



Using Enterprise Architect by multiple users

by Dermot O'Bryan and SparxSystems Japan

複数人数による設計開発での利用法 マニュアル

(2017/09/12 最終更新)



1	はじめに.....	3
2	プロジェクトの配置	3
2.1	開発環境の例.....	3
2.2	プロジェクトの配置に関する機能	3
2.3	プロジェクトの選択.....	4
2.4	プロジェクトの具体的な利用例	5
2.4.1	開発環境が単一	6
2.4.2	開発環境が複数(分散している場合).....	6
2.4.3	単一の開発環境+いくつかの分散環境	7
2.4.4	単一の開発環境 - 複数のモデル.....	7
2.5	プロジェクトファイルと DBMS リポジトリの間のモデル情報のやり取り.....	8
3	複数人数における開発に役立つ機能.....	8
3.1	複製と XMI の入出力	8
3.2	バージョン(履歴)管理.....	8
3.3	セキュリティ(アクセス権).....	9
3.4	ライセンスの共用	10
4	具体的な環境構築例	11
4.1	初期段階(評価段階)・少人数の場合	11
4.2	初期段階(評価段階)・中規模以上の場合	12
4.3	設計開発段階における環境構築	12
5	個別の機能の情報.....	14
5.1	パフォーマンスの最適化	14
5.2	複製と XMI ファイルの入出力の違い	16
5.2.1	複製.....	16
5.2.2	XMI ファイルの利用とバージョン管理	16
5.2.3	差分比較・マージツール「LemonTree」	17
5.3	パッケージの管理とバージョン管理.....	17
5.3.1	手動でのパッケージの管理	17
5.3.2	バージョン管理ツールを利用した管理	17
5.4	ベースライン・比較・マージ.....	18
5.5	リファレンス情報の入出力.....	19
5.6	Enterprise Architect のリモートインストール	19

1 はじめに

Enterprise Architect では、特に大規模での開発に役立つさまざまな機能があります。大規模の開発においては、その状況に応じて適切な方法でプロジェクト(モデルの情報などを含む、Enterprise Architect のファイルまたは DBMS リポジトリ)を配置する必要があります。このドキュメントでは、Enterprise Architect の機能と、そのプロジェクトの配置方法について紹介します。

なお、このドキュメント内での「プロジェクト」とは、Enterprise Architect の全てのエディションで利用できる「プロジェクトファイル(*.EAP ファイルあるいは*.FEAP ファイル)」と、コーポレート版で利用できる「DBMS リポジトリ」の両方を指すための言葉として利用しています。ご注意ください。それぞれの内容を指す場合には「プロジェクトファイル」「DBMS リポジトリ」と明記しています。また、ファイルの種類を明確に区別する必要がある場合には、「EAP ファイル」「FEAP ファイル」と表現しています。

Enterprise Architect Suite の各エディションは、コーポレート版の機能を利用できます。このドキュメントでは、コーポレート版に該当するものとしてお読みください。

2 プロジェクトの配置

2.1 開発環境の例

Enterprise Architect を大規模な環境で利用する場合にプロジェクトを配置する例として、次のような環境が考えられます。

- ・ 単一環境
 - LAN が敷設されている 1 つのフロア
 - 高速な WAN で結ばれている複数のフロア
- ・ 分散環境
 - 低速な WAN で結ばれている複数のフロア
 - 常設の WAN が存在しない複数のフロア
- ・ 組み合わせ
 - 大きな単一環境(例:本社)と複数の分散環境(例:お客様のフロア)
- ・ 単一環境・複数のプロジェクト
 - 複数のプロジェクト
 - 単一のプロジェクトで、中に複数のモデルを含む場合

2.2 プロジェクトの配置に関する機能

Enterprise Architect では、こうしたさまざまな状況に応じて組み合わせて利用可能なさまざまな機能があります。利用できる機能の数と範囲は Enterprise Architect の種類(デスクトップ版・プロフェッショナル版・コーポレート版)によります。デスクトップ版では複数人数での同時開発はできませんので、このドキュメントの対象外とします。

Enterprise Architect では、基本的には 2 種類のプロジェクトの形式が選択できます。

- ・ プロジェクトファイル
 - 手軽
 - 単一ファイルなのでコピーや持ち運びが簡単
 - EAP ファイルの場合には、複製機能が利用可能

- ・ DBMS サーバのリポジトリ(コーポレート版でのみ利用可能)
 - 堅牢
 - データのコピーや持ち運びを制限できる
 - 大きなサイズのモデルにも対応

この 2 種類の形式間については、自由にデータの転送が可能です。そのため、最初はプロジェクトファイル形式で運用し、状況が変化した場合に DBMS リポジトリに転送して継続利用することも可能です。

その他、両者で共通に利用可能な複数人数での開発に便利な機能は次の通りです。これらの機能については後述します。

・パッケージ単位の操作

- ✓ パッケージの内容を XMI 形式で一括入出力

・バージョン・履歴・管理

- ✓ 複数人数における排他処理と履歴管理を実現

・アクセス権の設定 (コーポレート版のみ)

- ✓ Enterprise Architect のさまざまな機能を利用可能かどうか指定可能
- ✓ パッケージやダイアグラム・要素をロックして排他制御

・ライセンスの共用 (フローティングライセンスのみ)

- ✓ Enterprise Architect の利用時のみライセンスを確保・終了したら開放
- ✓ 一定期間ライセンスを「持ち出す」設定にすることも可能

2.3 プロジェクトの選択

Enterprise Architect の主要な機能として、ローカルのファイルにモデル情報を保存することが可能であるだけでなく、Oracle や SQL Server などの DBMS(データベース管理システム)リポジトリにも保存可能なことが挙げられます。また、クラウドサーバという機能もあります。それぞれの方法には、異なる特長があります。主な項目の比較は次のとおりになります。

機能	EAP ファイル	FEAP ファイル	DBMS・クラウドサーバ
複製機能の利用	可能	不可能	不可能
想定している 利用人数	1 名から多くても 10 名程度	1 名のみ (ネットワークドライブに 配置しての共有不可)	それ以上の場合にも対応
堅牢性	なし	あり	あり

・ EAP ファイル

Enterprise Architect 標準の形式である EAP ファイルは、Microsoft の JET データベースエンジン(MS-Access の MDB ファイル)と同じ形式です。この形式の特長は、処理が早いことと、「複製」機能が利用できることです。ローカルマシンにファイルを配置して利用する場合にはメモリの使用量も少なく高速に動作します。以下の点に注意が必要です。

- ・ 多人数で利用する場合やモデルの内容が多い(ファイルサイズが大きい)場合には速度の低下が著しい
- ・ ネットワーク経由で利用している場合やマシンの予期せぬシャットダウンによりデータベースが破壊される可能性がある
- ・ Enterprise Architect で利用しているバージョンの JET データベースエンジンはすでに Microsoft のサポートは終了しており、多

数のバグが放置されたままになっている

・ FEAP ファイル

Enterprise Architect のバージョン 11.0 から利用できる FEAP ファイルは、オープンソースのデータベースエンジン Firebird の形式です。この形式の特長は、EAP ファイルに比較すれば処理が早いこと、EAP ファイルよりは堅牢性・性能が高いことです。また、オープンソースということもあり、JET データベースエンジンに比べると品質が高いです。以下の点に注意が必要です。

- ・ ネットワークドライブに FEAP ファイルを配置して、複数人で同時にファイルを開いて編集することはできない
- ・ バージョン 11.0 からサポートされた形式なので、予期しない不具合や未対応のアドインが残っている可能性がある (EAP ファイルの方が、いわゆる「枯れた」環境である)

FEAP ファイルを利用して複数人数で設計する場合には、バージョン管理機能との併用が必須になります。

・ DBMS リポジトリ・クラウドサーバ

DBMS サーバを利用する場合、プロジェクトファイルの問題点をクリアすることができます。堅牢性を実現するためにトランザクションの機能があるなどの関係で、速度の面ではプロジェクトファイルよりも遅くなります。しかし、多人数で大規模なモデルを利用する場合には、プロジェクトファイルよりも高速に動作する場合があります。また、ネットワーク経由で利用する場合にも、不慮のトラブルでデータベースが破壊され設計情報を失う危険性はありません。

しかし、リポジトリに接続を行う各マシンでは、ODBC などの接続設定が必要になります(初回に一度のみ)。また、DBMS リポジトリはコーポレート版で利用できます。

Enterprise Architect で利用可能な DBMS は Microsoft SQL Server, MySQL, Oracle, PostgreSQL, MSDE, Adaptive Server Anywhere です。なお、各 DBMS の性格や環境(OS)・設定の差異などにより、一部の DBMS については動作に問題が起きる場合もあります。弊社で全ての DBMS の全てのバージョン・ビルドの全ての OS での動作確認を行うことは難しく、組み合わせによっては何らかの問題がある可能性があります。この点はご了承・ご注意ください。なお、設定時にトラブルが少ないと思われる(=サポートへのご質問が少ない)DBMS は、SQL Server および MySQL です。

「クラウドサーバ」機能を利用することで、クライアント(Enterprise Architect をインストールするマシン)に ODBC ドライバをインストールする必要がなくなります。この場合には、DBMS リポジトリに直接接続する場合に比較すると通信速度は遅いですが HTTP や HTTPS でサーバと通信することもできます。クラウドサーバ機能を利用する場合には、クラウドサービスが常駐しクライアント(Enterprise Architect)からの通信を受け取る Windows マシンが必要になります。通常、このマシン内あるいは別のマシンに SQL Server などの DBMS を利用し、その DBMS にデータを格納します。この意味で、クラウドサーバを利用することと、DBMS リポジトリを利用することとは特徴や使うべき局面が似ています。このドキュメントで DBMS リポジトリを利用できると説明している局面では、クラウドサーバを利用することも可能です。クラウドサーバではサーバ側のプロジェクトの保存先として FEAP ファイルを利用することもできますが、EAP ファイルを利用することはできません。

(EAP ファイルからの移行時には、FEAP ファイルあるいは DBMS リポジトリに転送して利用します。)

なお、この「クラウドサーバ」機能は、いわゆる「クラウド」(外部のサーバ)にサーバを構築する必要はなく、社内で利用することもできます。ODBC ドライバの設定の手間を省けるなどのメリットがありますので、クラウドサーバの機能を利用することも検討して下さい。

クラウドサーバの詳細は、PDF ドキュメント「クラウドサーバ 設定と利用ガイド」をご覧ください。

2.4 プロジェクトの具体的な利用例

ここでは、上記の特長を考慮したプロジェクトの利用方法の具体的な例を 4 つご紹介します。

2.4.1 開発環境が単一

コーポレート版を利用する場合、単一のプロジェクトで利用するには次の 2 つの方法があります。

- ・ プロジェクトファイルをファイルサーバに配置して利用する
- ・ DBMS サーバを利用する

プロジェクトファイルを利用する方法は、とても簡単に利用することができます。利用者が 1 名から 5 名程度の場合には効果的です。ファイルサーバにあるプロジェクトファイルを利用するので、LAN などファイルサーバのファイルを操作できる環境であることが必要です。この方法でも、コーポレート版が提供するさまざまな機能を利用することができます。(コーポレート版で利用できる機能については、PDF ドキュメント「コーポレート版 機能ガイド」をご覧ください。)ただし、この方法の場合には EAP ファイルを利用してください。

この方法の問題点は次のとおりです。

- ・ プロジェクトファイルにアクセスする場合に処理速度が低下する可能性が高い
- ・ ネットワークに問題がある場合などにプロジェクトファイルが破損する(モデルの情報の一部または全部が失われる)可能性がある

これに対して、DBMS リポジトリを利用する場合には、大人数で利用する場合にも堅牢性が保たれます。複製機能を利用することはできませんが、バージョン管理機能を利用することで分散環境でも開発を進めることは可能です。ただし、プロジェクトファイルを利用する場合よりは性能面で劣る部分がありますので、サーバの性能やネットワークの速度などの環境を適切に整える必要があります。

2.4.2 開発環境が複数(分散している場合)

- ・ 低速の WAN で結ばれている複数の開発環境
- ・ 常設の WAN が存在しない複数の開発環境

このような場合には、主となる開発環境にメインのプロジェクトを配置して利用することになります。しかし、ネットワークの速度が遅いか、あるいは常に存在しない場合には、分散環境から主となる開発環境にあるプロジェクトを利用するのは現実的ではありません。それゆえ、分散された環境には別のプロジェクトを作成する必要があります。この別のプロジェクトは、定期的にメインのプロジェクトと同期する必要があります。

この同期を実現する方法の例は次のとおりです。

- メインのプロジェクトを EAP ファイルとし、分散環境の EAP ファイルはメインの EAP ファイルの複製として作成する。
- ターミナルサーバにプロジェクトファイルを配置し、利用者はターミナルサービスを通して利用する
(注意:この場合には利用する人数分だけ Enterprise Architect のライセンスが必要です。)
- メインのプロジェクトと分散環境にそれぞれ別々のプロジェクトを作成する。
 - XMI ファイルで情報を同期する。
 - バージョン管理機能を利用して同期する。(お勧め)

これらの方法の詳細について説明します。

a) EAP ファイルをメインのリポジトリにし、複製機能で同期する

ひとつの組織の中に部署などのいくつかの細かいグループがあり、それぞれのグループには 1 名から 5 名程度が所属しているような場合を想定します。このような場合に EAP ファイルで管理をする場合、マスターとなるメインリポジトリをまず決め、その後複製機能を利用してサブリポジトリを配置し、定期的に同期します。

この方法の場合、マスターのリポジトリとサブリポジトリの内容は、容易に同期可能で、編集内容は双方向に反映可能です。この点が c)の方法と異なる点です。

ただし、同じ要素をそれぞれの拠点で異なるように変更した場合、複製機能では差分を確認し同期することは困難です。あまりお勧めできない方法です。

b) ターミナルサーバを利用する

サイトが分散している場合でもサイト間の通信速度が比較的速いが通信量に制限がある場合には、ターミナルサーバを利用して Enterprise Architect を利用するという方法もあります。

この方法であれば、サーバとクライアント間の通信は画面の情報のみとなり、モデルなどの Enterprise Architect の情報は流れませんので、通信量を抑えて設計開発をすることができます。

なお、このような形式で Enterprise Architect を利用する場合、インストールするマシンは 1 台になりますが、ライセンスは利用する人数分必要となりますのでご注意ください(同時利用する人数ではなく、利用する可能性のある全ての人数分のライセンスが必要となります。フローティングライセンスの場合には、同時利用する人数分のライセンスが必要となります。)

c) 複数のプロジェクトを配置

複製機能に代わる機能として、バージョン管理機能があります。バージョン管理機能を利用して、排他制御を行うと同時に同期処理を行う方法が、一番優れています。このバージョン管理機能を利用して分散開発する方法が、世界中で広く利用されています。バージョン管理機能は全てのエディションで利用できます。

(クラウドサーバ機能のリリース後は、分散開発だけが目的でバージョン管理機能を利用することは徐々に減ってきており、クラウドサーバの利用が広がっています。)

この方法の具体的な例は、このドキュメントの後半で紹介します。

2.4.3 単一の開発環境+いくつかの分散環境

本社のようなメインになる開発環境と、いくつかの分散環境から構成される組織というのはそれほど珍しいものではありません。このような場合、メインになる開発環境では、モデルデータが大きくなり、利用する人数が多くなる傾向があるため、DBMS リポジトリを利用するのが効果的です。さらに、分散環境ではそれぞれにプロジェクトファイルを配置して利用するとよいでしょう。

これらの間のモデル情報のやり取りについては、バージョン管理機能を利用するとよいでしょう。バージョン管理機能を利用すると、パッケージの編集中にはロックが自動的にかかりますので排他処理が可能になります。

構成としては、2.4.2 章で説明した c)の方法と同一です。

2.4.4 単一の開発環境 - 複数のモデル

1つの開発プロジェクトの中で、複数のモデルを利用し、それぞれのプロジェクトで共通のモデルについてはプロジェクトをまたがって利用したいという場合を考えます。

このような場合には、DBMS リポジトリを作成するのがよいでしょう。複数のプロジェクトルート(最上位のパッケージ)を作成し、それぞれのモデルに割り当てます。バージョン管理機能を利用して共通に利用されるモデルの情報を保護し、一貫性を保証する必要があります。このような共通部分を編集した場合、バージョン管理機能を利用して変更内容を厳密に管理する必要があります。

また、このような環境ではセキュリティ機能(アクセス権管理機能)も効果を発揮するでしょう。それぞれのプロジェクトルートへのアクセ

ス権を適切に設定することで、プロジェクトメンバー以外の編集を防ぐことができます。

2.5 プロジェクトファイルと DBMS リポジトリの間のモデル情報のやり取り

分散環境における設計開発で重要な点は、モデルの情報をどのようにしてやりとりをするか、という点です。Enterprise Architect では、以下のような場合にデータを転送するための「プロジェクトの転送」機能を備えています。

- ・ プロジェクトファイル⇄プロジェクトファイル
- ・ DBMS⇄プロジェクトファイル
- ・ DBMS⇄DBMS

この機能を利用する主な機会のひとつとして、DBMS リポジトリを作成した直後が挙げられます。作成直後には DBMS リポジトリ内は空なので、適当なプロジェクトファイルの内容を初期状態として転送する必要があります。また、プロジェクトを移転する場合にも、この機能が利用できます。

3 複数人数における開発に役立つ機能

3.1 複製と XMI の入出力

既に説明しましたように、Enterprise Architect では複製機能と XMI の入出力機能が、分散環境での設計開発とモデル情報のやり取りに大きな意味を持っています。複製は EAP ファイル間でのみ利用可能であり、単に 2 つの EAP ファイルの内容を同期させます。同じ要素に同時に変更が加えられている場合には、どちらかの変更内容を選択し同期させます。ただし、その際に差分を視覚的に確認できない点など、いくつかのデメリットがあります。複製機能はプロフェッショナル版以上で利用できます。

FEAP ファイルや DBMS リポジトリを利用している場合には、XMI ファイルを利用してモデル間の入出力が利用できます。XMI ファイルはパッケージ単位で作成することが可能です。XMI ファイルには、指定したパッケージの中に含まれる全てのダイアグラム・要素・子パッケージの情報が含まれています。この情報を他のプロジェクトで読み込むことで、モデルの情報を簡単に移動させることができます。

この内容については 5 章もご覧ください。

3.2 バージョン(履歴)管理

Enterprise Architect では、モデル内のパッケージの単位でバージョン管理(履歴管理)を行うことができます。管理については、Subversion や Visual SourceSafe などの外部のツールを利用します。推奨は Subversion です。バージョン管理機能は全てのエディションで利用できます。

バージョン管理を行うことで、万が一過去のモデルに戻す必要があった場合でも、簡単に戻すことができます。また、バージョン管理されているファイルを読み込むことで、現在のモデルとの差分を確認することもできます。

パッケージクラス図をベースラインバージョン1.0と比較中

モデルの要素	状態	プロパティ	モデル	ベースライン
クラス図		別名		
Class1		重複を許可	false	false
attribute2	変更	コレクション	false	false
operation1	ベースラインのみ	Const	false	false
接続		IsLiteral	false	false
関連	ベースラインのみ	コンテナ	未指定	未指定
ソース	ベースラインのみ	包含		
ターゲット	ベースラインのみ	初期値		
関連	モデルのみ	派生	false	false
ソース	モデルのみ	順序あり	false	false
ターゲット	モデルのみ	長さ	0	0
Class4	モデルのみ	下限	1	1
attribute1	モデルのみ	名前	attribute2	attribute1
attribute2	モデルのみ	ノート		
operation1()	モデルのみ	位置	0	0
operation2()	モデルのみ	全桁数	0	0
接続		小数点桁数	0	0
関連	モデルのみ	Static	false	false
ソース	モデルのみ	可視性	private	Private
ターゲット	モデルのみ	ステレオタイプ		
		型	int	int
		上限	1	1
		修飾子		

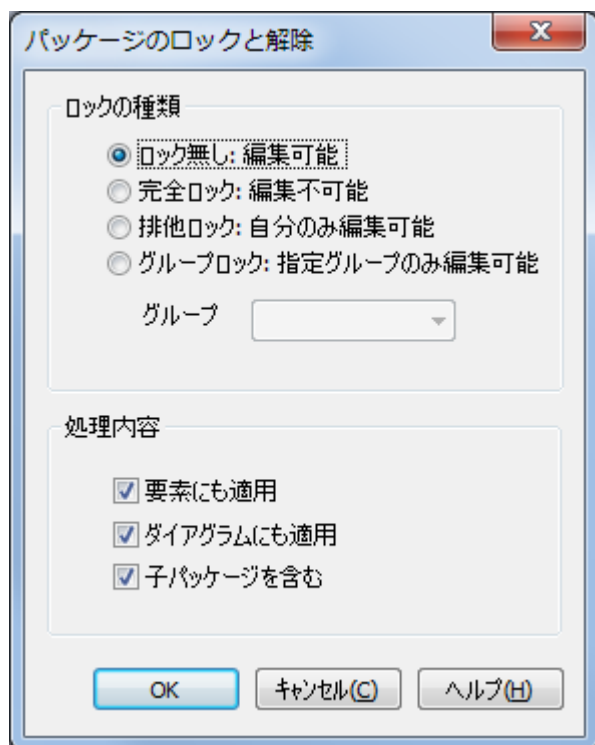
コーポレート版で利用可能なマージ機能を利用すると、過去のバージョンの一部の情報(要素や接続の単位)のみを最新版に反映することも可能になります。これにより、過去に削除された要素を最新のバージョンで復活させる、などの対応も可能になります。

この内容については 5 章もご覧ください。

3.3 セキュリティ(アクセス権)

Enterprise Architect ではセキュリティ(アクセス権)機能を利用することができます。この機能を利用すると、モデルに対して利用可能な機能(要素の編集・ソースコードの生成や読込・各種設定など)をユーザーに応じて制限することができます。また、要素やダイアグラム・パッケージ単位で、ユーザーやグループ(ユーザーが複数所属できる)に対して編集を制限することができます。この機能はコーポレート版でのみ利用可能です。

下の画面は、パッケージ単位でアクセス権を設定する場合のものです。



なお、Enterprise Architect では、2 種類のセキュリティモデルを提供しています。どちらかを選択して利用することができます。

a) 標準のセキュリティモデル

全ての要素とダイアグラムは既定の状態です。必要に応じて、ユーザーは要素・ダイアグラムやパッケージ(パッケージ自体と、パッケージに含まれる要素やダイアグラム)をロックすることができます。ロックする際に編集可能な対象をグループ単位で指定することもできます。

b) 厳密なセキュリティモデル

この場合には、全ての要素は既定の状態では編集することができません。要素のプロパティを参照することは可能です。この状態で排他ロックを行うと、ロックを実行した人だけがその要素の編集が可能になります。排他ロックとは、ロックを行った人だけが編集可能なロックです。

この方法の場合には編集時に手間がかかりますが、意図しない想定外の編集を防ぐことができます。また、複数のユーザーが同じ要素を同時に編集するという状況を防ぐことができます。

3.4 ライセンスの共用

Enterprise Architect のフローティングライセンスを利用することで、1つのライセンスを複数の利用者間で排他的に共有することができます。

「EA 終了時にキーを自動開放」の設定を利用する場合には、Enterprise Architect を起動したときにライセンスキーを取得し、終了すると開放します。つまり、同時に1ライセンスを複数人で共有することはできませんが、Enterprise Architect を終了させることで他の人が利用可能になります。多くの場合にはこの設定で利用することになるでしょう。



この設定を利用しない場合には、ライセンスの有効期間を日単位で指定することができます。例えば、有効期限を3日に指定した場合、3日間はネットワークから切り離されていても取得したライセンスをそのまま利用できます。しかしながら、Enterprise Architectを終了させても、指定された期間の間はライセンスが開放されません。

また、フローティングライセンスを利用している場合には、ライセンスマネージャーを利用することで、それぞれのライセンスキーをどのマシンの誰が利用しているかを把握することが可能です。利用状況をログとして出力し活用することもできます。

フローティングライセンスについての詳細は、PDFドキュメント「フローティングライセンス マニュアル」をご覧ください。フローティングライセンスは、コーポレート版および Enterprise Architect Suite の各エディションについて設定があります。また、いくつかのアドイン製品についてもフローティングライセンスがあります。

4 具体的な環境構築例

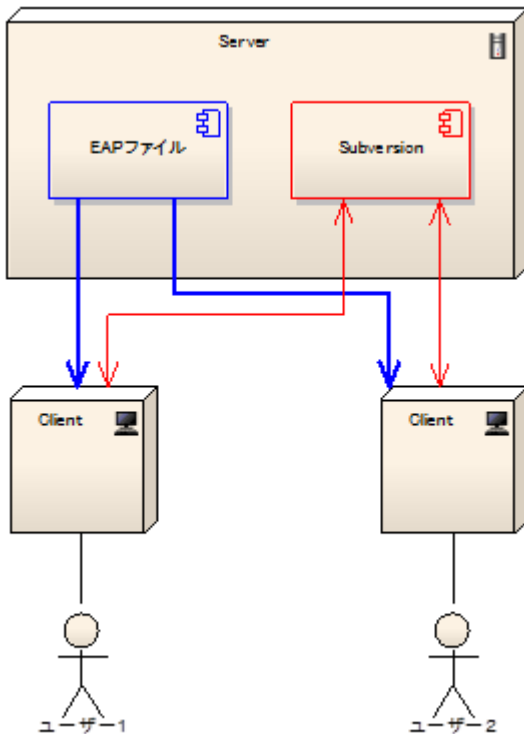
この章では、Enterprise Architect を利用して開発する場合を想定して、具体的な環境構築例をご紹介します。

4.1 初期段階(評価段階)・少人数の場合

Enterprise Architect を利用してどのように開発するか、という点が定まるまでは、手軽なプロジェクトファイルのみの運用をお勧めします。新しいプロジェクトファイルを作成し、そのプロジェクトファイルの内容を編集します。

また、将来的に規模が拡大することが予定されている場合には、この段階でバージョン管理機能についても試しておくといでしょう。Subversion や Visual SourceSafe を利用したバージョン管理機能は、複数人数で開発している人の多くが利用しています。Subversion を利用する場合には、TortoiseSVN などでリポジトリを作成し、チェックイン・チェックアウトできる段階まで作業をした段階で Enterprise Architect から利用設定を行います。これらの設定については、スパークスシステムズジャパンの Web サイトからダウンロードできる「バージョン管理機能 機能ガイド」をご覧ください。

Subversion などバージョン管理ツールの設定が完了したら、適当なパッケージを管理対象にし、その機能を確認してください。複数人数の場合には、以下の環境を構築して確認するとよいでしょう。下の図の「EAP ファイル」は DBMS リポジトリを利用することも可能です。また、Subversion とプロジェクトファイルが同じマシンに存在しなくても問題ありません。



複数人数で開発する場合に問題となるのは、同じパッケージを編集する場合にどのように排他処理を行うか、という点です。バージョン管理機能を利用している場合には、チェックアウトしない限り編集できませんので、結果的に確実な排他処理が可能になります。チェックアウトしていると他の人はそのパッケージの内容を編集できなくなりますので、作業の分担や範囲に応じて、適切にバージョン管理するパッケージを指定してください。

4.2 初期段階(評価段階)・中規模以上の場合

Enterprise Architect を利用した設計開発が中規模(10人以上)の場合には、プロジェクトファイルではなく、SQL Server などのデータベースサーバ上にプロジェクトを構築することも検討してください。なお、どのデータベースサーバを利用するか、については、SQL Server・MySQL のいずれかを選択することをお勧めいたします。

データベースサーバを利用する場合には、スパークスシステムズジャパンの Web サイトからデータベース作成用の SQL スクリプトをダウンロードし、テーブルを作成します。その後、既存のプロジェクトファイルを転送することで、利用可能になります。データベースサーバを利用している場合も、Subversion などのバージョン管理ツールの利用は可能です。このバージョン管理機能の利用の有無は、プロジェクトの形態がプロジェクトファイル・DBMS リポジトリのどちらであるかとは関係ありません。

複数人数での開発において、それぞれに役割を定義する場合があります。例えば、Aさんはソースコードとの同期(出力と読込)を担当し、それ以外の人はこの操作を禁止したい場合を考えます。このように人によって役割が変わる場合には、「セキュリティ機能(アクセス権)」を活用してください。この機能の詳細は、スパークスシステムズジャパンの Web サイトからダウンロードできる「アクセス権 機能ガイド」をご覧ください。

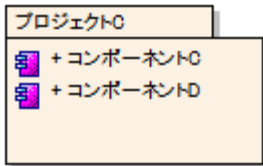
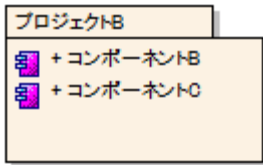
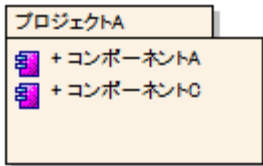
4.3 設計開発段階における環境構築

実際の設計開発段階における環境については、小規模なものであれば上記の 1 や 2 と同じようなシンプルな構成で問題ないでしょ

う。それ以上の規模の場合には、複数のプロジェクトが存在する場合もあるでしょう。

新規にプロジェクトを作成する場合、既存のプロジェクトからコピーして作成することで、各種設定(メインメニューの「設定」で変更可能な内容やアクセス権のユーザー設定など)などをそのまま引き継ぐことができます。

ここでは、実際の設計開発で良くある例として、複数のプロジェクトが存在し、その間に共通の情報が存在する場合を考えます。例として、次のような場合を考えます。



この図にあるように、プロジェクトが3つあり、中のコンポーネントCは3つのプロジェクトで共通に利用しているパッケージである場合を考えます。

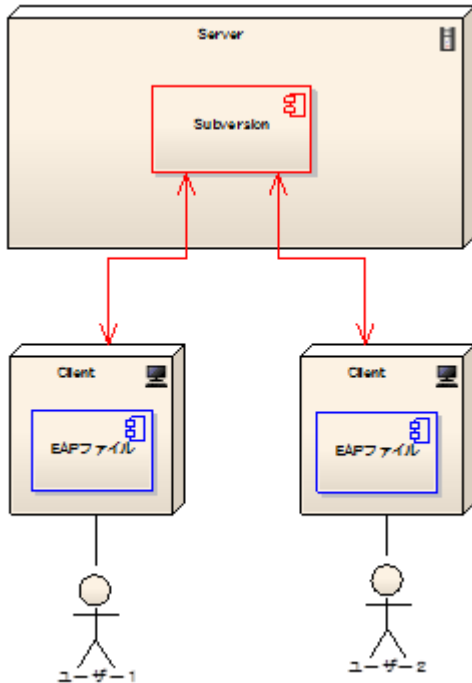
(ここでは、「コンポーネント」が Enterprise Architect のパッケージに該当します。)

このような場合には、まずプロジェクトAでコンポーネントAとコンポーネントCをバージョン管理の対象にします。その上で、プロジェクトBおよびプロジェクトCからそのバージョン管理されているXMIファイルを「パッケージを指定して追加」で読み込みます。(このときのバージョン管理の設定・設定識別IDはプロジェクトAと同じにします。)

このようにすることで、プロジェクトAで編集した内容について、他のプロジェクトでも「最新バージョンを全て取得」することで変更内容を反映させることができます。

このときに、それぞれの要素のGUIDは共通ですので、プロジェクトBやCにおいても、プロジェクトAの編集内容が適切に反映されます。

この方法を利用すれば、複数のプロジェクト間で情報を共有して活用することができます。この内容を図で表現すると、次のようになります。



なお、この構成で Enterprise Architect を利用する場合、パッケージの排他制御はバージョン管理ツール側の排他管理機能を利用して行います。つまり、排他ロックを行うことのできない CVS ではこの構成は利用できません。Subversion など他のバージョン管理ツールをご利用ください。

また、上記の構成の場合、EAP ファイルの代わりに FEAP ファイルを利用することもできます。

5 個別の機能の情報

5.1 パフォーマンスの最適化

LAN や WAN などのネットワークでプロジェクトを共有している場合に、パフォーマンスを最適化できる場合があります。一般的には、パフォーマンスは以下の条件に依存します。

- ✓ 利用しているリポジトリの種類(プロジェクトファイルか DBMS リポジトリか)
- ✓ バージョン管理機能を利用しているかどうか
- ✓ ネットワーク自身の速度
- ✓ ネットワークの負荷
- ✓ WAN の遅延時間 (このドキュメントにおいて、WAN が「早い」という場合には、40-50ms くらいの遅延時間を想定しています。)
注意: クラウドサーバ機能で改善できる場合もあります。クラウドサーバ機能を利用する場合には、通信内容を自動的に圧縮します。(バージョン 10 までの「WAN 高速化機能」はクラウドサーバ機能に統合されました。)

次の表は、高いパフォーマンスを出すための、ネットワークとリポジトリの種類を示しています。最初の表はバージョン管理機能を利用していない場合で、2 番目の表はバージョン管理機能を利用している場合の表になります。

バージョン管理機能を利用しない場合

リポジトリの種類	ネットワークの種類	利用者数	推奨	コメント
プロジェクトファイル	利用者のマシンにプロジェクトファイルを配置	1	○	高速に動作します。
プロジェクトファイル	LAN	1~10	○	LAN の速度・負荷に依存します。FEAP ファイルは利用できません。
プロジェクトファイル	WAN	1~10	×	推奨しません。DBMS を利用してください。遅延時間の短い WAN の場合には利用できるかもしれませんが、FEAP ファイルは利用できません。
EAP (複製機能)	利用者のマシンに EAP ファイルを配置	∞ *	△	他のファイルとの同期の対象となる EAP ファイルが必要になります。(複製機能の利用はあまりお勧めしません。)
DBMS	LAN	∞ *	○	大規模開発に適しています。LAN の負荷は DBMS の種類に依存します。(MySQL や SQL Server は Oracle に比べると負荷が低いようです。)
DBMS	WAN	∞ *	△	<ul style="list-style-type: none"> ➢ 遅延速度が短い WAN が必要です。 ➢ WAN の負荷は DBMS の種類に依存します。
XMI パッケージ (プロジェクトファイル・DBMS)	LAN ⇔ LAN	∞ *	△	XMI の入出力機能を利用して同期を行う必要があります。

* 同時に利用可能な利用者数はデータベースサーバの物理的な制約に依存します。

バージョン管理機能を利用する場合

バージョン管理機能を LAN や WAN で利用する場合には、パフォーマンスはネットワークの速度だけでなく、利用するバージョン管理ツールに依存します。

リポジトリの種類	バージョン管理	利用者数	推奨	コメント
プロジェクトファイル - 利用者のマシン I	LAN	1~10	○	パフォーマンスは最適です。
プロジェクトファイル - LAN 上	LAN	1~10	○	LAN の負荷にも依存します。FEAP ファイルは利用できません。
EAP ファイル - 複製	LAN・WAN	1~10	×	推奨しません。複製機能とバージョン管理機能を同時に利用することは非常に困難ですし、意味(メリット)もありません。
DBMS - LAN	LAN	∞ *	○	10 名以上の場合には最適の選択肢です。
プロジェクトファイル - 利用者のマシン or LAN	WAN	∞ *	○	プロジェクトファイルは各自のマシンに配置し、バージョン管理機能を利用して WAN 経由で更新を行います。 <ol style="list-style-type: none"> 1. バージョン管理ツールによっては、この方法が最適です。 2. プロジェクトファイルを LAN に配置し、複数人数で利用する方法も可能です。
DBMS - LAN	WAN	∞ *	○	LAN 上の DBMS リポジトリを共有で利用し、WAN 経

				<p>由でバージョン管理ツールを利用します。この方法を利用すると、プロジェクトの情報を分散環境で参照・利用できます。</p> <ol style="list-style-type: none"> EA での作業速度は比較的高速です。WAN による遅延はバージョン管理機能の機能呼び出しを実行した時に限られます。 WAN 上で DBMS リポジトリを利用するためのパッケージが流れることを防ぐことができます。
WAN DBMS	LAN/WAN		×	<p>推奨しません。以下のいずれかの方法を推奨します。</p> <p>a) プロジェクトファイル - LAN バージョン管理 - WAN DBMS - LAN バージョン管理 - WAN</p>

* 同時に利用可能な利用者数はデータベースサーバの物理的な制約に依存します。

5.2 複製と XMI ファイルの入出力の違い

分散設計開発を実現するためのモデルの情報交換の手段として、Enterprise Architect では 2 つの機能を提供しています。

- ✓ EAP ファイルの複製機能
- ✓ XMI ファイルの利用

また、サードパーティー製の製品として、差分比較・マージツール LemonTree を利用する方法もあります。これについても概要を紹介します。

5.2.1 複製

複製機能は、EAP ファイル間のモデル情報の交換・同期のためのシンプルな方法です。一般的には、マスターとなる EAP ファイルを中心的な場所に配置します。マスターの複製を必要に応じて作成し、物理的に離れた分散環境に配置します。この複製とマスターを同期させることで、モデル情報の同期と交換が可能になります。複製機能はプロフェッショナル版以上で利用できます。FEAP ファイルでは利用できません。

一般的には、1 つのマスターと複数の複製で構成します。詳細な設定方法についてはヘルプファイルの「複製」のページをご覧ください。

なお、複製機能は、同じ要素を別々に編集した際に、その差分を確認して同期する作業が困難です。同期することはできますが、それぞれの複製ファイルでは、異なる位置のモデルを変更することが前提です。(同じ要素を同時に編集しないことが前提です。) そのため、複数の人が同じモデルを編集したい場合には適さない方法です。現在は、バージョン管理機能を利用することを推奨しています。

5.2.2 XMI ファイルの利用とバージョン管理

DBMS リポジトリや大きなプロジェクトファイルを利用する場合には、XMI ファイルを介して、単一あるいは複数のパッケージを入出力することが少なからずあります。この方法により、設計チームを超えてモデル情報を共有することができます。この方法は、複製を利用する方法に比べて以下の利点があります。

- ✓ 必要な部分のみを取得し、モデルを構成することができます。
- ✓ 必要に応じて、すべてのモデル情報を構築することもできます。
- ✓ XMI ファイルを手動でバージョン管理しておくことで、目的に応じて、バージョンの異なるパッケージを取得し、モデルを構成することもできます。

- ✓ 設計者が異なるパッケージで作業を行うことで、設計者間の編集作業の衝突を防ぐことができます。

この方法は、バージョン管理機能を利用することでより効率的に行うことができます。バージョン管理機能を利用することをお勧めしますが、分散環境間で常設のネットワークがない場合には、この XMI ファイルを利用してモデル情報をやりとりする方法が最も有力な選択肢です。XMI ファイルの入出力機能とバージョン管理機能は、いずれも全てのエディションで利用できます。

5.2.3 差分比較・マージツール「LemonTree」

オーストリア LieberLieber 社の有償製品として、Enterprise Architect が持つ差分比較・マージの機能よりも高機能なツール LemonTree があります。このツールは、以下のような特徴があります。

- ✓ EAP ファイル単位での差分比較・マージが可能
- ✓ 接続を含めた、差分の可視化
- ✓ 大きな EAP ファイルでの比較・マージ作業の効率化
- ✓ 3 点マージ機能による、複数人の同時編集におけるマージが可能
- ✓ バージョン管理ツール Subversion, Git と連携し、ソースコードと同じような使い方が可能
(各自がそれぞれに編集し、コミット時に衝突があれば LemonTree が起動し、視覚的に差分を確認しながら衝突を解決しコミットが可能)

この製品の詳細は下記ページをご覧ください。

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

5.3 パッケージの管理とバージョン管理

Enterprise Architect では、2 種類の管理方法を提供しています。

- ✓ 手動でのパッケージの管理
- ✓ バージョン管理ツールを利用した管理

いずれの方法も、設計者間でパッケージを共有するための手段という意味では共通です。バージョン管理機能では、さらに変更の履歴を保持し、必要に応じて過去のモデルに戻ることができるようになります。しかし、バージョン管理機能を利用するためには、Enterprise Architect が対応するバージョン管理ツールをインストールし、設定を行う必要があります。

5.3.1 手動でのパッケージの管理

手動でのパッケージの管理では、あるプロジェクトから別のプロジェクトにパッケージの単位でモデル情報を移動します。複数のパッケージが対象の場合には、一括して出力あるいは入力することができます。一括での操作の場合には、対象となるパッケージを事前に指定する必要があります。

この手順は次の通りです。

1. 対象のパッケージを指定する
2. 一括入力あるいは一括出力の機能を実行する

それぞれの詳細については、ヘルプファイルの「コントロールパッケージ」のページをご覧ください。

5.3.2 バージョン管理ツールを利用した管理

Enterprise Architect では、バージョン管理ツールを利用してバージョン管理リポジトリにモデル情報を格納し、利用することができます。この方法には 2 つの利点があります。

- ✓ 設計者間でのパッケージの共有を簡単に行うことができる
- ✓ パッケージの編集内容を保存し、必要に応じて過去の内容に戻すこともできる

バージョン管理ツールとして、以下のツールに対応しています。

- ✓ SCC 標準に対応したツール(Visual SourceSafe など)
- ✓ Subversion
- ✓ Microsoft TFS
- ✓ CVS (非推奨)

モデルを共有する場合にバージョン管理ツールを利用することにはさまざまなメリットがあります。

バージョン管理ツールを利用することで得られる主なメリットとして、次の 4 項目が挙げられます。

利用方法	説明
モデルの共有	プロジェクトファイルや DBMS リポジトリを利用するだけでも、モデルの共有は可能です。このような場合には、明示的にモデル情報の取得を実行しなくても最新の情報を確認できます。バージョン管理機能を利用することで、排他制御が行えるほか、変更履歴を取得・管理することができます。
モデルの分散	まず、マスターとなるプロジェクトファイルを作成し、バージョン管理の設定を行います。その後、そのファイルをコピーし、それぞれの設計者に渡します。それぞれの設計者はそのプロジェクトファイルで作業を行います。他の設計者の情報は「パッケージを指定して追加」「最新バージョンを取得」機能でバージョン管理ツールを通して取得します。
共有パッケージ	プロジェクトファイルをそれぞれの設計者が保持します。1 つまたは複数のパッケージを、バージョン管理ツールを通して共有します。
標準パッケージ	組織や会社で共通に利用するパッケージ(編集せず参照のみを行うパッケージ)がある場合、そのパッケージをバージョン管理ツールで共有します。それぞれのプロジェクトでは、「パッケージを指定して追加」機能でそのパッケージを取り込み、標準パッケージの内容が更新されたら「最新バージョンを取得」機能を実行して最新の内容を取り込みます。

バージョン管理の設定の詳細についてはヘルプファイルの「バージョン管理」のページをご覧ください。設定方法については、PDF ドキュメントも提供しています。スパークスシステムズ ジャパンの Web サイトからダウンロードできます。

5.4 ベースライン・比較・マージ

ベースライン機能は、パッケージを対象に「スナップショット」(ベースライン)を作成・保存することのできる機能です。ベースライン機能を利用すると、2 つのベースラインを比較することもできます。さらに、比較した結果から差分を確認し、差分(の一部)をマージすることもできます。ベースライン機能はコーポレート版のみで利用できます。ベースライン機能は、バージョン管理ツールの設定が必要ない、簡易的なバージョン管理機能と言えるかもしれません。

ベースライン機能を使う一般的な例としては、定期的にベースラインを保存し、最新のモデルと過去のベースラインを比較することで更新内容を把握するために利用します。場合によっては、モデルの一部をロールバックすることもあります。

また、この機能を利用すると、「ブランチ分けとマージ」(複数の異なる設計の同時開発とマージ)も可能になります。基準となるプロジェクト(「ベース」)をコピーし、「ブランチ」とします。設計中の何らかのタイミングで、必要に応じて「ベース」と「ブランチ」のそれぞれのプロジェクトの差分を確認し、また変更内容をマージします。

ベースライン・比較・マージの機能の詳細はヘルプファイルの「ベースラインと比較・マージ」のページや、PDF ドキュメント「差分比較とマージ 機能ガイド」をご覧ください。

5.5 リファレンス情報の入出力

UML のステレオタイプやタグ付き値の定義など、プロジェクトに保存されている、設計開発に関連する情報(リファレンス情報)を複数のプロジェクト間で共有する方法には次の 3 つがあります。

- a) 1 つのプロジェクトに複数のプロジェクトルートを作成し、パッケージなどと同様にリファレンス情報を共有する
- b) プロジェクトごとに異なるプロジェクトファイルを作成し、リファレンス情報(共通の情報)は定期的に更新する
- c) 共有リポジトリ機能を利用して、1 つのリファレンス情報を参照するようにする
(DBMS リポジトリを利用している場合のみ利用可能な機能です)

リファレンス情報の入出力の機能は、2 番目の方法を実現するための機能です。一般的には、マスターとなるプロジェクトの内容を随時最新になるように更新し、それ以外のプロジェクトへ定期的に内容をコピーします。

リファレンス情報には次のような情報が含まれます。

- ✓ 用語集
- ✓ さまざまな種類の定義(状態・要求など)
- ✓ リソースや顧客などの情報
- ✓ RTF ドキュメントや HTML ドキュメントのテンプレート
- ✓ ソースコード生成や MDA 変換のテンプレート
- ✓ セキュリティ(アクセス権)関連の情報
- ✓ ステレオタイプ

出力したリファレンス情報は XML ファイルになります。

リファレンス情報の入出力の手順の詳細は、ヘルプファイル「リファレンス情報の入出力」のページをご覧ください。

共有リポジトリ機能は、リファレンス情報を格納している DBMS のテーブルを、他のデータベーステーブルと共有する方法で実現しています。データベースに対して、共有リポジトリを設定するためのスクリプトを実行する必要があります。詳細は、ヘルプファイルの「リファレンス情報をリポジトリ間で共有する」のページをご覧ください。

5.6 Enterprise Architect のリモートインストール

Enterprise Architect をネットワーク上のマシンにインストールする場合には、いくつかの方法があります。Windows Server ツールや Microsoft SMS のようなインストールソフトウェアを利用することもできます。このような場合に利用するための MSI ファイルの作成方法を説明します。

Enterprise Architect のセットアップファイルは、MSI 形式のインストーラです。この MSI ファイルを利用して、Enterprise Architect のリモートインストールに利用します。

一般的なりモートインストールのためのソフトウェアはこの MSI ファイルに対応しています。

○改版履歴

2007/07/11 Oracle の対応するバージョン情報を変更。その他細かい修正。

2007/12/06 バージョン管理ツールを利用する場合の具体的な方法についての図を追加。その他細かい点についての説明を追加・変更。

2009/08/31 ドキュメントのタイトルを変更。

2009/11/30 第 5 章の内容を追加。

2010/07/22 フローティングライセンスの取得について、バージョン 8.0 の情報を追加

2011/06/06 レジストリの SKT の型が間違っていた点を修正

2011/12/06 いくつかの説明を追加。ドキュメント名などの修正。

2013/08/29 説明の内容・メニュー名を最新の情報に更新。EAP ファイルの複製機能よりもバージョン管理機能の利用を推奨することを明記。エディションによって利用できない機能については、利用可能なエディションを明記。

2013/10/04 説明中の用語を調整・統一。

2014/04/22 バージョン 11.0 のリリースに伴い、内容の加筆修正。

2015/02/09 クラウドサーバについての説明を追加。

2017/09/12 フローティングライセンスに関する設定の情報を「フローティングライセンス マニュアル」に移動。その他古い内容を最新の内容に修正。