How to use SQL Query in Enterprise Architect

by SparxSystems Japan

Enterprise Architect 日本語版

SQL クエリの活用 機能ガイド

(2022/03/11 最終更新)

目次

1 はじめに 3
2 概要の把握 4
3 Enterprise Architect のプロジェクトの内部形式 6
3.1 利用できる形式 6
3.2 情報が格納されている位置の調べ方 7
3.3 クエリ定義時の注意点 8
4 利用できる機能とその利用例 8
4.1 検索機能 8
4.2 ドキュメント生成テンプレート 10
4.3 カンバン 11
4.4 一覧オブジェクト 11
4.5 チャート 12
5 API からの利用 13

Copyright (c) 2021-2022 Sparx Systems Japan

1　はじめに

Enterprise Architect で作成する内容は、EAPX ファイルなどのプロジェクトに格納します。このプロジェクトは独自形式ではなく、データベースになっています。例えば、バージョン 16.0 以降での標準形式となる QEA ファイルは、SQLite 3 形式のデータベースファイルです。それ以前のバージョンでの標準形式である EAPX ファイルは、Microsoft Access が利用している MS-JET データベースエンジンのファイルです。そのため、Access がインストールされているマシンであれば、拡張子を eapx から mdb に変更することで、Access で Enterprise Architect のプロジェクトファイルを開き、内容を直接参照することができます。

プロジェクトがデータベースになっているということは、SQL のクエリで情報を取得することが可能ということでもあります。例えば、「SELECT ＊ FROM t_object」のクエリを実行することで、そのプロジェクトに含まれる要素の情報を取得することができます。このような仕組みがありますので、作成した内容を活用したり分析したりする場合には、この仕組みを活用することもできます。Enterprise Architect の内容を取得するための手段として、API や XMI ファイル出力などの方法がありますが、直接参照する方法も活用できることは、他のツールと比較した際の大きなメリットです。また、いくつかの機能では、処理内容に SQL のクエリを設定することも可能です。

このドキュメントでは、こうしたクエリについての基本的な知識と利用できる機能を紹介します。

なお、このドキュメントで説明している範囲については、標準サポートの範囲外となります。範囲外とはなりますが、それぞれのお客様ご自身での責任での利用のための情報として提供します。特に、以下の点にご注意ください。

- UPDATE 文など、SELECT 文以外の処理を実行しプロジェクトの内容を変更する場合には、内部の情報の一貫性が失われるなどで、正しく動作しなくなる可能性があります。
  特に、不適切・不整合となるデータの追加や編集は、Enterprise Architect が正しく動作しなくなる可能性があるだけでなく、プログラムの不正終了など致命的な問題を引き起こす可能性があることにご注意ください。
  SELECT 文以外を利用しデータの編集が行われたプロジェクトは、その編集に関係がない場合・機能だとしても、サポートの対象外となります。
- Enterprise Architect の内部のデータ構造や含まれるデータの内容・位置などは、さまざまな理由で予告なく変更される可能性が十分にあります。
  (新しい機能への対応・対応する記法の仕様の更新への対応・バグ修正等)

同様に、このドキュメントの内容および以下のような質問も標準サポートの範囲外となり、サポート窓口では回答できません。

- 特定の情報がどこに格納されているのか
- テーブルに格納されている情報の意味

Copyright (c) 2021-2022 Sparx Systems Japan

なお、「有償追加サポート」を利用することで、上記の制限を解除することも可能です。

有償追加サポート: https://www.sparxsystems.jp/advanced.htm

なお、スパークスシステムズ ジャパンで公開しているヘルプやドキュメントには、この SQL クエリに関して記載している内容もあります。公開している内容についての不明点や問題点などがあればサポートにご連絡いただけますと助かります。これらの内容には不備・不足などがある可能性があり、ご指摘に応じて修正・改善いたします。

SELECT 文を利用して作成したモデルの内容を取得するだけであれば、プロジェクトの内容の編集や破壊を引き起こすことはありません。API を利用して情報を取得するよりも高速であることが多く、特に大規模なプロジェクトに対する情報取得には効果的です。

上記に記載の注意事項を十分に頭に入れて、うまく活用してください。

2 概要の把握

このドキュメントで説明する内容について、以下の内容を試してみてください。このドキュメントでお伝えする、Enterprise Architect の特徴的な仕組みの概要を把握できるのではないかと思います。

まず、Enterprise Architect を起動し、拡張子が QEA のプロジェクトファイルを開いてください。該当する適切なプロジェクトがない場合、「ホーム」リボン内の「ヘルプ」パネルにある「ヘルプ」ボタンを押すと表示されるメニューから「サンプルプロジェクトを開く」を実行してください。(バージョン 16.0 でのサンプルプロジェクトは、Enterprise Architect のインストールディレクトリにある EAExample.qea です。)

次に、Ctrl+F キーで、モデルの検索タブを開きます。(Ctrl+F は検索機能を呼び出すためのショートカットキーです。)

開いたら、ツールバーにある「エディタ」のボタンを押します。(下の画像の赤枠)

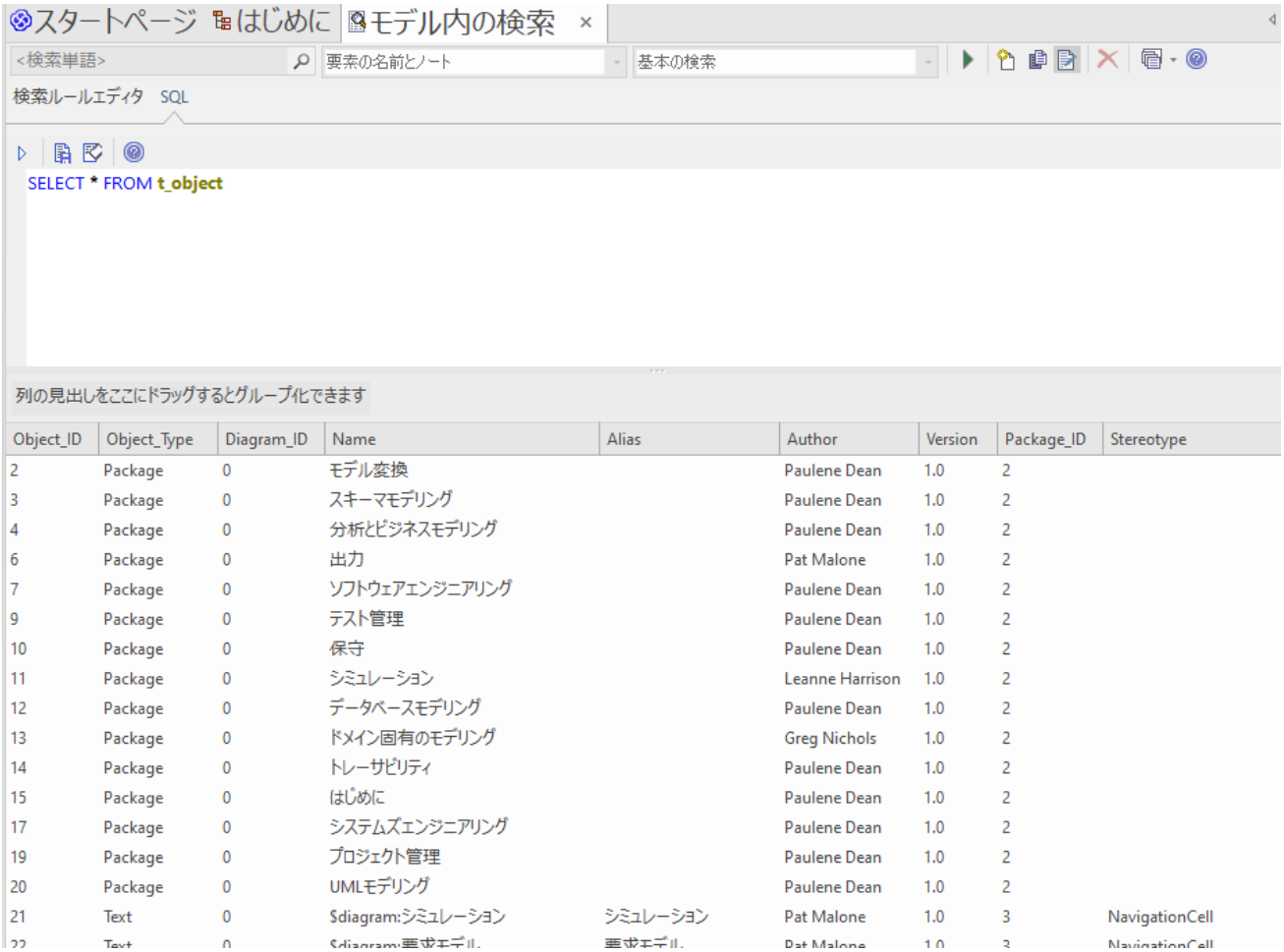
Copyright (c) 2021-2022 Sparx Systems Japan

検索ルールのエディタが開きますので、「SQL」タブを開きます。(次の画像の赤枠)

そして、エディタ部分に、以下のように入力後、実行ボタン(前の図の緑枠)を押してください。

```
SELECT * FROM t_object
```

すると、エディタ部分の下の一覧に、以下のように、開いているプロジェクトに存在する全ての要素の情報が表示されます。



Object_ID	Object_Type	Diagram_ID	Name	Alias	Author	Version	Package_ID	Stereotype
2	Package	0	モデル変換		Paulene Dean	1.0	2	
3	Package	0	スキーマモデリング		Paulene Dean	1.0	2	
4	Package	0	分析とビジネスモデリング		Paulene Dean	1.0	2	
6	Package	0	出力		Pat Malone	1.0	2	
7	Package	0	ソフトウェアエンジニアリング		Paulene Dean	1.0	2	
9	Package	0	テスト管理		Paulene Dean	1.0	2	
10	Package	0	保守		Paulene Dean	1.0	2	
11	Package	0	シミュレーション		Leanne Harrison	1.0	2	
12	Package	0	データベースモデリング		Paulene Dean	1.0	2	
13	Package	0	ドメイン固有のモデリング		Greg Nichols	1.0	2	
14	Package	0	トレーサビリティ		Paulene Dean	1.0	2	
15	Package	0	はじめに		Paulene Dean	1.0	2	
17	Package	0	システムズエンジニアリング		Paulene Dean	1.0	2	
19	Package	0	プロジェクト管理		Paulene Dean	1.0	2	
20	Package	0	UMLモデリング		Paulene Dean	1.0	2	
21	Text	0	$diagram:シミュレーション	シミュレーション	Pat Malone	1.0	3	NavigationCell
22	Text	0	$diagram:要求モデル	要求モデル	Pat Malone	1.0	3	NavigationCell

Copyright (c) 2021-2022 Sparx Systems Japan

先ほど実行した内容の「t_object」を「t_diagram」「t_package」「t_connector」に変更することで、それぞれ、ダイアグラム・パッケージ・接続の情報を得ることができます。一覧部分の項目をクリック後、Ctrl+A で全て選択し、Ctrl+C でコピーすることで、表示している内容を CSV 形式のテキストでクリップボードにコピーできます。コピーした内容を Excel などで編集することで、モデル内の情報を簡単に取得できます。

このように、Enterprise Architect のプロジェクトの実体は、SQL を実行可能なデータベースになっています。つまり、Enterprise Architect で作成した全ての情報は、このような SQL 文を実行することでも取得することができます。Enterprise Architect で作成した情報を取得する方法として、API を利用してプログラムを作成する等の方法がありますが、情報を取得するだけであれば、このような SQL を作成・実行することで、必要な情報を素早く得ることができます。

3 Enterprise Architectのプロジェクトの内部形式

3.1 利用できる形式

Enterprise Architect では、利用できるプロジェクトの形式として、以下のようなさまざまな選択肢があります。

- ファイル形式
  - 拡張子: EAP・EAPX
    →Microsoft JET データベース形式（Microsoft Access の mdb と同じ）
  - 拡張子: FEAP
    →Firebird 形式
  - 拡張子: QEA・QEAX
    →SQLite 3 形式
  - 拡張子: EADB
    →Access2007 以降の ACCDB 形式
- データベース形式（コーポレート版以上で利用可能）
  - Microsoft SQL Server
  - MySQL
  - MariaDB
  - PostgreSQL
    (スパークスシステムズ ジャパンとしては利用非推奨)
  - Oracle
    (スパークスシステムズ ジャパンとしては利用非推奨)

上記のようにさまざまな形式が利用可能となっていますが、いずれも SQL が利用できるリレーショナルデータベースです。そのため、「SELECT * FROM t_object」のような SQL クエリを実行し、Enterprise Architect で作成したモデルの情報を取得することが可能です。

Enterprise Architect の機能によっては、上記に記載している形式以外での情報の出力も可能です。例えば、プロジェクトの内容を XML 形式で出力することも可能です。ただし、このドキュメントでは、Enterprise Architect がプロジェクトとして扱うことができる形式についてのみ対象とします。

Copyright (c) 2021-2022 Sparx Systems Japan

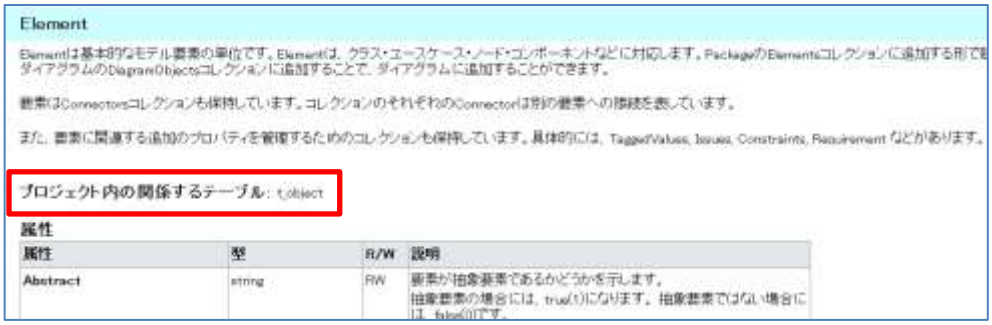
3.2 情報が格納されている位置の調べ方

第 1 章で説明したように、「特定の情報がどこに格納されているのか」「テーブルに格納されている情報の意味」のような質問は、標準サポートの範囲外となります。そのため、例えば「要素の別名を取得したいが、どこに格納されているのか」「t_object テーブルの PDATA1 列はどのような意味があるのか、どのように使われるのか」というようなご質問には標準サポートの範囲では対応できません。

しかし、実際にはこれらの情報をある程度理解しないと、必要とする情報を取得するための SQL クエリを作成することはできません。ここでは、その情報を自分自身で得るためのヒントを紹介します。

まず、API の構成およびヘルプが参考になります。API は、内部のデータベース構造と類似した構造になっている部分があります。例えば、要素の名前などの情報は Element クラスの属性となっていますが、要素のタグ付き値は TaggedValue クラスで、別クラスになっています。同様に、Element クラスの属性の情報は t_object テーブルに、TaggedValue クラスの属性の情報は t_objectproperties テーブルに格納されているというように類似の構成になっています。

API のヘルプには、「関係するテーブル」としてテーブル名が記載されている場合があります。例えば、Element クラスのページには以下のような情報があります。


Element

Elementは基本的なモデル要素の単位です。Elementは、クラス・ユースケース・ノード・コンポーネントなどに対応します。PackageのElementsコレクションに追加する形でもダイアグラムのDiagramObjectsコレクションに追加することで、ダイアグラムに追加することができます。

要素はConnectorsコレクションも保持しています。コレクションのそれぞれのConnectorは別の要素への接続を表しています。

また、要素に関連する追加のプロパティを管理するためのコレクションも保持しています。具体的には、TaggedValues, Issues, Constraints, Requirement などがあります。

プロジェクト内の関係するテーブル: t_object

属性

属性	型	R/W	説明
Abstract	string	RW	要素が抽象要素であるかどうかを示します。 抽象要素の場合には、true(1)になります。抽象要素ではない場合には、false(0)です。

これにより、Element クラスの属性(Name や Alias など)は、t_object テーブルに格納されている可能性が高いことがわかります。あとは、実際のテーブルの内容を見ることで、情報が保存されている位置を推測することができます。

別の方法として、知りたい情報だけを持つプロジェクトファイルを作成し、実際にその中身を探索するという方法もあります。例えば、要素の別名がどこに格納されているかを知りたい場合には、新規にプロジェクトファイルを作成して 1 つの要素のみを作成し、別名に探しやすい文字列を設定します。(なお、この例では、モデルルートの直下には要素は作成できないため、パッケージの作成も必要です。)

その上で、先ほどの検索機能を利用したりプロジェクトファイルの拡張子を変更して関係するツールで開いたりして、設定した文字列がどのテーブルのどの列に格納されているかを探すことで、格納位置を知ることができます。

なお、海外では、Enterprise Architect のユーザーがプロジェクトの内容を実際に解析・推測してまとめて電子書籍を発行しています。内容として完全ではなく、誤りがないとも言えませんが、参考になるかと思います。

https://leanpub.com/InsideEA

Copyright (c) 2021-2022 Sparx Systems Japan

3.3 クエリ定義時の注意点

SQL は、対象のデータベースによりさまざまな「方言」があります。例えば、いわゆるワイルドカードでの部分一致検索の場合、EAPX ファイル(JET データベース)では、「Name Like ‘Hoge*’」となりますが、FEAP ファイル(Firebird)では、「Name Like ‘Hoge%’」となり、ワイルドカードを示す文字が異なります。そのため、次の章で説明する機能について SQL クエリを利用する場合には、例えば EAPX ファイルでは問題ないが FEAP ファイルではエラーになる、というような問題が発生する可能性があります。

このような差異を吸収するために、Enterprise Architect はいくつかのマクロを提供しています。例えば、前述の例であれば、「Name Like ‘Hoge#WC#’」とマクロ#WC#を利用することで、どちらの場合にも対応できます。

詳細は、ヘルプ「検索ルールの新規作成」をご覧ください。ただし、ヘルプに記載のように、このマクロが利用できない機能もありますので、ご注意ください。

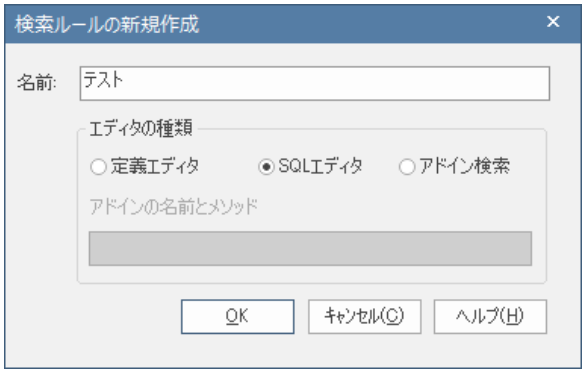
4 利用できる機能とその利用例

Enterprise Architectには、いくつかの機能でSQLクエリを利用できます。その目的と使い方についての概要を説明します。

4.1 検索機能

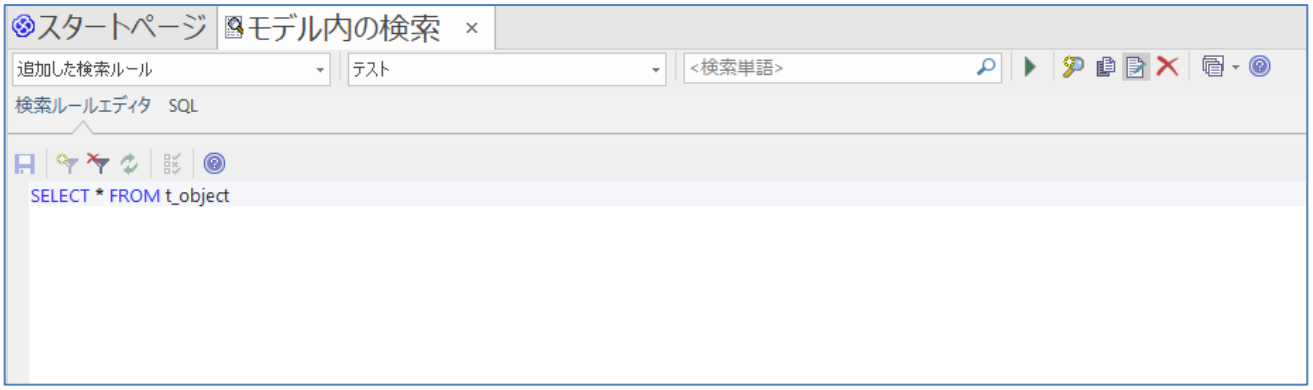
Enterprise Architectの検索機能を利用すると、さまざまな条件を指定して、条件に該当する要素を発見することができます。この条件の指定には専用のエディタがあり、状態やバージョンなど、要素が持つさまざまなプロパティに対して条件を指定することができます。

しかし、このエディタは、例えば「状態が実装済」のようなシンプルな条件のみ指定できます。そのため、例えば「インターフェース要素と実現の関係で結ばれているクラス要素」のような複雑な条件を指定することはできません。このような場合には、「SQLエディタ」の検索ルールを作成します。



「SQLエディタ」を選択した場合には、検索する内容を取得するためのSQLを直接記入する必要があります。下の画像はその一例です。(第2章で説明した内容と同じエディタです。)

Copyright (c) 2021-2022 Sparx Systems Japan



なお、この場合には、Enterprise Architect独自のマクロを利用することができます。例えば、モデルブラウザで選択されているパッケージ(およびその子パッケージ)に含まれている要素のみを対象としたい場合には、マクロ#Branch#を利用します。

```
SELECT Name FROM t_object WHERE Package_ID IN (#Branch#)
```

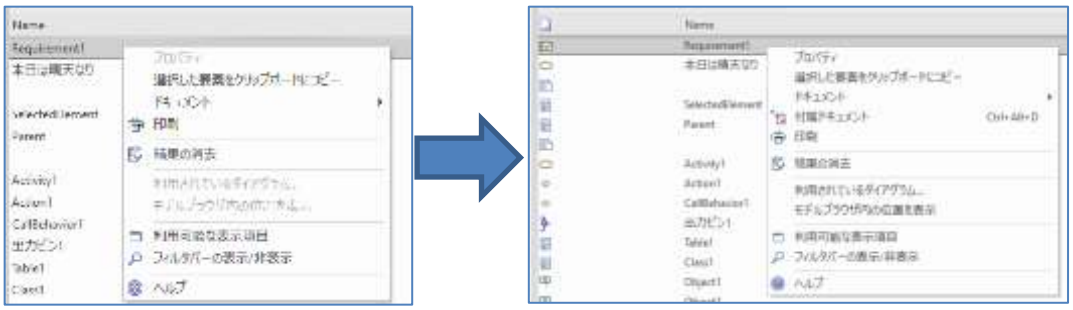
利用できるマクロの詳細は、ヘルプ「検索ルールの新規作成」をご覧ください。

なお、モデル内の検索タブ内のエディタを表示し、「SQL」タブに切り替えることで、SELECT文を実行しその結果を得ることができます。目的のSQLクエリを得るために試行錯誤する必要がある場合には、このSQLタブを利用することをお勧めします。

また、検索結果の一覧からEnterprise Architectが提供するさまざまな機能(例: 利用されているダイアグラムに移動)するためには、ヘルプに記載のようにCLASSGUIDとCLASSTYPEの情報を含める必要があります。上記のSQLを、検索機能で適切に動作するようにするためには、以下の赤枠部分のようにEnterprise Architectが対象を識別するための情報を含める必要があります。

```
SELECT ea_guid AS CLASSGUID, Object_Type as CLASSTYPE, Name FROM t_object WHERE Package_ID IN (#Branch#)
```

上記の赤枠の内容がない場合(左)とある場合(右)の検索結果の違いです。要素の種類を示すアイコンが表示されたり、「プロパティ」など右クリックメニューの機能が実行できるようになっていたりする点が異なります。



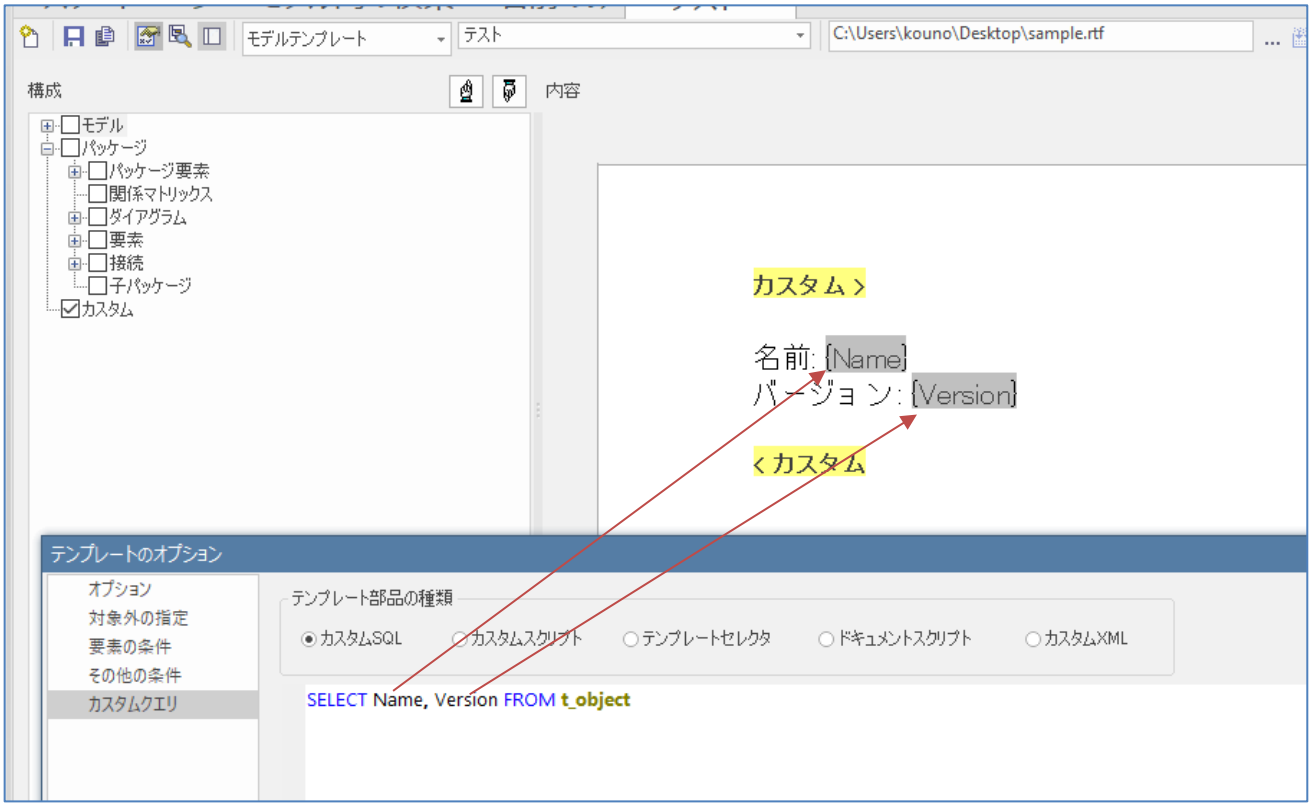
Copyright (c) 2021-2022 Sparx Systems Japan

4.2 ドキュメント生成テンプレート

Enterprise Architect で作成した内容を Word(docx)ファイルに出力するドキュメント生成機能でも、SQL クエリを利用することができます。利用できる場面は、大きく 2 つに分かれます。

- クエリを利用し、出力対象を絞り込む
- クエリで得られた結果をドキュメントに含めて出力する

このドキュメントでは、後者の内容について紹介します。この内容を実現するには、「テンプレート部品」を作成し、「カスタム」セクション内にカスタムフィールドを挿入する必要があります。下の画像で概要が把握できるかと思います。SELECT 文で得た結果と同じ名前のカスタムフィールドを、右クリックメニューの「カスタムフィールドの挿入」でエディタ内に追加することで、希望する内容を出力することができます。



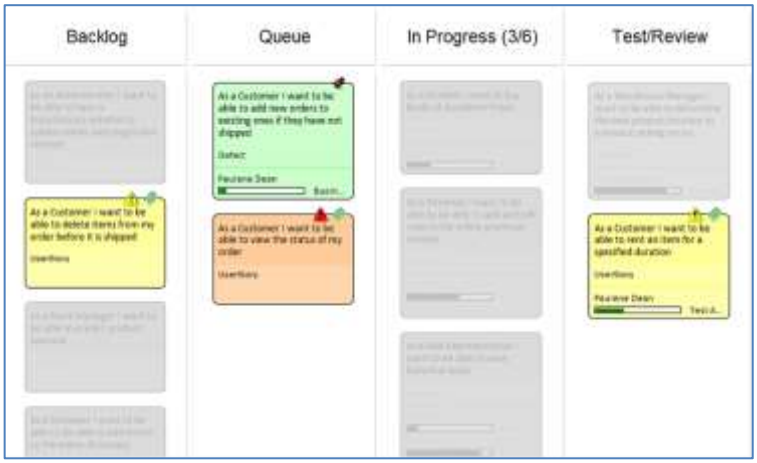
カスタムクエリを利用することで、複雑な条件を満たす要素のみを出力したり、Enterprise Architectの通常のフィールドには含まれない情報を出力したりすることができます。

Copyright (c) 2021-2022 Sparx Systems Japan

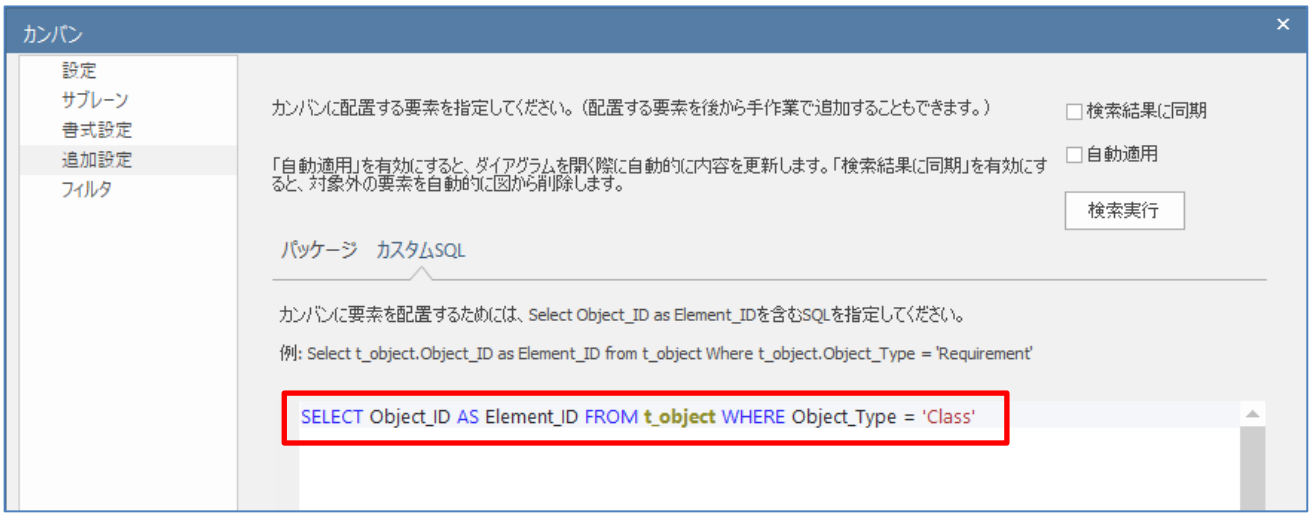
4.3 カンバン

Enterprise Architect では、ダイアグラムに配置される要素を、アジャイル開発などで利用されるカンバンの項目として利用することができます。このカンバン機能の詳細は、ヘルプや動画デモ「カンバン機能の紹介」をご覧ください。

動画デモ: https://www.sparxsystems.jp/products/EA/ea_demo.htm



このカンバン機能に配置する要素は、手作業で 1 つ 1 つ対象を指定する(カンバンのダイアグラムのドロップする)こともできますし、指定したパッケージ内の要素をまとめて配置することもできます。それ以外の方法として、SQL クエリで条件を指定して、条件に合致する要素をまとめて配置することができます。



4.4 一覧オブジェクト

一覧オブジェクトとは、指定した条件に合致する要素を一覧形式でダイアグラム内に表示することのできる要素で

Copyright (c) 2021-2022 Sparx Systems Japan

す。ツールボックスの「拡張要素」グループから作成できます。ステレオタイプ<<model view>>が付加された成果物要素です。

ModelView1

名前	状態
Class1	設計中
SelectedElement	設計中
Parent	設計中
Table1	Proposed
Class1	設計中
Concept B	Proposed
Concept A	Proposed
Concept C	Proposed
Table B	Proposed
Table C	Proposed

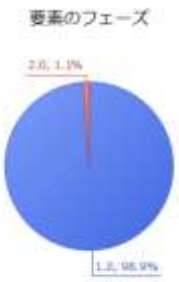
1番目から10番目までを表示中：19個の項目

この要素をダブルクリックすると表示されるプロパティ画面で、表示する要素を SQL で指定します。

一覧オブジェクト内の項目は、「操作モード」に切り替えることで内部の項目を右クリックし、いくつかの操作を実行することができます。SQL での条件を工夫することで、例えば ToDo リストのようにモデル内で何らかの条件を満たす要素がどの程度あるのか、あるとすれば何個か、などをダイアグラム内に可視化することができます。

4.5 チャート

チャート要素は、指定した条件に合致する要素について、円グラフや棒グラフなどでわかりやすく状況を可視化できます。チャート要素も、一覧オブジェクトと同じく表示対象の要素を SQL で指定することができます。



一覧オブジェクトと同様に、要素をダブルクリックすると表示されるプロパティ画面で、対象の要素を指定できます。パッケー

Copyright (c) 2021-2022 Sparx Systems Japan

ジを指定し、そのパッケージに含まれる要素を対象とするのが基本ですが、SQL で条件を指定することができます。

5 APIからの利用

このドキュメントで紹介する SQL クエリは、API からも利用することができます。API から利用することで、必要な情報の取得処理を高速化したり、API で未対応の情報を取得したりすることができるようになります。

API から実行する場合には、Repository オブジェクトの SQLQuery メソッドを実行します。以下は呼び出しの一例です。

repository.SQLQuery("SELECT Object_ID FROM t_object WHERE Object_Type = 'Class'");

実行した結果は、以下のような構造の XML 形式の文字列で返されます。<Row>が、クエリで指定された内容に合致する文だけ繰り返されます。(プロジェクトのデータベースの形式により、タグ名が全て大文字になる場合もあります。)

```
<EADATA>
    <Dataset_0>
        <Data>
            <Row>...</Row>
            <Row>...</Row>
            <Row>...</Row>
        </Data>
    </Dataset_0>
</EADATA>
```

XML 形式の情報を扱うことのできるライブラリはそれぞれのプログラム言語で提供されているかと思いますので、そうしたライブラリを利用することで、取得した情報を効率よく扱うことができるでしょう。

なお、Repository オブジェクトの SQLQuery メソッドの引数に、SELECT 文以外の SQL を指定することはできません。

Copyright (c) 2021-2022 Sparx Systems Japan