

メタモデルによる設計情報定義とマルチビューを活用した トレーサビリティ記録方式の提案

Proposal of traceability recording method using design information definition by meta model and multi view

株式会社デンソークリエイト
DENSO CREATE INC.

○西村 隆 山路 厚¹⁾ 伊藤 善博¹⁾ 原 健三¹⁾
○Takashi Nishimura Atsushi Yamaji¹⁾ Yoshihiro Ito¹⁾ Kenzo Hara¹⁾

Abstract

Traceability management can check for omissions in design and verification, and is expected to prevent defects. In order to get the most out of this effect, it is important to record trace information at the same time as the design. However, in the field where repeated design changes occur, the actual situation is that trace information is recorded collectively after development due to the cost of recording trace information. In this paper, we propose a "traceability recording method using design information definition by meta model and multi view" as a traceability recording method that can store trace information at the same time as designing even in the field where repeated design changes occur. The application of this method is expected to significantly reduce the cost of traceability management and reduce the risk of large rework due to the discovery of design omissions at the end of development.

1. はじめに

ソフトウェア開発において、段階的に進む開発の一貫性を担保し、製品の品質保証や安全性確保のためにトレーサビリティ管理が広く用いられている。CMMI や機能安全規格 ISO26262 でもトレーサビリティ管理が求められ、その有効性が認知されている^{[1][2]}。トレーサビリティ管理により以下のような効果が期待できる。

【効果1】設計・検証の抜け漏れ確認による不具合予防

【効果2】設計変更や不具合改修時の影響範囲特定

【効果3】安全に対する説明責任履行

この効果の内、特に【効果1】の利益を最大限享受するには、設計と同時進行でトレース情報記録することが重要である。しかし、度重なる設計変更が発生する現場では、トレース情報記録にかかるコストのため、開発後にまとめてトレース情報記録しているのが実情であった。

本稿では、度重なる設計変更が発生する現場でも、設計と同時進行でトレース情報が記録できるトレーサビリティ記録方式として、「メタモデルによる設計情報定義とマルチビューを活用したトレーサビリティ記録方式」を提案する。

提案手法の有用性を確認するために、実験では以下の評価を実施した。

- ・実際の開発で提案手法の導入・運用は容易であるか
- ・トレース情報記録コストが十分小さいか

株式会社デンソークリエイト
DENSO CREATE INC.

愛知県名古屋市中区錦 2-14-19 Tel: 052-728-0771 e-mail: takashi.nishimura.j5c@jpgr.denso.com
2-14-19, Nishiki, Naka-ku, Nagoya-shi, Aichi Japan

1) 株式会社デンソークリエイト
DENSO CREATE INC.

【キーワード：】トレーサビリティ, トレース情報, メタモデル, マルチビュー, 機能安全

評価の結果、提案手法により、ほとんど工数を使わずに設計と同時進行でトレース情報を記録できることが確認できた。これにより、トレーサビリティ管理にかかるコストが大幅に削減され、かつ、開発終盤で設計の抜け漏れが発覚し大きな手戻り工数が発生するリスクを減らす効果が期待できる。

これ以降の本稿の構成は次のとおりである。2章では、従来のトレーサビリティ記録方式の問題点と解決すべき課題を示す。3章では、メタモデルによる設計情報定義とマルチビューを活用したトレーサビリティ記録方式による解決策を提案する。4章では、提案手法の評価結果と考察を示す。5章では、まとめと今後の課題について述べる。

2. 課題設定

2.1 従来のトレーサビリティ記録方式

ソフトウェア開発では、要求仕様設計から基本設計、基本設計から詳細設計のように段階的に進められる。全ての要求が正しく実現されていることを確認するために、設計要素間の導出関係をトレース情報として記録することで、トレーサビリティ管理が行われる。トレーサビリティ記録方式として、一般的に知られているものとして以下がある。[2]

A) トレーサビリティマトリクス方式

設計書などの設計成果とは別に、トレーサビリティマトリクスを用意する。トレーサビリティマトリクスには、工程単位で列を用意する。マトリクス左端に最上位工程である「要求仕様」を記述し、その右側に後工程である「基本設計」の記述を行う。(後工程も同様)これにより、工程間のトレース情報を記録していく。図1に、トレーサビリティマトリクス方式のイメージを示す。

| 要求仕様 | 基本設計 | 詳細設計 | ソースコード | 単体検査仕様 | 結合検査仕様 | システム検査仕様 |
|-------------------------------|----------|------------|--------|--------|--------|----------|
| ドライバのスイッチ操作からクルコンON要求の有無を判断する | クルコン制御状態 | スイッチ状態判定 | | | | |
| | | クルコン状態判定 | | | | |
| | | 制御状態取得 | | | | |
| | スイッチ状態 | スイッチ信号AD変換 | | | | |
| 先行車の有無をドライバに通知する | 先行車有無 | 先行車検出 | | | | |
| | 先行車情報 | 車間距離計算 | | | | |
| | 外部システム情報 | 状態取得 | | | | |
| | | 表示情報変換 | | | | |

図1 トレーサビリティマトリクス方式のイメージ

B) タグ付与方式

設計成果そのものに、設計要素単位でタグを埋め込み付番する。設計要素に関連している上位工程設計要素(導出元)のタグ番号を記入する。タグ番号を辿ることでトレーサビリティが確認できる。図2に、タグ付与方式のイメージを示す。

| | | | | | | |
|-----------------|----|--|--|--|--|--|
| REQ-ACC-01-B-7 | 説明 | 特になし。 | | | | |
| | 理由 | 現在のクルコンモードをドライバへ通知する。 | | | | |
| | 説明 | 現在巡航モードになっているのか定速モードになっているかをドライバに知らせるため。 | | | | |
| REQ-ACC-01-B-9 | 説明 | 特になし。 | | | | |
| | 理由 | クルコンが on の時にドライバが車間設定の変更を要求した場合、車間設定(長、中、短)を変更する。 | | | | |
| | 説明 | ドライバによって快適であると感じる車間距離や、運転状況によって適切な車間距離は異なるため設定により変更できるようにする。 | | | | |
| REQ-ACC-01-B-10 | 説明 | 特になし。 | | | | |
| | 理由 | 現在の車間設定をドライバへ通知する。 | | | | |
| | 説明 | 現在どのような車間設定になっているかをドライバに知らせるため。 | | | | |

導出元のタグ記入別表への記載でも可

設計成果への付番

図2 タグ付与方式のイメージ

A)方式は、トレーサビリティマトリクスにより表で一覧視できるため、情報の整合性や網羅性の確認が容易になる。しかし、設計書の見出しを複製するなどして一覧表を作成する必要がある、設計を進めるのとは別で、工数が必要になる。B)方式は、設計成果そのものにタグ付与するため、設計書を複製することなくトレース情報が記録できるという利点がある。しかし、設計要素へタグ付与するための工数が別途必要となる。また、そのままでは一覧視することが難しいため、タグ情報から別途一覧表を作成することもある。

2.2 現状分析

トレーサビリティ管理の効果の内、特に1章【効果1】の利益を最大限享受するには、設計と同時進行でトレース情報を記録することが重要である。全ての設計を完了した後にまとめてトレーサビリティを確認した場合、設計の抜け漏れ発覚が開発終盤になってしまい、手戻り工数が大きくなる。また、トレース情報記録の効率面においても、作成した設計書をもう一度見直すことになり、設計書作成と同程度の工数が必要と考えられる。しかし、多くの開発現場では、全ての設計を完了した後にトレース情報を記録しているのが実情である。2.1章 A)B)いずれのトレーサビリティ記録方式も、設計を進めるために必要な工数とは別に、トレーサビリティ管理するための工数が必要となるためである。実開発の事例では、B)タグ付与方式を採用し、市販されているトレーサビリティ専用ツールにより部分的に効率化を図ってはいるが、3万行程度の規模のソフトウェア開発において、トレース情報記録に1人月程度の工数を必要とした。開発現場を取り巻く状況として、「開発中に要求変更が多発する」、「インクリメンタルな開発のため、開発初期は設計構造が固まっていない」があり、従来方式では、「多発する変更開発に追従し続けるコストが高すぎる」、「開発完了後に一度だけトレース情報を記録した方が、コストが低いように見える」と考えるためである。

2.3 課題提起

本研究では、度重なる設計変更が発生する現場でも、設計と同時進行でトレース情報が記録できるトレーサビリティ記録方式を開発する。

3. 解決策の提案

3.1 課題の解決方針

度重なる設計変更が発生する現場でも、設計と同時進行でトレース情報が記録できるようにするためには、以下の要件を満たすことが必要と考える。

- A) 僅かなコストでトレース情報記録を可能とする
設計対象となるシステムは年々大規模化しており、設計要素は数万オーダーを超えることが多い。タグ付与などのトレース情報記録操作一つ一つは大きくないが、これが大規模になると積み重なり、無視できないコストになる。これを無視できる程度に小さくする必要がある。
- B) 複数工程の設計要素が一覧視できる
トレース情報記録では、開発工程を跨ぐ設計要素間の導出関係を記録する。このとき、複数工程の設計要素が一覧視できると効率が良い。また、記録されたトレース情報を確認し、抜け漏れ確認するためにも、同様の必要がある。
- C) 設計情報を2重持ちしない
従来のトレーサビリティマトリクス方式では、設計書とは別にトレーサビリティマトリクスを作成することで一覧視できるようにしている。これは設計情報を2重持ちすることになり、設計変更時の保守コストが大きいという欠点がある。タグ付与方式では、付与されたタグ情報から一覧視できる表などを別途作成してこれを実現している。この部分はツール活用により自動生成可能^[3]であるが、設計を更新するたび再生成する必要があり、設計と同時進行で確認するには向かない。開発現場では、工程やフェーズ区切りなどのまとまった単位で実施されることが多い。

3.2 メタモデルとマルチビューによる解決策

本研究では、メタモデルによる設計情報定義とマルチビューを活用したトレーサビリティ記録方式を考案した。考案した手法を以下に示す。

(1) メタモデルによる設計情報構造体の定義

トレース情報を記録するためには、次の二つの前提が必要である。

- ・トレース対象の設計要素がトレース情報記録する粒度で分別できること
設計要素が明確に分離されていないとトレース対象を決められないし、粒度がばらつき、設計の抜け漏れ検証が困難になる。
- ・設計要素間の導出関係が定義されていること
どの工程間、どの粒度の設計要素間で導出関係を結ぶのか、定義されていることである。これが定義されていないと、どの工程間のトレース情報なのか判別できないし、粒度が揃うこともない。

よって、本手法では、設計を開始する前に、設計要素を分別するための設計情報構造体と設計要素間の導出関係を予め定義する。これらの定義が予め決まっていれば、設計要素を追加した時点でトレース情報を記録することが可能になる。この定義には、図3に示すようなメタモデル言語（モデルを記述するためのモデル）を用いる。これは、設計成果の情報構造体を定義するための言語であり、MOFのM2層^[4]に相当するものである。設計要素を矩形で表し、トレース情報（導出関係）を矢印破線で表している。図4は、要求仕様開発工程と基本設計工程について、メタモデルを用いて表現した例である。この例では、「ソフトウェア要求」から「コンポーネント」が導出されるという導出関係が定義されている。

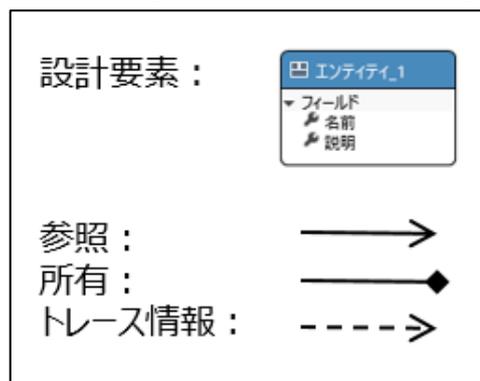


図3 メタモデル言語の定義

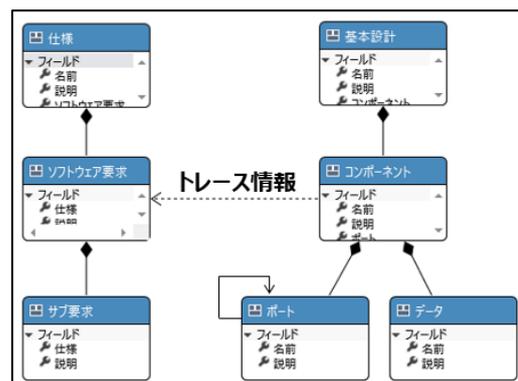


図4 設計成果のメタモデル例

(2) メタモデルに対するマルチビューの定義

設計を効率よく進めるためには、設計の関心事に応じた見た目（ビュー）を用意することが重要である。単一工程の設計を進めるためには、単一工程の設計ビューが必要となる。トレース情報を記録するためには複数工程が併記された一覧視可能なビューが必要となる。どちらのビューも必要だが、それぞれのビューを用意し、設計情報を2重で持つと変更設計時の保守性が低下する。

よって、本手法では、設計情報が一元化された状態を維持しつつ、設計の関心事に応じた複数の見た目（マルチビュー）を利用できるようにする。(1)のメタモデル定義に続き、設計を開始する前に、メタモデルに対して設計に必要なビューを複数定義する。

(3) 設計情報の一元化

図5に示すように、設計情報はメタモデルによって型を規定されたデータベースに一元化して保存する。それぞれのビューは、その設計情報を編集・参照するユーザーインターフェースとなる。よって、一つのビューで設計情報を編集した場合、それ以外のビューで同じ設計情報を表示している個所も自動で情報更新されるようにする。これにより、単一工程の設計ビューとトレーサビリティ管理のための複数工程の一覧視可能なビューの両方を持ちながら、設計情報を2重で持つことはなくなる。

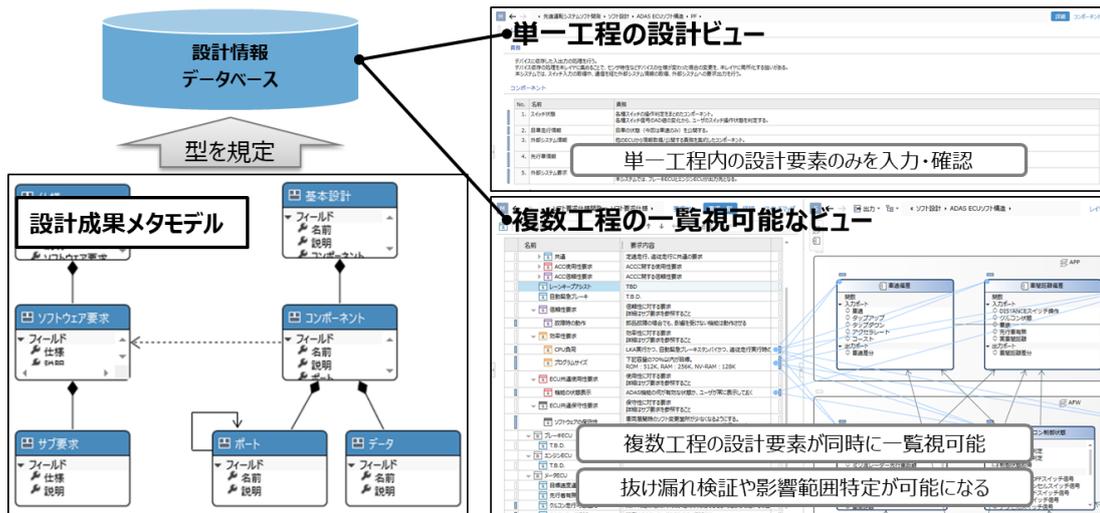


図5 メタモデルに対するマルチビューの定義

3.3 設計と同時進行でトレース情報記録

3.2章で述べた準備ができれば、設計と同時進行でトレース情報記録が可能となる。本手法では、マルチビューを活用することで、導出元の設計情報を確認しながら、導出先の設計情報を追加していくことができる。導出先の設計情報を新規追加したり、変更したりする場合に、導出元の設計情報と関連付けて編集する。この関連付け情報をトレース情報として記録する。これらの操作はツール化することで、トレーサビリティ管理しない場合と比べても、ほぼ追加コストなしで設計を進めることができると考える。具体的には、関連付けの操作は、導出元から導出先へのドラッグ&ドロップのような簡単な操作とし、操作完了時にこの関連付け情報をトレース情報として自動で記録する。図6の例で述べると、導出元工程である要求「定速走行要求」を実現するための「車間距離偏差コンポーネント」を「APPレイヤ」へ新規追加する場合、要求「定速走行要求」をドラッグして、「APPレイヤ」へドロップ操作し、「車間距離偏差コンポーネント」を新規追加する操作となる。このとき、「定速走行要求」から「車間距離偏差コンポーネント」へのトレース情報を自動で記録する。

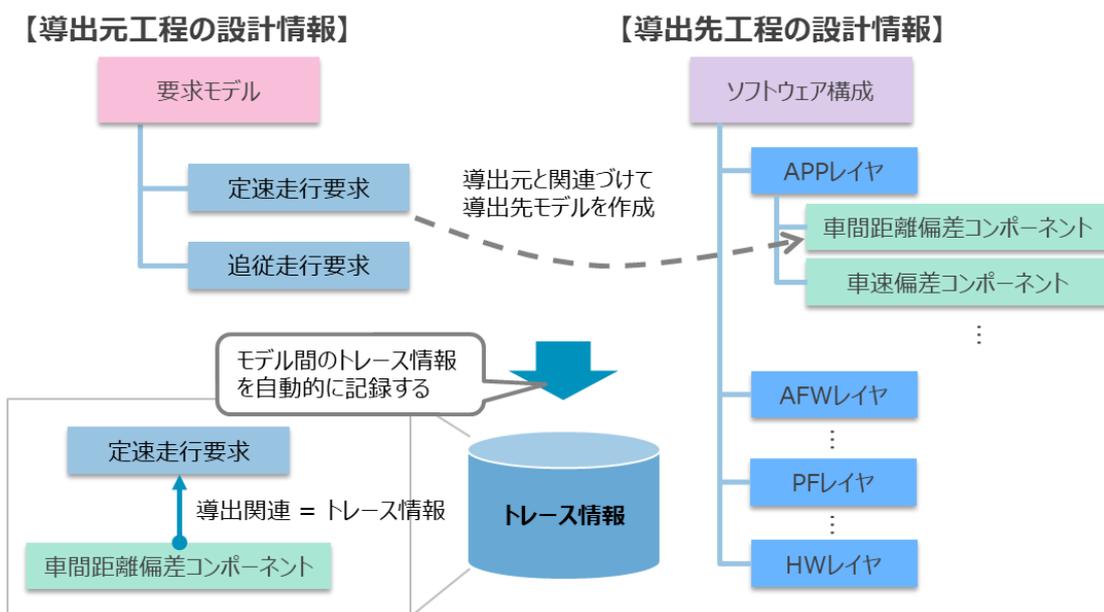


図6 設計と同時進行でトレース情報記録

4. 解決策の評価

提案手法の有用性を確認するために、実際の開発プロジェクトに提案手法を適用して評価する。今回の評価では、ユーザーインターフェースを有する大規模な Windows アプリケーションの開発プロジェクトに提案手法を適用する。本プロダクトは、高品質な組み込みソフトウェア開発に利用される設計ツールであり、本プロダクト自体にも高品質が求められるため、トレーサビリティ管理を必要とする。

提案手法適用前（トレーサビリティ管理に従来方式であるタグ付与方式を採用）と適用後のプロジェクトを比較し、以下の観点で評価する。

- A) 実際の開発で提案手法の導入・運用は容易であるか
- B) トレース情報の記録コストが十分に小さいか

4.1 ツールの選定

提案手法の実現には、3.2 章に示すようなメタモデルによる設計情報構造体の定義とそれに基づく設計情報の一元化（データベース化）、そして、マルチビューを活用したトレース情報記録の実現が必要である。これらの仕組みを僅かなコストで運用するために、ツールを導入する必要がある。導入するツールは、提案手法の実現を目的の一つとして開発した自社開発のツール「Next Design」^[5]を用いる。「Next Design」はメタモデルとビュー定義により設計情報のデジタル化を実現する設計ツールである。

4.2 メタモデルとマルチビューの定義

対象プロジェクトでは、外部仕様から外部テスト仕様へのトレーサビリティ管理に提案手法を適用する。このために必要なメタモデルとマルチビューを定義する。

- (1) メタモデルによる設計情報構造体の定義

対象プロジェクトの外部仕様設計では、ユーザーインターフェースとなる画面部品を定義し、それぞれの画面部品に対して表示仕様と操作仕様を定義する。開発対象のアプリケーションは大規模、かつ、インクリメンタルに開発を進めているため、前バージョンからの変化点に着目した外部テストにより効率よく品質確保を進めている。そのため、画面部品の表示仕様と操作仕様から今回の仕様変化点を導出し、この仕様変化点にテスト基準を照らし合わせて外部テスト仕様を設計するプロセスとしている。図7に、本プロセスを実現可能とするメタモデル事例を示す。

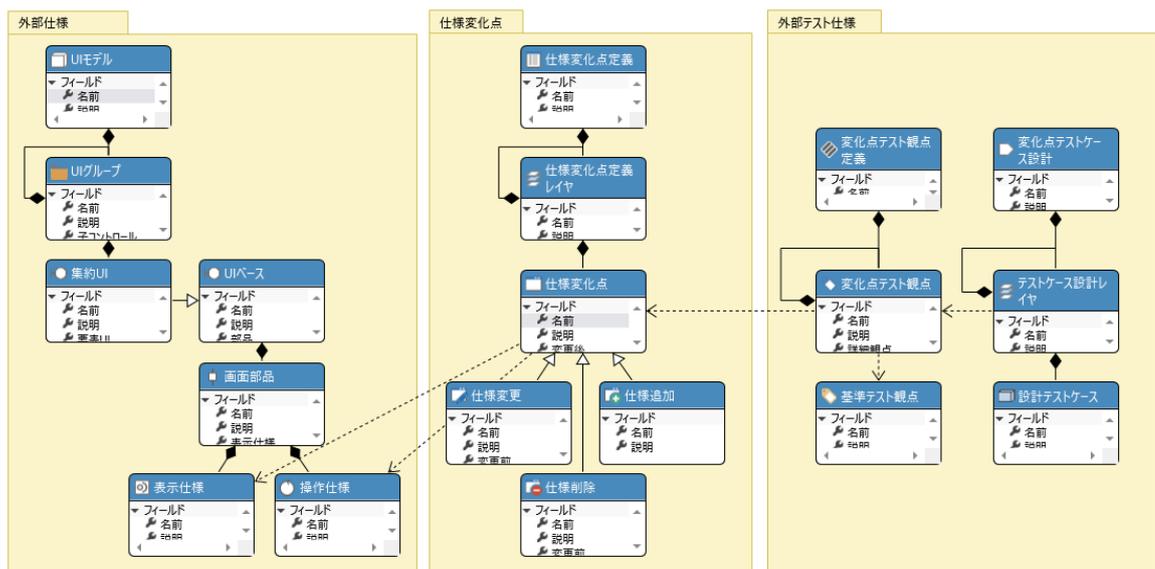


図7 メタモデル事例（外部仕様から外部テスト仕様）

- (2) メタモデルに対するマルチビューの定義

設計を効率よく進めるために、設計の関心事に応じた複数の見た目（ビュー）を用意する。

対象プロジェクトでは、単一工程の設計ビューとトレーサビリティ管理のための複数工程の一覧視可能なビューを定義する。

図8に、外部仕様工程で画面部品を定義するためダイアグラムビュー事例を示す。これは、画面部品を階層的に定義可能なビューであり、単一工程設計ビューの一例である。この他にも、振る舞いを設計するビューやデータ定義を設計するビューも定義することで様々な観点で設計が進められる。図9に、外部仕様の変化点一覧を左側に表示しながら、外部テスト仕様を導出追加するための表形式のビュー事例を示す。2つの表の間に結ばれている線は、2つの設計情報の間に導出関係があることを意味する。

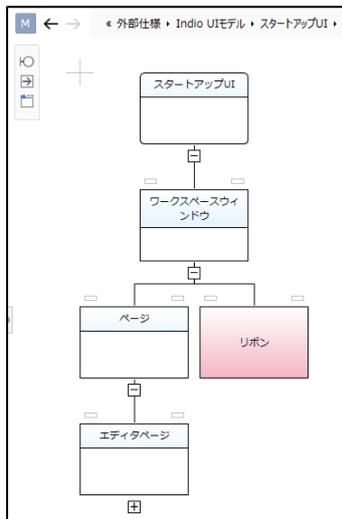


図8 単一工程ビュー

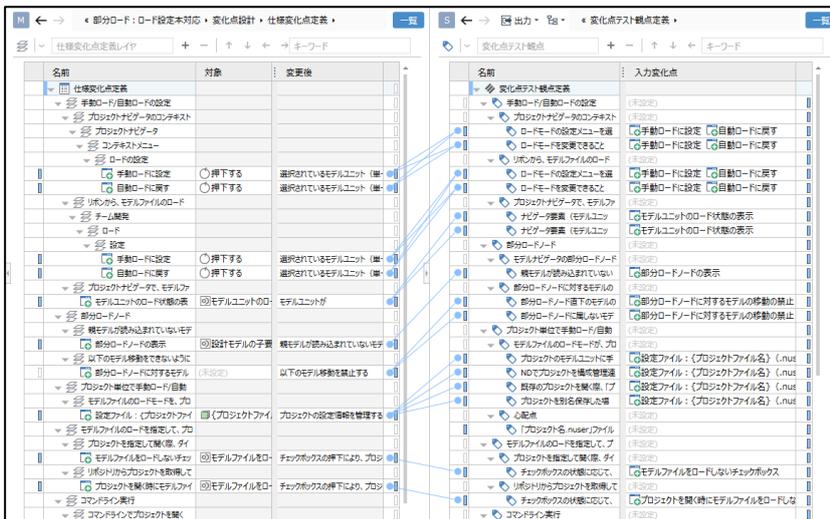


図9 外部テスト仕様導出のための複数工程一覧ビュー

4.3 評価結果

各観点に対して、評価結果を示す。

- A) 実際の開発で提案手法の導入・運用は容易であるか
 提案手法を対象プロジェクトに初回導入時は、メタモデルとビューの定義を一から始める必要があり、まとまったコストを必要とした。しかし、2回目以降の開発では、メタモデルとビュー定義が再利用可能であり、提案手法の導入が容易であった。ツールの支援を得ることで、設計過程では、ドラッグ&ドロップ操作等を覚える必要はあるが、一般的なモデリングツールと大差なく、特別なスキルは不要であり、運用容易であった。
- B) トレース情報の記録コストが十分小さいか
 表1に、提案手法適用前（トレーサビリティ管理に従来方式であるタグ付与方式を採用）と適用後プロジェクトの評価結果を示す。テスト生産性は、時間当たりのテストケース作成項目数である。トレース記録工数割合は、該当工程におけるトレース情報記録に必要な工数割合である。不具合密度は、テストケース当たりのテスト網羅性に関する不具合件数である。
 提案手法を適用することで、設計を進めるだけでトレース情報記録できるようになり、トレース情報記録のための工数は0Hであり十分に小さいといえる。さらに、提案手法を適用することで、適用前と比較してテスト仕様設計の生産性が43%向上し、不具合密度も低減した。

表1 提案手法適用前と適用後の評価結果

| トレーサビリティ記録方式 | テストケース数(項目) | テスト仕様工数(H) | トレース記録工数(H) | 不具合件数(件) | テスト生産性(項目/H) | トレース記録工数割合(%) | 不具合密度(件/項目) |
|--------------|-------------|------------|-------------|----------|--------------|---------------|-------------|
| 従来手法 | 472 | 84.5 | 4.5 | 7 | 5.3 | 5% | 0.015 |
| 提案手法 | 345 | 45.5 | 0.0 | 0 | 7.6 | 0% | 0.000 |

4.4 評価結果の考察

対象プロジェクトの開発プロセスに整合するメタモデルとビューを定義することで、ほとんど工数を使わずに、設計と同時進行でトレース情報を記録できることが確認できた。さらに、設計と同時進行でトレース情報を記録することで、上流工程からの導出関係や網羅性を確認しながら下流工程を設計できることから、設計そのものの生産性向上や品質向上に寄与できるようになると考える。

提案手法適用の準備であるメタモデルとビューの定義は、何をどうやって設計するかによって決定されるものであり、同一製品に対するバリエーション開発など、開発方法が成熟したプロジェクトでは、メタモデルとビューの定義をどのように定義すべきか明確であり、提案手法の導入が比較的容易である。一方で、これまでに開発経験のないドメインの新製品開発など、定義すべき設計情報や開発プロセスが定まっていないプロジェクトでは、適切なメタモデルとビューの定義が困難であると考えられる。また、同じようなシステムを繰り返し開発する場合は、メタモデルとビュー定義が再利用可能であり提案手法導入のためのコストは下がる。しかし、設計の関心事が毎回変わるような開発では、その都度メタモデルとビュー定義をし直す必要があり、提案手法導入のためのコストが大きくなってしまう。

5. おわりに

トレーサビリティ管理の効果を最大限享受するには、設計と同時進行でトレース情報記録することが重要であるが、度重なる設計変更が発生する現場では、トレース情報記録にかかるコストのため、全ての設計を完了した後まとめてトレース情報記録しているのが実情であった。

本研究では、メタモデルによる設計情報定義とマルチビューを活用したトレーサビリティ記録方式を考案し、トレース情報記録にかかるコストを十分に小さくできた。これにより設計と同時進行でトレース情報を記録可能とした。本手法を適用することで、トレーサビリティ管理にかかるコストが大幅に削減され、かつ、開発終盤で設計の抜け漏れが発覚し大きな手戻り工数が発生するリスクを減らす効果が期待できる。

本手法の最も大きな導入障壁は、開発対象ドメイン適用の初回に必要なメタモデルとビューの定義である。メタモデルとビューの定義は、何をどうやって設計するかによって決定されるものであり、開発対象のドメイン毎にある程度標準化することが可能であると考えている。今後の取り組みとして、メタモデルとビューの定義についてドメイングループ毎の標準化を進め、これを活用することで本手法の導入障壁を下げていく。

6. 参考文献

- [1] 宇田川佳久, 「情報システム開発におけるトレーサビリティの事例と今後」, 情報処理, 51.2:150-158, 2010
- [2] IPA/SEC, 「トレーサビリティ確保におけるソフト開発データからの効果検証実施報告書」, 2013
- [3] IPA/SEC, 「製品マニュアルと製品テスト結果のトレーサビリティ確保に係る コスト評価」, 2013
- [4] 堀内一, 大林正晴, 藤川泰之, 「メタモデル標準化の意義と最新動向, 前編: -基本的概念と歴史的経過-, 後編: -MOF, XMI 仕様と応用-」, 情報処理 Vol. 43 No. 11, No. 12, (社) 情報処理学会, 2002
- [5] システム・ソフトウェア設計ツール Next Design: <https://www.nextdesign.app/>