



How to use Enterprise Architect by multiple users

by Sparx Systems Japan

複数人数による設計での利用法 マニュアル



目次

1	はじめに	3
2	5 つの主な利用方法	4
3	プロジェクトファイルを共有	5
4	複製機能を利用して分散設計	7
5	Subversion を利用して分散設計	9
6	Pro クラウドサーバでプロジェクトを共有	11
7	LemonTree + Git/Subversion で分散設計	13
8	方式を選択するためのヒント	14
9	パッケージリポジトリ	16
10	複数プロジェクトの利用	17
11	その他の注意点	19

1 はじめに

Enterprise Architect は、チーム(複数人)での設計で活用することを想定したツールです。個人が UML などの記法に即した絵を描くことよりは、チームでの設計を支援することを主な目的としています。しかし、一口にチームと言っても、物理的に同じ環境・同じフロアに集まって利用する場合もあれば、複数の拠点に分かれている場合もあります。人数も数名の場合もあれば、100 名を超える場合もあります。それぞれの設計者がリモートワークを行っている場合もあるでしょう。

このドキュメントでは、チームで Enterprise Architect を利用する場合について一般的で多くの状況に対応できる 5 つの利用方法を紹介します。このドキュメントで説明する内容に関連する情報を「Enterprise Architect 活用セミナー」でも紹介しています。

Enterprise Architect 活用セミナー

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

なお、このドキュメント内での用語は以下の通りです。

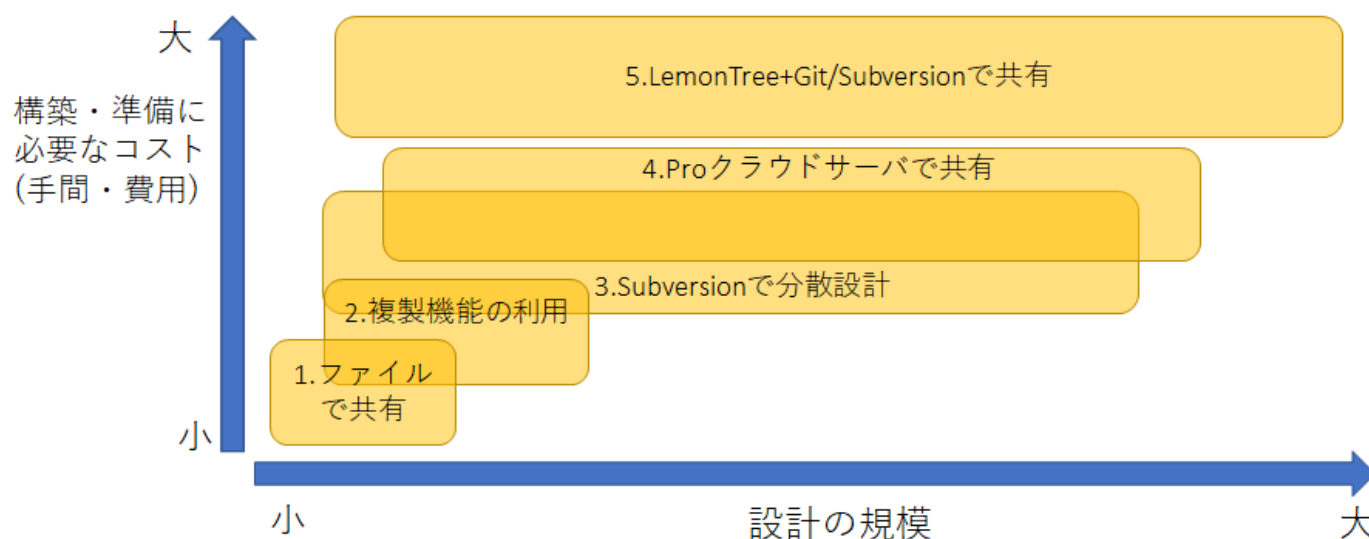
- 「プロジェクトファイル」
 - ファイルに作成した情報を保存します。
 - Enterprise Architect の全てのエディションで利用できます。
 - いくつかの形式があり、拡張子により区別できます。このドキュメントでは、QEA/QEAX ファイルを対象とします。EAPX ファイル・FEAP ファイルにはさまざまな制約・制限・バグなどがあるため、利用は推奨しません。
(これらの制約・制限・バグなどは、今後のバージョン・ビルドの Enterprise Architect でも解消はされません。QEA/QEAX ファイルを利用する前提です。)
 - QEA ファイルと QEAX ファイルの中身(形式)は同一です。Enterprise Architect は、ファイルの拡張子を見て動作を変えています。これらのファイルの形式を変換するためには、Windows のエクスプローラで拡張子を変更します。
- 「DBMS リポジトリ」
 - SQL Server や MySQL などの DBMS (データベース管理システム)に作成した情報を保存します
 - コーポレート版以上のエディションで利用できます。
- 「Pro クラウドサーバに設定するプロジェクト」
 - 別途インストールする Pro クラウドサーバに対して、HTTP/HTTPS で情報をやりとりします。
 - Pro クラウドサーバのバックエンドとして作成した情報を保存するためには、QEA ファイル・DBMS リポジトリのいずれかが必要です。
- 「プロジェクト」
 - 「プロジェクトファイル」「DBMS リポジトリ」「Pro クラウドサーバに設定するプロジェクト」の 3 つのすべてを指します。

2 5つの主な利用方法

チームで Enterprise Architect を利用する場合のプロジェクトの配置とその利用方法は、関係する Enterprise Architect の機能の活用方法も含めて考えると、さまざまな組み合わせが考えられます。しかし、一般的には、次の5つのいずれかを基本として検討し、個々のチームごとの事情を考慮して内容を調整する方法をお勧めします。

1. プロジェクトファイルを共有
2. 複製機能を利用して分散設計
3. Subversion を利用して分散設計
4. Pro クラウドサーバでプロジェクトを共有
(DBMS リポジトリでプロジェクトを共有する場合も含む)
5. LemonTree + Git/Subversion で分散設計

これらの5つの方法の概要は、このドキュメントで説明します。これらの5つの方法について、適用できる設計の規模と、構築・準備に必要なコストとで整理すると、次のような関係になります。



なお、LemonTree は有償(別売り)の製品です。そのため、構築・準備に必要なコストが高くなります。製品の詳細は以下 URL をご覧ください。

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

また、これらの 5 つの方法をプロジェクトが 1 つか複数かという軸で分類すると、以下の 2 つに分類できます。

	プロジェクト共有型	プロジェクト分散型
	1. ファイルで共有 4. Pro クラウドサーバで共有	2. 複製機能の利用 3. Subversion で分散設計 5. LemonTree + Git/Subversion で共有
プロジェクトの数	全体で 1 つ	基本的には、設計者の人数分
特徴	設計者全員が 1 つのプロジェクトを共有するので、全員が常に最新で一元化された情報を参照できる	各個人の手元にプロジェクトを保持するので、プロジェクトの参照・編集時にネットワーク通信が発生せず、性能劣化しにくい

また、これらの方式とは別に、「パッケージリポジトリ」機能でモデルを部品として共有する方法について、9 章で説明します。パッケージリポジトリ機能は、他の方法と併用して利用できます。

以下、それぞれの方法の概要と長所・短所について説明します。

3 プロジェクトファイルを共有

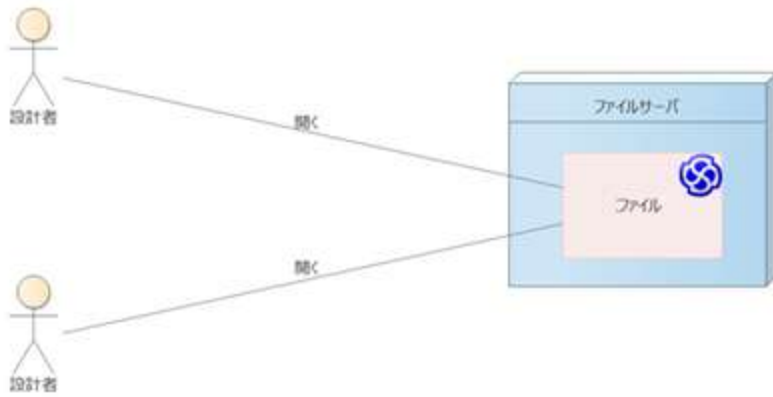
プロジェクトファイルを共有する方法は、チームでの設計を行う場合の最初の方法として最適です。このドキュメントで紹介する方法のどれを採用すれば良いか判断ができない場合には、まずはこの方法で作業を始め、状況に応じて他の方法に切り替えていくことをお勧めします。

この方法を実行するまでの手順は次の通りです。

1. QEAX 形式のプロジェクトファイルを作成し、チームの全員が利用できるネットワークドライブにそのファイルを配置します。
QEAX 形式は QEA 形式と完全に同一です。拡張子 QEA のファイルを、拡張子を QEAX に変更することで QEAX 形式となります。拡張子が QEAX の場合には、複数人での同時利用を前提とし、編集した内容をその都度ファイルに書き込みます。QEA の場合には、処理速度を考慮し、一定時間ごとにその間の編集内容をまとめて書き込みますので、複数人での利用の場合にはその間に競合が発生する可能性があります。
2. それぞれの設計者が、そのネットワークドライブのファイルをダブルクリックして開きます。

このように、事前の設定は全く必要なく、単純に 1 つのファイルを複数名で同時に開くだけで、1 つのプロジェクトをチームで共有することができます。

次の図は、この方法の構成のイメージを示しています。



なお、Google Drive・OneDrive・Dropbox などのファイル共有のシステムを利用してファイルを共有し同時編集・同時利用することはできません。これらのシステムは、同一のファイルを同時に編集しているわけではなく、それぞれのユーザーの環境にあるファイルのコピーを編集していますので、上記のような同一ファイルの編集とは技術的に異なります。また、双方で編集されたプロジェクトファイルのマージの仕組みはこれらのファイル共有システムにはありませんので、片方の編集内容は他方で上書きされてしまうか、別々の異なる 2 つのファイルとして保存されます。

この方法のメリットとデメリットは次の通りです。

○メリット

- 簡単に環境を構築し実行できる
- 1 つのプロジェクトを共有するため、他の人の作業状況が見える

○デメリット

- Windows のネットワークドライブ・ファイル共有の仕組みを利用するので、設計者が異なるネットワーク環境に属する場合など、環境によってはこの方法は使えない
 - 多くの企業では、外部ネットワークとのファイアウォールにて Windows のファイル共有の通信を遮断している
- プロジェクトファイルを利用する場合の制約として、利用人数・プロジェクトの規模(ダイアグラムや要素の数など)・ネットワークの通信速度により、Enterprise Architect の動作速度が劣化する
 - VPN などネットワークドライブが参照可能としても、通信速度が高速ではないことが多いため、実用的ではないことが多い
 - 利用人数や規模が大きくなり、以前に比べてダイアグラムやプロパティを開くのが遅くなったと感じることがあれば、他の方法への移行を検討する目安となる
(プロジェクトファイルの圧縮を行っていないなど、他の理由で動作速度が劣化する場合もあるので注意が必要です。)
 - 他の方法に移行すべき具体的な利用人数・プロジェクトの規模などの目安は、人数や規模だけでなくネットワークの通信速度やマシンの性能などにも依存するため、唯一の目安を提示すること

は困難

(参考情報として、過去の経験やお客様からの情報などでは、5 名以上が同時に 1 つのプロジェクトファイルを編集する状況が、移行の検討のきっかけになることが多いです。)

- 同じ要素に対して複数名が同時に同じ要素を編集した場合、いわゆる「後勝ち」になり、先に編集した人の編集内容は上書きされる
 - ファイル単位ではなく要素単位なので、別の要素であれば問題ない
 - ダイアグラムは、同時に複数名が編集することは想定していない
(「Enterprise Architect 続・入門セミナー」で説明しています)
 - セキュリティ(アクセス権)機能で、排他制御が可能

4 複製機能を利用して分散設計

QEA ファイルでは、「複製」の機能が利用できます。複数の QEA ファイルを作成してそれぞれの利用者が保持し、変更内容をマージできる仕組みです。複製の機能を有効にすると、変更を行ったかどうかの情報を記録する領域が内部に追加され、その漁期の情報を参考にして差分を判定します。複製機能には長所と短所がありますので、他の方式も含めて採用の可否を検討してください。

複製機能を利用する場合には、1 つの「デザインマスター」と、人数分の「レプリカ」のファイルを作成します。作業の流れの概要は以下の通りです。

1. 既存の QEA ファイルを「デザインマスター」に変換する
(「プロジェクト」リボンの「ツール」パネルにある「転送」ボタンを押し、「QEA ファイルの複製」→「デザインマスターの作成」を実行。変換対象のファイルをバックアップしておくが良い。)
2. デザインマスターから、「複製」を作成する
(1 と同じ「QEA ファイルの複製」の「新しい複製の作成」を実行)
3. 複製のファイルをコピーし、各設計者に配布する
4. 各設計者は内容を編集後、デザインマスターと同期する
(デザインマスターを開き、1 と同じ「QEA ファイルの複製」の「複製の同期」を実行)
5. 同期は双方向に行われ、衝突(同じ内容を編集している場合)には「同期中の衝突の解決」画面が表示されるので、解決する必要がある

それぞれの設計者は複製ファイルを持ち作業をしますので、他の作業者の進捗状況などはわかりません。また、結果的に同じ位置づけの要素を別々の設計者が追加してしまうということもあり得ます。そのため、この機能を利用する場合には、基本的には QEA ファイル内に複数のパッケージを作成し、各パッケージの担当者を割り当て、それぞれの担当者が自分のパッケージの範囲内で作業をする方式を推奨します。

(各設計者が設計モデルを自由に変更して良いとすると、衝突や重複などさまざまな問題が発生しやすくなります。)

各作業者は作業が完了したら、デザインマスターと同期処理を行います。同期を行うと、各設計者の変更内容がデザインマスターに追加されると同時に、他の設計者がデザインマスターに同期した内容が自分の複製ファイルに反映されます。

万が一何らかの理由で変更内容が衝突する、例えば同じ要素を編集してしまっている場合には、衝突したことを示す「同期中の衝突の解決」画面が表示されます。

同期中の衝突の解決

衝突しているレコード

説明	衝突の一覧	種類	行ID
Diagrams	t_diagram	マスターと複製の両方で...	6e2d389ad857a7436ce0c... ..
Object Details	t_object	マスターと複製の両方で...	c0318fe9de4946369a5100... ..

マスター側の値を全て採用
複製側の値を全て採用

衝突の詳細

フィールド	マスターの値	複製の値
Name	☒ aaa	☐ bb
ModifiedDate	☒ 2022-03-08 09:02:23	☐ 2022-03-08 09:02:42

マスター側の値を採用
複製側の値を採用
選択した項目を採用

閉じる
ヘルプ

この場合には、どちらの内容を採用するかを決めなければなりません。しかし、上の画像のように、単純な差分のみで判断する必要があり、影響範囲などを考慮してどちらを採用するかを検討はできません。また、ダイアグラムを参照して差分の把握もできません。そのため、できる限り衝突しないようにパッケージ構成や作業範囲を明確にするか、あるいは複製機能ではなく後述する LemonTree を利用する方法を検討してください。

この複製機能は、追加投資の必要が無く、準備も比較的容易ですので分散設計を行う場合の「最初の一步」としては有用です。しかし、人数が増えモデルの構成が複雑になるにつれ、パッケージ単位で明確に作業範囲を分割することは容易ではなく、結果的に衝突が増えその解決が困難になります。将来的には、LemonTree を利用する方法など他の方法に移行する必要があるでしょう。

なお、この複製機能を利用する場合には、複製に関する情報を格納するためファイルサイズが大きくなり、また

処理時間も多少増加します。複製機能の利用を終了する場合には、複製情報の削除が必要です。

5 Subversionを利用して分散設計

Enterprise Architect は、作成したモデルの履歴を管理するためのバージョン管理機能を搭載しています。バージョン管理機能は、Subversion 等の外部のバージョン管理ツールと連携し、機能を実現しています。バージョン管理機能の主目的は名前の通りバージョン(履歴)を管理することがですが、この機能を活用して異なる拠点(ネットワーク)での分散設計を実現できます。

このバージョン管理機能を利用して分散設計を行う方法は、Pro クラウドサーバがリリースされるまでは、複数の拠点がある場合の分散設計を実現する方法として世界中で広く利用されていました。現在は、Pro クラウドサーバを利用する方法の採用が増えていますが、ネットワークの通信速度が十分でない場合など、まだ適用できる場面は残されています。

この方法で利用するバージョン管理ツールは、Subversion がほぼ全ての場合で採用されています。安定性・利便さ・費用(無償)の点で Subversion が利用されます。なお、Subversion 以外にも、バージョン管理ツールに関する事実的な標準である SCC(Source Code Control) API に対応するツールも利用できます。Enterprise Architect のバージョン管理機能は Git には対応していません。Git を利用する場合には第 7 章で説明する LemonTree の利用が必要です。

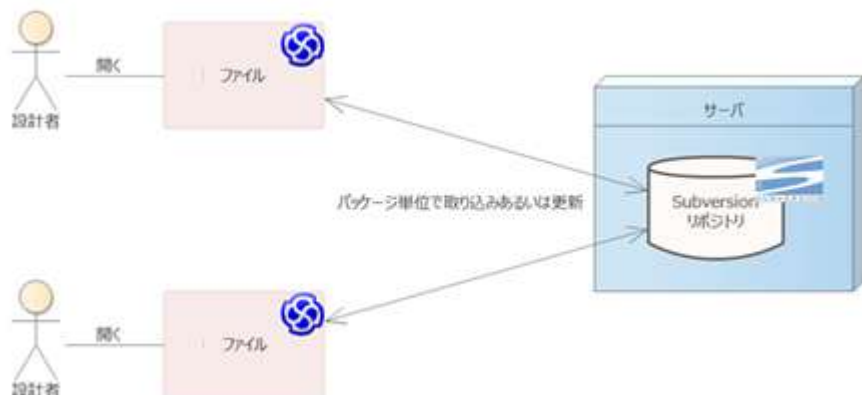
Subversion を利用し、この方法を実行するまでの手順は次の通りです。なお、手順 3 までの内容は、Enterprise Architect に依存しない一般的な Subversion の使い方となります。手順 3 までの内容が不明な場合には、Web サイトでの検索などで情報を入手してください。

1. Subversion のリポジトリを構築し、ユーザーなどの設定を行う
2. Enterprise Architect を利用する各設計者のマシンに Subversion クライアントをインストールする
(作業の手間を考えると、TortoiseSVN を利用することをお勧めします。TortoiseSVN のインストール時に、コマンドライン クライアント(svn.exe) をインストールするように指定してください。)
3. 各設計者が、Subversion のリポジトリからチェックアウトできるように設定
(パスワードなどの入力なく利用できるように、認証情報は保存するように設定してください。)
4. チームのうちの 1 名が、Enterprise Architect のプロジェクトファイルを新規作成し、パッケージなどの構成を作成する
5. そのプロジェクトファイルに対してバージョン管理機能を有効にして、手順 3 で設定したディレクトリを作業ディレクトリとして指定する
6. 作成したパッケージのうち、共有するパッケージをバージョン管理の対象のパッケージとする
7. 設定したプロジェクトファイルを、他の設計者に渡す
8. それぞれの設計者がプロジェクトファイルを開く

(それぞれの設計者ごとのバージョン管理に対する設定を要求されますので、手順 4 と同様に作業ディレクトリを指定します。)

上記のように、プロジェクトファイル自身をバージョン管理対象にするのではないことに注意してください。バージョン管理機能についての設定・操作などの基礎知識は、ドキュメント「バージョン管理機能 機能ガイド」をご覧ください。

次の図は、この方法の構成の概要を示しています。



この方法のメリットとデメリットは次の通りです。

○メリット

- 各設計者のローカルプロジェクトファイルに対して 1 名のみで作業を行うので、ネットワークの速度や同時利用人数に影響されず、快適な動作速度になる
- Enterprise Architect の標準のバージョン管理機能を利用するため、バージョン管理に関する機能は、Enterprise Architect 内の右クリックメニューから実行可能で、設定が完了すれば Subversion(および TortoiseSVN) の操作は不要
- 利用するプロジェクトファイルの形式に制限がない
- プロジェクト全体ではなく一部のパッケージのみを共有対象にできる

○デメリット

- リポジトリの設定など、Subversion の利用開始に関係する基礎設定が必要となる
- それぞれのメンバーが Subversion に関するツール(TortoiseSVN)のインストールと設定を行う必要がある
- それぞれのパッケージに対して、同時に 1 名のみが編集できる
(他の設計者の編集作業が完了しコミットされるまで、他の設計者は編集できない)
- コミットされ、その内容を取得するまで、他の人の作業状況が見えない
- 異なるネットワークに属する設計者で利用する場合、Subversion に関連する通信がファイアウォールで

遮断されている場合がある

- リファレンス情報の扱いに注意が必要（第 8 章参照）

6 Proクラウドサーバでプロジェクトを共有

Pro クラウドサーバを利用する方法は、基本的にはプロジェクトファイルで共有する方法と似ています。ただし、Windows のファイル共有の仕組みではなく、HTTP や HTTPS の形式で通信するためファイアウォールなどの制約を受けづらく、分散環境での利用に適しています。多くの企業では、HTTP や HTTPS はファイアウォールを通して外部と通信できます。また、HTTPS での通信とすることで、通信内容が暗号化されます。

Pro クラウドサーバは有償版を利用することで追加の機能が利用できますが、このドキュメントの内容は無料で利用できる範囲のみを利用します。

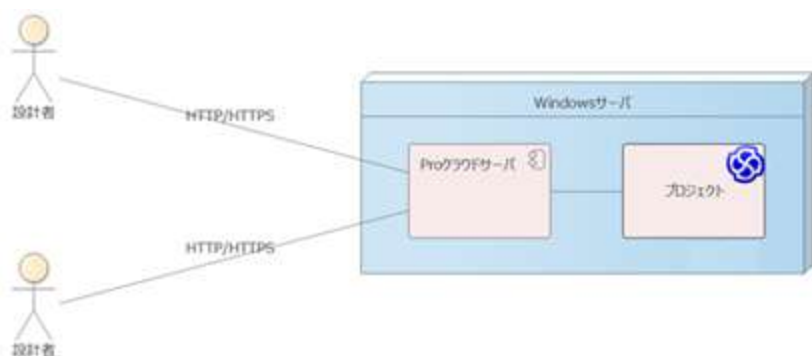
なお、Pro クラウドサーバは「クラウド」という名前が付いていますが、Windows マシンであればどのマシンにでもインストール可能です。（Amazon Web Services や Microsoft Azure のようなパブリッククラウドサービスだけでなく、プライベートクラウド・Windows ホスティングサービスなどでも利用できます。ローカルネットワークの Windows マシンにもインストールできます。）

この方法を実行するまでの手順は次の通りです。

1. Pro クラウドサーバのインストーラをダウンロードし、サーバとして利用する Windows マシンにインストールする
2. Pro クラウドサーバが利用するポートに対して、外部と通信可能にする
3. Pro クラウドサーバの設定ツールを利用し、プロジェクトを追加する
4. 各設計者の Enterprise Architect から、Pro クラウドサーバに接続する
（サーバ名あるいは IP アドレス・ポート・ユーザー名・パスワード・プロジェクト名を入力する）

Pro クラウドサーバのインストールや設定は、ドキュメント「Pro クラウドサーバの利用のための手順書」をご覧ください。

次の図は、この方法の構成の概要を示しています。



この方法のメリットとデメリットは次の通りです。

○メリット

- HTTP/HTTPS での通信が可能のため、通信が遮断されず場所に依存せず利用できることが多い
- HTTPS での通信を利用する場合には、通信内容が暗号化される
- 通信内容は圧縮されるので、環境によってはプロジェクトファイルの共有より通信速度が早い
- それぞれの設計者は、自分のマシンに対する設定や他ツール等のインストールは不要
- 他の人の作業状況が見える

○デメリット

- サーバのセットアップが必要 (Windows 環境が必要)
- 動作性能は通信速度に大きく影響を受ける
- ある程度の人数・規模になると、サーバの性能も影響する
 - 人数・規模が大きくなる場合には、バックエンドのプロジェクトとしてファイル形式ではなく、MySQL などの DBMS を利用することを推奨
 - ただし、各設計者は Enterprise Architect コーポレート版以上のエディションが必要
 - バックエンドに DBMS を利用する場合でも、各設計者からの利用方法は変わらない
- 同じ要素に対して複数名が同時に編集した場合、いわゆる「後勝ち」になり、先に編集した人の編集内容は上書きされる

(注意点などの補足事項は、プロジェクトファイルで共有する場合と同じです。)

なお、Enterprise Architect では、サーバに MySQL などの DBMS をインストールし、その DBMS のデータベースにプロジェクトを配置して共有する方法もあります。Pro クラウドサーバを提供する前は、DBMS を利用する方法はある程度の規模での Enterprise Architect の利用の場合の基本的な方法でした。現在でも、Enterprise Architect ではこの方法での利用にも対応しています。Pro クラウドサーバを利用する方法と DBMS を利用する方法は基本的には大きな差はありませんが、以下の違いがあります。

- Pro クラウドサーバはサーバに対して HTTP/HTTPS で接続するのに対して、DBMS を利用する場合には、それぞれの DBMS ごとのプロトコルで通信する。外部との通信では、これらの DBMS ごとの通信の

ためのポートは通信不可となっていることが多い。

(例: MySQL であればポート 3306 を利用することが多いが、多くの企業ではポート 3306 で外部とは通信できない)

- Pro クラウドサーバは、サーバと Enterprise Architect との間の通信内容を圧縮して送受信するため、通信速度が速くない環境では、DBMS を利用する場合に比べて動作速度が上がることが多い。一方で、ローカルネットワークの場合には、通信内容の圧縮・展開の処理がない分、DBMS を利用の方が早くなる場合がある。
- Pro クラウドサーバには、独自の接続制御やログの記録の仕組みなどがあり、問題発生時にもログから状況や問題点を推測しやすい。DBMS のログは、Enterprise Architect からの利用を前提としていない一般的な内容のため、状況や問題点の把握には Enterprise Architect の内部動作の理解が必要。

現在は、多くの場合には Pro クラウドサーバを利用することを推奨しますが、ローカルネットワーク内でプロジェクトを共有する場合など、場合によっては DBMS リポジトリに直接接続することも検討する価値があります。DBMS リポジトリの利用については、ドキュメント「DBMS リポジトリ 機能ガイド」もご覧ください。

7 LemonTree + Git/Subversionで分散設計

最近は、Git を利用したソースコードの分散管理・分散開発が広まってきています。ソースコードと同様に、設計情報についても Git を利用して分散環境で行いたいという声は、以前に比べると増えてきています。Enterprise Architect の標準機能では Git に対応していませんが、有償の別売り製品 LemonTree を利用することで、Git を利用した分散設計が可能です。

(LemonTree は Subversion を利用した分散設計にも対応します。Subversion の場合も、このドキュメントで説明する内容と同じ方法で利用できます。)

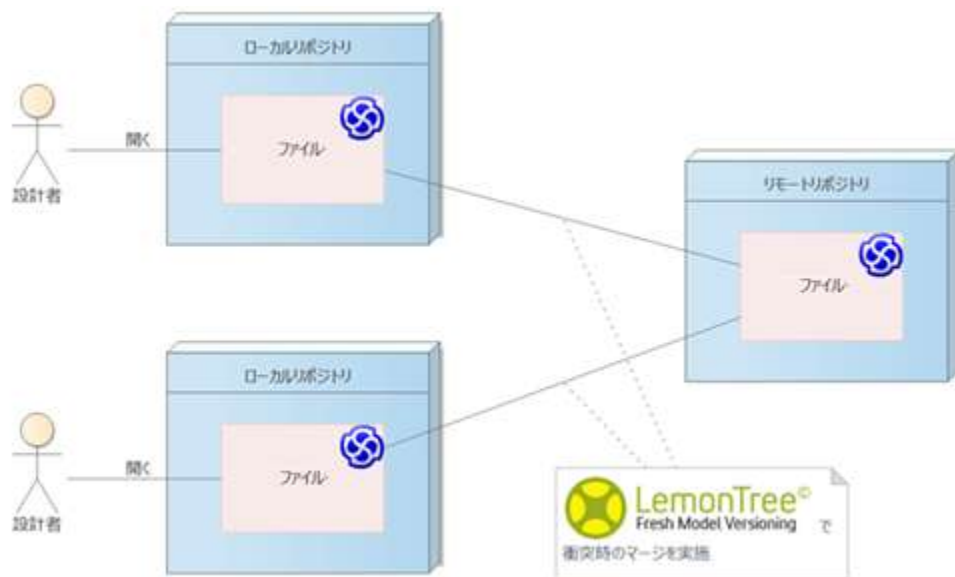
この方法を実行するまでの手順は次の通りです。なお、手順 2 までの内容は、Enterprise Architect に依存しない一般的な Git の使い方となります。手順 2 までの内容が不明な場合には、Web サイトでの検索などで情報を入手してください。

1. Git のリポジトリ環境を構築する
2. 各利用者は TortoiseGit をインストールし、利用できるようにする
3. 各利用者は LemonTree をインストールし、利用できるようにする
(TortoiseGit がインストールされている場合、LemonTree のインストール時に TortoiseGit との連携機能が自動的にインストールされる)
4. チームのうちの 1 名が、Enterprise Architect のプロジェクトファイルを新規作成し、パッケージなどの構成を作成する。そのプロジェクトファイルを Git のリポジトリに追加する
5. ソースファイルの場合と同様に、管理されているプロジェクトファイルを編集する

プッシュ・プルの段階で対象のプロジェクトファイルが編集されている場合には、TortoiseGit を通して自動的に

LemonTree が起動します。自動的にマージを行い、衝突がない場合にはマージ後の内容を編集できます。衝突がある場合には LemonTree の画面が表示され、どのようにマージをするか指定する必要があります。

次の図は、この方法の構成の概要を示しています。



この方法のメリットとデメリットは次の通りです。

○メリット

- Git を利用する場合では、Git の思想に合致した分散・同時並行での設計が可能
- ローカルのプロジェクトファイルに対して1名のみで作業を行うので、ネットワークの速度や同時利用人数に影響されず、快適な動作速度になる
- TortoiseGit/TortoiseSVN と連携するので、マージは自動的に行われる
(マージ時に自動的に LemonTree が起動し、衝突がある場合には LemonTree の画面が開く)

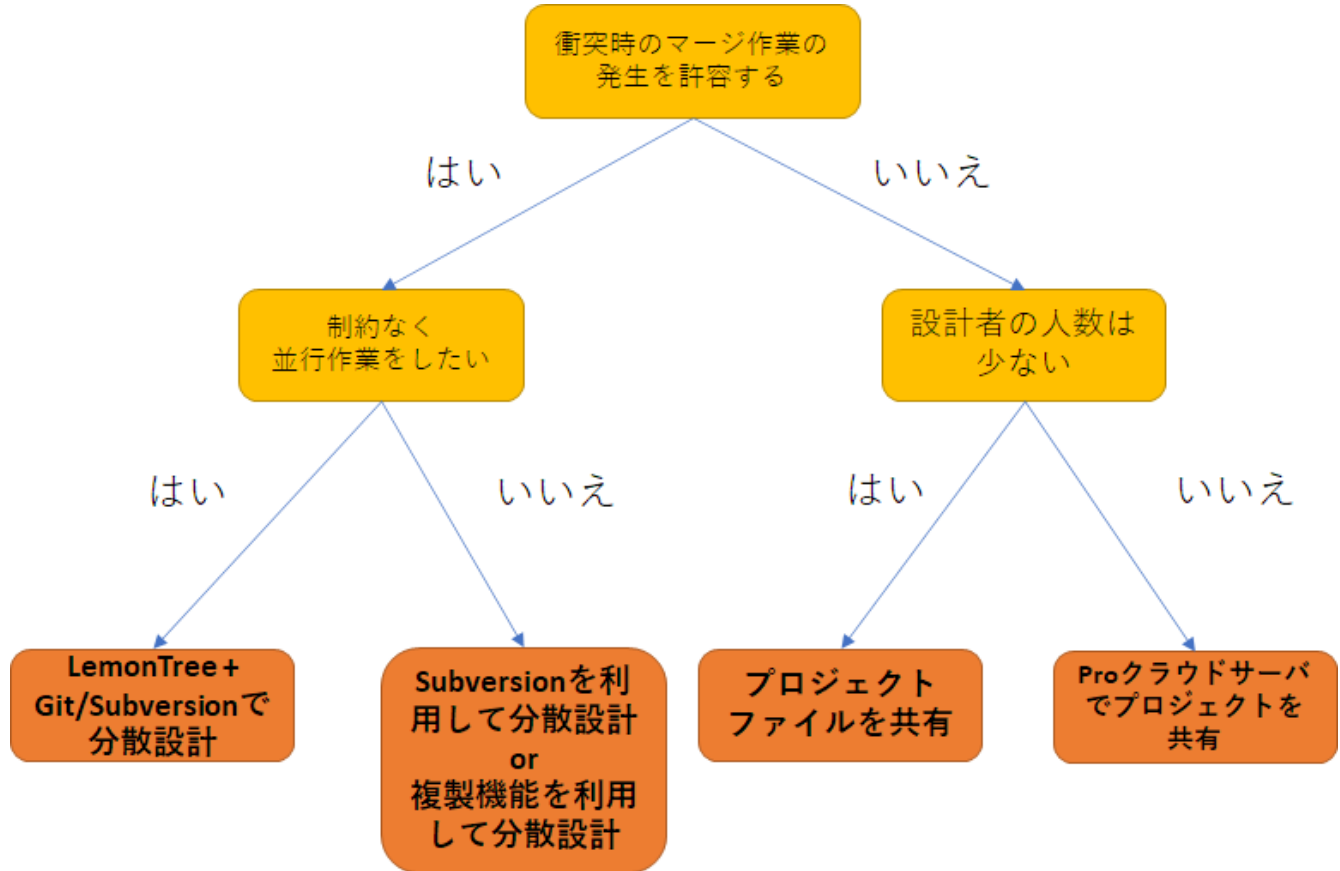
○デメリット

- LemonTree を購入する必要がある
- Git/Subversion 環境を構築する必要がある
- 他の人の作業状況が見えない
- バイナリファイルであるプロジェクトファイルを直接管理対象にするので、バージョン管理ツールのリポジトリが肥大する

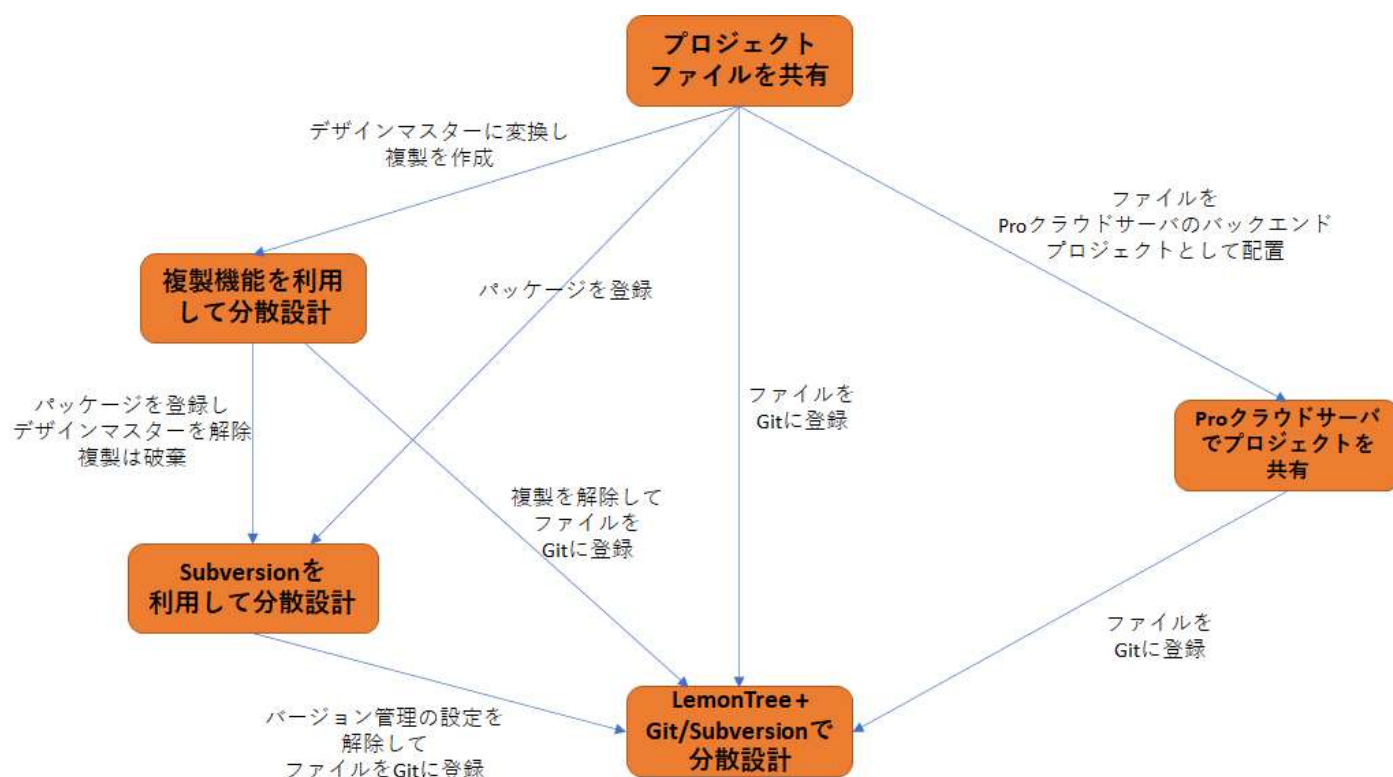
8 方式を選択するためのヒント

ここまで、5 つの方式を紹介しました。どの方式を採用するのかを検討する際のヒントになるチャートを作成し

ました。



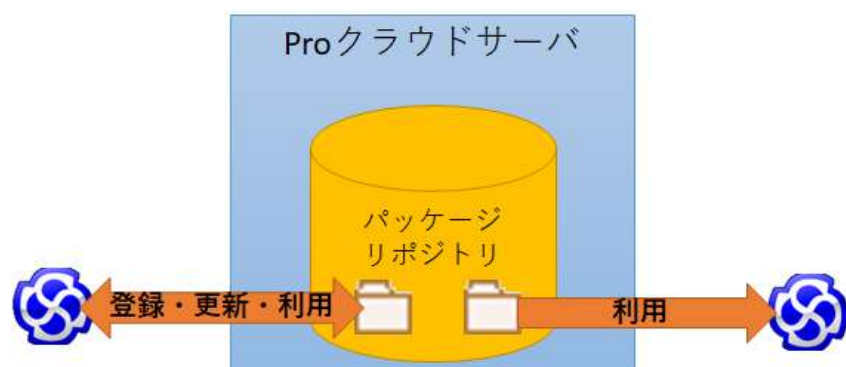
また、ここまで紹介した 5 つの方法は、途中で別の方法に切り替えることも可能です。実際の発生する可能性が高い切り替えについての概要は以下の通りです。



9 パッケージリポジトリ

ここまでに説明した方法とは異なる観点の機能として、パッケージリポジトリ機能があります。この機能は、指定したパッケージを Pro クラウドサーバに保存し、保存されているパッケージをプロジェクトに取り込みます。つまり、Pro クラウドサーバをストレージとして、パッケージの内容を保存し共有する仕組みです。

パッケージリポジトリは、比較的変更が少なく、かつ多くの人やさまざまなプロジェクトで共有されるパッケージを、パッケージ単位で共有する方法として便利です。具体的には、共通部品・ライブラリなどをパッケージ単位でパッケージリポジトリに保存し、プロジェクトの開始時などにパッケージリポジトリから必要なパッケージを取り込む、という流れが可能になります。



パッケージリポジトリは、パスワードを設定することで登録および更新を特定の設計者のみに限定できます。登録されているパッケージの内容を更新することも可能であり、その場合にはそれぞれの利用者は、パッケージを再度取得し、同期更新できます。

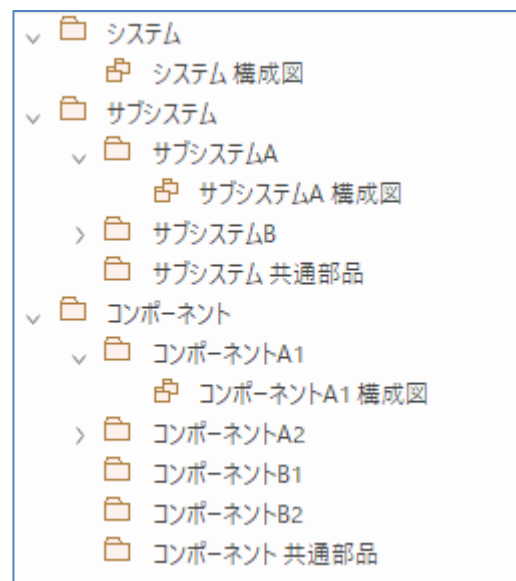
パッケージリポジトリの詳細や具体的な設定方法については、ヘルプをご覧ください。

10 複数プロジェクトの利用

この章では、応用編として複数のプロジェクトを利用する際の考え方やヒントを紹介します。

このドキュメントでは、5 つの主な利用方法を紹介しましたが、それぞれに長所と短所があります。例えば、LemonTree を利用する方法では、プロジェクトファイルが管理対象のためわかりやすく、また Git の思想に合った運用が可能です。一方で、常にプロジェクトファイル全体が管理対象となるため、プロジェクトファイルが 1 つの場合にはそれぞれの設計者と直接関係がない内容も含まれ、ある意味で必要のない情報も管理対象となっていると言えるかもしれません。

一例として、ドキュメント「階層化した設計を実現するための機能ガイド」で紹介している以下のモデルブラウザの構成を考えます。



この例で、例えばコンポーネント A1 の設計担当者は、「コンポーネント A1」と、上位となる「サブシステム A」および共通部品の「コンポーネント共通部品」「サブシステム共通部品」のみが必要となります。プロジェクト全体は不要です。

こうした場合には、以下のような方法で必要な部分のみを持つプロジェクトを作成・利用することで、関係がな

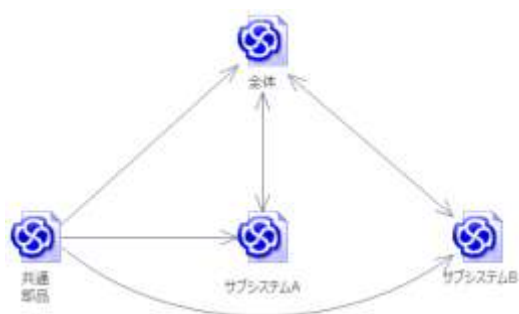
い部分を誤って編集・削除することがなくなり、またプロジェクトのサイズを減らすことができます。

- 必要なパッケージのみを XML ファイルなどで出力し、それぞれの利用者のプロジェクトファイルに取り込む
 - シンプルな方法だが、手順を誤ると既存の内容の削除や古い内容での上書きにつながる
 - コーポレート版以上で利用できるベースライン機能を利用すると、取り込み時の操作の簡易化や簡易的な差分比較が可能
 - 複数の利用者の同時編集が可能だが、この方法の場合には差分比較・マージの調整が困難 (LemonTree を利用すれば差分比較・マージは可能だが、誰と誰のファイルをどのような順序でマージしていくのか、等の調整が困難)
- バージョン管理機能でそれぞれのパッケージを管理対象にするのと同時に、それぞれの利用者が必要なパッケージのみを取得し利用する
 - 更新内容の同期などは Enterprise Architect の機能を利用できるため、同期操作の失敗がない
 - 1 つのパッケージにつき、1 名のみが編集可能のため並行作業はできない
- LemonTree 付属の MPMS (Model Package Management System) を利用する
 - MPMS の概要は以下の URL をご覧ください。
https://www.sparxsystems.jp/LemonTree/change_v30_features.htm
 - 操作をグラフィカルな画面で行うことができるため、取得や更新の手順を間違えにくい
 - パッケージの依存関係を自動で解析し、取り込み時に考慮してくれる



- 同時編集で衝突が起きた場合、LemonTree が自動的に起動して解決可能
- MPMS のファイル自体の履歴は保持されないため、履歴を保持するには MPMS ファイル自体と MPMS の出力元のプロジェクトをバージョン管理する必要がある

前述のモデルブラウザの構成の場合に、複数プロジェクトで構成の一例は以下のようになります。



最初は1つのプロジェクト(「全体」)で、「システム」の内容を中心に設計します。要求などをサブシステム単位に割り当てができた段階で、サブシステムAおよびサブシステムBのプロジェクトファイルを作成し、それぞれ関係する内容の設計を進めます。

サブシステム間で共通の部品は別プロジェクトで定義・管理し、それぞれのプロジェクトに読み込みます。それぞれのプロジェクトでの共通部品の編集を禁止することで、共通部品の差分比較やマージの手間を削減できます。(MPMSを利用する場合には「Read Only」のコンポーネントとする。バージョン管理機能の場合には、共通部品のプロジェクトでチェックアウト状態のまま保持すれば、他の人がチェックアウトできなくなる。)

それぞれのサブシステムのプロジェクトにおける設計内容は、適宜「全体」のプロジェクトに読み込み、全体としての整合性やトレーサビリティの確認を行います。

この説明は一例ですが、「作業用のプロジェクト」「全体を参照するためのプロジェクト」「共通部品を管理するプロジェクト」のような形で分割し、できるかぎりモデルの読み込みを一方向とする(定義するプロジェクトと利用するプロジェクトを分ける)ことで、モデルの内容の衝突を避けられます。

さらに、このドキュメントで紹介した方法を組み合わせる、例えば上記の分割したプロジェクトのそれぞれをバージョン管理したり LemonTree + Git で管理したりするなどの方法で、それぞれの方法の価値を得られます。

11 その他の注意点

以下、ここまでに説明した内容の補足や注意事項です。

- ネットワークを経由してプロジェクトを利用する場合には、通信量を減らすことが動作速度の改善に大きく影響します。特に、多くのアドインが有効になっている場合には、アドインが原因で動作速度の劣化が発生する場合があります。利用していないアドインが有効になっている場合がありますので、アドインは常に利用するもののみ利用・有効にすることを強くお勧めします。
(アドインは、Enterprise Architect 内での様々な操作について、自分のアドインが対象かどうかのチェッ

クを行う場合があります。通信速度が遅い場合には、このチェック処理で全体的な動作速度の低下として現れる場合があります。Enterprise Architect 内の操作には、ダイアグラム内での要素の選択など、ほぼすべての操作が含まれます。)

- プロジェクトには、「モデル」と呼ばれる、パッケージ・ダイアグラム・要素などの設計を構成する情報(モデルブラウザに表示される情報)の他に、プロパティの選択肢など「リファレンス情報」と呼ばれるさまざまな設定情報が含まれます。第 5 章の Enterprise Architect 標準のバージョン管理機能を利用する方法は、バージョン管理される対象はモデルの内容のみで、リファレンス情報は管理の対象外です。

具体的には、例えば要素のプロパティの「状態」の選択肢をカスタマイズした場合に、そのカスタマイズ結果は他の設計者には反映されません。ソースコードやドキュメントのテンプレートもリファレンス情報の一部です。

リファレンス情報は、XML 形式で入出力できます。リファレンス情報をカスタマイズした場合には、他の設計者に出力した XML ファイルを配布し、それぞれの設計者が読み込む必要があります。