

情報システム設計における トレーサビリティ管理モデル

宇田川 佳久[†]

開発プロセスにおける成果物間のトレーサビリティ管理は情報システムの品質を高めるための基本である。本文は、Web アプリケーションの開発を事例として要求定義および設計工程におけるトレーサビリティ管理について論じる。要求および設計のモデル化言語としては、システム開発プロセスの成果物に対し有効なモデリング記法を提供している SysML (Systems Modeling Language) を採用している。

SysML は、要求や設計項目を関連付ける豊富な機能を提供している。一方、成果物のどの組み合わせに対して関連性を管理するかは、プロジェクトごとに異なる。本文は、SysML をベースとしたシステム開発において、プロジェクトの目的を反映したモデルフレームワークに即してトレーサビリティ関連を管理する手法を論じた。

実務のシステム開発においても、データベース設計の段階でテーブル名やデータ項目名を厳密に定義することが定着しているが、厳密に定義された用語は、要求、設計、実装間のトレーサビリティの管理に有効な手段となることを指摘した。

Traceability management model in information system design processes

Yoshihisa Udagawa[†]

Managing traceability on artifacts among development processes is essential to improve the quality of information systems developed. This paper discusses an approach to manage traceability in requirement and design phases using an example of "longon function" to a Web system. SysML (Systems Modeling Language) has used as a modeling language since it provides effective constructs for modeling artifacts in system development processes.

SysML provides several relationships that enable a modeler to relate model elements in SysML. On the other hand, it depends on each of development project on what items traceability relations should be defined. This paper discusses a Sys-ML technique to manage traceability with referring to traceability model framework which reflects the purpose of the project.

It is now an industry practice in a database design to define strictly names of objects such as tables and data items etc. We have pointed out that we can take advantage of these strictly-defined names to manage traceability among items on database in requirement, design and implementation phases.

1. はじめに

情報システムは社会活動の安心、快適、効率を実現するための基盤として必要不可欠なものとなっている。社会活動における情報システムの重要性が増すに連れ、情報システムの障害によって引き起こされる社会活動への影響は、広範囲かつ深刻なものとなっている。このため、情報システムの安全性、信頼性に対する要求は高まっており、幅広い分野での取り組みが行われている。本研究も、このような取り組みの一翼を担うことを目指したものである。

総務省統計局の報告によれば¹²⁾、「ソフトウェア」と「情報処理・提供サービス」を合わせた「情報サービス業」の実質 GDP は、1998 年度が約 7.1 兆円、2008 年度が約 12.9 兆円で、この 10 年間で約 181% 増加している。なお、名目 GDP でも 189% 増加している。雇用者数では、1998 年度が約 88.9 万人、2008 年度が約 116.3 万人であり、この 10 年間で約 131% 増加している。これらの統計から、情報サービス業は堅調に拡大してきた産業であり、社会における情報システムの役割の拡大と定着を裏付けるものと考えている。

一方で、情報システムのトラブルが後を絶たない現実がある。トラブルによる影響は、交通、金融、流通業務など、我々の日常生活に直結するものに及んでいる。情報システムは、経済活動および日常生活を効率化するための手段を提供すると同時に、潜在的なリスクにもなりつつある。近い将来、情報システムの利用環境は、益々、複雑、高性能かつミッションクリティカルなものとなり、これまでの情報システムの単純な延長線上にあるものではないとの指摘がある¹⁰⁾。

学術においても、高信頼な情報システムを構築する技法に関する研究が行われてきた。これらの研究は、大きく以下の 3 種類に分類できる。

(1) システム開発方法論

要求に合致した信頼性の高いシステムを期限内に開発するための方法論に関する研究で、ウォーターフォールモデル、オブジェクト指向設計モデルなどが提案されている。

(2) 情報・物理セキュリティ

情報システムを外部攻撃、内部漏洩から守る技法に関する研究で、ユーザ認証、アクセス制御、システム異常検知など、数多くの技法やツールが提案されている。

(3) 耐故障アーキテクチャ

情報システムを構成する機器の故障への対応力を高める研究で、データレプリケーション、冗長記憶装置、バーチャル OS など、数多くの技法やツールが提案されている。

[†] 東京工芸大学工学部コンピュータ応用学科

Tokyo Polytechnic University, Faculty of Engineering, Department of Applied Computer Science

本研究は、システム開発方法論に関するもので、特に、システム開発ライフサイクル全体（要求定義、設計、実装、テスト）に渡るトレーサビリティ（追跡可能性）を確保することにより、情報システムの品質を高めるための開発技法に関するものである。我々は研究報告⁹⁾で、情報システム開発におけるトレーサビリティの枠組を提案した。本文は、Webアプリケーションの開発を事例として、提案したトレーサビリティの枠組みの実施方式を、要求定義および設計フェーズの範囲で検証している。要求および設計の記述言語としては、国際的に普及している SysML (Systems Modeling Language)⁴⁾を採用している。また、データベースの設計では、業界で広く使われている IDEF1X による記述を採用している¹⁾。

本文の構成は以下の通り。2章では、情報システム開発モデルとトレーサビリティ管理へのアプローチについて、これまでの研究の概要を述べ、本研究の位置づけを明らかにしている。3章では、記述対象としたログオン機能と、このログオン機能の SysML による記述結果を示す。4章では、成果物間のトレーサビリティについて考察している。

2. 情報システム開発とトレーサビリティ管理

2.1 情報システム開発モデル

Pohl 氏らは、システム開発を、仕様 (Specification)、表現 (Representation)、合意 (Agreement) を軸とする 3次元空間において、合意が得られていない曖昧な仕様と非形式的な表現を、合意が得られた完全な仕様と形式的な表現に変換する過程としてモデル化している。その上で、この変換過程を追跡できるようにすることがトレーサビリティの役割としている (図1)⁵⁾。

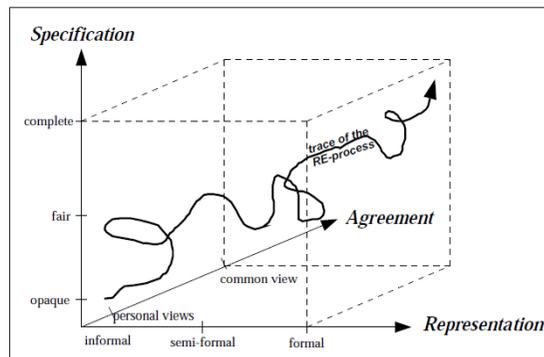


図1. 情報システム開発モデル (文献5より引用)
Figure 1. Information system development model (from reference 5)

実務におけるソフトウェアシステムの開発モデルについては、1970年に米国の Royce 氏が、ソフトウェアの作成から廃棄までのライフサイクル・プロセスの概念を提唱した論文が最初と考えられている⁷⁾。このモデルは、後にウォーターフォールモデルと呼ばれている。ウォーターフォールモデルでは、開発サイクルを複数の工程 (プロセス) に分けることを前提としている。ひとつの工程が終わるごとにその工程での成果が文書化されるので、品質、コミュニケーションやリスク管理が容易である。また、各工程では、プロジェクトメンバーの全員が同じレベルの作業をするため、工程の進捗を資源の配分によって調整・管理することが容易である特徴もある。ウォーターフォールモデルは、大多数の業務向け情報システム開発で採用されているとの報告がある¹¹⁾。

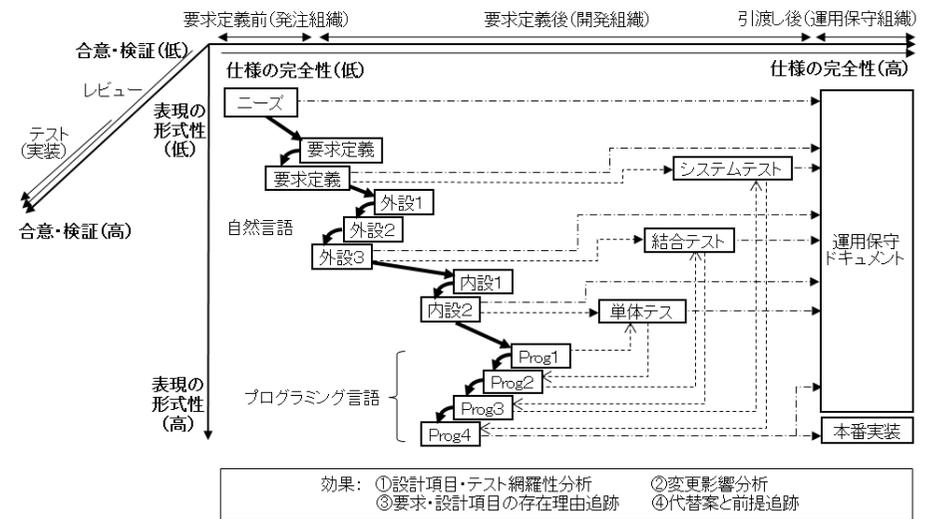


図2. トレーサビリティの枠組み
Figure 2. A traceability framework

図2は、筆者らが提案するトレーサビリティの枠組みであり⁹⁾、Pohl 氏らのモデルとウォーターフォールモデルの概念に基づき、実務における業務および成果物の関係をモデル化したものである。

Pohl 氏らの情報システム開発モデルと図2の枠組みには、下記の解釈によって関係付けることができる。

- ・仕様は、要求定義、外部設計、内部設計という一連の工程を通して完全なものに変換される。
- ・表現は、一連の工程を通して、自然言語、図、表による形式性の低い表現から、プログラミング言語などによる形式性の高い表現に変換される。
- ・要求定義書、設計書などはレビューによって、また、プログラムはテストによって、関係者との合意が高められる。

ウォーターフォールモデルと図2の枠組みは、一連の成果物（要求定義、設計、プログラム、テスト結果、運用保守ドキュメント）によって対応付けることができる。

2.2 トレーサビリティ管理のアプローチ

情報システム開発における成果物間のトレーサビリティ情報を取得する方法は、自動取得、半自動取得、手動取得に分類できる⁸⁾。

(1)自動取得

成果物を情報検索手法（Information Retrieval）によって分析し、分析によって得られた情報（類似度）にもとづいてトレーサビリティ情報を再構築する。

(2)半自動取得

成果物を作成する段階でトレーサビリティ情報を取得する、または、開発者があらかじめ定義したトレーサビリティ情報を付与する。

(3)手動取得

開発者が成果物の内容を確認し、トレーサビリティ情報を付与する。多くの場合、トレーサビリティ情報の付与を支援するツールを使用する。

Lormans 氏らは、LSI(Latent Semantic Indexing)と呼ばれている情報検索手法を用いて、トレーサビリティ情報の自動取得の程度を検証している³⁾。その結果によれば、最も精度が良かった Pacman ゲームの要求定義書、設計書、テストケース間のトレーサビリティでも、回想率(Recall)が100%から74%の時に精度(Precision)が10%から30%であった。一般に、情報検索手法では、検索漏れを低めた場合(回想率を高めた場合)、本来関係がない多くの項目が検索される(精度が低下する)。従って、本来関係がない項目にトレーサビリティがあると判断される傾向が強まる。そのため、例えば、変更影響分析を実施した場合、本来は関係がない多くの項目がチェック対象となる。

Ramesh 氏らは⁶⁾、より高い精度のトレーサビリティ情報を取得するためにはモデルフレームワークに基づくアプローチが必要であることを論じている。モデルフレームワークには、トレーサビリティ情報の用途や対象とする成果物の構造が含まれる。本研究は、モデルフレームワークに関する研究であり、業界での普及を考慮し、成果物の記述方法として SysML を採用している。なお、Spanoudakis 氏らは、これらの研究を半自動取得に分類している⁸⁾。

多くの商用の要求仕様管理ツールは、トレーサビリティ情報を手動で取得する方法を採用している。この場合、トレーサビリティ情報を取得するためのコストが高価で

あり、情報システム開発におけるトレーサビリティ管理の普及の妨げになっているとの指摘があり、トレーサビリティ情報の自動取得または半自動取得が研究テーマとなっている。

3. システムモデリングとトレーサビリティの事例

3.1 SysML について

SysML は、システムの要求 (Requirement)、振舞い (Behavior)、構造 (Structure) をモデル化するために必要な意味的基盤とその図式表現を提供する汎用モデリング言語である。SysML は、UML 2.0 の拡張付サブセットとして位置づけられるものであり、多くの図が UML 2.0 と同じまたは修正したものとなっている (図 3)。SysML の最新版は、2010 年 6 月にリリースされたバージョン 1.2 であり、OMG のホームページから規格書をダウンロードできる⁴⁾。SysML はシステム要求定義とアーキテクチャ設計への適用を、UML 2.0 はソフトウェア要求定義と設計への適用を想定している。なお、図 3 で塗り潰した部分が、今回使用した SysML の図式表現を示す。

No	分類	SysML図式表現	UML2.0との関連
1	要件図	要件図	SysMLで新規に追加
2	振る舞い図	アクティビティ図	UML2.0と同じ
3		シーケンス図	UML2.0から修正
4		ステートマシン図	UML2.0から修正
5		ユースケース図	UML2.0から修正
6	構造図	ブロック定義図	UML2.0から修正
7		内部ブロック図	UML2.0から修正
8		パッケージ図	UML2.0と同じ
9		パラメトリック図	SysMLで新規に追加

図 3. SysML の図式表現一覧

Figure 3. List of SysML diagram

3.2 モデル化の対象とするログオン機能について

情報システムのログオン機能は、情報システムのユーザと付随する権限を特定するために必須の機能である。パスワードによるユーザ認証方法に限定しても、パスワードの変更、有効期限および失念への対応など、実装に際しては考慮すべき点も多い。この章では、パスワードの変更、有効期限管理を含むログオン機能の要求定義と設計を対象として取り上げ、要求項目と設計項目間のトレーサビリティについて考察する。

図 4 は、想定している「プロジェクト情報管理システム」とログオン機能の概要を

示している。このプロジェクト情報管理システムでは、複数のプロジェクトの情報を管理対象としており、それぞれのプロジェクトに対し、管理ユーザと業務ユーザを定義できる。従って、プロジェクト情報管理システムへのログオン機能としては、ユーザIDとパスワードに加えプロジェクトIDを指定するものとする。管理ユーザと業務ユーザの区別は、ユーザのプロジェクト内での役割をデータベースで管理する。データベース設計については、本文「3.4.3 データベース設計」で述べる。

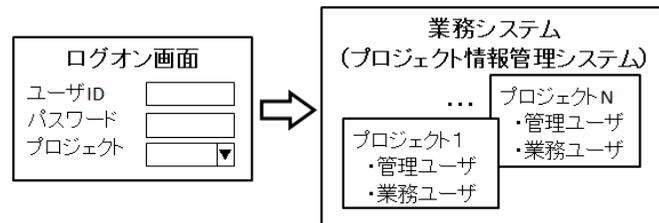


図 4. ログオン機能と業務システム
 Figure 4. Logon function and application system

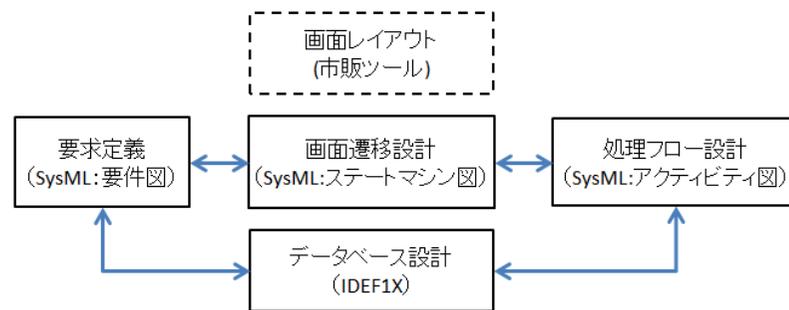


図 5. 成果物とトレーサビリティ
 Figure 5. Artifacts and traceability

3.3 モデルフレームワーク

図 5 は、成果物と成果物間のトレーサビリティを示している。成果物としては、要求定義、画面遷移設計、処理フロー設計とデータベース設計である。要求定義、画面遷移設計、処理フロー設計は、SysML が提供する図で表現しているが、データベース設計は SysML に適切な図がないため、従来からデータベース設計で使われてきた

IDFE1X で表現している¹⁾。また、SysML および UML には、画面レイアウトを記述する図が見当たらない。本研究では、実務での作業を考慮し、市販のホームページ作成ツールによる画面設計と対応する HTML を成果物としているが、本文ではこれ以上は述べない。

トレーサビリティについては、要求定義から設計項目への網羅性の管理を想定し、要求定義と画面遷移設計（外部設計に相当）、画面遷移設計と処理フロー設計（内部設計に相当）、要求定義とデータベース設計、処理フロー設計とデータベース設計の 4 個のトレーサビリティを管理するものとした。

3.4 SysML によるモデル記述

3.4.1 SysML 要求定義

SysML の要求定義 (Requirements Diagram) では、要求を文章によって表現する機能と、他のモデル要素（詳細化やテストケース）を関係付けるための要素を提供している。これにより、要求だけでなく要求と設計要素との関係を記述することができる。要求には、機能要求だけでなく非機能要求も記述できる。要求の階層構造を構成する機能 (Requirement Containment Relationship) も提供されている。

図 6 は、ログオン機能の要求を、要求の階層化機能を使って記述したものである。SysML の要求定義では、要求の充足を表現するために充足関連 (Satisfy Relationship) が提供されている。図 6 の «satisfy» は充足関連の例であり、要求のすべてを後述する画面遷移設計と関連づけている。従って、要求定義から設計へのトレーサビリティは 100% である。

この他に、SysML の要求定義では、以下の関連付け機能が提供されている。

«deriveReq»: 元になった要求とその要求から導出された要求を表現するために、導出依存関連 (Derive Requirement Relationship) が提供されている。

«verify»: 要求の検証を表現するために検証関連 (Verify Relationship) が提供されている。一般に、検証関連は要求と分析、デモンストレーション、テストケース間で定義される。

«refine»: 要求と別の側面から表現した成果物との関連を表現するために洗練化関連 (Refine Requirement Relationship) が提供されている。洗練化関連は、要求定義と SysML の他の図（ユースケース図、アクティビティ図など）との関連を表現するために使われる。

«trace»: 汎用トレース関連 (Generic Trace Requirement Relationship) は、要求定義と他の要素との関連付けを行うもので、意味的な制約は他の関連に比べて乏しい。利用局面固有の意味を付けた関連を定義する場合に用いる。

また、存在理由 (rationale) 要素は、任意の要求、または、任意の要求間に定義された関連に付与することができ、その要求または関連の背景や存在理由を記述するために用いられる。

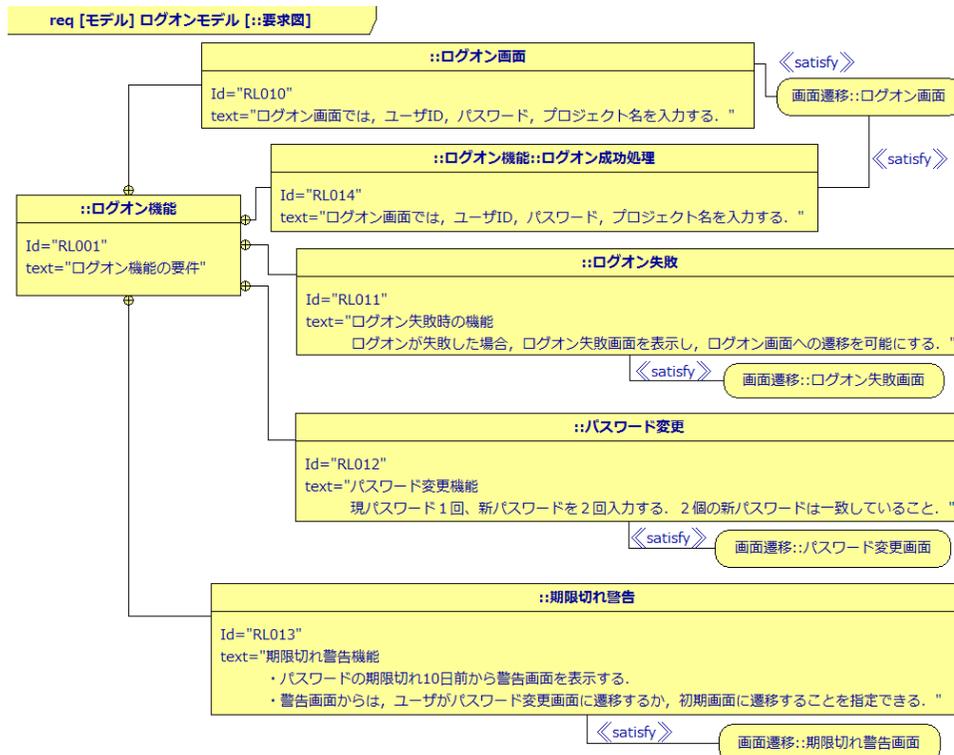


図 6. SysML による要求定義
 Figure 6. Requirement definition by SysML

3.4.2 画面遷移設計

SysML のステートマシン図 (State Machine Diagram) は、オブジェクトの状態と、状態の遷移によってオブジェクトの動作を表現する。状態は矩形で表現する。遷移は状態を表す矩形間を結ぶ線で表現する。遷移の契機となる動作、分岐条件、開始、終了を記述する記法が提供されている。

画面遷移は、画面と画面遷移のイベント (ボタン押下など) でモデル化できることから、ログオン画面の遷移をステートマシン図で記述した (図 7)。ログオン画面、ログオン失敗画面、期限切れ警告画面、パスワード変更画面は、それぞれ、要求定義に対応する。一方、要求には明記されていないが、画面遷移の観点からは、パスワード

変更が成功した場合は、成功した旨を利用者に知らせる必要がある。このために、パスワード変更成功画面を定義した。パスワード変更成功画面では、ログオン成功のメッセージとともに「ログオン」ボタンを表示し、このボタンを押下することで、「プロジェクト情報管理システム」にログオンするものとした。

以上より、画面遷移設計から要求定義へのトレーサビリティは 100%にはならない。要求を分析して設計が進められるので、要求項目に該当しない設計項目が存在するのは、一般的なことである。しかし、要求項目に該当しない設計項目がどれであるかを特定することは、設計内容の確認作業を効率的かつ確実に支援する前提である。

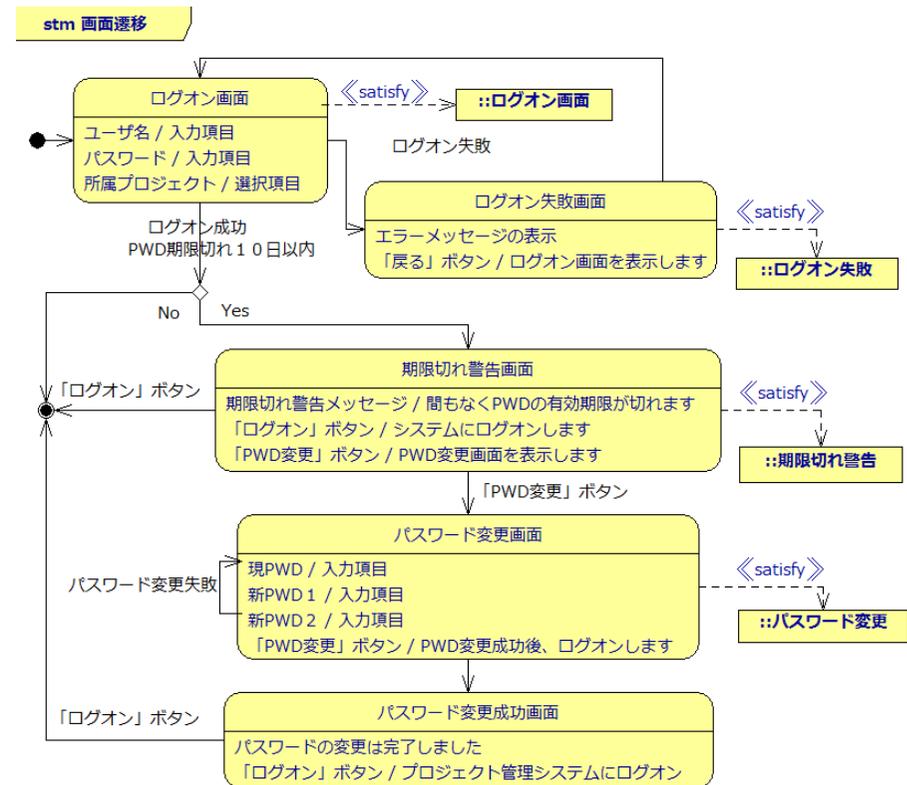


図 7. SysML による画面遷移設計
 Figure 7. Screen transition design by SysML

3.4.3 データベース設計

図8は、「プロジェクト情報管理システム」のログオンに関する情報を管理するためのデータベース構造を IDEF1X で表記したものである¹⁾。

図8に示したデータベースは、テーブル R_User とテーブル R_Project から構成されている。テーブル R_User は、基本的に、ユーザ ID、パスワードおよびパスワードの有効期限を管理するものである。テーブル R_Project は、ユーザ ID がプロジェクトで果している役割を表現している。ユーザは複数のプロジェクトに属することから、テーブル R_User とテーブル R_Project は 1 : N の関係がある。これを表現するため、テーブル R_Project は、テーブル R_User の UserID (ユーザ ID) を外部キーとして含んでいる。

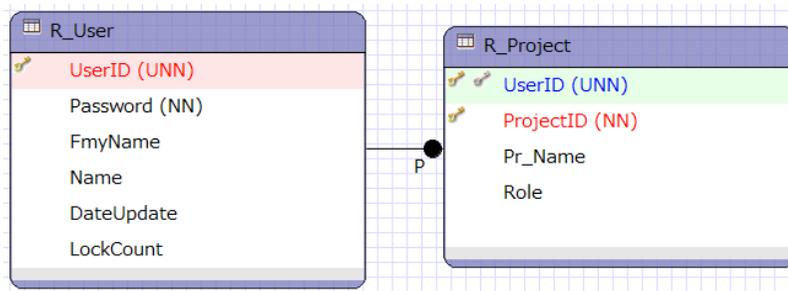


図8. IDEF1X によるデータベース構造の記述
 Figure 8. Database structure described by IDEF1X

物理名 R_User 論理名 ユーザ表

PK	FK	物理名	論理名	型	NOT N...	UNIQ...
🔑		UserID	ユーザID	varchar(12)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		Password	パスワード	varchar(16)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		FmyName	苗字	varchar(6)	<input type="checkbox"/>	<input type="checkbox"/>
		Name	名前	varchar(6)	<input type="checkbox"/>	<input type="checkbox"/>
		DateUpdate	更新日	timestamp	<input type="checkbox"/>	<input type="checkbox"/>
		LockCount	ロック回数	smallint	<input type="checkbox"/>	<input type="checkbox"/>

図9. テーブル R_User の論理設計
 Figure 9. Logical design of table R_User

物理名 R_Project 論理名 プロジェクト表

PK	FK	物理名	論理名	型	NOT N...	UNIQ...
🔑	🔑	UserID	ユーザID	varchar(12)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
🔑		ProjectID	プロジェクトID	varchar(12)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		Pr_Name	プロジェクト名	varchar(16)	<input type="checkbox"/>	<input type="checkbox"/>
		Role	役割	varchar(16)	<input type="checkbox"/>	<input type="checkbox"/>

図10. テーブル R_Project の論理設計
 Figure 10. Logical design of table R_Project

図9と図10は、それぞれ、テーブル R_User とテーブル R_Project の論理設計の詳細を示している。一般にデータベース設計では、実装で使用する物理名と設計書で言及される論理名を厳密に定義している。従って、設計工程の後半では、データベース設計で定義された物理名を使った SQL 文を定義し検証することができる。

図9と図10に示したように、物理名と論理名は厳密に対応づいていることから、ソースプログラムに含まれる SQL 文と設計書とのトレーサビリティは、物理名と論理名の文字列一致によって確認することができる。

3.4.4 処理フロー設計

SysML のアクティビティ図 (Activity Diagram) では、入力、出力、処理、条件分岐、同期などの要素を使って処理フローを記述することができる。

図11は、本文で対象にしているログオン機能の処理フローを記述したものである。ログオン機能の処理フローは、ログオン情報 (ユーザ ID、パスワード、プロジェクト ID) を取得する処理 (LonOnForm)、ログオン情報に基づいて実行する処理内容 (LogOnProcess) および、処理結果に応じた遷移先の定義に大別できる。これらの分類は、アクティビティ図の「アクティビティ区画」で表現している。

図11の LonOnForm では、ユーザ ID、パスワード、プロジェクト ID を取得するが、これらは、対応する SysML のアクティビティで表現している。図11の LogOnProcess では、ログオン処理の内容を記述する。データベース設計の結果を参照して、データベースの検索で使用する SQL 文を正確に定義している。パスワードの一致判定の結果が Fasle であれば「ログオン失敗画面」に遷移する。パスワードが一致していて、そのパスワードの期限が 10 日以内の場合「期限切れ警告画面」に遷移する。パスワードが一致していて、そのパスワードの期限が 11 日以上の場合は「プロジェクト情報管理システム」の初期画面を表示する (ログオン成功)。各アクティビティに対応する、要求項目、画面名へのトレーサビリティは 《satisfy》 充足関連で表現している。

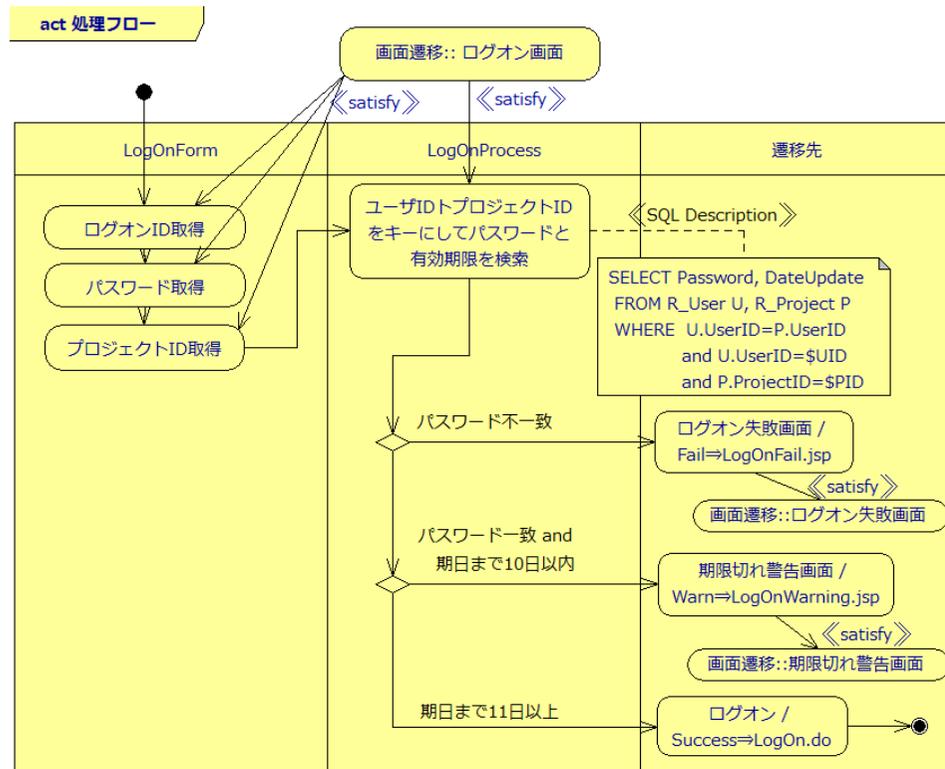


図 11. SysML による処理フロー設計
 Figure 11. Process flow design by SysML

4. 成果物間のトレーサビリティ管理についての考察

4.1 データベース設計とトレーサビリティ

データベース設計は、通常、外部設計の早い段階で実施される。多くの実務向け情報システム開発では、テーブル名、項目名、キー項目、データ型などの情報を開発メンバーで合意・共有しながら、後続の設計が実施される。注目すべきは、各項目の名称（半角全角の区別を含め）が厳密に定義されることである。

データベース設計の理念は、業務向け情報システム設計で実践されているだけでなく、データベース設計支援ツールでも実装されている。図 8 に示したテーブル関連図（IDEFIX による表記）と図 9 と図 10 のテーブル定義は、無償のデータベース設計支援ツールで作成したものである。これらの図では、変更した内容がすべて他の図に伝搬しているという、強い意味でのトレーサビリティが管理されている。

データベースに関する用語やキー項目、データ型などをデータベース設計ツールから抽出すれば、処理フロー設計で作成された SQL 文やソースプログラムに実装されている SQL 文との整合性をチェックすることも可能である。また、実装の対象とするデータベース管理システム（RDBMS）で装備されているデータ型が厳密に管理されていることから、データベース設計からデータベースを実装するための「データベース定義言語（DDL）」の自動生成も可能である（設計から実装へのトレーサビリティが管理されている）。なお、詳細については、別稿にて報告する予定である。

4.2 SysML とトレーサビリティのモデルフレームワーク

「3.4.1 SysML 要求定義」で述べたように、SysML の要求定義では、要求要素間に数種類の関連を定義することができる。これに加え、SysML の要求定義要素を、SysML の他の要素（振舞い図、構成図）と関連付けることができる。関連付け可能な要素は、パッケージ(Package)やフレーム(Frame)によって定義された名前空間(Namespace)に含まれるものに限定される。これらの機能を組み合わせることにより、仕様の上では、SysML の任意の構成要素間にトレーサビリティ関連を定義することが可能である。

しかし、システムモデリングを行う立場からは、この仕様は広すぎると考える。例えば、多くの場合、要求定義からブロック図の関連と、ブロック図から内部ブロック図の関連を組み合わせることで関連性を管理する。このようなケースでは、要求定義から内部ブロック図への関連は参照できないように制限することが望ましいと考える。

成果物のどの組み合わせに対して関連性（トレーサビリティ）を管理するかは、プロジェクトごとに異なる。従って、SysML の体系をベースに、本文「3.3 モデルフレームワーク」で述べたような、モデルフレームワークに基づいて名前空間を絞り込み、トレーサビリティ関連を管理する機能が必要と考える。

4.3 要求定義の粒度

SysML の要求定義では、要求項目がテキストで表現されていることを前提としている。本文の図 6 は、要求項目を、原則として、画面単位で定義している。一般に、1 つの画面に関する記述は、一般的な技術文章である。この技術文章を機械的に解析して、要求を抽出することは至難であり、要求項目の細かさ(粒度)と記述ルールなどを定める必要がある。要求項目の粒度と記述ルールは、SysML の規格対象外であり、開発工数、モデリングの目的、プロジェクトの文化にも依存する問題である。しかし、この問題を議論することなくしては、要求項目のトレーサビリティは語れない。

一つの切り口として、ISO STEP AP233 のモデルがある²⁾。ISO STEP AP233 では、

トレーサビリティ関連は「要求ビュー定義」(REQUIREMENT_VIEW_DEFINITION) に対して定義される。要求ビュー定義と要求定義のテキスト表現 (PLAIN_TEXT_ITEM) とは、図 12 に示した構造がある。すなわち、要求定義の内容が反映されるように要求ビュー定義を行い(人間の解釈が介入)、要求ビュー定義に対してトレーサビリティを定義することを前提としている。要求項目の内容を機械的に解析することが至難であることを考慮すると、これは、現実的な解であると考えられる。

SysML 規格書によれば、SysML で記述したモデルを複数のツール間でデータ交換する場合、STEP AP233 を使用することを推奨している。SysML と STEP AP233 のモデル要素間の対応付け(Mapping) については、OMG で検討中であり、成果に注目している。

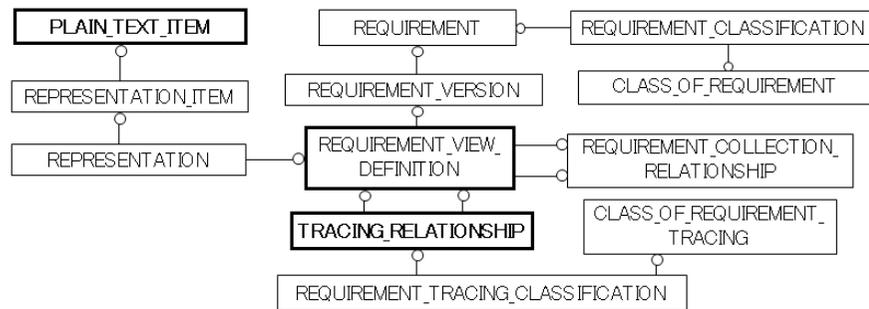


図 12. STEP AP233 のエンティティ構造
Figure 12. Entity structure of STEP AP233

5. おわりに

トレーサビリティの管理は、品質管理の基本であり、情報システム開発も例外ではない。一方で、ソフトウェアを含む情報システム開発では、管理対象が多様、物理形状がない、変更が発生する、利用者が少ないといった特質があり、物流業、流通業、製造業などと比較してトレーサビリティの管理を困難する要因が認められる⁹⁾。

情報システム開発では、データベース設計をはじめとする一部の設計手法には、実務レベルで確立した設計手法があるものの、システム全体を記述する手法が確立していない状況がある。これは、主に情報システム開発の多様性が背景にあると考えられるが、品質管理や開発のグローバル化を考慮すると、統一した記述方法の確立が期待されている。OMG SysML は、このような期待に応えるために開発されてきた国際的に利用されている記述方法であり⁴⁾、本研究でもシステムモデリングの中心的な記述

言語としている。

本文は、ログオン機能をサンプルとして、SysML の定義機能について調査した結果を報告した。SysML では、要求や設計項目を関連付ける豊富な機能があり、項目間のトレーサビリティ管理にも十分に対応できる汎用性の高い機能が提供されている。

一方、SysML を現実のシステム開発に適用するためには、関連性の管理対象とする成果物の構造(モデルフレームワーク)を定義し、この構造に基づいて成果物の項目間の関連性(トレーサビリティ)を半自動的に定義する機能の開発が重要であると考えられる。また、SysML では、要求定義はテキストによる表現を前提としていることから、要求項目の記述ルールの整備も必要である。

実務における情報システム開発では、要求、設計、テストなどの項目は数百から数千に及ぶ。これらの項目を手動で管理することは、工数と精度の観点から現実的ではない。情報システムの設計・開発環境が、項目間のトレーサビリティ管理機能を備える事が重要である。今後も本研究を継続する予定である。

参考文献

- 1) Bruce, T.: Designing Quality Databases with IDEF1X Information Models, New York Dorset House(1992).
- 2) ISO 10303 STEP, (2008). http://ja.wikipedia.org/wiki/ISO_10303.
- 3) Lormans, M. and van Deursen, A. : Can LSI help Reconstructing Requirements Traceability in Design and Test?, Proc. of the IEEE International Conference on Software Maintenance and Reengineering, pp. 47-56(2006).
- 4) Object Management Group: OMG Systems Modeling Language Version 1.2 (2010).
- 5) Pohl K.: PRO-ART: Enabling Requirements Pre-Traceability, Proc. of the IEEE Intl. Conference on Requirements Engineering(ICRE), pp76-84(1996).
- 6) Ramesh B. and Jarke M.: Toward Reference Models for Requirements Traceability, IEEE Transactions on Software Engineering Vol.27, No. 1, pp.58-93 (2001).
- 7) Royce W.W.: Managing the development of large software systems, Proc. of the IEEE WESCON, pp1-9 (1970).
- 8) Spanoudakis, G. and Zisman A.: Software Traceability: A Roadmap, Handbook of Software Engineering and Knowledge Engineering. World Scientific Publishing, pp.395-428 (2005).
- 9) 宇田川: トレーサビリティの定義・効果・課題に関する考察, 情報処理学会, 研究報告 2010-IS-111-8(2010).
- 10) 産業経済省商務情報政策局(編): 情報システム・ソフトウェアの信頼性及びセキュリティの取組み強化に向けて, オーム社, 2010年3月.
- 11) (独)情報処理推進機構 ソフトウェア・エンジニアリング・センター著: ソフトウェア開発データ白書 2009, 日経 BP 社(2009).
- 12) 総務省: 平成 22 年版情報通信白書, 2010年7月, <http://www.soumu.go.jp>.