



## Version Control Feature Guide

---

*by SparxSystems Japan*

Enterprise Architect 日本語版

バージョン管理機能 機能ガイド

(2023/12/01 最終更新)



## 目次

1	はじめに.....	3
2	バージョン管理の対象について .....	3
3	使い方の流れ .....	4
3.1	バージョン管理ツールのインストールと設定.....	4
3.2	バージョン管理ツールのリポジトリの作成 .....	5
3.3	バージョン管理機能の構成.....	5
3.4	バージョン管理機能の有効化 .....	6
3.5	対象パッケージの指定と設定 .....	8
3.6	チェックアウトとコミット .....	10
3.7	複数人での利用の場合の流れ .....	11
4	役に立つ機能・利用方法 .....	14
4.1	パッケージの共有.....	15
4.2	入れ子のパッケージの管理.....	15
4.3	過去のバージョンの参照.....	16
4.4	過去のバージョンとのマージ .....	17
4.5	過去のバージョン間の差分比較.....	17
4.6	チェックアウトしているユーザーを知る .....	18
4.7	強制的にチェックアウトを解除する .....	19
5	差分比較・マージツール <b>LemonTree</b> .....	20
6	最後に .....	21

## 1 はじめに

ここでは、Enterprise Architect で利用可能な機能のうちの一つ、バージョン管理機能との統合機能について、具体的に説明していきます。このドキュメントでは、バージョン管理ツールとして Subversion を利用しています。他のバージョン管理ツールでは設定方法が異なりますが、Enterprise Architect からのバージョン管理機能の使い方は変わりません。

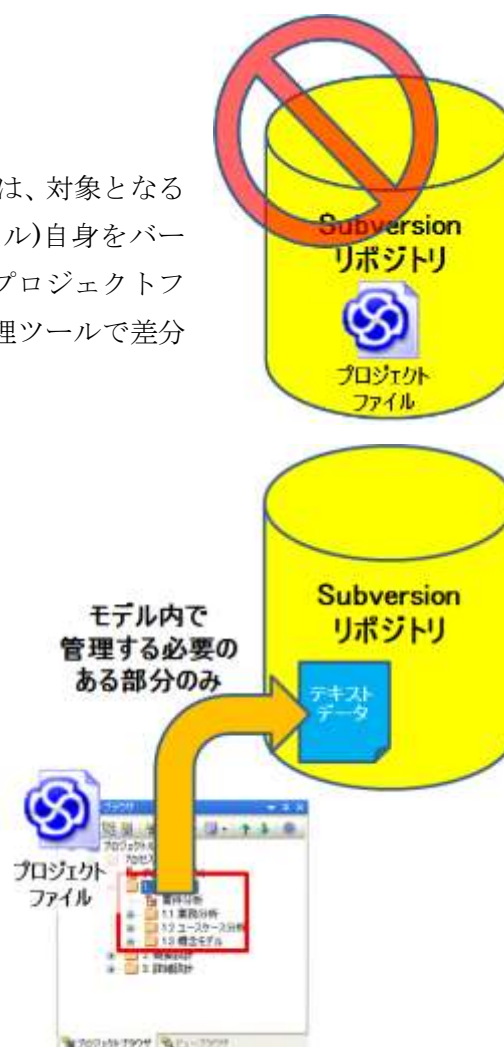
なお、Enterprise Architect のモデルのバージョン管理のためにバージョン管理ツールを選択できる状況にある場合には、Subversion のバージョン 1.6 以降を利用することをお勧めします。

## 2 バージョン管理の対象について

バージョン管理機能を利用する際には、一般的には、対象となるプロジェクトファイル(EAPX/FEAP/QEA ファイル)自身をバージョン管理ツールに登録するものではありません。プロジェクトファイルはバイナリファイルのため、バージョン管理ツールで差分のみを管理することができません。

Enterprise Architect のバージョン管理機能を利用する場合には、プロジェクトファイル内のパッケージが管理対象になり、そのプロジェクトファイルそのものは管理の対象にはなりません。パッケージの内容を XML ファイルとして出力し、そのファイルを管理対象にします。

また、上記のように、プロジェクトファイル自身を管理するわけではありませんので、プロジェクトファイル以外の形式、つまり MySQL や SQLServer などの DBMS リポジトリを利用する場合や Pro クラウドサーバを利用する場合でも、バージョン管理機能は利用できます。



なお、Enterprise Architect が搭載するバージョン管理機能を利用せずにバージョン管理をおこなう方法もあります。差分比較・マージツール「LemonTree」を利用すると、Git や Subversion にプロジェクトファイルそのものを格納し、ファイル全体を1つとしてバージョン管理する方式を実現することができます。この方法の場合、TortoiseSVN や TortoiseGit と組み合わせることで、コミット時に競合が発生していた場合には、自動的に LemonTree が起動して競合を解決し、解決できない内容については差分を視覚的に確認し解決方法を指定することができます。

次の表は、ここまでの内容をまとめたものです。

	Enterprise Architect のバージョン管理機能	LemonTree
連携するツール	基本的には Subversion	Subversion あるいは Git
管理対象	パッケージの内容を出力した XML ファイル	プロジェクトファイル
長所	Enterprise Architect の標準機能	TortoiseSVN や TortoiseGit と連携し、ソースファイルなどと同じように並行で編集・コミット・マージが可能
短所	図の差分の確認が限定的、特にシーケンス図など振る舞い図の比較は不可能	別途購入が必要

LemonTree については第 5 章もご覧ください。

### 3 使い方の流れ

#### 3.1 バージョン管理ツールのインストールと設定

まず、バージョン管理ツールをインストールする必要があります。このドキュメントの説明で利用するのは、Subversion です。なお、Enterprise Architect が現在対応しているバージョン管理ツールは、Subversion の他には、Microsoft が提唱している SCC API に対応しているツール(Visual SourceSafe など)・Team Foundation Server です。

Subversion のインストール方法および設定方法についてはここでは触れませんので、これ以降はインストールが終了しているという前提で話を進めます。このドキュメントで利用している svn.exe のバージョンは 1.6.16 です。Subversion のクライアントである svn.exe を含む、最新の TortoiseSVN をインストールすると設定が容易です。32 ビット版・64 ビ

ット版のどちらも利用可能です。

(TortoiseSVN のインストール時のオプションで、コマンドラインクライアントをインストールするように選択してください。)

Team Foundation Server を利用する場合には、ヘルプファイルの「排他チェックアウトのための設定」のページをご覧ください。排他的にチェックアウトを行うように設定する必要があります。

## 3.2 バージョン管理ツールのリポジトリの作成

Enterprise Architect 専用のリポジトリを作成するには、次の手順で作業を行います。

まず、リポジトリの元となるディレクトリ(ここでは **sample** という名前のディレクトリとします)を作成します。その後、この空のディレクトリを読み込み、リポジトリを新規に作成します。TortoiseSVN を利用すれば、簡単にリポジトリを作成できます。その後、作成したリポジトリを別の場所にチェックアウトして作業場所を作成してください。これらの操作は Enterprise Architect に固有のものではなく、TortoiseSVN の通常の操作となります。不明点があれば、TortoiseSVN に関するドキュメントをご覧ください。

これで、準備は完了です。ここで作成した **work** ディレクトリ(c:\¥svn¥work とします)を、Enterprise Architect の設定で利用します。

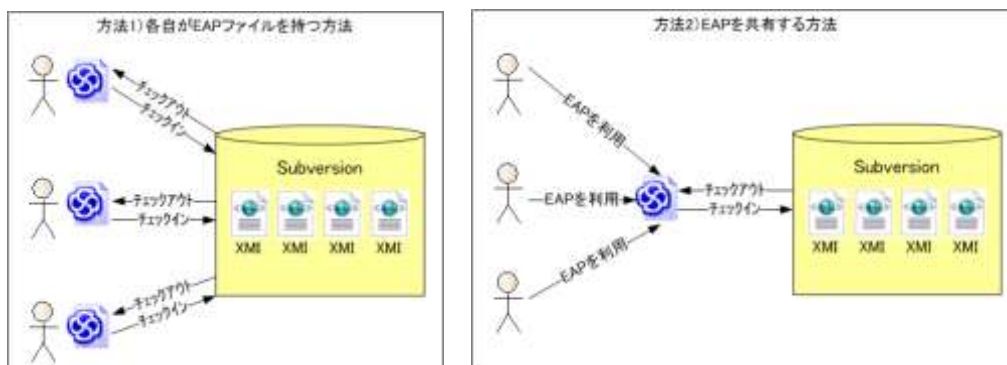
## 3.3 バージョン管理機能の構成

バージョン管理機能を利用する場合は、モデルブラウザ内のパッケージの単位でバージョン管理することができます。

(要素やダイアグラムなどの単位ではバージョン管理できません。)

パッケージをバージョン管理すると、そのパッケージに含まれるモデルの情報が XMI ファイルとして自動的にエクスポートされ、その XMI ファイルがバージョン管理ツールのリポジトリに保存されます。(XMI ファイルはテキストベースになっているため、バージョン管理ツール側で効率的な差分管理が可能です。)

また、バージョン管理機能を利用する場合、各自がプロジェクトファイルを持つ方法と、プロジェクトファイルや DBMS リポジトリを共有してバージョン管理機能を利用する方法の 2 通りがあります。



方法 1) 各自がプロジェクトファイルを持つ方法：

この方法の場合、格差業者の PC でプロジェクトファイルを開いて作業することと、各自が必要な情報のみバージョン管理から取得して作業しますので、モデルへのアクセス速度が速く快適に編集できるというメリットがあります。ただ、他の担当者が編集・コミットした最新の情報は、最新情報を取得する機能を実行しないと知ることができません。

方法 2) プロジェクトファイルや DBMS リポジトリを共有する方法：

この方法の場合、プロジェクトをネットワーク上で共有していること、またすべてのモデルが格納されている場合はモデルのサイズも大きいことから、モデルへのアクセスに時間がかかる可能性があります。ただ、他の担当者が編集したモデルの情報は、同じプロジェクト内にありますので、ダイアグラムを開いたり、要素のプロパティ画面を表示したりするタイミングで、最新の情報を知ることができます。

バージョン管理機能を利用する際には、上記のどちらの方法を取るかをまず決める必要があります。

どちらの構成の場合も、以下で説明している操作方法は同じです。

### 3.4 バージョン管理機能の有効化

バージョン管理機能を利用する場合は、最初にバージョン管理機能を有効にします。対象のプロジェクトを開き、「プロジェクト」リボン内の「バージョン管理」パネルにある「基本設定」を実行します。すると、次のような画面が表示されます。

バージョン管理の基本設定

プロジェクトの設定

- このプロジェクトでは、同時に複数の人がプロジェクトを開くことはありません
- 入れ子でバージョン管理されている場合を考慮する(推奨)
- パッケージ間の参照情報を常に保持

設定の詳細

設定識別ID:

種類:  SCC  CVS  Subversion  TFS

作業用ディレクトリ:  ...

ローカルマシンの設定

SVN.Exeのパス:  ...

タイムアウトの値:  秒

定義済みの設定

設定識別ID	種類	ファイル	位置
--------	----	------	----

バージョン管理の基本設定が未完了の場合でも確認しない

新規(N) 保存(S) 削除(D) 開じる(O) ヘルプ(H)

まず、利用するバージョン管理ツールの種類を選択します。今回の例では「Subversion」を選択します。

次に、「設定識別 ID」の欄に、任意の文字列を入力します。この設定をわかりやすく表現している文字列がよいでしょう。この ID は、このバージョン管理されているプロジェクトが複数のマシンで利用される場合に重要になります。各マシンには、それぞれ「ID とバージョン管理ツールの設定(バージョン管理ツールのユーザーなど、利用者ごとの情報)」という組で設定情報が保存されます。ID と設定を結びつけることにより、Enterprise Architect 側では異なる設定情報を適切に管理・参照することができます。(この ID は、Subversion 自身に関する設定とは関係ありません。)

また、ID を変えることで、複数の設定を利用することができます。たとえば、複数の Subversion リポジトリを利用したり、異なるバージョン管理ツールを併用したりすることができます。

ここで設定した ID およびは、設定を行ったマシンのレジストリに保存され、他のプロジェクトからも参照・利用することができます。

設定識別 ID の設定後、「...」ボタンを押します。すると、作業ディレクトリの指定画面になりますので、Subversion からチェックアウトしているディレクトリの位置を指定します。例えば、「C:\svn\work\sample」とします(C:\svn\work ディレクトリが存在し、その中に Subversion に関する設定ファイルがあることになります)。

その後、利用する svn.exe のパスを指定します。TortoiseSVN を利用している場合には、インストール時の選択によっては svn.exe のファイルが含まれていない場合があります。コマンドクライアントをインストールするように設定し、インストールをしてください。  
(例: C:\Program Files\TortoiseSVN\bin\svn.exe)

すべて設定が終わると「保存」ボタンを押すことができるようになります。「保存」ボタンを押すと、環境確認のため Subversion のコマンドが内部的に実行され、設定が正しく行われていれば、設定内容が画面最下部の一覧に追加されます。

「保存」ボタンを押したときに Subversion のエラーが表示される場合には、Subversion の設定が正しくないか、作業用ディレクトリの位置がおかしいか、などの原因が考えられます。エラーメッセージに応じて修正してください。

なお、Subversion を利用する場合には、「タイムアウトの値」を指定できます。Subversion サーバとの通信が指定時間内に完了しない場合に、操作をキャンセルすることができます。

### 3.5 対象パッケージの指定と設定

次に、実際にバージョン管理を行うためのパッケージを指定します。

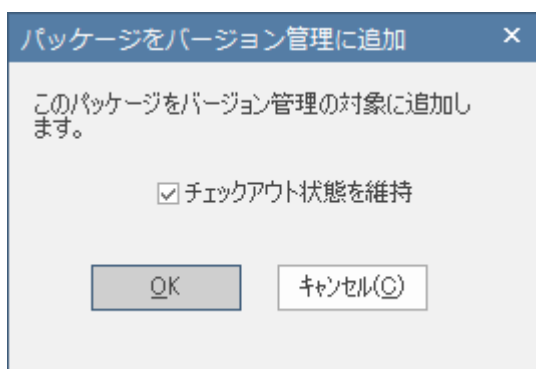
まず、モデルブラウザで対象のパッケージを選択します。そして、「プロジェクト」リボン内の「バージョン管理」パネルにある「パッケージの設定」を実行します。次の画面が表示されます。





まず、「バージョン管理あるいは一括入出力の対象にする」にチェックを入れます。すると、他の設定項目が操作可能になりますので、「バージョン管理」の一覧から先ほど設定したものを選択します。また、「XMI ファイル名」の欄にバージョン管理ツールで管理されるファイルのファイル名が設定されますので、必要に応じて修正します。  
(既定値はパッケージ名です。通常は同名のファイルを複数バージョン管理することはできませんので、同名のパッケージがモデル内に複数ある場合には、修正が必要です。)

この状態で **OK** ボタンを押すと、次のような確認画面が表示されます。今回は、説明の都合上「チェックアウト状態を維持」のチェックボックスを外します。多くの場合には、そのまま「**OK**」を選択し、モデルの編集を行うことになるかと思います。ここでこの画面が表示されない場合には、「最後に」の章にあるヒントもご覧ください。その後、初期バージョンに対するコメントを入力します。



この状態でモデルブラウザを確認すると、次の図のように「sample」パッケージのアイコンが変わります。その下にある「sample2」のパッケージのアイコンと比較すればわかり

やすいと思います。

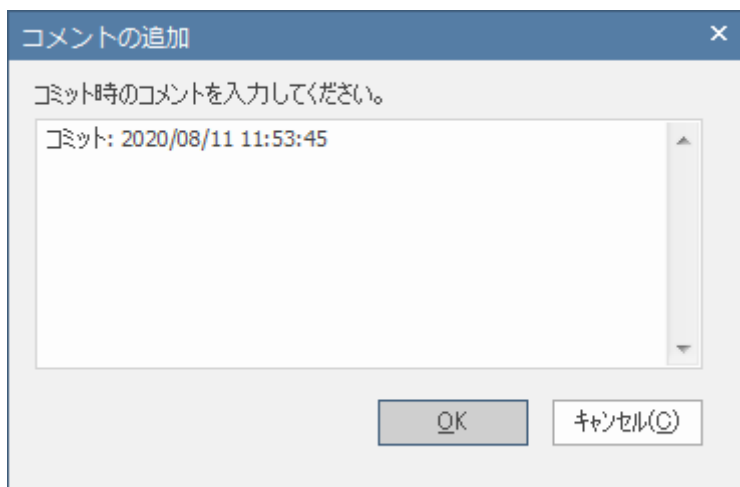


このアイコンは、バージョン管理されていて、チェックアウトしていない(ので編集できない)ことを示しています。バージョン管理されているかどうかは、パッケージの右側に表示される、鎖ないしは数字の 8 のようなアイコンです。鍵マークのアイコンは、チェックアウトしていないことを示しています。

### 3.6 チェックアウトとコミット

次に、編集するためにチェックアウトの作業を行います。パッケージのコンテキストメニューから「パッケージの管理」→「チェックアウト」を選択してください。すると、モデルブラウザのアイコンが変化して、自由に編集できる状態(チェックアウト状態)になります。

作業が終わってコミットする場合にも同様です。パッケージのコンテキストメニューから「パッケージの管理」→「コミット」を選択してください。コメントの入力画面が表示されますので、必要に応じて入力してください。



コミットした後は、再び編集不可能の状態に戻ります。入力したコメントは、パッケージのコンテキストメニューから「パッケージの管理」→「パッケージの履歴」を実行すると、コミットの情報とともに表示されます。また、この画面から、以前のバージョンのモデルに一時的に戻したり、過去のバージョンのモデルを編集して最新版として更新したりすることもできます。(4.2章をご覧ください。)

なお、Enterprise Architect にはブランチを作成して利用する機能はありません。

なお、このパッケージをチェックアウトしているかの情報は、バージョン管理ツールのユーザー名をモデル内にチェックアウトしたユーザーとして保存し、メニュー項目の有効・無効のチェックなどに利用しています。そのため、異なるマシンから利用する場合には、異なるユーザーでアクセスするようにしてください。

(同じユーザーで異なるマシンから利用すると、チェックアウトの状況などが不適切になる場合があります。ご注意ください。)

### 3.7 複数人での利用の場合の流れ

バージョン管理機能を利用して、チーム(複数名)で利用する場合の基本的な流れを紹介しします。この例では Subversion を利用します。

1. 3.5 章までの説明を元に、プロジェクトを作成して管理対象のパッケージをバージョン管理可能な状態にします。「設定識別 ID」は、チーム全員で共通に利用するものですので、プロジェクト名などチーム全体での共通の文字列とします。ここでは、「ProjectASetting」とします。すべてのパッケージはコミットされた状態にしてください。

この例では、以下のような表示・構成になります。

### バージョン管理の基本設定画面:

バージョン管理の基本設定

プロジェクトの設定

このプロジェクトでは、同時に複数の人がプロジェクトを開くことはありません

入れ子でバージョン管理されている場合を考慮する(推奨)

パッケージ間の参照情報を常に保持

設定の詳細:

設定識別ID: ProjectASetting

種類:  SCC  CVS  Subversion  TFS

作業用ディレクトリ: C:\svn\work\sample

ローカルマシンの設定:

SVN.Exeのパス: C:\Program Files\TortoiseSVN\bin\svn.exe

タイムアウトの値: 30 秒

新規 保存 削除

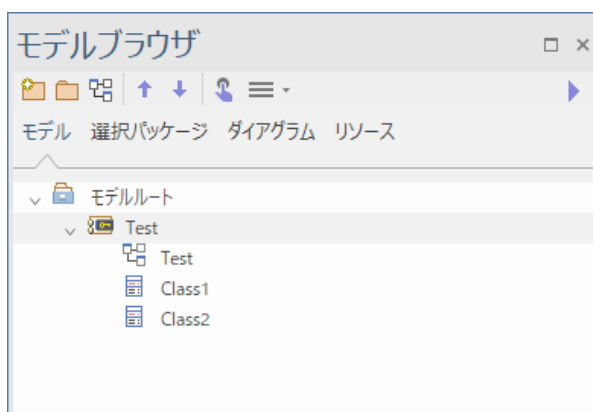
定義済みの設定

設定識別ID	種類	ファイル	位置
ProjectASetting	SVN	1	%ProjectASetting%

バージョン管理の基本設定が未完了の場合でも確認しない

閉じる ヘルプ

### モデルブラウザの構成:



## Test パッケージの設定:

- ここまで設定したプロジェクトファイルを、他のメンバーに配布します。それぞれのメンバーは **Subversion** の設定を行い、リポジトリからチェックアウトし作業用ディレクトリを作成します。
- 他のメンバーがプロジェクトファイルを開くと、その環境では、今回設定した設定識別 ID「ProjectASetting」に関する設定が未定義ですので、以下のメッセージが表示されます。

「はい」ボタンを押して、バージョン管理の基本設定画面を表示させます。

- 「作業用ディレクトリ」および「SVN.exe のパス」を指定し、「保存」ボタンを押します。

バージョン管理の基本設定

プロジェクトの設定

このプロジェクトでは、同時に複数の人がプロジェクトを開くことはありません

入れ子でバージョン管理されている場合を考慮する(推奨)

パッケージ間の参照情報を常に保持

設定の詳細

設定識別ID: ProjectASetting

種類:  SCC  CVS  Subversion  TFS

作業用ディレクトリ: C:\svn\work\_another

ローカルマシンの設定

SVN.Exeのパス: C:\Program Files\TortoiseSVN\bin\svn.exe

タイムアウトの値: 30 秒

新規 保存 削除

定義済みの設定

設定識別ID	種類	ファイル	位置
ProjectASetting	SVN	1	(設定が必要)

バージョン管理の基本設定が未完了の場合でも確認しない

閉じる ヘルプ

保存ボタンを押すと設定内容を確認します。正常に設定が完了できれば、上の画像の「定義済みの設定」一覧内で「(設定が必要)」となっている箇所が、「%ProjectASetting%」に変わります。

- バージョン管理されているパッケージ(今回の例では「Test」パッケージ)をモデルブラウザ内で右クリックし、「パッケージの管理」→「チェックアウト」でパッケージをチェックアウトして編集できるかどうかを確認してください。

## 4 役に立つ機能・利用方法

この章では、バージョン管理機能を利用している際に役に立つ機能・利用方法をいくつか紹介します。

## 4.1 パッケージの共有

バージョン管理機能を利用することで、複数のプロジェクト間でパッケージを共有することができます。

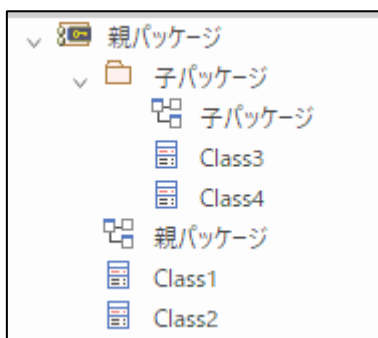
すでにバージョン管理されているパッケージは、「パッケージを指定して追加」コマンドを実行することで、別のプロジェクトで読み込むことができます。これにより、1つのパッケージを異なるプロジェクト間で共有することができます。

例えば、複数のプロジェクトで共有される共通部品などがある場合には、その共通部品が含まれるパッケージをバージョン管理の対象として、バージョン管理ツールに登録します。その共通部品を利用するそれぞれのプロジェクトからは、「パッケージを指定して追加」コマンドでそのパッケージを取得し、パッケージ内の要素をそれぞれのプロジェクトのダイアグラムなどで利用できます。

その共通部品が含まれるパッケージの内容が更新された場合には、「最新バージョンを取得」コマンドを実行することで、作業を行っているプロジェクトで利用されている内容を同期更新できます。この場合には、ダイアグラムに配置している要素の位置や大きさなどはできる限り維持されます。また、他の要素との接続も維持されます。

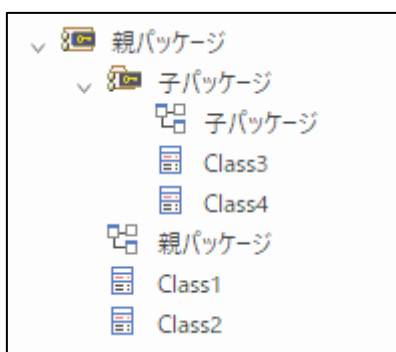
## 4.2 入れ子のパッケージの管理

パッケージ内に別の子パッケージが含まれる場合には、基本的には子パッケージも含めてバージョン管理の対象となります。そのため、下の例では、親パッケージをチェックアウトした人が子パッケージも含めてすべての内容を編集可能になり、他の人が子パッケージの内容のみを変更する、ということはできません。



このような場合に、親パッケージだけでなく子パッケージもバージョン管理対象とするこ

とで、別々にチェックアウトが可能になります。下の例では、子パッケージをチェックアウトした人は子パッケージ内のダイアグラム「子パッケージ」とクラス要素 Class3,Class4 を、親パッケージをチェックアウトした人は親パッケージ直下のダイアグラム「親パッケージ」とクラス要素 Class1,Class2 を編集できます。



### 4.3 過去のバージョンの参照

バージョン管理されているパッケージは、過去のバージョンの情報を参照することができます。対象のパッケージをモデルブラウザ内で右クリックして「パッケージの管理」→「パッケージの履歴」を実行してください。過去の履歴の概要を確認できます。（利用しているバージョン管理ツールによって表示される画面は異なります。）

Subversion を利用している場合には、「取得」ボタンを押すことで、過去のバージョンの内容を Enterprise Architect 内で確認することができます。「チェックアウト」ボタンを押すと、選択したリビジョンのモデルを編集可能な状態で取得しますので、過去のバージョンに戻す(ロールバックする)ことができます。





なお、Enterprise Architect のバージョン管理機能では、バージョンのブランチやタグ付けを行うことはできません。

#### 4.4 過去のバージョンとのマージ

バージョン管理されているパッケージについて、過去のバージョンに含まれる、一部の要素のみを最新バージョンに反映したい場合などには、マージを行うことができます。(コーポレート版以上のエディションの場合には、4.4 章で説明する方法が簡単です。)

この手順にはいくつかの方法がありますが、Enterprise Architect のみで行う場合の手順は次の通りです。

1. 対象のパッケージを選択し、「プロジェクト」リボン内の「バージョン管理」パネルにある「パッケージの設定」を実行します。表示される内容のうち、「XMI ファイル名」の欄に表示されているファイル名(拡張子.xml)を確認します。
2. 1 番のファイルについて、バージョン管理時に設定した「作業ディレクトリ」に含まれる、XMI ファイルをデスクトップなどの別の場所にコピーしてください。
3. マージしたい過去のバージョンについて、4.2 章で紹介した方法で、「過去のバージョンからの編集」を行います。チェックアウト状態になります。
4. 対象のパッケージをモデルブラウザを選択した状態で、「生成・入出力」リボン内の「モデルの入出力」パネルにある「XML 読み込み」ボタンを押し「XML ファイルの読み込み」を選択します。
5. 2 番でコピーしたファイルを指定し、読み込みます。これで、内容が最新バージョンのものになりますが、バージョン管理ツール側での「最新(カレント)」は、マージしたい過去のバージョンになっています。
6. 対象のパッケージを右クリックし「パッケージの管理」→「最新バージョンと比較」を実行します。バージョン管理ツールによっては、ファイルを上書きするかどうか確認の画面が表示されますが、上書きしないようにしてください。
7. 比較ツールが起動し、差分が表示されます。必要な情報について右クリックし、「モデルに追加」や「ベースラインとマージ」を実行してください。なお、「状態」の欄が「Baseline only」になっている項目が、過去のバージョンにのみ存在する情報です。
8. コミットすると、マージした内容が最新バージョンとなります。

#### 4.5 過去のバージョン間の差分比較

バージョン管理されているパッケージについて、過去のバージョン間の差分を比較することもできます。

(この方法は、Enterprise Architect のベースライン機能を利用するため、コーポレート版以上のエディションで利用できます。)

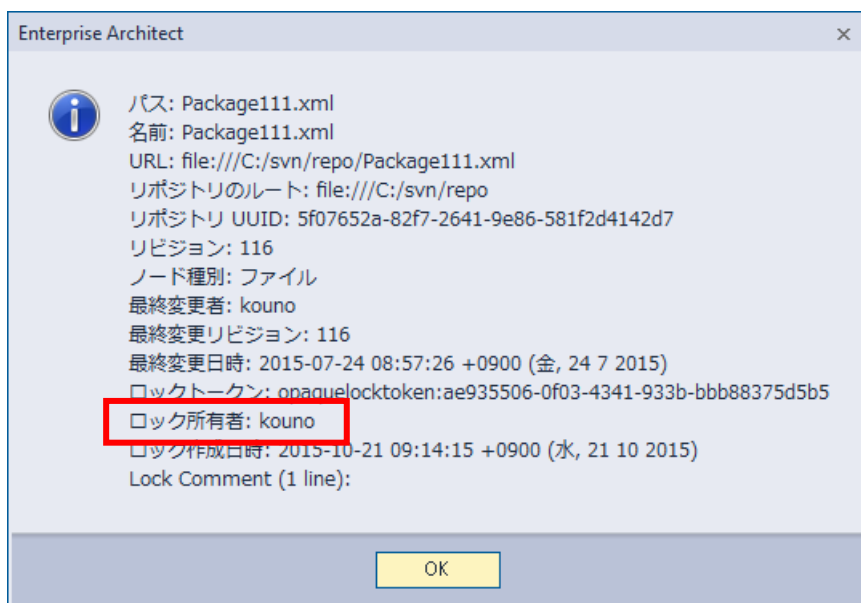
手順は次の通りです。

1. 対象のパッケージを右クリックし「パッケージの管理」→「パッケージの履歴」を実行し、パッケージの履歴画面を表示させます。比較対象となるリビジョンを選択して「取得」ボタンを押して下さい。
2. モデルブラウザで対象のパッケージを選択した状態で、「モデル」リボン内の「パッケージ」パネルにある「操作」ボタンを押してメニューを表示し、「ベースラインの管理」を実行します。「新規ベースライン」ボタンを押して、現状の内容を保存します。
3. 1 番の手順と同じように、もう 1 つの比較対象の履歴を取得します。  
(最新バージョンと比較・マージしたい場合には、「パッケージの管理」→「最新バージョンを取得」を実行します。)
4. 対象のパッケージを再度モデルブラウザで選択し、「モデル」リボン内の「パッケージ」パネルにある「操作」ボタンを押してメニューを表示し、「ベースラインの管理」を実行します。手順の 2 番で作成したベースライン情報を選択した状態で「差分の表示」ボタンを押します。
5. これにより、2 つの履歴の差分が表示されます。「モデル」側の内容が、手順 3 で取得した内容になり、「ベースライン」側の内容が、手順 1 で取得した内容になります。
6. 手順 2 で作成したベースラインを削除して下さい。

#### 4.6 チェックアウトしているユーザーを知る

バージョン管理機能を利用すると、パッケージがチェックアウトされている場合にはそのパッケージをチェックアウトできず、編集ができない状態となります。対象のパッケージを誰が(どのアカウントが)チェックアウトしているかを知るためには、対象のパッケージを右クリックして「パッケージの管理」→「パッケージのバージョン管理情報」を選択してください。

表示される内容は利用しているバージョン管理ツールにより異なります。一例として、Subversion を利用している場合には以下のような内容が表示されます。



上の画像の赤枠部分に、該当パッケージをチェックアウトしているアカウント(バージョン管理ツールのユーザー名)が表示されています。

#### 4.7 強制的にチェックアウトを解除する

誰かがパッケージをチェックアウトしている場合には、コミット操作が行われるまでは他の人はチェックアウトできません。すなわち、該当パッケージを編集することはできません。しかし、チェックアウト中のユーザーが長期不在の場合など、例外的に強制的にチェックアウトを解除する必要が発生するかもしれません。

このような場合の手順は次の通りです。

1. 対象のプロジェクトを開いていない状態でバージョン管理ツール側の機能で、チェックアウトされているパッケージに紐づく XML ファイルのロックを解除します。例えば、Subversion を利用している場合には、TortoiseSVN を利用し、該当ファイルのロックを解除します。  
(この操作はバージョン管理ツールにより異なります。ツールによっては「排他ロック」など別の名称の場合もあります。)
2. Enterprise Architect を起動し、対象のプロジェクトを開きます。「プロジェクト」リボン内の「バージョン管理」パネルにある「状態をすべて同期」ボタンを押します。

この機能は、バージョン管理されているすべてのパッケージについて、バージョン管理されている XML ファイルの状態を取得します。もし、Enterprise Architect

で保持している内容とバージョン管理ツール側での設定状況が異なる場合、バージョン管理ツール側の状況を正しいものとして、**Enterprise Architect** の管理情報を更新します。

3. これにより、**Enterprise Architect** でのチェックアウト状況が解除され、別の人がチェックアウトできるようになります。

なお、内部の状態保持などに影響を与える可能性がありますので、上記のような例外的な場合のみ、この操作を行うようにしてください。また、チェックアウト中のユーザーが編集集中の内容がある場合、すべて破棄されることになる点にもご注意ください。

## 5 差分比較・マージツールLemonTree

このドキュメントで説明する、バージョン管理の機能は Enterprise Architect の標準機能です。この標準機能では、いくつかの問題点があります。

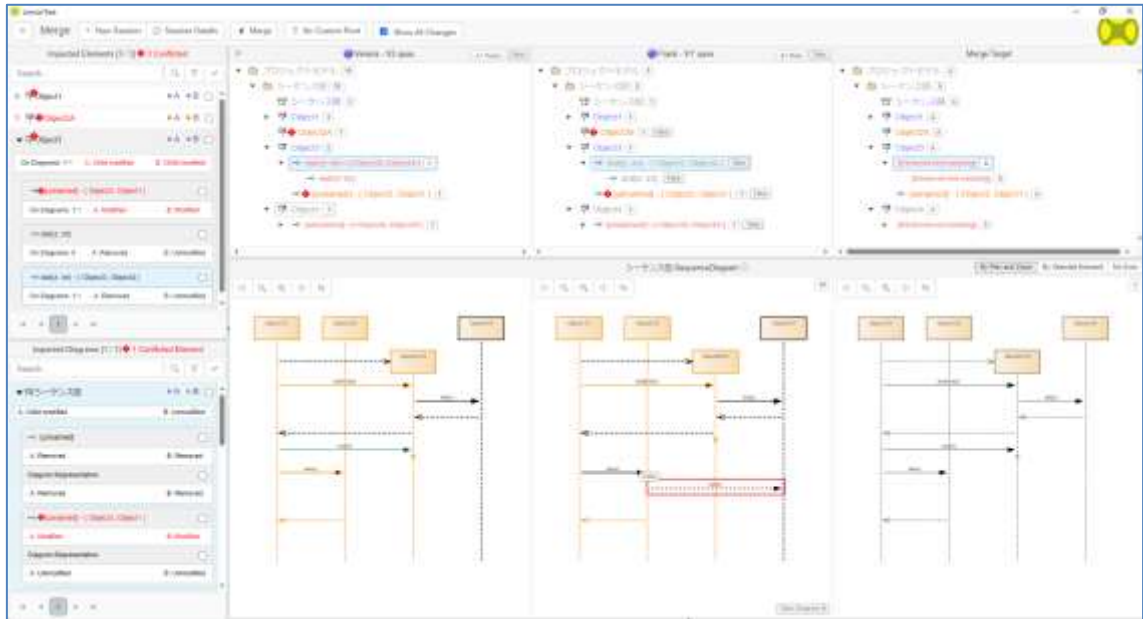
- ✓ パッケージ単位での管理となり、チェックアウト・コミットの操作が煩雑  
(図と、図に配置されている要素が同じパッケージにあるとは限らないため、編集対象の図や要素が含まれるパッケージを意識する必要があります)
- ✓ モデルの編集作業を並行して行うことができない
- ✓ Git に対応していない
- ✓ 指定した 2 つの履歴の差分比較の手順が面倒  
(TortoiseSVN 等を利用する場合には、履歴の一覧で 2 つの履歴を選択し、右クリックメニューから比較を実行すれば良い)

こうした問題に対応するために、サードパーティー製(オーストリア LieberLieber 社製)の高機能な差分比較・マージツール「LemonTree」をスパークスシステムズ ジャパンでも販売しています。

LemonTree では、EAPX/QEAX ファイルを Subversion や Git に格納し、複数人で同時に編集してコミットするような、ソースコードでの利用と同じような利用方法が可能となります。コミット時に衝突(モデルの同時編集)がある場合には LemonTree が起動し、視覚的に衝突内容を確認し、問題を解決できます。もし、同じモデル内の異なる場所を編集していて同じ内容を編集していない場合には、自動的に内容をマージし、その結果をコミットできます。

LemonTree の詳細は下記ページをご覧ください。

<https://www.sparxsystems.jp/LemonTree/>



## 6 最後に

以上で、Enterprise Architect で利用できるバージョン管理ツールとの連携機能の簡単な説明になります。この機能を利用することで、履歴を管理したり、あるいは複数のユーザーで分散して開発したりすることが容易になります。ぜひ実際に試してみて、ご利用ください。