求の名前”、“text属性”を割り当てることができる。別の要求として切り出せなかった場合は、SysMLでは、元の要求の“text属性”の中にも含めることができる。

2.3.3. 要求の関係記述

USDM の要求記述では，“上位要求”と“下位要求”の階層関係として要求間の関係を記述できる。さらに，2.3.2 で述べたように要求の構成要素として記述する“理由”が個別の要求である場合は，元の要求と“理由”の間に階層関係があると考えられる。従って，USDM では明示的に示されている“上位要求”と“下位要求”の間の階層関係と，“要求”とその“理由”の間の階層関係が表現できると考えることができる。これら，USDM で表現されている要求間の関係は，SysML における“階層関係”と“派生関係”で記述することができる。SysML では，要求間の関係に“階層関係”と“派生関係”以外の関係を定義することができるが，USDM では記述できない。

2.4. 提案手順

本稿で提案する手法の手順を以下に示す。

1. 【手順 1】

USDM の要求の構成要素における，上位要求の“振る舞い”，“理由”，及び下位要求の“仕様”，“理由”から，dependability（安全性）に関する記述を選択する。

2. 【手順 2】

（手順 1）で選択した各々の記述を，SysML の要求記述へ変換する。

（ア）（手順 1）で選択した記述から，安全性に関する記述のみを切り出す。これを SysML における要求の“text 属性”とする。

（イ）（ア）で切り出した要求の“text 属性”に適切な名前を付与する。これを SysML における“要求の名前”とする。

（ウ）（ア）で切り出した要求の USDM における“id”を，SysML の要求記述における“id 属性”とする。

3. 【手順 3】

（手順 2）で作成された SysML の要求記述を SysML の一覧表に纏める。

4. 【手順 4】

（手順 3）で纏められた要求記述について，USDM の要求仕様書から要求間の関係を抽出し SysML の一覧表に纏める。USDM 記述情報から抽出できる要求間の関係は下記の 2 パターンとなる。すなわち，“上位要求”と“下位要求”の階層関係と，要求とその“理由”の関係である。一覧に纏める段階で，SysML における要求間の関係のうち，“階層関係”と“派生関係”のどちらにあたるのかを決定する。

5. 【手順 5】

（手順 3）で纏められた要求の一覧と，（手順 4）

で纏められた要求間関係の一覧より，SysML のグラフを作成する。

6. 【手順 6】

（手順 5）で作成された SysML のグラフを参照して，既にグラフ中に存在する要求の間の関係について，漏れている関係を検討し追加する。追加する関係は“階層関係”と“派生関係”のどちらかとなる。また，追加した要求間の関係は，一覧表の方にも追加する。

7. 【手順 7】

（手順 6）で作成された SysML のグラフの関係一つ一つについて検討し，要求の関係に飛躍がある場合は適当な要求を追加し，要求間の関係の飛躍を減じる方向に修正する。一つの要求に複数のサブ要求が関係している場合，同程度の関係になるよう間に要求を追加する。また，追加した要求間の関係は，一覧表にも追加する。

8. 【手順 8】

（手順 7）で作成された SysML のグラフの，要求一つ一つについて他に下位の要求あるいは上位の要求が存在しないか検討し，存在すると思われる場合は追加する。追加した要求は，一覧表にも追加する。

3. 手順の適用評価

本稿で提案した手順を，USDM で書かれた要求仕様書に適用することで，手順の有効性を評価した。

3.1. USDM の要求仕様書

手順の適用には，インターネット上で公開されている“話題沸騰ポット”に関する要求仕様書を利用した [7]。この要求仕様書は，電子ポットを題材にした組込みシステム分析・設計のために作成されたものである。複数の版が存在するが，本稿では，できるだけ曖昧さをなくした仕様書の例として作成された“話題沸騰ポット要求仕様書（GOMA-1015 型）第 7 版”を利用した。この要求仕様書には 18 個の上位要求と 57 個の下位要求が含まれている。

3.2. 手順の適用

3.1 の要求仕様書に本稿で提案した手順を適用し，USDM から SysML への変換を試みた。

3.2.1. 手順適用の様子

以下に手順毎の具体的な手順適用の様子を示す。

【手順 1】 USDM の要求記述から 30 個の dependability（安全性）に関係する要求記述を選択した。選択した要求記述は，全体（上位要求の“振る舞い”，“理由”，及び下位要求の“仕様”，“理由”を記述している個所 93 個所）の 32%にあたる。

【手順 2】 及び【手順 3】（手順 1）で選択した 30 個の要求記述を SysML の要求記述に変換し，一覧表に

纏めた。

【手順 4】(手順 3) で一覧表にまとめた要求記述について、それら要求記述間の関係を USDM の要求記述仕様から 20 個抽出し、SysML の要求間関係記述に変換し、一覧表に纏めた。

【手順 5】(手順 3) と (手順 4) で纏められた一覧表から SysML のグラフを作成した。

【手順 6】(手順 5) で作成した SysML のグラフに、要求間の派生関係を新規に 8 個の追加した。

【手順 7】(手順 6) で作成した SysML のグラフに、新規に 2 つの要求を追加した。この要求の追加に伴い、要求間の階層関係を 2 個追加した。(手順 7) では、全く新しい要求の追加を行うのではなく、要求記述をより厳密に行うために要求を追加すると考えることができるので、基本的に追加される関係は“階層関係”になる。つまり、ここで追加する要求は、要求を“階層関係”でより具体的な下位要求に分割することで、さらにその下位にある要求が、実際に何を充足しようとしているのかを認識するための補助となる。

【手順 8】(手順 7) で作成した SysML のグラフに、新規に 1 つの要求を追加した。具体的に追加した要求は、(手順 7) で追加した要求である“危険回避”から、派生する要求として“ポット転倒時の危険回避”である。

3.2.2. 評価結果

本稿で提案した手順を適用することで、USDM の要求記述に (手順 7) において 2 個、(手順 8) において 1 個の要求を追加することができた。(手順 7) において追加された要求は、要求記述の構造を補足するためのものなので全く新しい要求ではなかったが、(手順 8) で追加された要求は、新規に追加された要求であり、USDM の記述では抽出できなかった要求であると考えられる。従って、USDM で書かれた要求仕様書を SysML に変換することは、USDM の要求仕様書では不足している要求記述の抽出に有効であることがわかった。ただし、要求の抽出にドメイン知識が必要となるのは、一般的な要求抽出と同じであると考えられる。本稿で示した適用の例でも、ドメイン有識者の解析であれば (手順 8) において追加した要求の下位要求 (あるいは他の要求) を抽出して追加できる可能性が高いと考えられる。

4. 関連研究

IT システムの要求記述に関する研究は、USDM, SysML 以外にも数多く実施されている。代表的な手法に、NFR Framework[2], KAOS[3], i*framework[4]などがある。また、これらの手法を拡張した手法として、Tropos[8]などがある。また、安全性要求については、

安全性ケース (Safety case) や保証ケース (Assurance case) の研究が活発化している [9][10][11][12][13]。本稿では、USDM からの要求記述の変換先として、SysML の要求図を利用した。SysML には、要求図以外に振る舞い図、構成図が定義されており、要求図は後から追加された。構成図では、IT システムのアーキテクチャを表現することができる。本稿では、IT システムの要求は、IT システムが持つアーキテクチャと密接な関係を持つという立場から SysML の要求図を採用した。今後、保証ケースとアーキテクチャとの関係について研究していく予定である。

5. 考察

本稿で提案する手順の検討及び適用に関して下記の考察を実施した。

5.1. 要求間に関する記述能力差

USDM と SysML における記述能力の違いの一つに、要求間関係の記述方法がある。USDM では要求間関係は、“上位要求”と“下位要求”の関係、及びそれぞれの要求の構成要素である“理由”との関係となっており記述形式が固定されている。また、“上位要求”と“下位要求”の関係以外の関係を表現することは難しい。それに対して、SysML では、要求を定義すれば、それら要求の間に自由に関係を定義することができる。これは、“要求”を“ゴール”で置き換えて考えると、NFR Framework や KAOS のような一般的なゴールグラフが持っている特徴と同じである。この特徴により、SysML の記述では上位の要求と下位の要求を任意の数の階層で表現することができる。今回の結果では、この任意の数の階層を許すという特徴により、USDM の要求仕様記述では漏れていた要求を、発見することができたと考えられる。

任意の数の階層構造を許すという SysML の特徴は、USDM に対する優位性であると考えられる。

5.2. 要求記述の記述能力差

USDM では“下位要求”は“仕様”として記述されることになっており“上位要求”は“振る舞い”として記述されることになっている。機能要求と非機能要求の種別で考えると、これらの記述はどちらも機能要求に関するものである。本稿の評価で利用した USDM の要求仕様書からの要求抽出でも、抽出された 30 件の要求記述のうち 25 件は機能要求に関するものであった。後の 5 件は非機能要求に関するものであるが、“上位要求”の“理由”から抽出したものであり、USDM では明確に要求とは考えられていないものである。このように、USDM は機能要求に重点をおいた述方法であると考えられる。SysML では、機能要求と非機能要求の区別が存在しないので、機能要求の抽出まで至らずに分析を終えてしまう可能性がある。つま

り、確実に機能要求の記述を促すということが USDM の SysML に対する優位性となると考えることができる。ただし、SysML では、振る舞い図や構成図など要求図以外の構成要素と要求図を連携させる際、機能要求の漏れが発見できるとも考えられる。

逆に、本稿の評価において追加した 3 件の要求が、いずれも非機能要求に関するものであったことを考えると、USDM では非機能要求に関する分析が不十分になってしまうと考えることができる。本稿の評価の(手順 8)で追加した要求については、ドメインの知識を持つ分析者が、要求の分割あるいは派生を使って分析を進め機能要求まで抽出していく必要がある。

5.3. USDM における理由記述に関する考察

USDM では、要求ごとに「理由」が記述できる。しかし、理由が書けることと、正しく理由が書いてあることには大きな差がある。本稿で利用した“話題沸騰ポット”の要求仕様書では、下記のような理由を明確に記述していない例が存在した。

[要求] pot-330: 沸騰行為も保温行為もできないときは温度制御はしない

[理由] 温度制御行為ができないポットの状態ではヒータを on/off させると危険だから

これは、非常に分かり辛い表現であり、次のような疑問がわく。

- そもそも「温度制御行為ができないポットの状態」が「ヒータを on/off できない状態」のことではないのか？
- ヒータを on することは危険かもしれないが、off するのは、必ずしも危険とは言えないのではないか？
- 「沸騰行為も保温行為もできないとき」と「温度制御行為ができないポットの状態」とは、どのような関係にあるのか？

説明の中で<アイドルになる判断>が記述されていることと、「温度制御行為をしない(アイドル)」という記述が節の題名になっていることから、“アイドル状態”というポットの状態があることが推察できる。次のように記述を変更すると、理解が容易になると考察した。

[要求] pot-330: ポットの危険な操作を感知して、アイドル状態に遷移することにより、温度制御を停止する

[理由] ポットの安全性を確保するため

一方、pot-221 では、次のように記述されている

[要求] pot-221: 蓋を開けたら(開いていたら)、ロックは解除され、温度制御行為はしない

[理由] 安全確保のため

pot-330 と pot-221 で同じような要求を記述している

ようである。このように USDM による要求記述では、日本語の文だけ見ると一見異なる類似要求が混在する可能性が高いことが分かる。これらの類似要求は、SysML では、構造的に把握することが可能となる。

また、話題沸騰ポットの“理由”欄では、次のような環境制約が、理由として記述されることもある。

[要求] pot-320: 設定されたモードの温度にポット内の水温を保持する

[理由] そのままにしておく、水温は自然に気温近くまで下がってしまうので

水温を保持することが目的であり、地球環境では外気温まで水温が下がる制約があるので、一定水温を保持するために加熱する機能要求が必要になるのである。なぜ、このような混乱が生じるかといえば、「要求」と「理由」という項目しかないために、この 2 項目に情報を無理に詰め込むためであると推察される。そうであるならば、「要求」「目的」「制約」と分けて記述するか、あるいは、便宜的に、USDM の「理由」の中で、「目的」と「制約」を明示的に示したうえで記述することが望ましい。しかし、わざわざ、明記するくらいなら、理由欄をやめて、目的欄と制約欄を用いたほうがいい。USDM の理由欄は、このように、記述された内容が、目的なのか制約なのか分かりにくいので注意する必要がある。

5.4. 最上位の要求について

本稿で行った評価では、手順適用の過程で“安全確保”という SysML における最上位の要求を一つに絞ることができた。しかし、最上位の要求を毎回一つに絞ることができるとは限らない。前述したように USDM では、非機能要求の抽出に重点がおかれていないと考えられるので、最上位の要求を一つに絞り切れない状況も十分考えられる。本稿の手順では、最上位の要求が一つに絞れなかった場合の対処については言及していない。無理に最上位の要求を一つに絞る必要はないという考えからである。最上位の要求を一つに絞れなかった場合でも、そのまま分析を進めることができる。しかし、USDM が非機能要求に重点をおいていない状況を考慮すると、最上位の要求が抽出できていない場合、(手順 8)で要求を追加する際、積極的に上位方向の要求について検討を行った方が良いと言える。

本稿の手法は、dependability (安全性) の要求分析に限定しているため、最上位の要求として“安全確保”という要求を置くのも一つの考え方である。

5.5. dependability (安全性) の脅威分析

dependability (安全性) の要求を分析するためには、dependability (安全性) に対する脅威を、対として検討する方が良い。本稿で提案した手順では、(手順 8)において要求を検討抽出する際に、“危険回避”に関する

る脅威の分析を利用すると分析が抜け漏れなくスムーズに進むことが考えられる。本稿で行った評価では、(手順 8)において“ポット転倒時の危険防止”という要求を“危険回避”という要求から派生させて抽出した。これは、“危険回避”という要求の“危険回避”の対象を想定している方がわかりやすいと考えられる。元々“危険回避”から派生している要求には、“温度制御行為不可時の危険防止”と“火災等の危険回避”があった。前者は温度制御が不可になるという脅威に対する要求、後者は火災等の脅威に対する要求と考えられる。本稿で行った評価では、ポットが転倒するという異常事態の脅威を想定し、“ポット転倒時の危険防止”という要求を抽出した。他にも適当な脅威を想定できれば、それに対応した要求を抽出できる。

SysML の要求図で安全性に対する脅威のモデル化を行い検討できるようにするためには、SysML の拡張が必要になる可能性がある。

5.6. ドメイン知識が必要な手順

本稿で提案した手順は、ドメインに対する知識が必要となる手順と、そうでない手順が存在する。ドメインに対する知識とは、“話題沸騰ポット”のポットに対する知識や沸点などの液体の知識に関する知識のような、システムの対象に依存する知識である。

本稿で提案した手順のうち、(手順 1)～(手順 5)は、MSDM で書かれた要求仕様書に存在する要求を、SysML の記述へ変換する手順なので、ドメインに対する知識をほとんど必要としない。それに対して(手順 6)～(手順 8)は、新しい要求や要求間の関係を抽出する必要があるのでドメインに対する知識が必要となる。特に(手順 8)は、(手順 6)、(手順 7)と比較して、要求抽出に関する手掛かりが少ないので、より深いドメインに対する知識が求められると考えられる。

6. まとめと今後の課題

本稿では、USDM で記述された要求仕様記述を、SysML に変換する手順を提案し、実際に適用することで、その有効性について検討した。

提案手順を適用では、USDM では抽出できなかった要求を 3 件抽出することができた。本稿で実施した評価はドメインの知識をほとんど有しない作業者が実施したが、ドメイン知識を有する作業者が手順を実施すると、さらに多くの要求が抽出できることが期待できる。

抽出できた要求が非機能要求であったことを考えると、提案手順を適用することで USDM の記述では抽出することが難しい非機能要求についての検討を十分実施できる環境が整うということが言える。

今後は、SysML における要求図以外の要素も含めた

手順の検討が課題となる。USDM では、ある程度具体的なアーキテクチャを想定した要求仕様記述を実施していると考えられるが、SysML の要求図だけでは、アーキテクチャが対象範囲に入れられていない。SysML における要求図以外の要素を検討に加えることで、より厳密な分析が実施できることが期待できる。これは、保障ケースとアーキテクチャの関係を整理することとも関係する。

また、SysML の要求図自体も、脅威の分析及びモデル化など、拡張する余地がある考えることができ、これらも今後の課題となる。

文 献

- [1] 山本修一郎, ～ゴール指向による!!～システム要求管理技法, 株式会社ソフト・リサーチ・センター, 東京, 2007.
- [2] Mylopoulos, J., L.Chung, et al.(1992).”Representing and Using Nonfunctional Requirements : A Process-Oriented Approach.” IEEE Transactions on Software Engineering SE-18(6): 483-497.
- [3] A. Dardenne, A. van Lamsweerde and S.Fickas, “Goal-Directed Requirements Acquisition”, Science of Computer Programming, Vol.20, 1993, 3-50
- [4] Eric S. K. Yu. Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering. In 3rd IEEE Int. Symp on Requirements Engineering, pages 226-235, Washington DC, Jan 1997.
- [5] 山本修一郎, SysML の要求図, ビジネスコミュニケーション 2009, Vol46, No7
- [6] 清水吉男, テストの質を上げるための要求仕様書, JaSST2007 講演資料
- [7] 組込みシステム教育教材 話題沸騰ポット GOMA-1015 型 要求仕様書, http://www.sesame.jp/workinggroup/WorkingGroup2/POT_Specification.htm
- [8] Paolo Bresciani, Paolo Giorgini, Fausto Giunchiglia, John Mylooulos, and Anna Perini, TROPOS: AN AGENT-ORIENTED SOFTWARE DEVELOPMENT METHODOLOGY Autonomous Agents and Multi-Agent Systems Volume 8, Number 3, 203-236
- [9] Kelly, Tim P. “Arguing Safety - A Systematic Approach to Safety Case Management,” DPhil Thesis, York University, Department of Computer Science Report YCST, May 1999.
- [10] Ankrum, T. Scott and Alfred H. Krombolz. “Structured Assurance Cases:Three Common Standards,” Slides presentation at the Association for Software Quality (ASQ) Section 509 meeting, the MITRE Corporation, 25, January 2006
- [11] Bishop, P., Bloomfield, R., Guerra, S.: The Future of Goal-Based Assurance Cases. In:Proc. Workshop on Assurance Cases, pp. 390-395 (2004)
- [12] Mark-Alexander Sujana, Floor Koornneef, and Udo Voges, Goal-Based Safety Cases for Medical Devices: Opportunities and Challenges, F. Saglietti and N.Oster (Eds.): SAFECOMP 2007, LNCS 4680, pp. 14-27, 2007.
- [13] Thomas Rhodes, Frederick Boland, Elizabeth Fong, Michael Kass, Software Assurance Using Structured Assurance Case Models, NIST Interagency Report 7608,