



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

(2018/05/16 最終更新)



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。

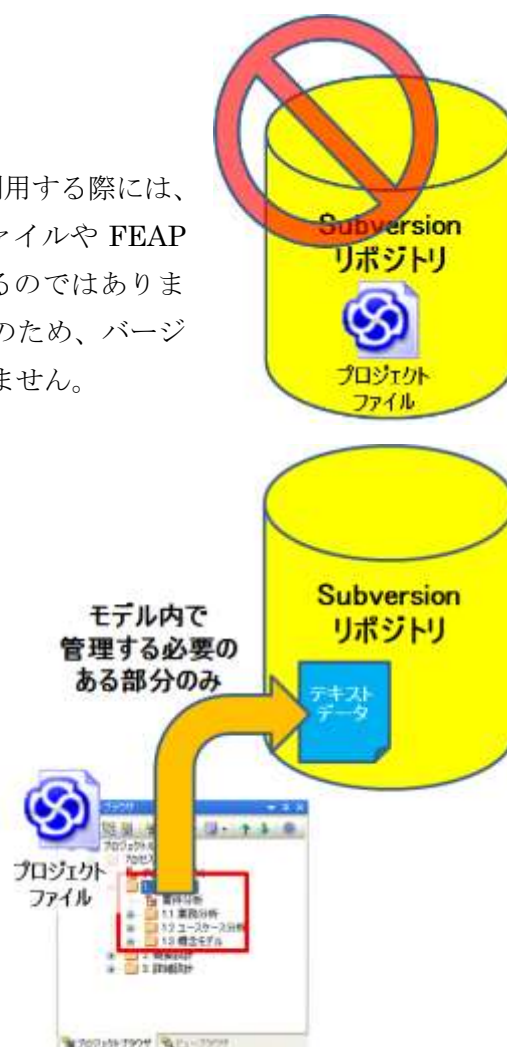
2 バージョン管理の対象について

Enterprise Architect のバージョン管理機能を利用するには、対象となるプロジェクトファイル(EAP/EAPX ファイルや FEAP 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。

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モデル(の一部)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。

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、プロジェクトが存在しない場合、つまり Oracle や SQLServer などの DBMS リポジトリを利用する場合でも、バージョン管理機能は利用できません。

なお、Enterprise Architect の標準の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 EAP 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。

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、Microsoft が提唱している SCC API に対応しているツール (Visual SourceSafe など)・CVS・Subversion・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。

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。

コマンドは、次のようになります。この例では、c:\¥svn というディレクトリを作成し、

その中にリポジトリや作業ディレクトリを作成しています。上記の **sample** ディレクトリは、この例では `c:\svn\sample` です。下記コマンド実行時のカレントディレクトリは `c:\svn` です。

```
svnadmin create c:\svn\repo (リポジトリの新規作成)
svn import sample file:///c:/svn/repo -m "initial" (リポジトリへの登録)
```

その後、一度チェックアウトして作業場所を作成してください。

```
svn checkout file:///c:/svn/repo work
```

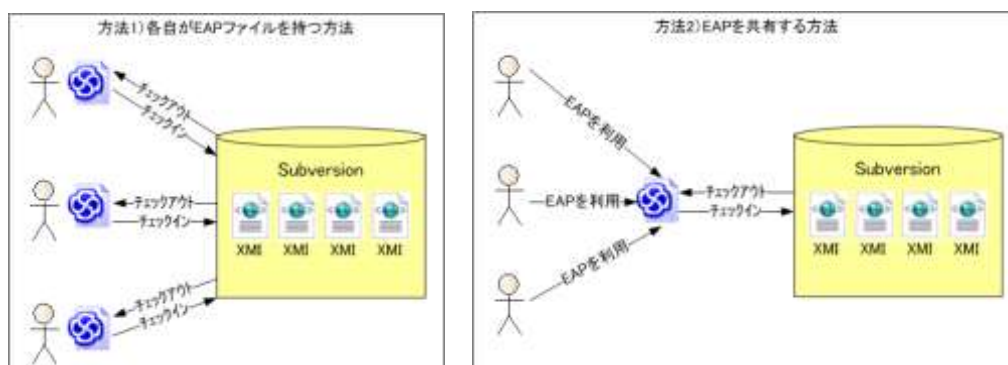
これで、準備は完了です。ここで作成した **work** ディレクトリ (`c:\svn\work`) を、**Enterprise Architect** の設定で利用します。なお、ここまでの手順は、**TortoiseSVN** を利用することで、より簡単に実行できます。(コマンドプロンプトからの操作が不要になります)

3.3 バージョン管理機能の構成

EA でバージョン管理機能を利用する場合は、EA のプロジェクトブラウザ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自動的に保存されます。(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場合と、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場合の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場合：

この方法の場合、ローカルの 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・チェックインした最新の情報は、手動で最新情報の取得をするためのメニューを実施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場合：

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、ダイアグラムをオープンしたり、要素のプロパティ画面を表示したタイミングで、最新の情報を知ることができます。

EA の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。

(注意:バージョン管理ツールに CVS を利用する場合は、プロジェクトを共有する方法を取ってください。理由として、各自がプロジェクトファイルを持つ方法にすると、同時に複数人が CVS から同じパッケージの情報を取得・編集できてしまい、後から、チェックインした人の情報で上書きされてしまうためです。CVS 以外の場合は、誰かがチェックアウトしている間は、ほかのユーザはチェックアウトできないようにバージョン管理ツール側ではじいてくれます。)

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。「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

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) 閉じる(C) ヘルプ(H)

まず、種類として今回は「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 とバージョン管理の設定」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まったく関係ありません。)

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 は、設定を行ったマシンに保存され、他のプロジェクトファイルからも簡単に参照することができます。この設定情報はレジストリの以下の場所に XML 形式で保管されます。

```
HKEY_CURRENT_USER\Software\Sparx Systems\EA400\EA\OPTIONS\VCConfigs
```

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、パッケージによっては svn.exe のファイルが含まれていない場合があります。その場合には、別途 svn.exe クライアントを入手してください。

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、Subversion のコマンドが実行され、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。

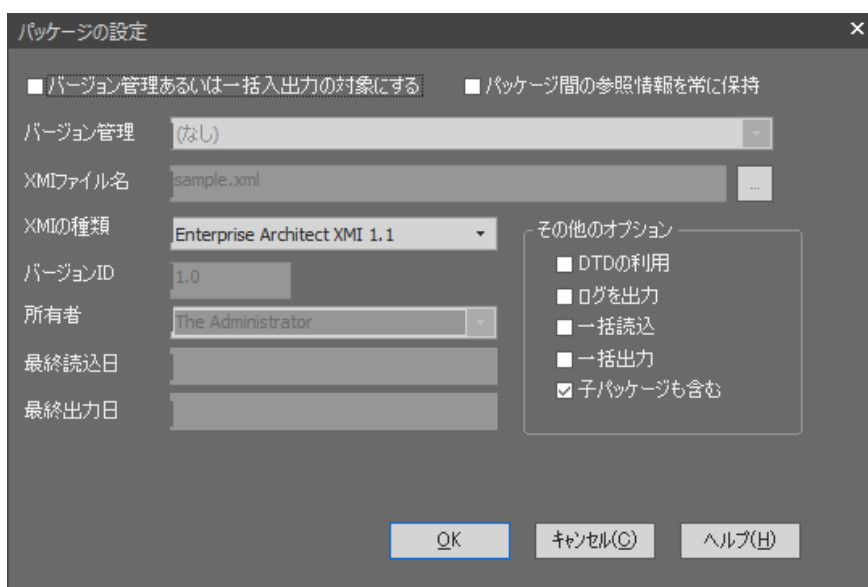
「保存」ボタンを押したときに Subversion のエラーが表示される場合には、Subversion の設定が正しくないか、作業用ディレクトリの位置がおかしいか、などの原因が考えられます。エラーメッセージに応じて修正してください。

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

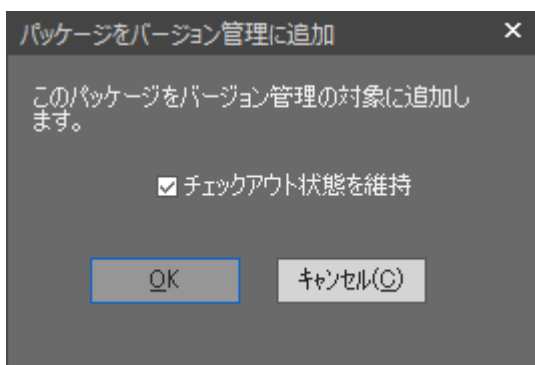
次に、実際にバージョン管理を行うためのパッケージを指定します。

まず、プロジェクトブラウザで対象のパッケージを選択します。「プロジェクト」リボン内の「バージョン管理」パネルにある「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、操作することのできなかつたチェックボックスやテキストボックスが利用できるようになりますので、「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」にパスが設定されますので、必要に応じてファイル名を修正します。

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」を選択することになります。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。



この状態でプロジェクトブラウザを確認すると、次の図のように以前と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。

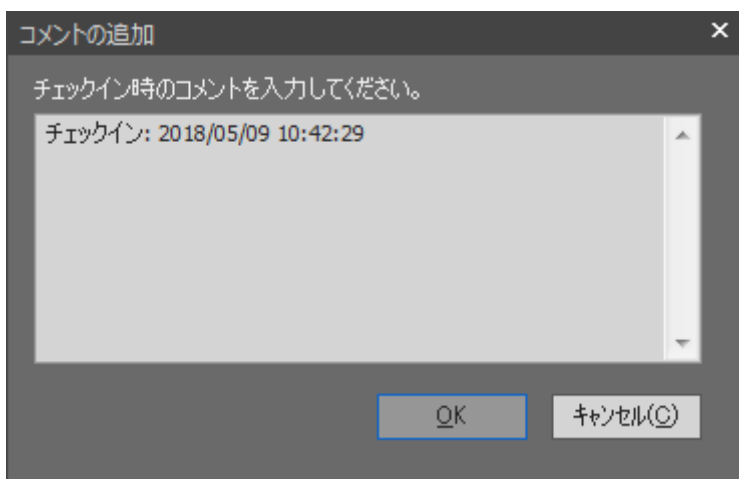


このアイコンは、バージョン管理されていて、チェックアウトしていない(ので編集できない)ことを示しています。

3.6 チェックアウトとチェックイン

次に、編集するためにチェックアウトの作業を行います。同じように、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、プロジェクト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。

作業が終わってチェックイン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」→「チェックイン」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。



チェックインした後は、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、チェックイン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することもできます。(4.2 章をご覧ください。)

ただし、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 Enterprise Architect の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。)

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 モデルの共有

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ファイル間でモデル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、別のプロジェクトファイル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ファイル間で共有することができます。

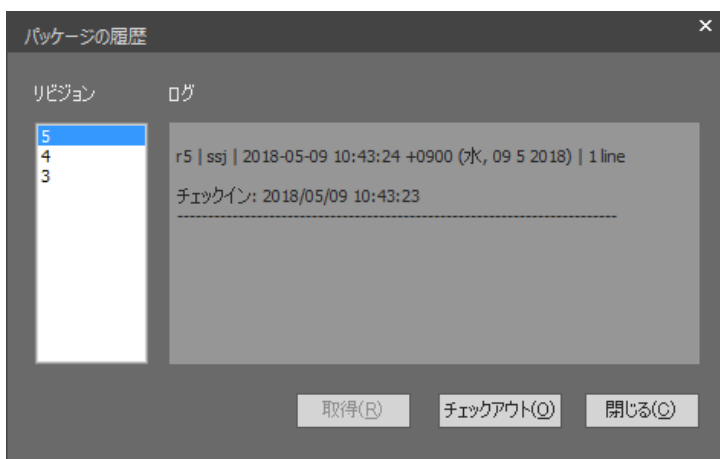
この機能の使い方は、ヘルプのほか PDF ドキュメント「複数のプロジェクトファイルを活用する設計開発 マニュアル」もご覧ください。

4.2 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プロジェクト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(枝分かれ)やタグ付けを行うことはできません。

4.3 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右クリックし「パッケージの管理」→「設定」を実行し、パッケージの設定画面を表示させます。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プロジェクト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「XMI 読み込み」ボタンを押し「XMI ファイルから読み込み」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. チェックインすると、マージした内容が最新バージョンとなります。

4.4 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. プロジェクト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」→「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プロジェクトブラウザで選択し、「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」→「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。

4.5 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が表示されています。

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・チェックインの操作が煩雑
(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

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 ジャパンから提供しています。

LemonTree では、EAP 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。

○ 改版履歴

2008/03/10 Enterprise Architect7.1の画像に更新。

2008/04/02リポジトリの作成におけるパスを一部修正。

2009/03/24 Enterprise Architect7.5の画像に更新。Subversionの推奨バージョンを更新。

2009/08/31 ドキュメントのタイトルを更新。

2010/04/16 Enterprise Architect8.0のリリースに伴い、内容を更新。

2011/05/18 Enterprise Architect9.0のリリースに伴い、内容を更新。Subversionクライアント
についての入手方法を追加。第3章を追加。

2011/08/01 ビルド909の変更点を反映。ユーザーについての説明を追記。

2011/09/12 「2.3 バージョン管理機能の構成」を追記。

2011/10/11 第2章としてバージョン管理の対象について追記。

2011/12/07 Enterprise Architect9.2 のリリースに伴い、内容を更新。

2012/03/07 Enterprise Architect9.3 のリリースに伴い、内容を更新。

2014/04/22 Enterprise Architect11.0 のリリースに伴い、内容を更新。

2015/05/29 指定した 2 つの履歴の差分を表示する方法を追記。

2015/09/04 差分比較とマージの説明を修正。その他、説明内容を最新の内容に更新。

2015/10/21 4.5 章を追加。

2016/10/07 Enterprise Architect13.0 のリリースに伴い、内容を更新。

2017/09/12 差分比較・マージツール LemonTree の情報を追記。

2018/05/16 Enterprise Architect14.0 のリリースに伴い、内容を更新。