



Enterprise Architect 日本語版

階層化した設計を実現するための機能ガイド



## 目次

1	概要 .....	3
2	「階層化した設計」とは？ .....	3
2.1	「階層」の種類 .....	3
2.2	Enterprise Architect での保持方法 .....	5
3	要素に子ダイアグラムを作成する方法 .....	5
4	パッケージを利用する方法 .....	6
4.1	ダブルクリックで移動するダイアグラムの指定 .....	8
4.2	要素間の関係の定義と参照 .....	9
5	個別の状況に関する機能 .....	10
5.1	他のダイアグラムへのリンク .....	10
5.2	SysML ブロックの階層化 .....	11
5.3	アクティビティ図の階層化 .....	12
5.4	パッケージ階層構造の複製 .....	13
5.5	一部の内容を編集不可能にしたい場合 .....	13
5.6	プロジェクトの扱いについて .....	14

## 1 概要

Enterprise Architectは、ツールを利用する際の記法や方法論は限定していません。そのため、ツールに起因する利用時の制約が少ない反面、利用者がどのようにツールを使うのかを考える必要があります。

ツールという観点で見た場合に、一例として以下のような項目についての検討が必要です。

- ✓ どのような方法論を採用し、どの範囲にEnterprise Architectを利用するのか
- ✓ どの記法を利用するか
- ✓ どの図を利用するのか  
それぞれの図に表現する内容はどのようなものにするのか
- ✓ 作成した内容は、どのようにして保持するのか  
(モデルブラウザ内のパッケージ階層・構造をどのようにするのか)

このドキュメントは、上記の項目の最後の「作成した内容は、どのようにして保持するのか」の項目の検討において、参考になるEnterprise Architectの機能を紹介します。

なお、ダイアグラムやパッケージの作成方法など、基本的な操作方法是このドキュメントでは触れません。「Enterprise Architect 入門セミナー」の受講や、ドキュメント「ゼロからはじめるEnterprise Architect」が役に立ちます。

Enterprise Architect 入門セミナー

<https://www.sparxsystems.jp/seminar/EASIntroduction.htm>

Enterprise Architect ドキュメントライブラリ

[https://www.sparxsystems.jp/products/EA/ea\\_documents.htm](https://www.sparxsystems.jp/products/EA/ea_documents.htm)

## 2 「階層化した設計」とは？

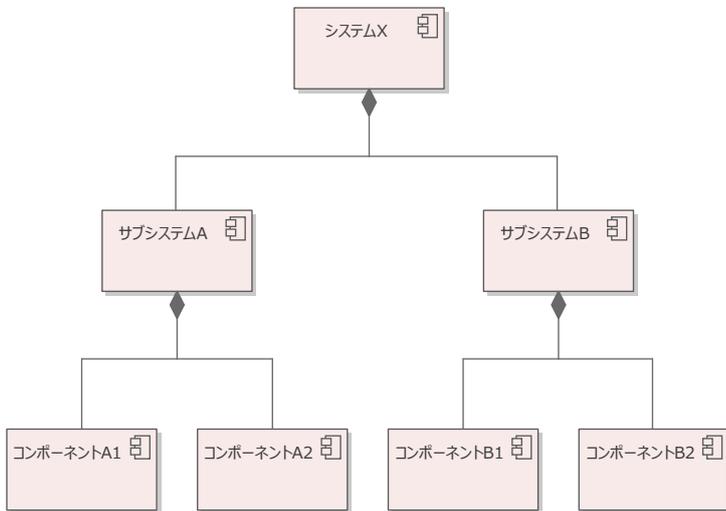
機能の説明の前に、このドキュメントのタイトルとなっている「階層化した設計」について、このドキュメントで指す内容を定義したいと思います。

### 2.1 「階層」の種類

システムやソフトウェアの設計を行う場合には、下記のいずれか(ないしは両方)の「階層」が存在することが多くあります。

## 1. ある構成要素が、別の構成要素の組み合わせとなる場合 (パターン 1)

一例として、「システム」がいくつかの「サブシステム」で構成され、それぞれの「サブシステム」が、いくつかの「コンポーネント」で構成されるような階層構造がある場合です。このような階層は、「詳細化」と呼ばれる場合もあります。



このような構造がある場合には、この例の場合には以下のような関係があります。

- ✓ システムは複数のサブシステムから構成される
- ✓ サブシステムは複数のコンポーネントから構成される
- ✓ システムに対する要求はいずれかのサブシステムに割り当てられ、それぞれのサブシステムでより詳細に定義される。その結果導出された要求は、さらにいずれかのコンポーネントに割り当てられる
- ✓ システムの振る舞いをアクティビティ図で表現する場合は、それぞれのアクションはサブシステムのアクティビティ図でより詳細な粒度で定義される

このような階層構造を、以降の説明では「パターン 1」と呼びます。

## 2. 同じ対象について、観点や抽象度などが異なる場合 (パターン 2)

もう 1 つのパターンとして、以下のような「階層」があります。

(このドキュメントでは、下記の言葉のそれぞれの定義や位置づけなどは対象外とします。)

- ✓ 「要求」と「設計」
- ✓ 「方式設計」と「詳細設計」
- ✓ 「外部設計」と「内部設計」

- ✓ 「基本設計」と「詳細設計」
- ✓ 「論理設計」と「物理設計」

これらの関係は、異なる設計工程間の関係とも言えます。これらの関係では、パターン 1 のように、ある構成要素が別の階層の複数の構成要素から成り立つ、というような関係になるとは限りません。一般的には、異なる設計工程の要素間には、N 対 N の関係があります。

いずれの場合でも、目的・観点・抽象度・粒度などが異なる複数の工程から設計が構成される場合に、Enterprise Architect でどのように関係を扱うのか、という点がポイントになります。

このような階層構造を、以降の説明では「パターン 2」と呼びます。

## 2.2 Enterprise Architectでの保持方法

前述の 2 つのパターンは、設計という観点では全く異なる概念ですが、ツールとしての扱い方を考える場合には、同じ方法・機能が活用できる可能性があります。

具体的には、次のいずれかの方法になります。

- ✓ ある要素の子ダイアグラム・子要素として、その要素の詳細情報を保持する。
- ✓ 設計上の構造・構成をパッケージで表現し、パッケージごとに情報を保持する。  
そして、パッケージ内の要素間に関係を定義する。

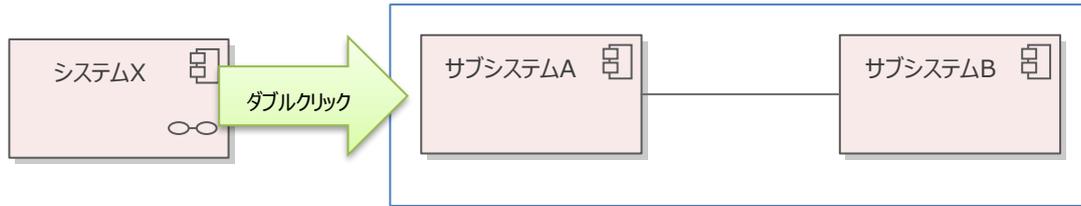
次の章で、この 2 つの方法に関する機能・手順を説明します。

## 3 要素に子ダイアグラムを作成する方法

1 つめの方法では、要素に子ダイアグラムを保持し、その子ダイアグラムでその要素の詳細情報を表現します。主に、前述の「パターン 1」で利用可能な方法です。

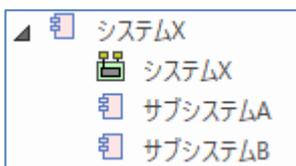
Enterprise Architect での具体的な操作では、ダイアグラム内の要素を右クリックして「子ダイアグラムの作成」→「子ダイアグラムの追加」を実行します。要素の右下に「」マークが表示され、ダブルクリックすると作成したダイアグラムに移動できます。

(要素の種類および有効になっているアドインによっては、ダブルクリックでプロパティ画面の表示など異なる動作になり、移動できない場合もあります。)



作成される子ダイアグラムの種類は、親となる要素の種類によって自動的に決まります。ダイアグラムの種類を変更する場合には、子ダイアグラムを開いた状態で、「モデル」リボン内の「ダイアグラム」パネルにある「操作」ボタンを押し、「種類を変更」を実行してください。

モデルブラウザでは、下の画像のように、ある要素の子要素・子ダイアグラムとして下位階層に配置されます。



この方法のメリット・デメリットは以下の通りです。

#### メリット

- ✓ ある要素と、その要素を構成する子要素・子ダイアグラムがまとまってわかりやすい
- ✓ 異なるパッケージに移動する場合に、親要素(上の画像の例ではシステム X)をドラッグすれば子要素も含めた全体を移動できるなど、まとめて管理することが容易

#### デメリット

- ✓ 階層が深くなると、孫要素など下位階層の参照が容易ではない
- ✓ 1つのパッケージに全ての内容が含まれることになるので、Enterprise Architect の機能との相性が悪い場合もある

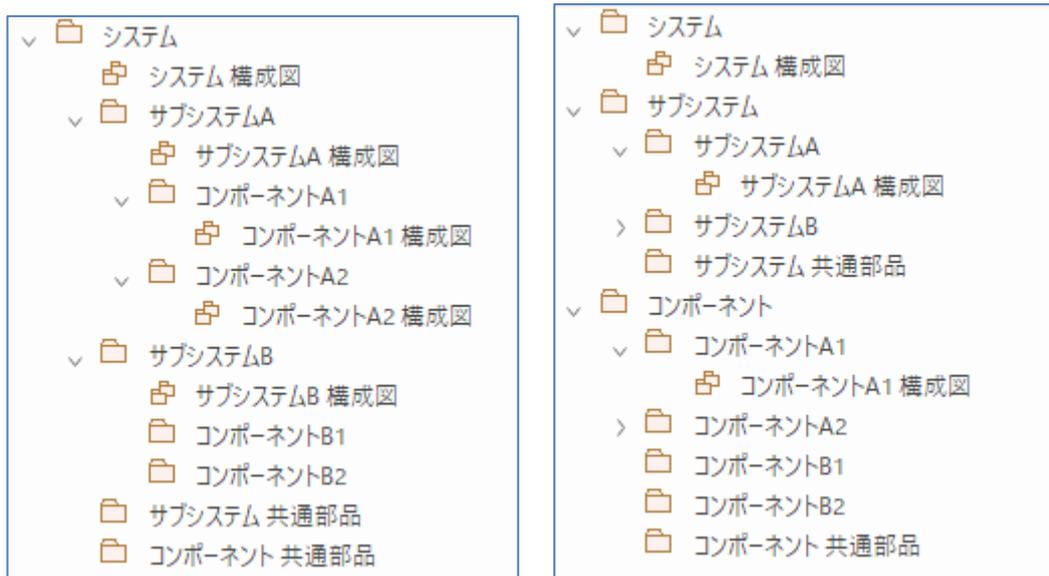
例:

- バージョン管理機能はパッケージ単位となるので、例えば上位担当者が編集中には、下位階層を他の人が編集できない
- XML形式の出力では常に全階層が出力対象となるので、対象となるデータのサイズが大きくなってしまいます。また、下位階層のみが必要でも全部の情報の入出力が必要となる

## 4 パッケージを利用する方法

2 つめの方法として、階層ごとにパッケージを分割する方法があります。この方法は、パターン 1 とパターン 2 の両方の場

合に利用可能です。パターン 1 の場合のモデルブラウザの定義例は次の通りです。

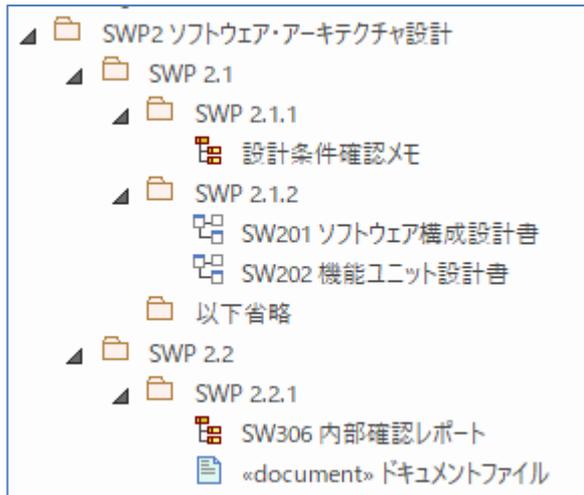


(上記の内容を作成時に、パッケージとダイアグラムの順序が上の画像と異なる場合で同一の内容にしたい場合には、ユーザーオプションの「全般」グループにある「バージョン 15.2 までの順序で表示」にチェックを入れてください。)

左側の例は、要素間の階層構造をパッケージの階層構造に反映した、つまり、「システム」パッケージ内に「サブシステム」パッケージを配置した例です。右側の例は、最上位は「サブシステム」「コンポーネント」などの種別ごとのパッケージを配置し、その中にそれぞれのパッケージを配置した例です。

これらの 2 つの構成は、ツールという観点で見ればどちらも同じで、利用できる機能や使い勝手の違いはありません。利用する立場から見て、パッケージが実際に階層構造を反映している方がわかりやすいか、あるいは分類ごとに分けられている方がわかりやすいか、という違いになります。ただし、例えばバージョン管理機能を利用する場合に、左側の例は入れ子のパッケージの管理となり、右側に比べると少し管理が複雑になるなど、利用する機能によって優劣が付く場合もあります。(上記の左右の例は、パッケージの追加やドラッグ&ドロップでの移動で構成を相互に変更できますので、状況によって構成を変えることも可能です。)

次の例は、パターン 2 の例として IPA(情報処理推進機構)が発行した ESPR 2.0 の一部分をパッケージの階層構造として示しました。このように、設計の工程をパッケージとして表現することで、設計プロセスとパッケージの関係が把握しやすくなります。



このような、パッケージ構造で階層を表現する場合には、次の 2 つの問題があります。

- ✓ 関係するダイアグラムがある場合でも、子ダイアグラムを作成する場合のように要素のダブルクリックでは移動できない
- ✓ 異なるパッケージ内の要素の関係をどう表現するのか  
(例えば、あるシステムがどのサブシステムを持っているのかが、3 章のモデルブラウザのように明確ではない)

この 2 点について、解決策となり得る機能を紹介します。

#### 4.1 ダブルクリックで移動するダイアグラムの指定

第 3 章にて、要素に対して子ダイアグラムを作成する方法を紹介しました。この方法の場合には、対象の要素に、新規に子ダイアグラムを作成します。作成した子ダイアグラムへは、ダブルクリックで移動できます。

一方で、子ダイアグラムとして保持しないダイアグラムに対しても、要素のダブルクリックで移動することができます。これにより、子ダイアグラムとしてダイアグラムを保持することなく、別のパッケージにダイアグラムを配置できます。

具体的には、次の手順になります。

1. 要素をダブルクリックした際の移動先となるダイアグラムを作成する  
(作成するパッケージには制限はありません。対象の要素が含まれるパッケージと別のパッケージでも構いません。)
2. 対象の要素をダイアグラム内で右クリックし、「子ダイアグラムの追加」→「子ダイアグラムの選択」を実行する
3. 表示される画面で、移動先となるダイアグラムを選択する

この設定を行うと、要素をダブルクリックした際に移動するダイアグラムを自由に指定することができます。複数のパッケージに分けて設計情報を格納する場合に、パッケージを超えて移動することが可能になります。

なお、いくつかの種類要素は子ダイアグラムを持つことができません。ダイアグラム内で要素を右クリックした場合に表示さ

れるメニューで「子ダイアグラムの追加」が表示されない要素は、この説明の方法は利用できません。

## 4.2 要素間の関係の定義と参照

ある要素が、(ダイアグラムではなく)別の要素と関係がある場合には、子ダイアグラムの機能は利用できません。例えば、要求要素に対して、その要求を実現する(複数の)設計要素があるような場合です。

このような場合には、Enterprise Architect のトレーサビリティの機能を利用します。次の例のように、関係する要素間にマトリクス形式で関係を定義することができます。

拡張マトリクス

ソース P ユースケース分析 ... 対象 要素 種類 ユースケース 接続 追跡 表示更新

ターゲット P クラス ... 対象 要素 ... 種類 クラス 方向 ターゲット→ソース 保存

プロファイル

ソース	ターゲット					
	クラス	書籍DB	利用者DB	書籍	貸出記録	利用者
ユースケース分析	12	2	3	1	3	3
図書の貸出処理をする	2				←	←
図書の返却処理をする	2				←	←
図書を検索する	1	←				
図書を追加する	0					
図書を破棄する	1	←				
図書情報を編集する	1			←		
貸出内容を印刷する	1				←	
利用者情報を検索する	1		←			
利用者情報を削除する	1		←			
利用者情報を登録する	1		←			
利用者情報を変更する	1					←

バージョン 1.6.0.1512 Copyright © 2019 Sparx Systems Japan

(この画面は「拡張マトリクスアドイン」の画面です。サポートが有効であれば無料で利用できます。)

このようなトレーサビリティについての詳細は、ドキュメント「モデルの追跡(トレーサビリティ) 機能ガイド」をご覧ください。上記のように関係づけた要素の情報を参照したり、関係する要素が配置されている図に移動したりできます。

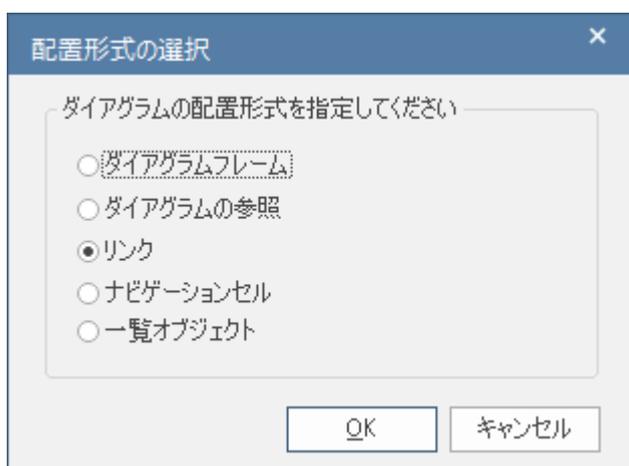
## 5 個別の状況に関する機能

いくつか具体的な状況・記法・図の場合に、役に立つ機能や方法を紹介します。

### 5.1 他のダイアグラムへのリンク

あるダイアグラムが別のダイアグラムと関係がある場合に、ダイアグラム間にリンクを作成し、容易に移動できます。

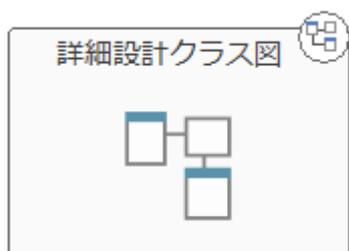
リンクを作成するには、移動元(リンクの配置先)となるダイアグラムを開き、モデルブラウザから移動先となるダイアグラムをドラッグし、移動元のダイアグラムにドロップしてください。以下の「配置形式の選択」画面が表示されますので、「リンク」あるいは「ナビゲーションセル」を選択してください。



「リンク」は、以下のようなシンプルな形式です。ダブルクリックで、リンク先の図を開きます。文字列の既定値は、「パッケージ名 : ダイアグラム名」となっていますが、変更可能です。

 詳細設計パッケージ : 詳細設計クラス図

「ナビゲーションセル」は、以下のようにダイアグラム名とアイコンを表示できます。ダブルクリックで、リンク先の図を開きます。ダイアグラム名として表示される情報は「別名」に格納されますので、表示内容をダイアグラム名以外にもできます。

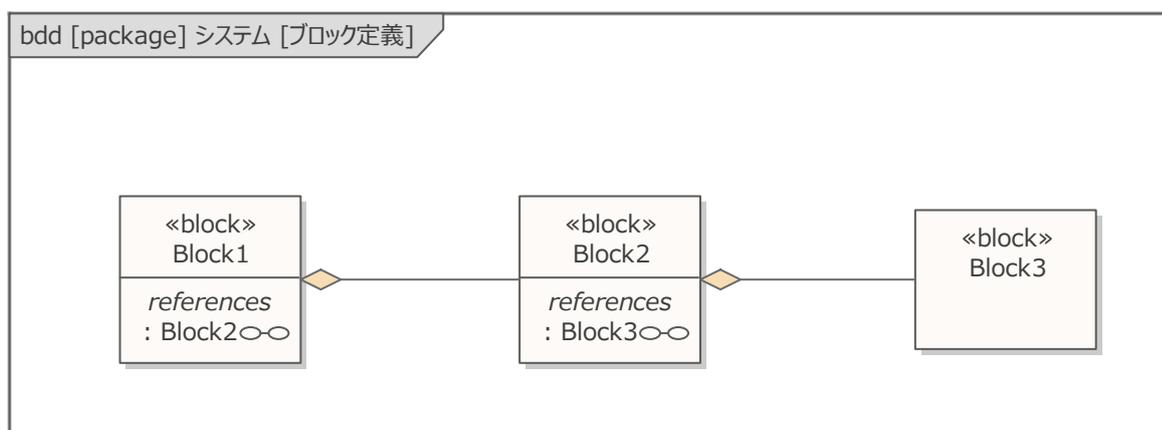


いずれの場合も、移動先の図からは BackSpace キー、あるいは、タブに表示される「←」アイコンで元の図に戻れます。

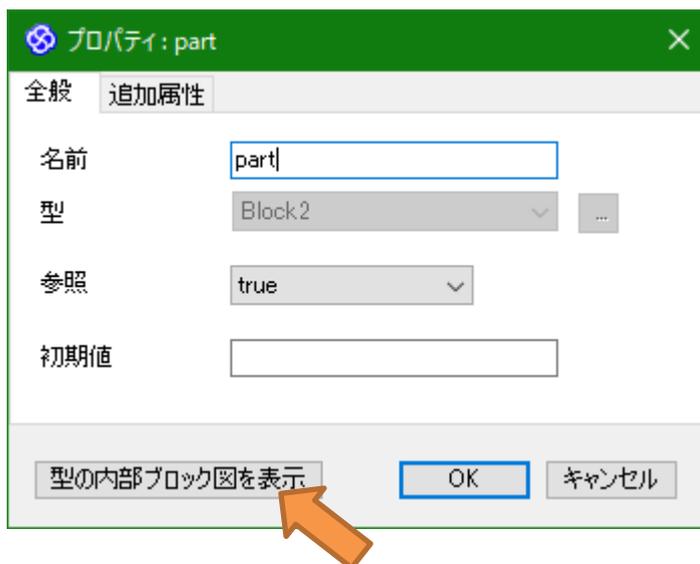
☞ 詳細設計クラス図 ←

## 5.2 SysMLブロックの階層化

SysML のブロック定義図では、以下のようにブロックが階層化するような関係になる場合があります。



このような場合で Block1 および Block2 がそれぞれ内部ブロック図を持つ場合には、Block2 を型とするプロパティ要素のプロパティ画面にある「型の内部ブロック図を表示」ボタンで、Block2 の内部ブロック図に移動できます。

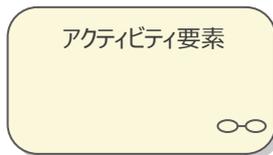


### 5.3 アクティビティ図の階層化

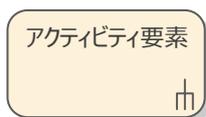
アクティビティ図で、あるアクションが別のアクティビティ(図)の呼び出しになる場合には、「振る舞い呼び出しアクション」として定義することが、仕様上は適切と言えます。

具体的には、以下のような操作になります。

1. 呼び出し先となるアクティビティ要素を作成し、その要素に対して子ダイアグラムを作成します。



2. モデルブラウザから、そのアクティビティ要素をアクティビティ図にドロップすると、振る舞い呼び出しアクション要素として配置されます。  
(要素の配置画面が表示される場合には、「配置形式」として「呼び出し(アクション)」を選択してください。)



3. この振る舞い呼び出しアクション要素をダブルクリックすると、アクティビティ要素が持つ子ダイアグラムを開けます。

なお、上記の手順では、「下位階層の振る舞いを示すアクティビティ要素を作成」→「上位階層のアクティビティ図に振る舞い呼び出しアクションとして配置」となりますが、多くの場合には「上位階層のアクティビティ図を作成」→「下位階層の振る舞いとして詳細化」となるはずです。つまり、設計の流れと操作の手順が逆転してしまいます。

このような場合には、「モデル調整アドイン」あるいは「アクティビティサポートアドイン」の「別の図として抽出」機能が有用です。上位階層のアクティビティ図内の(通常の)アクション要素を右クリックし、別の図として抽出を呼び出してください。以下の作業をまとめて行います。

- 同名の、下位階層の振る舞いを保持するアクティビティ要素を作成
- 上位階層のアクション要素を振る舞い呼び出しアクション要素に変換
- 振る舞い呼び出しアクション要素の振る舞いとして、作成したアクティビティ要素を設定
- 上位階層のアクション要素にアクションピンがある場合には、アクティビティパラメータに変換

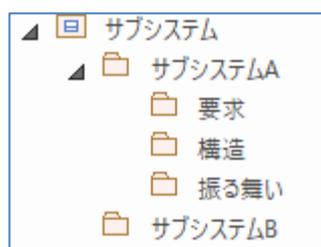
これにより、結果として上記の手順と同じ動作を簡単に実行できます。

モデル調整アドインの概要とダウンロードは以下 URL をご覧ください。アクティビティサポートアドインは、バージョン 17.0 以降では Enterprise Architect のインストール時に自動的にインストールされます。

<https://www.sparxsystems.jp/products/EA/tech/Addins.htm#Refactoring>

## 5.4 パッケージ階層構造の複製

以下のようなパッケージ構成を例にします。この例では、「サブシステム」パッケージ内に個別のサブシステムについての設計を格納することを想定しています。また、個別のサブシステムのパッケージ内には、「要求」「構造」「振る舞い」の3つのパッケージを作成し、そのサブシステムについての要求・構造・振る舞いを格納するような構成とします。



このような場合に、新たに「サブシステム B」を追加することを考えます。その場合に、「要求」「構造」「振る舞い」の3つのパッケージを手作業で作成するのは手間がかかります。こうした場合には「モデルテンプレート」の機能で利用できるような独自のテンプレートを作成する方法もありますが、お手軽にパッケージ構造を複製する方法もあります。

元となるパッケージ(今回の例では「サブシステム A」パッケージ)を右クリックし、「コピー・貼り付け」→「クリップボードにコピー」→「複製情報をクリップボードにコピー」を実行します。

(あるいは、ショートカットキー Ctrl+SHIFT+C も利用できます)

その後、作成先(親パッケージ)となるパッケージ(今回の例では「サブシステム」)を右クリックし、「コピー・貼り付け」→「クリップボードからパッケージを貼り付け」を実行します。

(あるいは、ショートカットキー Ctrl+V も利用できます)

この方法を利用すると、パッケージの構成(およびパッケージの中身)をすべて複製できます。名前などもそのまま複製されますので、パッケージ名などを編集します。

これにより、同じような構成・内容がある場合に、簡単に複製できます。

## 5.5 一部の内容を編集不可能にしたい場合

このドキュメントで説明したような階層化した設計を行う場合に、例えば「システム」に関する内容はシステムの設計を担

当する部門(あるいは担当者)、「サブシステム」に関する内容は、それぞれのサブシステムの設計を担当する部門のみが編集できるようにしたい、という場合があります。プロジェクトを完全に分けてしまうことで対応することもできますが、その場合には設計内容間のトレーサビリティの定義・参照ができません。

(トレーサビリティについては、以下の URL や関連するドキュメントをご覧ください。

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

プロジェクトとしては同一で、編集可能かどうかを制御するために利用できる機能としては、以下のような機能があります。それぞれの機能については、関連するドキュメントやヘルプをご覧ください。

- バージョン管理機能を利用する。バージョン管理が有効になると、各パッケージは「チェックアウト」の操作をしない限り編集不可能となり、また編集した場合でも履歴から元に戻すことができる。
- セキュリティ(アクセス権)機能を有効にし、それぞれのパッケージに編集可能なグループを割り当て、他のグループのメンバーを編集できないようにする。
- セキュリティ(アクセス権)機能を有効にし、排他ロックをしないと編集できないモードにする。排他ロックをすれば編集可能になるが、意図せず編集してしまう問題を防ぐことができる。

## 5.6 プロジェクトの扱いについて

このドキュメントで説明した内容は、主にモデルブラウザ内のパッケージやダイアグラムを構成する場合に役に立つ考え方について触れています。一方で、そのモデルブラウザの内容も含めて、プロジェクト全体としてどのように扱うのか、についてはいくつかの選択肢があります。1 つのプロジェクトを設計者全員で共有し、そのプロジェクトにこのドキュメントで説明した構成を構築するという選択もあれば、場合によってはこのドキュメントで説明した構成の一部のみを切り出して複数のプロジェクトで管理したい状況もあるかもしれません。

このような、プロジェクトをどのように扱うのか、については別のドキュメント「複数人数による設計での利用法 マニュアル」で説明しています。合わせてご覧ください。