



## Traceability of model

---

*by Sparx Systems Japan*

モデルの追跡（トレーサビリティ）機能ガイド



## 目次

|   |    |
|---|----|
| 1. はじめに.....                                | 3  |
| 2. モデルの追跡.....                              | 3  |
| 3. 要求を表現する方法について.....                       | 4  |
| 3.1. Enterprise Architect 独自の要求要素を利用する..... | 4  |
| 3.2. SysML の要求要素を利用する.....                  | 5  |
| 3.3. USDM を利用する.....                        | 5  |
| 4. 要素間の関係の定義.....                           | 6  |
| 4.1. 図で関係を定義する.....                         | 7  |
| 4.2. 関係マトリクスで関係を定義する.....                   | 8  |
| 4.3. 拡張マトリクスアドイン.....                       | 8  |
| 5. 関係の追跡.....                               | 10 |
| 5.1. 拡張マトリクスアドインの利用.....                    | 10 |
| 5.2. 単一の要素の追跡.....                          | 12 |
| 5.3. 影響範囲を図で確認.....                         | 13 |
| 5.4. 要素間の関係を一括確認.....                       | 15 |
| 6. まとめ.....                                 | 16 |

## 1. はじめに

このドキュメントは、設計モデルにおける追跡(トレーサビリティ)の確保と参照について、Enterprise Architect を利用して効率的に行うために役立つ機能を紹介しています。なお、このドキュメントの内容はUMLを想定していますが、SysML など他の記法でもこのドキュメントで紹介する機能を利用できます。

## 2. モデルの追跡

UMLを利用したソフトウェアの設計開発において重要なことのひとつとして、モデル内の一貫性・整合性を保つことが挙げられます。ソフトウェアには多くの要求がありますが、その要求を過不足なく確実に満たすソフトウェアを作成しなければなりません。この、ソフトウェアの要求を満たすことを確認し続けることはとても重要です。

実際には、要求が設計開発の途中で変更されることが少なからずあります。変更される理由はさまざまですが、要求に変更があった場合に、実装レベルでのクラスの範囲まで影響範囲を容易に確認できれば、要求の変更による影響範囲や対応工数の目処をつけられます。また、要求に対する変更依頼があった場合に、事前に影響範囲を確認できれば、依頼を受けるかどうかの判断材料にできます。

重要なことは、このような変更が発生する状況で、設計モデルの内容を追跡し、影響範囲を把握できるようになっていることです。つまり、トレーサビリティ(追跡可能性)が確保されていることが重要です。

モデルの内容が追跡可能な状況になっていない場合、つまり単なる絵の集合であるような場合には、以下のような問題が発生する可能性があります。

- ・ 要求の変更に対する修正漏れが発生する
- ・ 上位と下位の設計内容が一致していない
- ・ 仕様(UML/SysML などのモデル)の内部に矛盾が残る

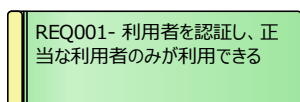
このドキュメントでは、上記のような問題の解決を効率的に行うために、Enterprise Architect を利用する方法を提案します。

### 3. 要求を表現する方法について

要求と設計要素を結びつけるためには、何らかの形で個々の要求を Enterprise Architect 内の要素として定義する必要があります。そこで、Enterprise Architect において要求を表現することのできるいくつかの方法について説明します。

#### 3.1. Enterprise Architect独自の要求要素を利用する

要求要素は、ツールボックスの「拡張」グループ内の「要求」グループにあります。要求要素は、ソフトウェアに対する要求項目になります。例えば、次のような形になります。



この例のように、要求を区別するための ID を付加するのもよいでしょう。この ID 付加は、「自動カウンター」機能を利用すると便利です。

なお、要求を効率的に管理するために、スパークスシステムズジャパンが提供する要求管理ツール RaQuest の併用をお勧めいたします。RaQuest を利用すると、要求項目についてこのドキュメントで説明している追跡の設定など要求の管理を効率的に行うことができます。さらに、要求管理を前提とした RaQuest 独自の機能を利用することで、より厳密な要求の管理を行えます。さらに、要求項目と UML などのモデルの要素を容易に関連づけ、管理できます。

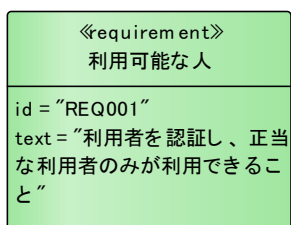
RaQuest についての概要は以下の Web サイトをご覧ください。

<https://www.raquest.jp/>

RaQuest で要求項目を作成すると、Enterprise Architect で上記の要求要素を作成するのと同じように要求要素として保持されます。つまり、Enterprise Architect で要求要素を利用して要求を定義する代わりに、RaQuest を利用して効率的に要求の定義・管理を行えます。また、ID の採番ルールを高度にカスタマイズしたり、独自の属性(プロパティ)の追加・管理などを効率よく行ったりできます。

### 3.2. SysMLの要求要素を利用する

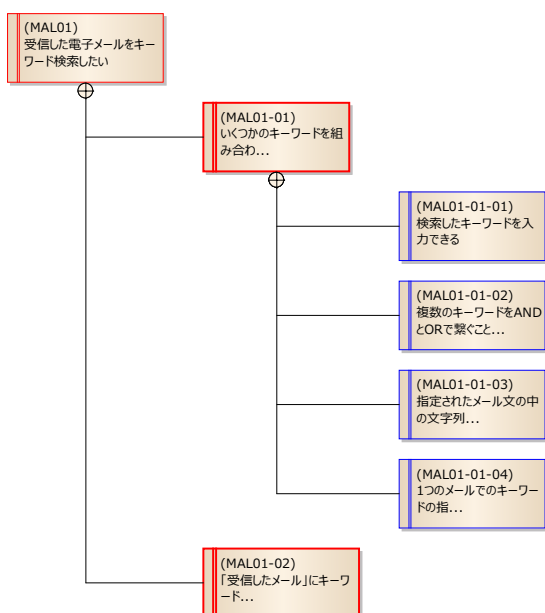
SysML には、対象のシステムの要求を定義・表現するための「要求図」があります。この要求図に要求要素として配置することで、要求を定義できます。



### 3.3. USDMを利用する

要求を記述する方法として、USDM (Universal Specification Describing Manner) があります。USDM は Excel を利用して表現することが多くありますが、Enterprise Architect の「USDM アドイン」を利用して、Enterprise Architect 内で要求・仕様を視覚的に定義できます。定義した内容を Excel 形式で出力したり、Excel で編集した内容を反映させたりできます。

次の図は、Enterprise Architect で表現した場合の一例です。



このような形式で表現した内容についても Enterprise Architect 内部では要素として保持されていますので、設計要素と結びつけられます。

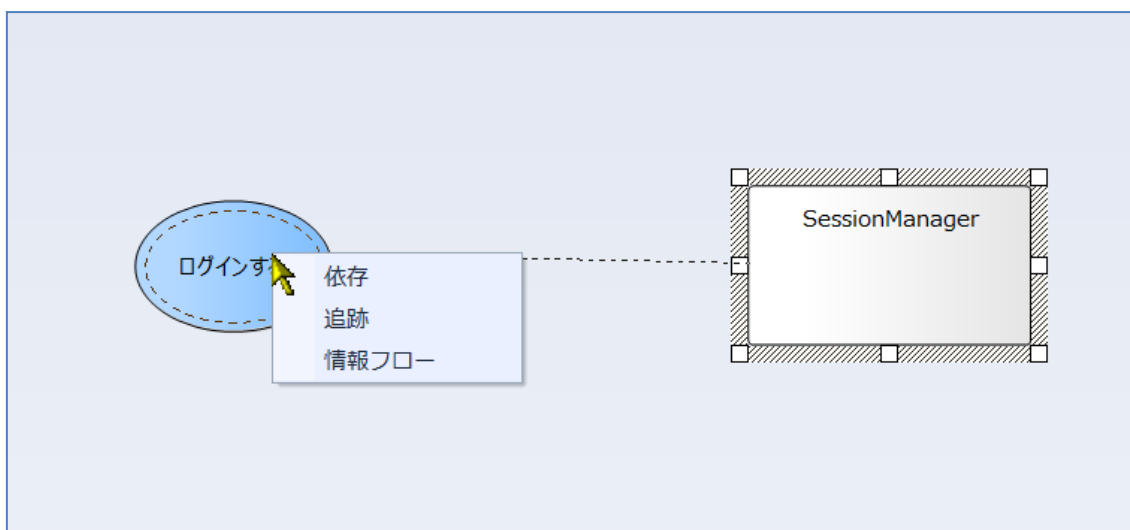
USDM アドインについては、以下のページをご覧ください。

<https://www.sparxsystems.jp/products/EA/tech/USDM.htm>

## 4. 要素間の関係の定義

3 章で説明した方法のいずれかで要求を定義した後は、要求と設計要素間の関係や設計要素同士の間関係を定義していきます。この関係を定義するには、「追跡」の関係を利用することをお勧めします。

「追跡」はステレオタイプ <<trace>> がついた「抽象化」という種類の接続です。「追跡」の関係を利用する一つのメリットは、Enterprise Architect の「クイックリンク」の機能を効率的に活用できる点です。「追跡」の関係は、ほとんどの要素間に対してクイックリンク機能の候補に表示されます。



(クラス要素からユースケース要素に対して追跡関係を定義する際にクイックリンク機能を利用する例)

なお、「追跡」以外の種類の関係を利用することもあります。例えば、要求要素とそれ以外の要素の間については、「実現」の接続を利用するという方法もあります。これは、その接続の名前の通り、「要求を実現する」ということを示しています。RaQuest で要求と設計要素との間に関係を作成した場合には、実現の接続で結ばれます。

その他、SysML の場合で要求要素と他の要素との関係を表現する際には、充足(satisfy)など

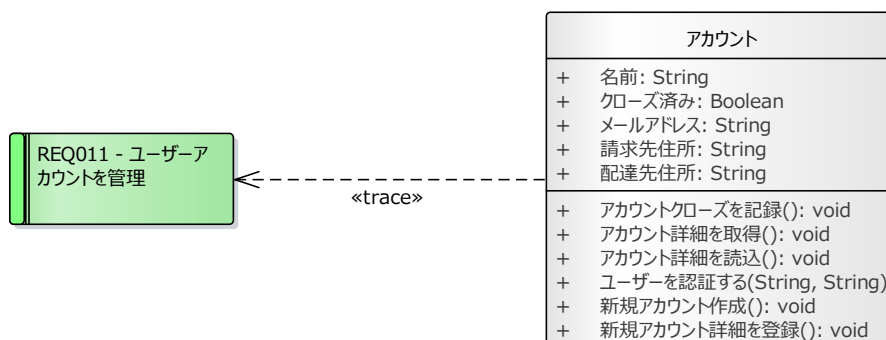
SysML で定義される種類の関係を利用します。

#### 4.1. 図で関係を定義する

このような関係を実際に定義する一例を紹介します。ここでは、要求要素と、その要求を実現する責務のあるクラス要素を結び付ける場合を想定します。

まず、対応を定義するためのダイアグラムを作成します。ダイアグラムの種類は、シーケンス図以外であればどの種類の図でも構いません。作成する位置は、要求要素が含まれるパッケージでもよいですし、対応を定義するための専用のパッケージを作成し、その中にダイアグラムを追加するという方法でもよいでしょう。

次に、作成したダイアグラムを表示し、対象の要求要素とクラス要素をプロジェクトブラウザからドラッグ&ドロップで配置します。その後、関係するクラス要素(下流の要素・実装に近い要素)から要求要素(上流の要素・要求に近い要素)に、「追跡」の関係を設定します。



このようにすることで、この例では要求とクラスの関係、つまり、要求を実現する責務を負うのがそのクラス要素であり、要求を変更すると影響を受ける可能性があるのがそのクラス用であることが示されます。

このようにダイアグラムに要素を配置して結ぶ方法は分かりやすいですが、要素の数が増えてきた場合に効率が下がります。そこで、後ほど説明する関係マトリックスの機能を利用して、マトリックス(表)形式で関係を設定することが一般的です。

## 4.2. 関係マトリックスで関係を定義する

関係を定義する別の方法として、Enterprise Architect が標準で搭載する「関係マトリックス」機能があります。この機能は、縦横に指定したパッケージ・要素間の関係をマトリックス(格子形式)で表示できます。関係マトリックスは、「モデル」リボン内の「トレーサビリティ」パネルにある「関係マトリックス」ボタンを押すと表示できます。

次の例は、マトリックスの表示の例です。

| ターゲット +          | 図書の貸出処理をする | 図書の返却処理をする | 図書を検索する | 図書を追加する | 図書を破棄する | 図書情報を編集する | 貸出内容を印刷する | 利用者情報を検索する | 利用者情報を削除する | 利用者情報を登録する | 利用者情報を変更する |
|------------------|------------|------------|---------|---------|---------|-----------|-----------|------------|------------|------------|------------|
| + ソース            |            |            |         |         |         |           |           |            |            |            |            |
| 010 - 図書の貸出処理をする | ↑          |            |         |         |         |           |           |            |            |            |            |
| 011 - 図書の返却処理をする |            | ↑          |         |         |         |           |           |            |            |            |            |
| 012 - 図書を検索する    |            |            | ↑       |         |         |           |           |            |            |            |            |
| 013 - 貸出内容を印刷する  |            |            |         |         |         |           | ↑         |            |            |            |            |
| 014 - 利用者情報を登録する |            |            |         |         |         |           |           |            |            | ↑          |            |
| 015 - 利用者情報を変更する |            |            |         |         |         |           |           |            |            |            | ↑          |
| 016 - 利用者情報を削除する |            |            |         |         |         |           |           |            | ↑          |            |            |
| 017 - 図書を追加する    |            |            |         | ↑       |         |           |           |            |            |            |            |
| 018 - 図書情報を編集する  |            |            |         |         | ↑       |           |           |            |            |            |            |
| 019 - 図書を破棄する    |            |            |         |         |         | ↑         |           |            |            |            |            |

ただし、この標準の機能を利用するよりは、次の「拡張マトリックス」を利用する方法を推奨します。

## 4.3. 拡張マトリックスアドイン

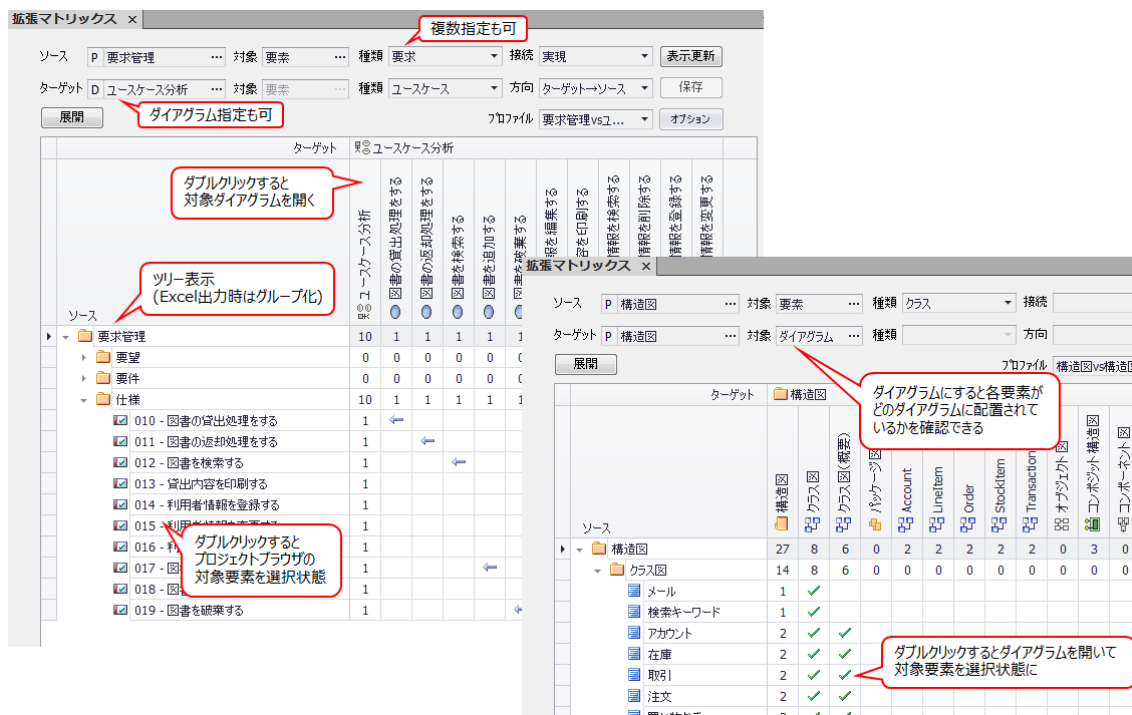
前節の関係マトリックスの機能は、Enterprise Architect の標準機能です。実際に設計の現場でトレーサビリティの定義・確保のためにはこの機能が重要な位置づけになりますが、一方で実際の現場での効率的な設計や分析のためには機能が不足している面もあります。実際に、この標準の機能について、お客様からのさまざまな要望がスパークスシステムズ ジャパンのサポートに届いています。

これらの要望は開発元に随時フィードバックしていますが、すべての要望が必ず反映されるわけではありません。そのため、日本のお客様の実際の現場からのご要望を元に、スパークスシステムズ ジャパンで独自



のアドイン「拡張マトリクスアドイン」を作成しました。このアドインは、Enterprise Architect の標準機能の関係マトリクスの上位互換の機能という位置づけです。

このアドインの外見および主な特徴は次の図をご覧ください。



このアドインは「動作期間限定アドイン」です。サポートが有効な期間中は無料で利用できます。

アドインの詳細とダウンロードは、下記ページをご覧ください。

<https://www.sparxsystems.jp/products/EA/tech/MatrixEX.htm>

標準のマトリクス機能も利用できますが、その標準機能では不足する点や効率が悪い点があり、そうした点がすでにトレーサビリティを実践しているお客様から寄せられ、拡張マトリクスアドインで対応しています。その意味では、標準機能を使うと、いずれ既実践している方と同じ問題に直面する可能性が高いです。そうした点が既に解決されている、拡張マトリクスアドインを利用することを強くお勧めします。

このドキュメントでは、標準の関係マトリクス機能の詳細は触れません。

## 5. 関係の追跡

ここでは、4章で定義した関係を分析・追跡するための方法を紹介します。

### 5.1. 拡張マトリクスアドインの利用

4.3節で紹介した拡張マトリクスアドインは、関係の定義だけでなく、関係の把握や追跡にも利用できます。必要に応じて、表示範囲や表示内容を絞り込むことで、さまざまな状況の確認や編集を効率よく行えます。

また、拡張マトリクスアドインで作成した内容は、CSV の他、Excel・RTF・画像などさまざまな形式で出力できます。

さらに、拡張マトリクスアドインは、要素がどの図で使われているかを知る手段としても活用できます。ターゲット側の対象としてダイアグラムを指定することで、ソース側のそれぞれの要素がどの図で利用されているかを把握できます。

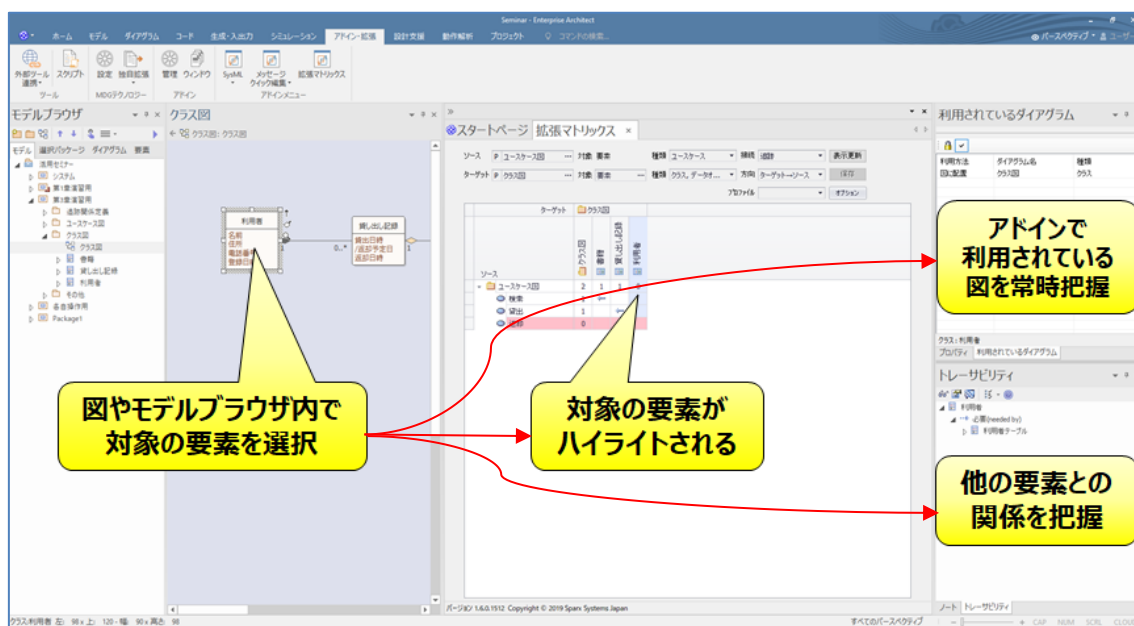
図を対象に指定

図に配置されているかどうか把握できる  
(ダブルクリックで図を開いて要素の位置を表示)

| ソース         | UML 構造図 | オブジェクト図 | 抽象クラスモデル | Controller | Conseq | GenApply | Integer |
|-------------|---------|---------|----------|------------|--------|----------|---------|
| コンポーネント構造   | 0       | 0       | 0        | 0          | 0      | 0        | 0       |
| オブジェクト図     | 0       | 0       | 0        | 0          | 0      | 0        | 0       |
| コンポーネント     | 0       | 0       | 0        | 0          | 0      | 0        | 0       |
| クラス         | 18      | 0       | 12       | 6          | 0      | 0        | 0       |
| アカウント       | 3       |         |          | ✓          |        |          |         |
| 在庫          | 4       |         |          | ✓          |        |          |         |
| 取引          | 1       |         |          | ✓          |        |          |         |
| 注文          | 4       |         |          | ✓          |        |          |         |
| 買い物かご       | 1       |         |          | ✓          |        |          |         |
| 品目          | 5       |         |          | ✓          |        |          |         |
| パッケージ図      | 8       | 0       | 0        | 0          | 2      | 2        | 3       |
| Controller  | 4       | 0       | 0        | 0          | 2      | 2        | 0       |
| Conseq      | 2       | 0       | 0        | 0          | 0      | 2        | 0       |
| ConnSeg     | 1       |         |          |            | ✓      |          |         |
| NetAbstract | 1       |         |          |            |        |          |         |
| Loader      | 1       |         |          | ✓          |        |          |         |
| Time        | 1       |         |          | ✓          |        |          |         |
| GenApply    | 3       | 0       | 0        | 0          | 0      | 3        | 0       |
| Loader      | 1       |         |          |            |        |          |         |
| Shape       | 1       |         |          |            |        | ✓        |         |
| Time        | 1       |         |          |            |        | ✓        |         |
| Integer     | 1       | 0       | 0        | 0          | 0      | 0        | 1       |
| Integer     | 1       |         |          |            |        |          | ✓       |
| リンクデータ      | 0       | 0       | 0        | 0          | 0      | 0        | 0       |

それぞれのセルをダブルクリックすると、そのダイアグラムが開き、対象の要素にカーソルがあたりますので利用状況の詳細を把握できます。なお、要素そのものの利用だけでなく、分類子や型としての利用も表示されます。例えば、SysML のブロック要素がアクティビティ図のレーンの型として利用されている場合も検出・移動できます。

さらに、「利用されているダイアグラム」アドインも組み合わせることで、以下のように便利なトレーサビリティ確認環境を構築できます。モデルブラウザやダイアグラム内で要素を選択すると、その要素の行や列が拡張マトリクス内でハイライトされます。それと同時に、利用されているダイアグラムアドインは選択されている要素についての利用状況を表示し、トレーサビリティサブウィンドウでは他の要素とのつながりを表示します。



このように、アドインを連携させることで、標準機能では実現できない便利な環境を構築できます。

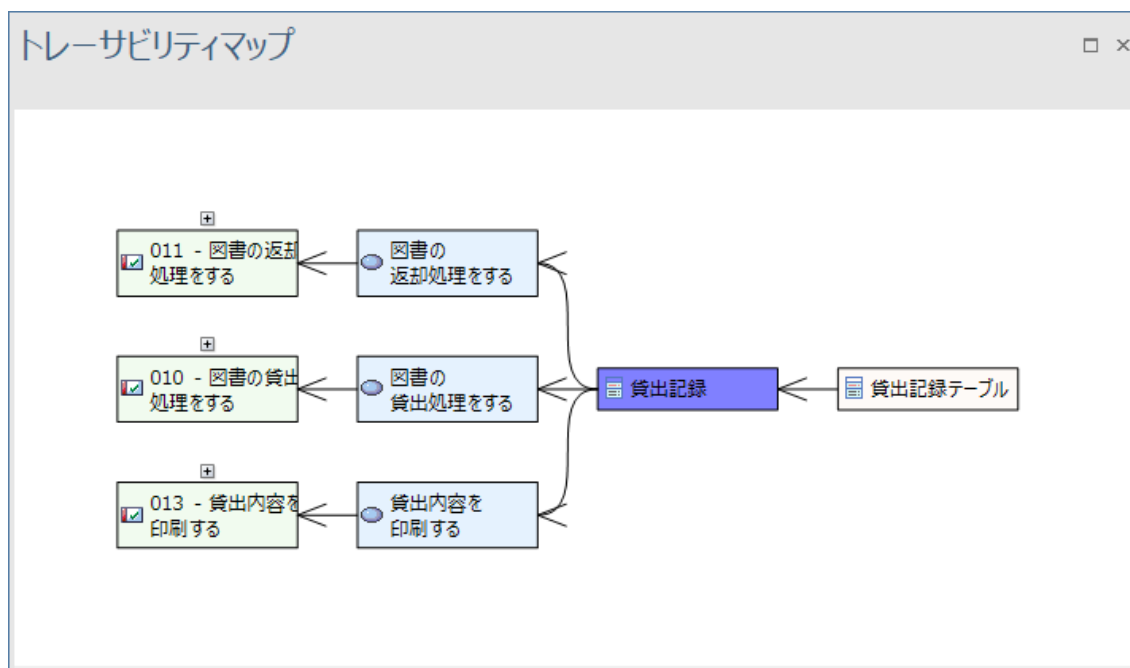
## 5.2. 単一の要素の追跡

次に、特定の要素を選択して、その要素に関する項目を確認する方法を紹介します。具体的な例としては、要求項目を選択して、その項目に関するクラス要素を確認したり、逆にクラス要素を選択して、そのクラス要素に関する要求項目を確認したりする、などがあります。

ここで利用できる機能は、Enterprise Architect の標準機能である「トレーサビリティサブウィンドウ」と、スパークスシステムズ ジャパンが提供する「トレーサビリティマップアドイン」があります。拡張マトリックスアドインの場合と同様に、標準のトレーサビリティサブウィンドウでもある程度利用は可能ですが、実際の現場での利用で不足していたり効率が良くなかったりする点があり、それを解消するのがトレーサビリティマップアドインです。拡張マトリックスと同様に、標準のトレーサビリティサブウィンドウではなくトレーサビリティマップアドインを利用することを推奨します。

トレーサビリティマップアドインでは、要素の前後関係が図として可視化されますので、関係の把握が容易です。

<https://www.sparxsystems.jp/products/EA/tech/TraceabilityMap.htm>



トレーサビリティマップで関係があり要素が表示されるということは、対象の要素は他の要素のいずれかが変更になる場合には影響を受けるということになります。また、もしこの要素からつながる下流の設計・実装に問題があった場合には、このマップに表示されている要求にも影響がある可能性があることもわかります。

トレーサビリティマップアドインで表示されている内容は、画像として保存したり印刷したりできます。右クリックメニューから機能呼び出せます。

なお、トレーサビリティマップアドインも「動作期間限定アドイン」です。サポートが有効な期間中は無料で利用できます。

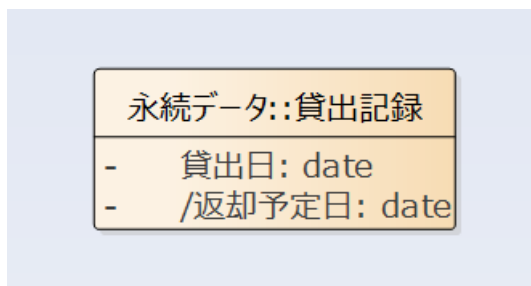
### 5.3. 影響範囲を図で確認

単一の要素の影響範囲については、トレーサビリティマップアドインが便利です。単一の要素の関係だけでなく、その内容からさらに、関係する要素を自由に追加して関係図として可視化する場合には「関係する要素の配置」機能の活用が便利です。

このような場合に、影響範囲図を作成します。なお、UML や Enterprise Architect のダイアグラム (図)の種類として、影響範囲図という名前の図があるわけではありません。「カスタム図」を利用することが多くありますが、ダイアグラムの種類は何でも構いません。

(ただし、シーケンス図は利用できません。)

そのダイアグラムに、対象の要素をモデルブラウザからドロップし、配置します。ここでは、下記のようにクラス要素を対象にします。



次に、この要素を右クリックし、「関係する要素の配置」を実行します。すると、関係する要素の配置画面が表示されます。

ここでは、次の3点の指定が必要です。

- ・ 探索範囲

この項目は、何階層まで関係をたどるか、を指定します。追跡の関係が階層構造になっている場合には、その階層を表示できるだけの数値を指定します。不明な場合には、とりあえず既定値の1のまま実行してみることをお勧めします。

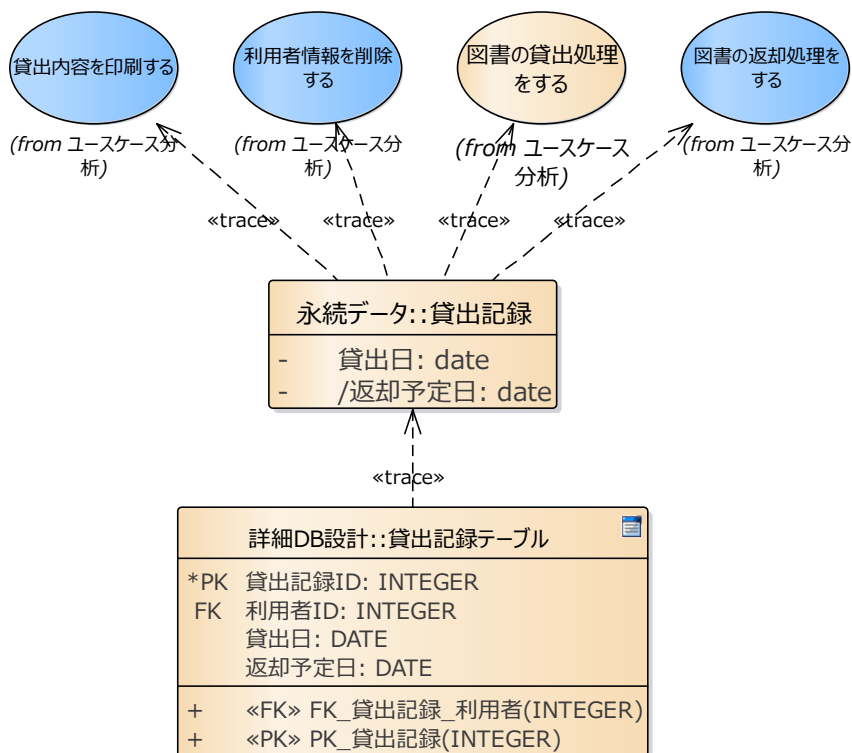
- ・ 接続の種類

追跡の接続のみをたどりたい場合など、接続の種類を指定する必要がある場合には、対象外の接続のチェックを外します。何も変更しない場合には全ての種類が対象になります。

- ・ 接続の方向

このドキュメントの方法の場合、上流(要求要素側)から下流に向かって関係をたどりたい場合には「対象の要素が元側」を、下流(実装クラス側)から上流に向かって関係をたどりたい場合には「対象の要素が先側」を指定してください。両方を対象にする場合には「<全て>」を選択します。

その他にも設定項目がありますので、必要に応じて設定します。実行すると、右クリックした要素を、開いているダイアグラムに追加配置します。



このように、特定の要素に影響する要素を、図として表示できます。このようにして自動配置された要素を右クリックして「関係する要素の配置」を実行することで、興味がある方向に、関係する要素をさらに追加配置できます。この内容を印刷したり、Word 等に貼り付けて資料として活用したりできます。

この影響範囲図は、作成時の内容が表示され、要素間の関係が変わった場合に自動的に更新されるわけではありません。そのため、一時的な利用方法を想定しています。上記のように印刷や画像としての利用が終われば図のみを削除し、また必要になれば新規に図を作成して対象の要素を配置し、「関係する要素の追加」機能を再度実行します。

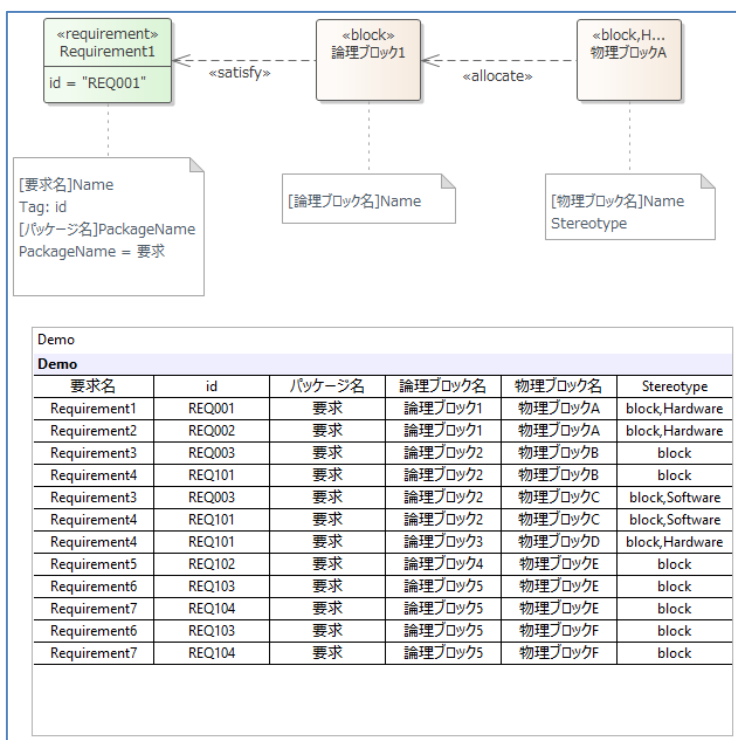
#### 5.4. 要素間の関係を一括確認

ここまで紹介した機能で、マトリックス形式を利用して2つのグループ間の関係の把握および、マップ形式を利用して特定の1要素に関する影響の把握が可能になります。しかし、実際にモデルを利用して設計を行うと、より複雑な関係について、一括で確認したい状況が発生します。

この状況への対応機能は Enterprise Architect の標準機能には搭載していません。スパークスシステムズ ジャパンでは「トレーサビリティ表出カードイン」を提供しています。

<https://www.sparxsystems.jp/products/EA/tech/TraceabilityTable.htm>

このアドインは、指定した条件を満たすモデル内の要素および接続について、Excel あるいは Enterprise Architect の表要素として出力します。次の例をご覧ください。



この例では、上部にある要求要素 - ブロック要素 - ブロック要素 という 3 つの要素間の関係をモデル全体から抽出し、表として表示します。抽出する際には、名前などいくつかの条件を指定して絞り込むことができます。また、表に表示する内容についてもカスタマイズすることができます。Enterprise Architect の検索機能とは異なり、抽出条件自体もモデルとして定義する点も特徴です。

このアドインにより、モデルを構築した後に、指定した条件を満たす関係および要素を抽出し、漏れ抜け・重複・矛盾などがなくどうかを効率的に把握できます。

## 6. まとめ

このようにして、モデル内の要素間の関係を定義してトレーサビリティを確保することで、変更発生時にモデル内での変更漏れを防ぎ、一貫性・整合性のある設計開発を行えます。



この方法で最も重要な点は、もれなく「追跡」の関係を構築することにあります。この関係の構築に漏れがある場合には追跡できませんので、それぞれのフェーズ(要求分析・ユースケース分析等)において確実に関係を設定してください。