【解説】

情報システム開発標準におけるトレーサビリティの事例と今後

宇田川佳久　（三菱電機インフォメーションシステムズ（株））

情報システムが社会活動のインフラとして普及するのに伴い，情報システムの障害は現実の社会活動に多大な影響を与えており，情報システムの品質に対する要求は高まっている．トレーサビリティ（追跡可能性）の管理は品質管理の基本であり，情報システム開発においてもその重要性が認識されている．本稿は，情報システム開発におけるトレーサビリティの概要と利用効果について事例を示しながら解説する．続いて，情報システム開発標準におけるトレーサビリティの事例として，米国カーネギーメロン大学ソフトウェア工学研究所の能力成熟度モデル統合（CMMI：Capability Maturity Model Integration），米国カリフォルニア州交通局の ITS システム構築ガイドブックを紹介する．また，トレーサビリティのデータ交換に関する国際規格の一例として ISO 10303 TC184/SC4 で規格化が進められている STEP AP233 の概要を述べる．

はじめに

トレーサビリティ（traceability）の原義は“追跡（trace）可能であること（ability）”であり，日本語では，追跡可能性，来歴などと訳される．一般に，トレーサビリティは，対象物の作成者，所在，構成，変更履歴などを後から確認できる手段または機能を意味しており，品質管理の基本的な手段である．

情報処理分野においても，1970 年代には，情報システム開発におけるトレーサビリティの重要性が認識され，以後，多くの研究と支援システムの開発が行われて今日に至っている[1]．ただし，品質を特に重視するケースを除けば，一般の情報システム開発プロジェクトでは，トレーサビリティの管理が十分に定着しているとは言い難い状況にある．主な理由としては，トレーサビリティの管理コストに対する効果が十分に認識されていないことが挙げられる．トレーサビリティの管理コストと効果については，情報システム開発プロジェクトにかかわる開発者だけでなく，管理者や発注者の品質管理に対する理解が不可欠である．本稿では，情報システム開発プロジェクトにかかわるすべての方々を対象とし，情報システム開発標準におけるトレーサビリティの事例を解説している．

図 -1 は，業務用の情報システム開発で広く採用されているウォーターフォールモデルを図示したものである．要求定義で定められた要求は，後に続く設計工程で詳細化され，コーディング工程でプログラムとして実装される．テスト工程では，プログラムが設計や要求を充足していることを検証する．ウォーターフォールモデルでは，要求定義の内容は，後に続く設計，コーディングおよびテスト工程に網羅的に引き継がれていることが前提である．この網羅的に引き継がれていることを検証する手段としてトレーサビリティがある．業務向けの情報システム開発では，設計項目やテスト項目は数百から数千におよぶことから表管理ツール（Microsoft Excel® や OpenOffice Calc など）などによる目視での管理は，筆者らの経験では，コストおよび網羅性の観点から限界に近づいていると認識している．

本稿は，情報システム開発におけるトレーサビリティ管理の定義，概要，利用効果，および，トレーサビリティの情報システム開発標準での採用状況を解説し，トレーサビリティ管理の重要性を，開発者だけでなく，管理者や発注者などの方々の理解の一助となることを目的としている．情報システム開発に関する標準化は，大学をはじめとする研究機関が公開している標準（CMMI[2]，PMBOK，SysML など），ISO 国際標準化機構の規格（ISO 10303 STEP[3]，ISO 9000，ISO/IEC 15288 など），INCOSE システムエンジニアリング国際評議会のガイドブック，および，大規模システムの運用機関（CalTrans[4]，NASA など）が公開しているガイドブックなどにより推進されている．本稿では，紙面の制約

から，数多くの情報システム開発標準のうち，比較的にトレーサビリティに関する要求が明快な下記の情報システム開発標準を対象としている．

(1) 能力成熟度モデル統合(CMMI)[2)]

(2) 高度道路交通システム（ITS：Intelligent Transportation Systems)システム構築ガイドブック[4)]

(3) ISO 10303 STEP AP233（システムエンジニアリングと設計：Systems Engineering and Design)[3)]

本稿の構成は次のとおり．最初に，トレーサビリティの定義と分類について概論する．次に，具体例を挙げながら，トレーサビリティの概要と利用効果について解説する．続いて，トレーサビリティの情報システム開発標準での採用事例として，能力成熟度モデル統合 CMMI[2)]，および，米国カリフォルニア州交通局が公開しているITS システム構築ガイドブック[4)] におけるトレーサビリティについて解説する．

情報システム開発が多数の企業の分業によって行われることも一般的になり，プロジェクト管理の一貫として，トレーサビリティを含むシステムエンジニアリングデータ交換のニーズが高まっている．ISO 10303 TC184/SC4 で規格化が進められている STEP AP233 は，このニーズに応えるために作成されている国際規格である[3)]．本稿では，特に，要求と設計との間のトレーサビリティに着目し，AP233 のデータ構造とツールによるトレーサビリティの確認方法について解説する．最後に情報システム開発におけるトレーサビリティの今後について展望する．

トレーサビリティの定義

◉ IEEE での定義

技術用語としてのトレーサビリティは，1990 年の IEEE 用語集[5)]で下記のように定義されている．

The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another.（開発工程における複数の成果物間に確立された関連性の程度であり，特に，前工程と後工程，または，主従関係がある成果物に関するものである）

この定義は，情報システム開発において一般的に採用されている．

◉ 要求定義前／後トレーサビリティ

Gotel 氏らは，要求トレーサビリティを下記のように定義している[6)]．

Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction.（要求トレーサビリティとは，要求のライフサイクルを記述し，前方にも後方にも探索できる能力である）

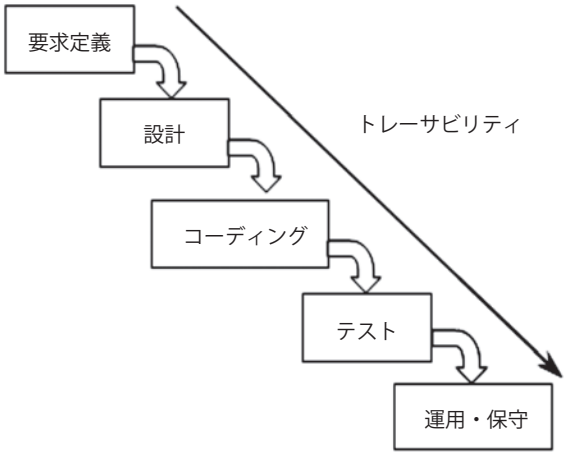

図 -1　ウォーターフォールモデルとトレーサビリティ

この定義は現在多くの研究論文などで引用されるものであり，要求のライフサイクルに着目している点と，要求を中心に前方および後方に探索することを明記していることが特徴である．

Gotel 氏らは，同じ論文で，要求定義前トレーサビリティ（Pre-RS：Pre-Requirements Specification）と要求定義後トレーサビリティ（Post-RS：Post-Requirements Specification）の概念を導入している（**図 -2**）．一般に，情報システム開発は要求定義の前に，ニーズと効果分析が実施される．この作業は，情報システムの発注者側で実施されるもので，分析結果は後続の要求定義に網羅的に引き継がれていることが前提である．一方，要求定義の後に続く，設計，コーディング，テスト工程は，情報システムの開発者側で実施されるものである．トレーサビリティの管理者や利用者が異なることから，要求定義以前のトレーサビリティと要求定義以降のトレーサビリティとを区別すべきであるというのが，Gotel 氏らの主張である．

トレーサビリティの利用の観点からは，要求定義前トレーサビリティは，要求定義以前の工程で作成された成果物と要求定義との間のトレーサビリティを示し，要求定義における要求の存在理由や優先度を提供するものである．一方，要求定義後トレーサビリティは，成果物としての要求定義，設計，プログラム，テスト間のトレーサビリティを示し，要求定義における要求が設計に反映され，コーディングされ，テストされ，要求が正しく実装されたことを検証する手段を提供している．

要求定義前トレーサビリティと要求定義後トレーサビリティは，後述する米国カリフォルニア州交通局の ITS システム構築ガイドブック[4)]でも採用されている．

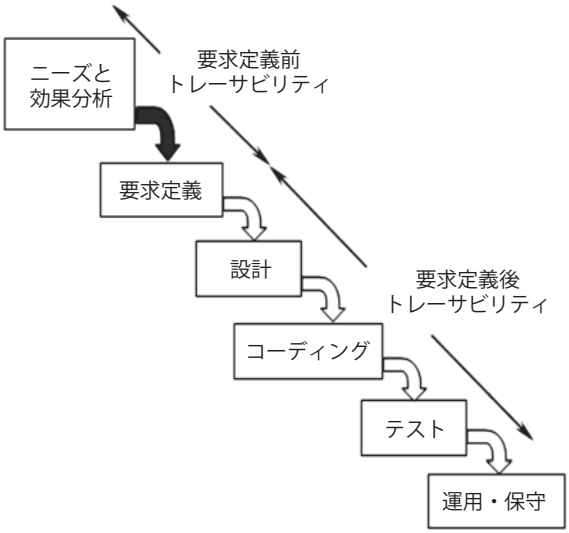

図 -2　要求定義前トレーサビリティと要求定義後トレーサビリティ

要求 1.　起動方法
・ポータル画面から申請画面を起動する.
要求 2.　申請画面
・申請画面で氏名とメールアドレス(申請情報)を入力する.
要求 3.　申請処理
・申請情報にエラーがなければ，会員 ID，仮パスワードと登録画面の URL を生成し，メールで送信する．次に，申請成功のメッセージと登録画面への遷移を可能にするボタンを表示する.
・申請情報にエラーがある場合は，エラーメッセージと申請画面への遷移を可能にするボタンを表示する.
要求 4.　登録画面
・登録画面で会員 ID，仮パスワード，本パスワード (2 回) (登録情報)を入力する.
要求 5.　登録処理
・登録情報にエラーがなければ会員登録し，登録成功メッセージとポータル画面への遷移を可能にするボタンを表示する.
・エラーがある場合は，エラーメッセージと登録画面への遷移を可能にするボタンを表示する.

図 -3　要求定義の例

◉ 定義のバリエーション

IEEE 用語集 [5]，CMMI[2]，ISO 10303 AP233[3] では，トレーサビリティを成果物間の関連（relationship または association）と定義している．一方，Gotel 氏ら [6] や ISO 9000 などでは，物の履歴などを管理（記述し探索）する能力 (ability) と定義している．この定義は，物にかかわる成果物間の関連が物のライフサイクルに渡って識別され，管理されている際にトレーサビリティという能力が備わっていると解釈できるものである．このほかにも，トレーサビリティの定義に関してはいくつかのバリエーションがあるが，本稿の主旨から逸脱するため，これ以上の解説は割愛する．以降，本稿で扱う情報システム開発標準に従い，トレーサビリティを成果物間の関連と定義し解説する．

トレーサビリティの概要

◉ トレーサビリティの事例

この節では，要求定義後トレーサビリティのイメージを Web サイトの会員登録機能に対する要求定義，外部設計とテストケースを例にして述べる．

図 -3 は，会員登録に関する要求定義を示している．会員登録は，申請と登録から構成されるものとする．申請の起動は，ポータル画面から行う（要求 1）．申請に必要な申請情報は，氏名とメールアドレスであり，申請画面で入力する（要求 2）．申請処理では，氏名とメールアドレスを分析し，エラーがなければ，会員 ID，仮パスワードと登録画面の URL を生成し，ユーザにメールを送信する（要求 3）．登録は，ユーザにメールで送信された会員 ID，仮パスワードと本パスワード（2 回）を登録画面で入力し（要求 4），登録処理を実行する（要求 5）ものとする．

要求定義に続いて，外部設計を行う．**図 -4** の左半分は，図 -3 の要求定義に対応する外部設計の概要を図示しており，矢印付きの実線は，項目間のトレーサビリティ (成果物間の関連) を示している．要求 1 の「起動方法」として，ポータル画面の「申請画面へ」ボタンをクリックし，申請画面に遷移するものとしている．要求定義の「起動方法」は，外部設計の「ポータル画面」と「申請画面」に関連していることから，「起動方法」から「ポータル画面」と「申請画面」に対するトレーサビリティが存在する．

要求 2 の「申請画面」に対応して，外部設計では「申請画面」を定義し，氏名とメールアドレスの入力項目および申請処理への遷移を行うための「申請実行」ボタンを定義している．したがって，図 -4 に示したように，要求定義の「申請画面」から外部設計の「申請画面」へのトレーサビリティがある．要求 3 の「申請処理」に関しては，申請処理が成功した場合，外部設計の「申請成功画面」を表示し，失敗した場合は「申請失敗画面」を表示するものとしている．したがって，要求定義の「申請処理」から，外部設計の「申請成功画面」と「申請失敗画面」へのトレーサビリティがある．要求 4 の「登録画面」は，要求 2 の「申

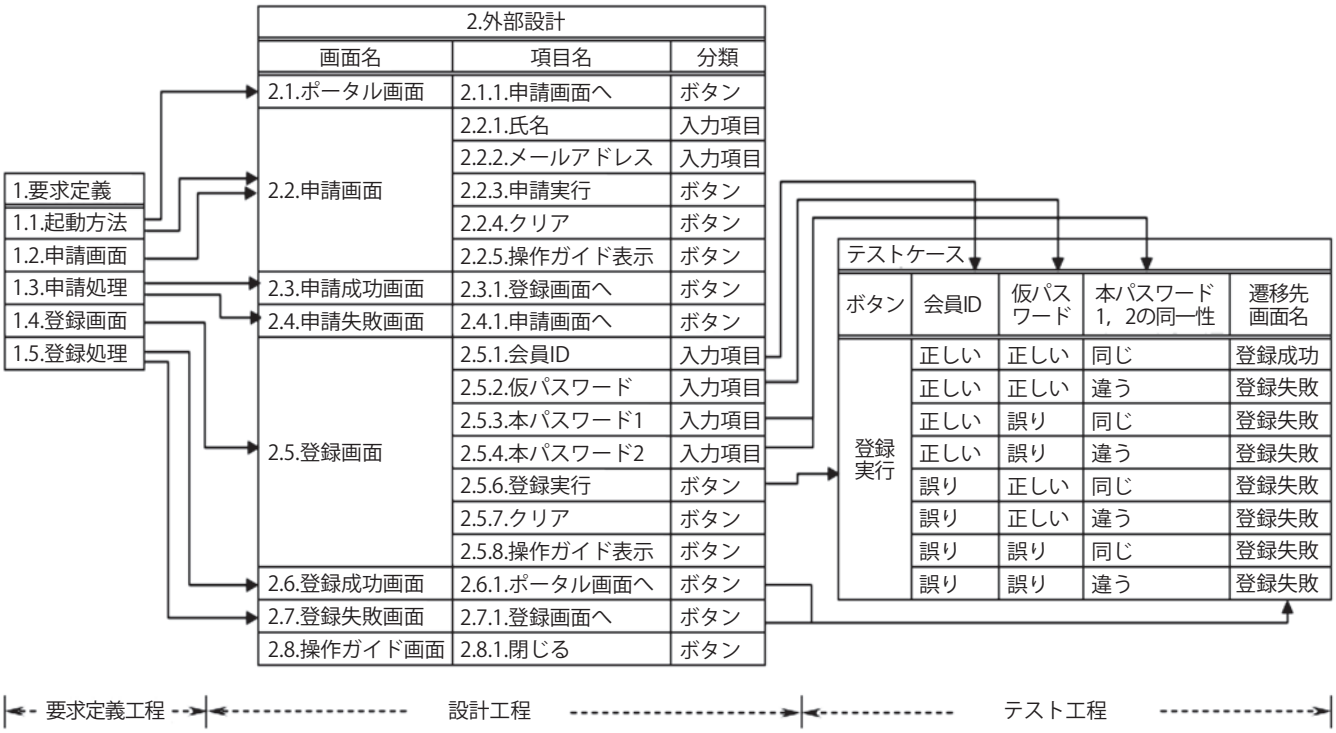

図 -4　トレーサビリティ（成果物間の関連）の例

請画面」と同様のトレーサビリティがある．また，要求 5 の「登録処理」は，要求 3 の「申請処理」と同様のトレーサビリティがある．

一般に，外部設計では，要求定義で定められた要求に加え，操作性の観点からの検討が行われる．図 -4 では，申請処理および登録処理の操作ガイドを表示する画面（操作ガイド画面）を追加している．この操作ガイド画面の追加を外部設計における追加設計項目として扱うか，要求定義の要求漏れとして扱うかは，開発プロジェクトのポリシーに依存する．本稿では，外部設計における追加設計項目として扱うこととしているため，要求定義から「操作ガイド画面」に対応するトレーサビリティはない．

図 -4 の右半分は，外部設計の「登録画面」の「登録実行」に関するデータ項目とテストケースとのトレーサビリティを示している．データ項目としては，会員 ID，仮パスワードと本パスワード 1 および 2 があることから，テストケースとしては，図 -4 に示した 8 個のパターンがある．これらのすべてのパターンに合格することがテストの完了条件になる．

工程にまたがる項目間のトレーサビリティを一覧表として表現したものをトレーサビリティ・マトリックス（Traceability Matrix）と呼んでいる [7]．**図 -5** は，図 -4 の要求定義と外部設計に対応するトレーサビリティ・マトリックスであり，横軸に要求定義の項目，縦軸に外部設計の項目を配置し，トレーサビリティが存在する場合は，交差する個所に矢印を記入したものである．図 -5 では，要求定義のすべての項目が外部設計に引き継がれていることを容易に確認できる．一方，外部設計の「操作ガイド画面」が要求定義のどの項目にも関連していないことから，外部設計で独自に取り入れた項目であることを確認することができる．

図 -5　トレーサビリティ・マトリックスの例

◉ トレーサビリティの利用効果

ここでは，トレーサビリティ（成果物間の関連）管理の利用効果として，工程に跨る要素間のカバレッジ（網羅性）分析と変更の影響分析に着目して解説する [7]．

・カバレッジ分析

トレーサビリティは，前工程と後工程の成果物に過不足がないことを確認する手段を提供しており（カバレッジ分析），開発工程の成果物の品質向上に効果がある．カバレッジは，トレーサビリティが定義されている項目数を全項目数で割った値をパーセントで表したものである．図 -4 の場合，要求定義のすべての項目は，外部設計の項目に関連付いていることから，要求定義から外部設計へのカバレッジは 100% である．これは，要求定義のすべての要素が外部設計に引き継がれていることを意味している．一方，外部設計から要求定義へのカバレッジは，87.5%（7/8 項目）である．

前工程と後工程でトレーサビリティがない項目が発見された場合，原因に応じて下記の対処が必要である．なお，図 -4 の「操作ガイド画面」の場合は，下記の (2) の対処を実施している．

(1) 前工程で該当する項目の検討が漏れていた場合は，前工程の成果物を修正する．
(2) 後工程で詳細化が行われ，前工程に関連しない新たな項目が追加された場合は，そのままとする．
(3) トレーサビリティの定義漏れの場合は，トレーサビリティを定義する．

要求定義からテストまでのトレーサビリティにより，要求に対応する設計が完結し，設計に対する実装が十分にテストされ，結果として，要求定義どおりの製品が完成したことを検証することが可能になる．

・変更の影響分析

現実のシステム開発では，設計変更が避けられない．一般に，設計変更は，後の工程で発生するほど変更の影響を受ける範囲が広がり，変更に要するコストが増大する．また，変更への対応の漏れは，ドキュメントを含めた製品の品質劣化につながる．

トレーサビリティにより，変更によって影響を受ける範囲を特定することが可能になり，変更コスト低減だけでなく変更漏れの防止により，品質を確保することができる．たとえば，図 -3 の要求定義の「申請画面」で入力すべき項目として「所属」を追加する場合，図 -4 のトレーサビリティにより，外部設計の「申請画面」に「所属」を入力する項目を追加すべきであることが判明する．障害が発生した場合も，トレーサビリティにより，テスト状況を効率的に確認し，テストケースの追加などの再発防止に向けた対策を実施できる．

CMMI とトレーサビリティ

ここでは，情報システム開発にかかわる多くの企業，組織においてプロセス品質改善のために導入が進められている CMMI[2] におけるトレーサビリティ（成果物間の関連）の扱いについて解説する．

CMMI は，米国カーネギーメロン大学ソフトウェア工学研究所（SEI：Software Engineering Institute）がソフトウェアの開発能力を客観的に評価するための指標として定めたものであり，開発のプロセスに注目した評価を行う．なお，CMMI の規格書は SEI の Web ページから無償でダウンロードすることができる．

CMMI は，ソフトウェアの開発能力を以下の 5 段階で評価している．

レベル 1 初期：プロセスがメンバに依存し，組織としては確立していない段階
レベル 2 管理：反復してプロセスが実行できる段階
レベル 3 定義：プロセスが組織で定義され共有されている段階
レベル 4 定量的管理：品質測定基準に基づきプロセスが管理されている段階
レベル 5 最適化：品質測定基準に基づき継続的にプロセスを管理し最適化している段階

レベル 1 は，組織としてのプロセスが確立されていない段階であり，実質的にはレベル 2 から 5 が，企業や組織が達成すべきレベルである．

表 -1 は，現時点で最新である CMMI バージョン 1.2（以降，単に CMMI と表記）におけるプロセス領域，レベルとトレーサビリティに関する規定の有無について一覧にまとめたものである．CMMI は，要求管理，プロジェクト計画策定，構成管理など 22 個のプロセス領域で構成されていて，レベルに応じてサポートすべきプロセス領域が定められている．このうちトレーサビリティに関する記述があるのは，表 -1 に示した 10 個のプロセス領域である．なお，トレーサビリティが関連するプロセス領域のレベルは 2 から 4 であり，CMMI においてトレーサビリティが基本条件となっていることが分かる．

以下，トレーサビリティの要求事項が顕著な要求管理プロセス（Requirements Management）に着目し，概要を解説する．要求管理の目的は，プロジェクトの進展に伴って発生する要求の変更を管理し，プロジェクト計画，成果物と要求の間に発生するすべての不整合を特定し，是正することである．トレーサビリティに関しては，原要求と成果物を構成するすべての要素との間で双方向の追跡が可能であることを規定している．要求管理プロセスでは，この目的を達成するための固有プラクティス（Specific Practices）として下記を規定している．

SP 1.1 要求の理解

要求の評価基準を定め，発注者との理解を共有する．

SP 1.2 要求の承認

プロジェクト参加者から要求に対する合意や承認を得る．

No	英語名称	日本語名称	レベル	トレーサビリティ
1	Causal Analysis and Resolution	原因分析と解決	5	
2	Configuration Management	構成管理	2	○
3	Decision Analysis and Resolution	決定分析と解決	3	○
4	Integrated Project Management	統合プロジェクト管理	3	○
5	Measurement and Analysis	測定と分析	2	○
6	Organizational Innovation and Deployment	組織改革と展開	5	
7	Organizational Process Definition	組織プロセス定義	3	
8	Organizational Process Focus	組織プロセス重視	3	
9	Organizational Process Performance	組織プロセス実績	4	
10	Organizational Training	組織トレーニング	3	
11	Product Integration	成果物統合	3	
12	Project Monitoring and Control	プロジェクトの監視と制御	2	
13	Project Planning	プロジェクト計画策定	2	
14	Process and Product Quality Assurance	プロセスと成果物の品質保証	2	
15	Quantitative Project Management	定量的プロジェクト管理	4	○
16	Requirements Development	要求開発	3	○
17	Requirements Management	要求管理	2	○
18	Risk Management	リスク管理	3	
19	Supplier Agreement Management	供給者合意管理	2	○
20	Technical Solution	技術解	3	○
21	Validation	妥当性確認	3	
22	Verification	検証	3	○

表 -1　CMMI Ver.1.2 のプロセス領域とトレーサビリティ

SP 1.3 要求変更の管理

要求変更に伴う影響を評価し，変更内容を管理する．

SP 1.4 要求の双方向トレーサビリティの維持

詳細化の各レベルにおける成果物と原要求との間の双方向トレーサビリティを維持する．

SP 1.5 プロジェクト作業と要求の間の不整合の特定

原要求とプロジェクト計画および成果物との不適合を特定し，必要な是正処置を実行する．

このうち，SP 1.4 がトレーサビリティに関係する項目であり，以下を実施することを規定している．

(1) 詳細化された成果物に対応する原要求を特定し，文書化する．
(2) 原要求から機能，インタフェース，オブジェクト，要員，プロセスや成果物に至るトレーサビリティを維持管理する．
(3) 要求のトレーサビリティ・マトリックスを作成する．

トレーサビリティは，要求変更によって発生するプロジェクト計画や成果物への影響を把握する手段として特に重要である．原要求と詳細化された要求の間の双方向トレーサビリティによって，原要求が網羅的に詳細化されていることを確認することができる．また，トレーサビリティにより，中間成果物と最終成果物，要求変更の内容，要求とテスト計画の対応についても探索可能になる．

ITS システム構築ガイドブックとトレーサビリティ

CalTrans（米国カリフォルニア州交通局：California Department of Transportation）は，24,000km に及ぶカリフォルニア州のハイウエイの管理を行う公的機関であり，ハイウエイの効率的な運用とサービス向上のため，ITS の導入と保守を推進している．

CalTrans は，ITS の開発に多数の企業が円滑に参画できるよう，システム開発プロセスを定めた ITS システム構築ガイドブック（System Engineering Guidebook for ITS）を公開している [4]．

このガイドブックでは，ウォーターフォールモデルとスパイラルモデルの特徴を取り入れた Vee と命名したモデル（以降，Vee モデル）を定めている（**図 -6**）．Vee モデルの左側は，システム要素の定義，および，詳細化の工程を示している．Vee モデルの底の部分は，実装工程であり，ソフトウェアコーディング，ハードウェア製作および製品購入が含まれる．Vee モデルの右側は，テストおよび検証により，システム要素を統合し，最終システムを作り上げる工程を示している．

業務要求とシステム要求の間に位置する「完了評価 1」は，CalTrans（発注者）による業務要求の作成作業が完了したことを確認し，システム開発者によるシステム要求の作成作業を開始することを判断するものである．また，システム検証とシステム検収の間に位置する「完了評価 2」は，システム開発者による一連の作業が完了したことを確認し，発注者による運用開始を判断するもの

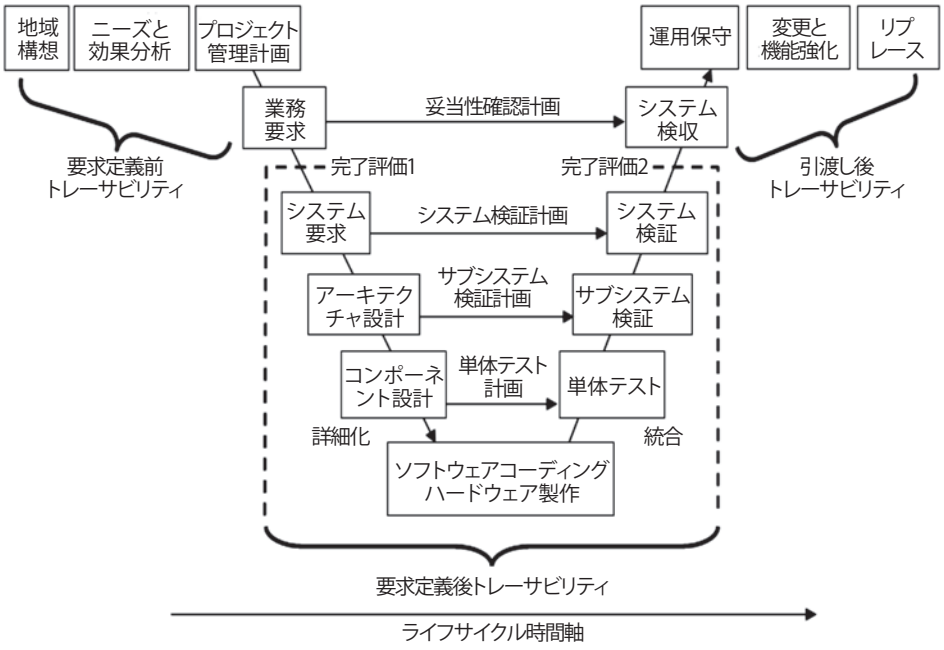

図-6 Vee モデル

である．

Vee モデルでは，下記の 3 種類のトレーサビリティを規定している．

(1) 要求定義前（Pre-RS）トレーサビリティは，業務要求とシステム要求との追跡を可能にするためのものである．要求の変更が発生した場合は，技術，費用，スケジュール面での影響を査定することを支援する．このトレーサビリティは双方向である．すなわち，すべての業務要求がシステム要求に関連付いていること，逆に，すべてのシステム要求が業務要求に関連付いていることを確認することを可能にする．

(2) 要求定義後（Post-RS）トレーサビリティは，システム要求，設計，ソフトウェアコーディング，ハードウェア製作，および，検証計画などでの成果物間の追跡を可能にするためのものである．要求定義後トレーサビリティは，双方向の追跡が可能であり，システム要求がすべて実装され，検証されたことを確認する手段を提供する．また，変更が発生した場合は，技術，費用，スケジュール面での影響を査定することを可能にする．

(3) 引渡し後（Post-Delivery）トレーサビリティは，システムが納入されてからリプレースされるまでの運用保守工程で使われるものである．このトレーサビリティは，双方向の追跡が可能であり，システム要素の変更や機能強化を行う場合の技術，費用，スケジュール面での影響を査定することを可能にする．

また，このガイドブックでは，トレーサビリティの実務的な管理方法について，下記のように記載している．

- 要求が 100 個以下の場合，表管理ツールによる管理が可能である．
- 要求が 100 個以上の場合，トレーサビリティ管理機能を提供している要求管理ツールを使用することを推奨する．

現在の技術では，要求定義書の中身を理解してトレーサビリティを自動的に定義し，要求定義書の変更に対して自動的にトレーサビリティを修正することはできない[7]．そのためトレーサビリティの定義は手動によって行われ，トレーサビリティの管理には多くのコストを要する．このガイドブックでは，トレーサビリティの管理コストと要求管理ツールのコストとを比較し，要求が 100 個以上の場合は，要求管理ツールによる管理が現実的であるとしている．

データ交換国際規格とトレーサビリティ

ISO 10303 は，通称 STEP（Standard for the Exchange of Product model data）と呼ばれている国際規格であり，特定のシステムに依存しないデータ形式で製品に関するあらゆるデータを交換する仕組みを提供することを目的としている[3]．STEP 全体としては数百の分冊から構成されている．主な分冊は以下のものがある．

分冊 10 番台：情報モデルの記述手法を規定している．

分冊 20 番台：データ交換の実装方法について規定している．

分冊 200 番台：応用に特化した製品データを規定している．応用プロトコル（Application Protocols）と呼ばれ，

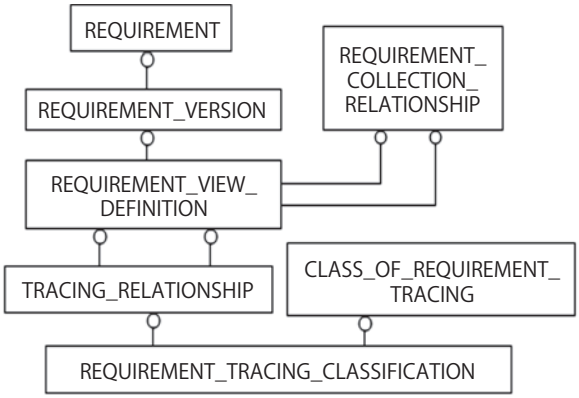

図 -7　AP233 のエンティティ（抜粋）

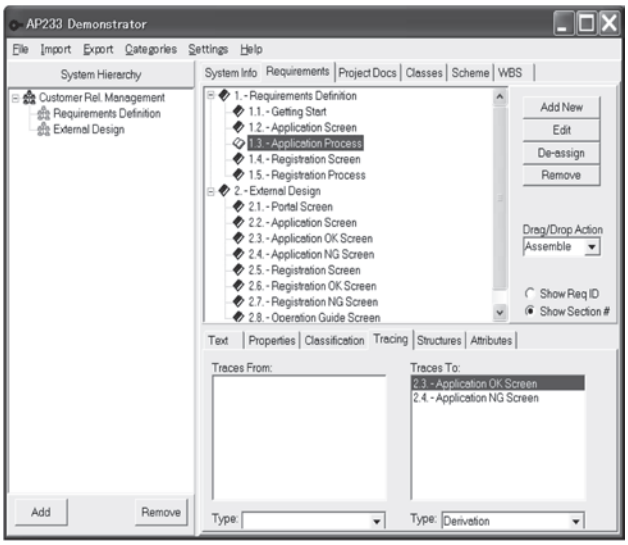

図 -8　トレーサビリティの表示例

AP2XX と表記される．

分冊 40 番台：分冊 200 番台で使う製品データの部品となる製品データを規定している．

AP233（システムエンジニアリングと設計：Systems Engineering and Design）は，システム開発における要求データの交換を目的とした応用プロトコルであり，現在，国際規格原案（Draft International Standard）を作成中である．業務向け情報システムの開発では，複数の組織による分業も珍しくなく，要求データの交換が必要となる．AP233 は，このような状況における要求データを交換するための規格である．AP233 は，要求をはじめとして，システムの構造，プロジェクト管理データ，スケジュール，リスク管理データ，問題管理データなども対象としているが，本稿の主旨に沿って，要求のトレーサビリティに着目して概要を述べる．

図 -7 は，AP233 における，要求とトレーサビリティに関連するエンティティ（Entity）を，EXPRESS-G[3] と呼ばれている STEP におけるモデリング言語で表現したものである．EXPRESS-G では，四角形でエンティティを示し，丸印付きの実線で丸印が付いている先のエンティティを参照していることを示す．要求（REQUIREMENT）には，要求バージョン情報（REQUIREMENT_VERSION）と要求ビュー定義（REQUIREMENT_VIEW_DEFINITION）が関連付いている．追跡関連（TRACING_RELATIONSHIP）は，2 つの要求ビュー定義の関連として定義されている．追跡関連には，追跡の分類（CLASS_OF_REQUIREMENT_TRACING）が関連付いていて，追跡関連に，任意の分類情報を付加することができる．

Eurostep（http://www.eurostep.com/）は，AP233 に準拠したビューア（正式名称は Eurostep AP233 Demonstrator，以降 AP233 ビューアと表記）を無償で提供している．**図 -8** は，図 -4 の要求定義と外部設計間のトレーサビリティを AP233 ビューアで表示したものである．ただし，AP233 ビューアは日本語に対応していないため，図 -4 の各項目名を英語で表記している．

要求定義または外部設計の項目をクリックすると Tracing タブにトレーサビリティ情報が表示される．なお，AP233 では双方向のトレーサビリティをサポートしている．図 -8 では，要求定義の「1.3. 申請処理」（1.3.-Application Process）をクリックしているためトレース先（Traces To：）に外部設計の「2.3. 申請成功画面」（2.3.-Application OK Screen）と「2.4. 申請失敗画面」（2.4.-Application NG Screen）が表示されている．一方，外部設計の項目をクリックするとトレース元（Traces From：）に要求定義へのトレーサビリティ情報が表示される．

次に図 -4 に示した要求と外部設計項目，および，項目間のトレーサビリティに関するデータが AP233 でどのように表現されるか，STEP ファイル[3] の内容を示して概要を解説する．

図 -9 は，STEP ファイルのデータを抜粋したものである．"#" に続く数字は，STEP ファイル内で一意な識別番号である．"=" に続く文字列は，エンティティの名称であり，図 -7 に示したエンティティの名称に対応している．

#400 ～ #402 の 3 レコードで，「1. 要求定義」を示す項目である「1.-Requirements Definition」を定義している．#409 ～ #411 は，要求定義で定義された「1.3. 申請処理」（1.3.-Application Process）を定義している．REQUIREMENT_COLLECTION_RELATIONSHIP エンティティは項目の階層構造を定義するものであり，#482

```
#400 = REQUIREMENT_VIEW_DEFINITION ((#3), #401, '', '1.', #3, '', $, .F.);
#401 = REQUIREMENT_VERSION ('1.0', 'version 1.0 of 1.', #402);
#402 = REQUIREMENT ('Requirements', '1.', 'Requirements Definition');
...
#409 = REQUIREMENT_VIEW_DEFINITION ((#3), #410, '', '1.3.', #3, '', $, .F.);
#410 = REQUIREMENT_VERSION ('1.0', 'version 1.0 of 1.', #411);
#411 = REQUIREMENT ('Requirements', '1.3.', 'Application Process');
#482 = REQUIREMENT_COLLECTION_RELATIONSHIP ($, 'Assemble', #409, #400, 'Assemble');
...
#500 = REQUIREMENT_VIEW_DEFINITION ((#3), #501, '', '2.', #3, '', $, .F.);
#501 = REQUIREMENT_VERSION ('1.0', 'version 1.0 of 1.', #502);
#502 = REQUIREMENT ('Requirements', '2.', 'External Design');
...
#509 = REQUIREMENT_VIEW_DEFINITION ((#3), #510, '', '2.3.', #3, '', $, .F.);
#510 = REQUIREMENT_VERSION ('1.0', 'version 1.0 of 1.', #511);
#511 = REQUIREMENT ('Requirements', '2.3.', 'Application OK Screen');
#582 = REQUIREMENT_COLLECTION_RELATIONSHIP ($, 'Assemble', #509, #500, 'Assemble');
#512 = REQUIREMENT_VIEW_DEFINITION ((#3), #513, '', '2.4.', #3, '', $, .F.);
#513 = REQUIREMENT_VERSION ('1.0', 'version 1.0 of 1.', #514);
#514 = REQUIREMENT ('Requirements', '2.4.', 'Application NG Screen');
#583 = REQUIREMENT_COLLECTION_RELATIONSHIP ($, 'Assemble', #512, #500, 'Assemble');
...
#906 = TRACING_RELATIONSHIP ($, 'Trace', #509, #409, 'Trace', $);
#907 = REQUIREMENT_TRACING_CLASSIFICATION (#906, #999);
#908 = TRACING_RELATIONSHIP ($, 'Trace', #512, #409, 'Trace', $);
#909 = REQUIREMENT_TRACING_CLASSIFICATION (#908, #999);
#999 = CLASS_OF_REQUIREMENT_TRACING ('Derived', 'DRV', 'Derivation');
```

図 -9　AP233 に準拠した STEP ファイル（抜粋）

は，#409 が #400 の下位の項目として存在することを定義している．

#500 ～ #502は，「2.外部設計」(2.-External Design)，同様に，#509 ～ #511 は「2.3. 申請成功画面」(2.3.-Application OK Screen)，#512 ～ #514 は「2.4. 申請失敗画面」(2.4.-Application NG Screen) を定義している．#582 と #583 は，#482 と同様に，これらの項目間の階層構造を定義している．

#906は#409（1.3.-Application Process）から#509 (2.3.-Application OK Screen) へのトレーサビリティを定義している．#907 は #906 のトレーサビリティの分類が #999 に示した展開 (Derivation) であることを定義している．同様に，#908 と #909 は，(1.3.-Application Process) から #512（2.4.-Application NG Screen）へのトレーサビリティとそのトレーサビリティの分類を定義している．

今後の展望

情報システムが社会活動のインフラとして普及するのに伴い，情報システムの障害は，現実の社会活動に多大な影響を与えている．情報システムの品質に対する要求はますます高まっている．一方，国内で開発されている業務向け情報システムの大多数がウォーターフォールモデルに基づいて開発されていることから，産業界へのインパクトを考慮した場合，ウォーターフォールモデル (またはこれに準じたモデル) におけるトレーサビリティに注目した研究が必要であると考える．

本稿で解説したように，多くの企業で採用されている CMMI[2] や，米国カリフォルニア州交通局の ITS システム構築ガイド[4] などでは，品質確保のためにトレーサビリティの管理を義務付けている．また，システム開発における要求データの交換に関する国際規格（ISO 10303 AP233[3]）でもトレーサビリティが規格化の対象に含まれている．これらの標準や規格への対応は，グローバルな情報システム開発への対応という意味においても重要である．

1970 年代から続く研究から，トレーサビリティ管理を困難にする課題として下記が指摘されている[1]．

(1) 成果物が異なる言語（自然言語，プログラミング言語）で記述されている．
(2) 成果物が異なる抽象レベル（要求，設計，実装）で記述されており，表現方法が異なる．
(3) 要求変更，設計変更に対し，トレーサビリティの変更および再定義が発生する．

これらの課題解決には，システム開発のモデル化を推し進めるアプローチなどがあり，今後の研究に注目していく必要があると考える．情報システム開発におけるトレーサビリティ管理に関する研究，および，実務レベルの活動が盛んになることを期待したい．

参考文献

1) Rilling, J., Charland, P. and Witte, R. : Traceability in Software Engineering - Past, Present and Future, CASCON (Centre for Advanced Studies Conference) 2007 Workshop Report (2007).
2) CMMI Product Team: CMMI for Development, Ver. 1.2 - Improving Processes for Better Products, CMU/SEI-2006-TR-008, Software Engineering Institute, Carnegie Mellon University (2006). http://www.sei.cmu.edu/
3) ISO 10303 STEP (2008). http://ja.wikipedia.org/wiki/ISO_10303
4) California Department of Transportation : Systems Engineering Guidebook for ITS, Ver. 2.0 (2007). http://www.fhwa.dot.gov/cadiv/segb/index.htm
5) IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE, New York (1990).
6) Gotel, O. and Finkelstein, A. : An Analysis of the Requirements Traceability Problem, Proceedings of the 1st International Conference on Requirements Engineering, pp. 94-101 (1994).
7) レフィングウェル，ウィドリング（著），石塚，荒川（監訳）：ソフトウェア要求管理，ピアソン・エデュケーション (2002).

（平成 21 年 9 月 9 日受付）

宇田川佳久（**正会員**）

udagawa-yoshihisa@mdis.co.jp

1982 年東京大学大学院博士課程修了．工学博士．同年三菱電機（株）入社．データベースの研究を経て，製造業および金融業向けの情報システム設計，開発，プロジェクト管理に従事．情報システムの品質向上，開発の効率化に興味がある．