

PROMCODE

次世代プロジェクト管理データ交換アーキテクチャ協議会

適用ガイド

第 1 版

2013 年 10 月 22 日

南山大学
日本アイ・ビー・エム株式会社
富士通株式会社
日本電気株式会社
株式会社 NTT データ
株式会社日立製作所
株式会社野村総合研究所

Copyright © Nanzan University 2013 All rights reserved.

Copyright © IBM Corporation 2013 All rights reserved.

Copyright © FUJITSU LIMITED 2013 All rights reserved.

Copyright © NEC Corporation 2013 All rights reserved.

Copyright © NTT DATA CORPORATION 2013 All rights reserved.

Copyright © Hitachi, Ltd. 2013 All rights reserved.

Copyright © Nomura Research Institute, Ltd. 2013 All rights reserved.

本書は、本書に記載した要件・技術・方式に関する内容が変更されないこと、および出典を明示いただくことを条件に、無償でその全部または一部を複製、翻訳、転載、引用および公衆送信することができます。なお、全体を複製、翻訳、転載または公衆送信する場合は、本書にある著作権表示を明示してください。

本書の著作権者は、本書の内容に関して、その正確性、完全性その他一切を保証するものではなく、その利用等により生じた損害について、法律上の構成のいかんを問わずいかなる責任も負いません。

Eclipseは、開発ツールプロバイダのオープンコミュニティであるEclipse Foundation, Inc.により構築された開発ツール統合のためのオープンプラットフォームです。

OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。

Microsoft, Windows, Microsoft Office および Excel は Microsoft Corporationの米国およびその他の国における商標です。

その他、記載されている会社名、商品名、またはサービス名等は、各社の登録商標、または、商標である場合があります。

目次

1	はじめに.....	4
1.1	本書について.....	4
1.2	想定利用ケース.....	4
2	プロジェクト管理の課題分析.....	5
3	プロジェクト管理の改善目標設定.....	5
4	PROMCODE データ交換の実装アーキテクチャ検討.....	5
5	プロジェクトデータの現状分析.....	6
6	PROMCODE 共通リソースモデルへのマッピング.....	7
6.1	帳票と PROMCODE モデルとのマッピング設計.....	7
6.1.1	報告元帳票と PROMCODE スキーマのマッピング.....	7
6.1.2	PROMCODE スキーマと報告先帳票／システムのマッピング.....	7
7	PROMCODE プロバイダ詳細設計・実装.....	9
7.1	PROMCODE プロバイダ実行環境の準備.....	9
7.2	EXCEL 帳票から PROMCODE スキーマへのマッピングルールの実装.....	9
7.3	PROMCODE プロバイダのデプロイと実行.....	10
8	PROMCODE コンシューマ詳細設計・実装.....	11
8.1	EXCEL VBA ベースのコンシューマの場合.....	11
8.1.1	Excel VBA ベースの PROMCODE コンシューマ実行環境の準備.....	11
8.1.2	PROMCODE プロバイダからのデータ読み込み部分の実装.....	11
8.1.3	読み込んだデータの加工・整形.....	13
9	(参考) PUSH モードの実装.....	13
9.1	PROMCODE プロバイダ.....	13
9.2	PROMCODE コンシューマ.....	13

1 はじめに

1.1 本書について

この文書は、PROMCODE 技術を用いてプロジェクト管理データの収集を実現する際の、概略手順を説明するものです。まずトライアルを始めてみよう、という人向けに書かれています。

1.2 想定利用ケース

PROMCODE のプロジェクト管理データ収集の利用ケースとしては、主に以下の 2 つのパターンを想定しています。

- ・各チーム→プロジェクトマネージャ（実績データの上位報告）
- ・各プロジェクト→PMO組織（実績データのスタッフ部門報告）

前者の場合、委託会社が異なる等の理由でチーム毎に管理帳票のフォーマットが異なることがあります。そのような状況に対して、PROMCODE 技術を使うことで帳票フォーマットの違いを許容したまま、プロジェクトマネージャへの上位報告が効率化できるようになります。PROMCODE 技術によってフォーマット差分を吸収してデータ収集できるためです。

後者の場合ですが、一般にプロジェクトが異なると管理帳票のフォーマットも異なります。こちらのケースも前のケースと同様に、PROMCODE 技術によって、異なるフォーマットの帳票のままプロジェクトから PMO 部門に報告することができるようになります。

2 プロジェクト管理の課題分析

現在のプロジェクト管理システムの仕組みを分析し、プロジェクト管理上の課題を特定します。

例) 現在、プロジェクトの開発フェーズ毎に何れかの協力会社が開発を担当し、それぞれのやり方で管理しています。そのため、発注会社側で、進捗データを統合してプロジェクト全体の進捗を管理・分析することが困難になっていました。また、Excel シートから発注会社内のプロジェクト管理システムに人手でデータを転記しています。そのため、プロジェクト管理システムで評価を行えるまでに日数を要していることが判りました。

3 プロジェクト管理の改善目標設定

次に、プロジェクト管理の改善目標と、そのアーキテクチャ (ToBe モデル) を定義します。

ここでは2つの課題に対応して、改善目標として、プロジェクト管理データの統合と、プロジェクト進捗データ集計の自動化を掲げます。

この改善目標を達成するために、プロジェクト管理システムアーキテクチャをオブザーバデザインパターンとし、PROMCODE 共通リソースモデルでデータモデルを定義することにします。各協力会社で扱う異なるフォーマットの Excel 帳票が、それぞれプロジェクト管理データのプロバイダとなり、これらをオブザーバが一元管理して、発注会社内のプロジェクト管理システムへ取り込むアーキテクチャです (図 1)。

このプロジェクト管理システムでは、2つのユースケースを想定します。1つめは、進捗状況を定期的に把握できるようにプロジェクト管理サーバでは、プロジェクト管理サーバからリクエストを送信し、各 Excel データを収集します。2つめは、進捗状況を即時に把握できるように、各 Excel データが作成・更新される度に、その都度、各 Excel データを受信します。

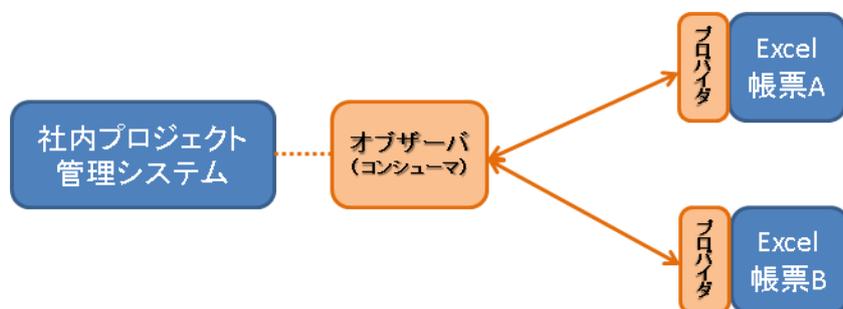


図1 プロジェクト管理システムのアーキテクチャ (ToBe モデル)

4 PROMCODE データ交換の実装アーキテクチャ検討

プロジェクト管理アーキテクチャに基づき、PROMCODE データ交換の実装アーキテクチャと、通信プロトコルを決定します。

図2に、実装アーキテクチャを示します。実装アーキテクチャには、プロキシサーバを導入し、進捗管理用 Excel 帳票に、Excel@用 PROMCODE アダプタ (以下 Excel 用アダプタ) を導入します。Excel 用アダプタは、PROMCODE プロバイダとしてプロジェクトデータを PROMCODE 共通リソースモデルのデータ (PROMCODE データ) に変換し、PROMCODE コンシューマに送信します。プロキシサーバは、PROMCODE コンシューマとして PROMCODE プロバイダからの PROMCODE データを受信します。

また、取得した PROMCODE データを、社内プロジェクト管理システム用のデータに変換します。

また、PROMCODE データ交換の通信プロトコルとして、①PROMCODE コンシューマから、各 PROMCODE プロバイダに HTTP 接続し、PROMCODE データを取得 (HTTP GET) する Pull モードと、②PROMCODE プロバイダから PROMCODE コンシューマに HTTP 接続し、PROMCODE データを送信 (HTTP POST) する Push モードをサポートします。

Pull モードは、PROMCODE プロバイダ側が、Push モードは、PROIMS コンシューマ側が HTTP サーバとして機能します。

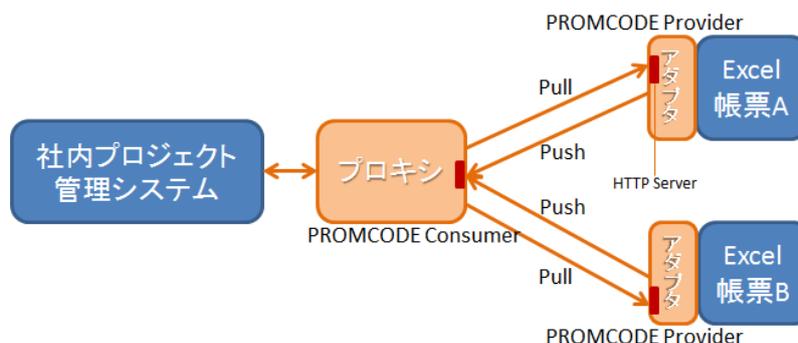


図 2 プロジェクト管理システムの実装アーキテクチャ

Push モードは、Pull モードを強化する位置づけのため、以降の章では Pull モードを基本実装として説明します。(Push モードの実装については、「9 (参考) Push モードの実装」をご覧ください)

5 プロジェクトデータの現状分析

プロジェクト管理用 Excel 帳票フォーマットの項目と、プロジェクト管理システムで管理する情報を確認します。

Excel 帳票 A のフォーマットを分析したところ、「区分 (機能)」と「作業タスク」で構成されていることが分かりました。この分析内容と PROMCODE 共通リソースモデルとの対応表を作成します。この結果、表 1 が得られました。

上記の対応の作成を Excel 帳票 B についても同様に行います。

そして、プロジェクト管理システムについても行い、プロジェクト管理データの報告側・収集側すべてについて、PROMCODE 共通リソースモデルとの対応関係を整理します。

表 1 Excel-PROMCODE 共通リソースモデル対応表

Excel 帳票	意味	PROMCODE 共通リソースモデル
区分 (機能)	管理範囲	ScopeItem
作業タスク	作業	WorkItem
タスク名		
担当者		
予定日 (予定, 実績)		
完了日 (予定, 実績)		

6 PROMCODE 共通リソースモデルへのマッピング

6.1 帳票と PROMCODE モデルとのマッピング設計

6.1.1 報告元帳票と PROMCODE スキーマのマッピング

PROMCODE の仕組みを使ってプロジェクト管理データをクライアントに送信するためには、報告元となる帳票を PROMCODE の定義するスキーマ(XML 電文)に変換する必要があります。その設計として、帳票の各項目が PROMCODE スキーマのどのリソース/プロパティに対応するか、をひとつひとつ定義します。(クライアントが欲しいデータだけを送信すればよく、帳票のすべての項目をマッピングする必要はありません。)

「5 プロジェクトデータの現状分析」の結果に基づいて項目マッピングを定義します(図3)。5章では、ScopeItem と WorkItem が対象リソースと分析しましたが、本章以降では簡単のため WorkItem だけのマッピングで説明します。

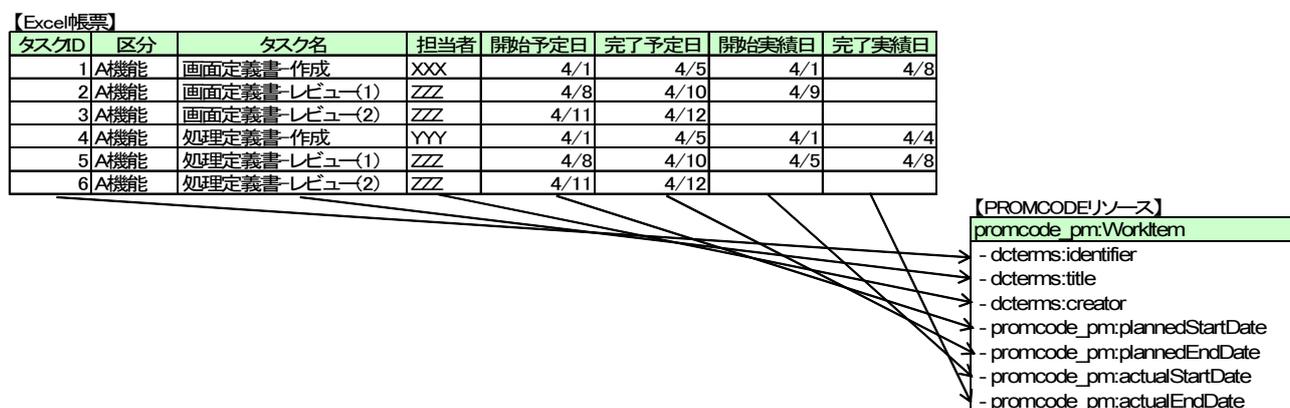


図3 帳票→PROMCODE リソースのマッピング

実際のケースでは、階層的なデータを扱うなど、もう少し複雑になることが多いです。その場合の項目マッピングは、複数のマトリクス表を組み合わせるなどの工夫が必要になります。

また、PROMCODE のリソース定義にない属性値を扱いたい場合は、独自のリソース属性を拡張定義することになります。その設計・定義もこのタイミングにて行います。

6.1.2 PROMCODE スキーマと報告先帳票/システムのマッピング

プロジェクト管理データを収集する側の仕組みとしては、サーバから PROMCODE リソースの XML 電文を受信し、その XML をパースして値を取得し、帳票やシステムにその値をセットしていくことになります。

こちらは「6.1.1 報告元帳票と PROMCODE スキーマのマッピング」とは逆の考え方で設計をすることになります。設計方法は 6.1.1 と同様のやり方で対応可能です。

ただし、集計値を出すような帳票やシステムの場合は、工夫が必要になります。いったんワークシート等に読み込み、その後集計値を算出して帳票化や DB 投入するよう設計します(図4)。

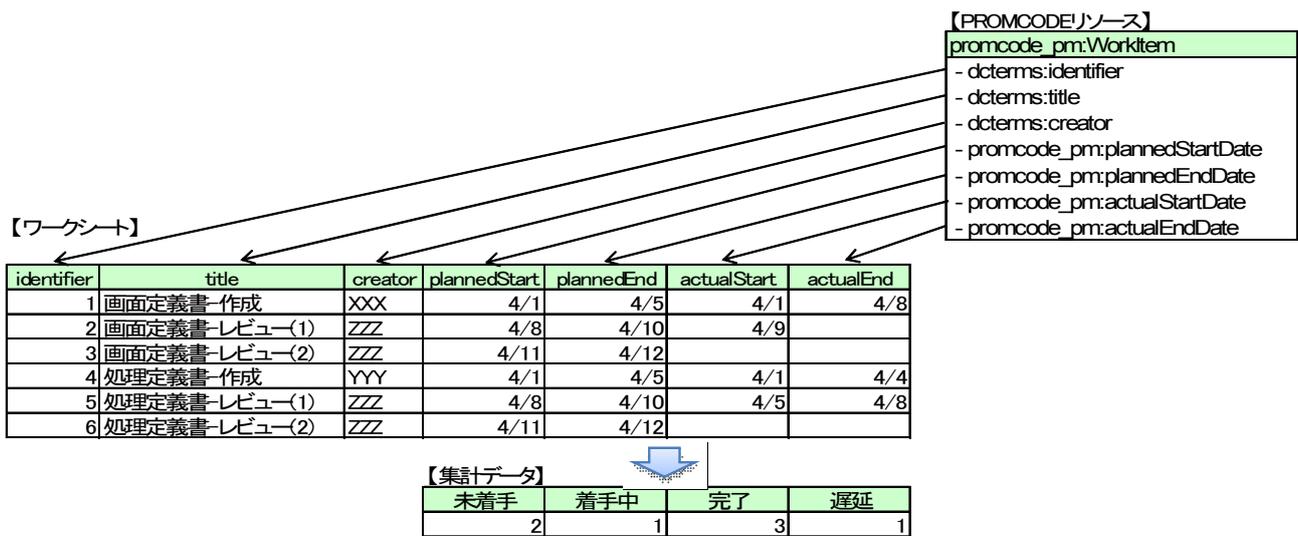


図 4 PROMCODE リソース→帳票/システムのマッピング

7 PROMCODE プロバイダ詳細設計・実装

7.1 PROMCODE プロバイダ実行環境の準備

PROMCODE プロバイダは、Excel 帳票のデータを PROMCODE コンシューマ（クライアント）に提供するサーバの役割を持ちます。ここでは、PROMCODE プロバイダを「Excel 用アダプタ」を用いて実現することを例にとって説明します。

最初に実行環境の準備を行う必要があります。セットアップしなければならない要素として、以下のものがあります。

- PROMCODE プロバイダ（含む Excel 用アダプタ）
- マッピングルールファイル
- データファイル（Excel 帳票）

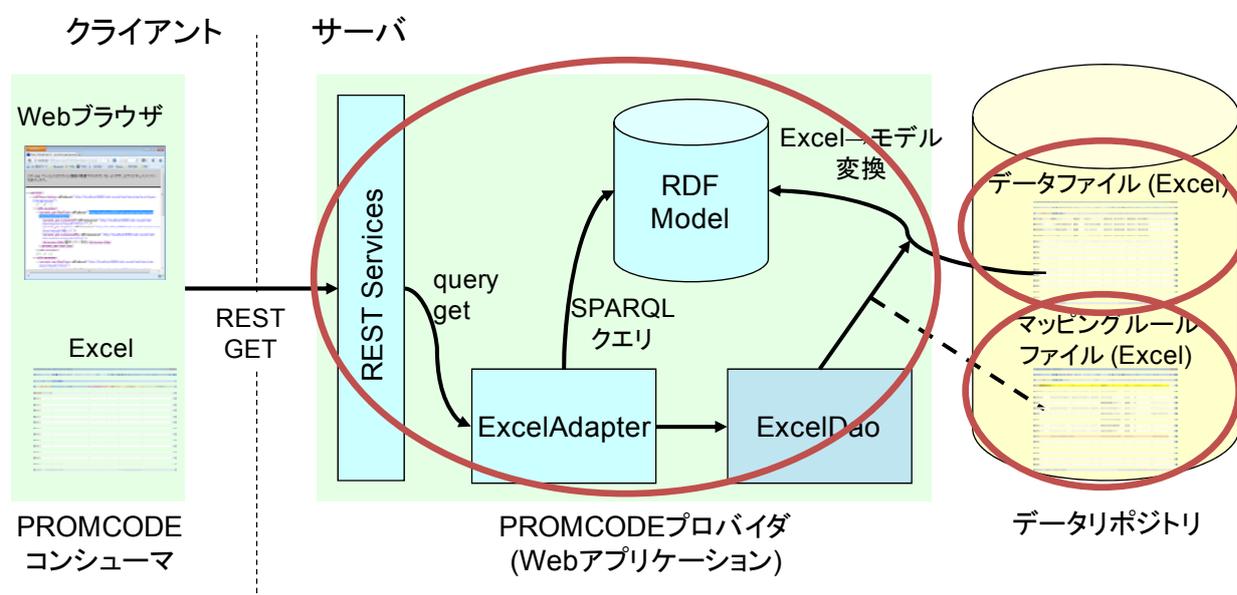


図 5 PROMCODE プロバイダ実行環境

具体的な設定手順については、「PROMCODE サービス開発ガイド」の「2 PROMCODE モデルの開発環境のセットアップ」をご参照ください。

7.2 Excel 帳票から PROMCODE スキーマへのマッピングルールの実装

PROMCODE プロバイダとして帳票データを PROMCODE コンシューマに提供するためには、「6.1.1 報告元帳票と PROMCODE スキーマのマッピング」で設計したものを、Excel 用アダプタのマッピングルールファイルである「mapper.xls」として、実装する必要があります。

mapper.xlsは、Excel用アダプタ規定の記法に則って記述します。先の設計例のものを実装してみると、以下のようになります。

Resource				Property			
name	type	line	uri	name	type	column	reference
WI	promcode_pm:WorkItem	0,1,*,exist(A)	WI_%s,A				
				dcterms:identifier	String	A	
				dcterms:title	String	C	
				dcterms:creator	String	D	
				promcode_pm:plannedStart	DateTime	E	
				promcode_pm:plannedEnd	DateTime	F	
				promcode_pm:actualStart	DateTime	G	
				promcode_pm:actualEnd	DateTime	H	

図 6 マッピングルールの実装例

左側 4 列「Resource」には、リソース（エンティティ）レベルでのマッピング情報を記述します。

name：マッピング定義に付与する名称を記載します。

type：マッピング先となる PROMCODE のリソース型名を記載します。

line：元となる Excel 帳票のどの範囲を検索対象にするかの条件式を記載します。

url：マッピング先になる URI 文字列の構築式を記載します。

右側 4 列「Property」には、プロパティ（属性）レベルでのマッピング情報を記述します。

name：マッピング先となる PROMCODE のプロパティ名を記載します。

type：マッピング先となる PROMCODE のプロパティ値のデータ型を記載します。

column：元となる Excel 帳票のどの列のデータを取得対象にするかの条件式を記載します。

reference：他リソースへの参照式を記載します。

上記は非常に簡単な例ですが、Excel 用アダプタは多様なマッピングルール記述が可能となっています。その詳細については、「PROMCODE サービス開発ガイド」の「4 マッピングルールの定義」をご参照ください。

もし、帳票から PROMCODE スキーマへの変換が Excel 用アダプタの機能で実現できない場合は、Excel 用アダプタの拡張が必要となります。方法については、「PROMCODE サービス開発ガイド」の「5 Excel 用アダプタ拡張のヒント」をご参照ください。

7.3 PROMCODE プロバイダのデプロイと実行

PROMCODE プロバイダの実行環境上に、mapper.xls と Excel 帳票を規定のディレクトリに配置し、httpd のサービスを実行します。

これで、PROMCODE コンシューマから帳票データを PROMCODE 形式の XML として HTTP/REST のインタフェースから取得できるようになります。

8 PROMCODE コンシューマ詳細設計・実装

8.1 Excel VBA ベースのコンシューマの場合

8.1.1 Excel VBA ベースの PROMCODE コンシューマ実行環境の準備

PROMCODE コンシューマは、PROMCODE プロバイダ（サーバ）から提供されるプロジェクト管理データを受信するクライアントの役割を持ちます。ここでは、PROMCODE コンシューマを Eclipse Lyo で提供されている「Excel クライアント」を用いて実現することを例にとって説明します。

実行環境の準備ですが、Eclipse Lyo パッケージに含まれる

- Excel クライアント：Client of RIO for Excel.xls

のファイルを取りだし、実行したいフォルダに配置するだけです。

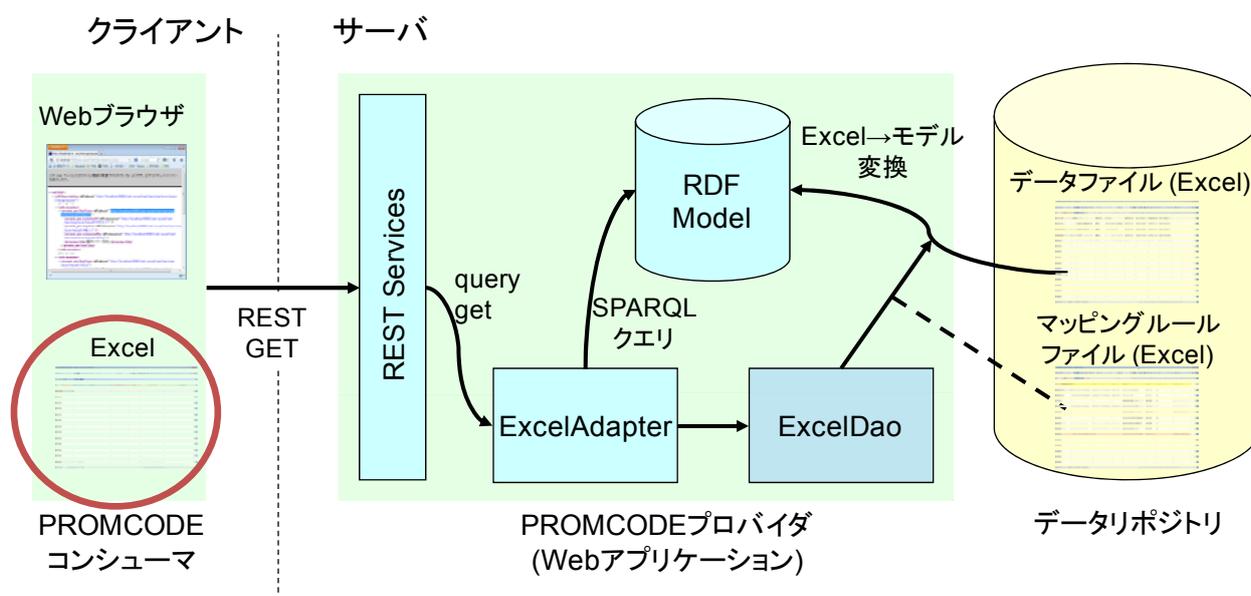


図 7 PROMCODE コンシューマ実行環境

詳細については、「PROMCODE サービス開発ガイド」の「2 PROMCODE モデルの開発環境のセットアップ」をご参照ください。

8.1.2 PROMCODE プロバイダからのデータ読み込み部分の実装

PROMCODE コンシューマとして、帳票データを PROMCODE プロバイダから読み込むためには、

「6.1.2 PROMCODE スキーマと報告先帳票／システムのマッピング」で設計したものを、Excel クライアントの設定として実装する必要があります。

Excel クライアントの設定は Client of RIO for Excel.xls ファイルの Setting シート上に記述します。

Server Parameters

Requirement Server URL:	http://localhost:8080/oslc-excel/rest/services/catalog/projectA
Jazz Project Name:	

Query Parameters

Where	
Select	dcterms:identifier,dcterms:title,dcterms:creator,promcode_pm:plannedStartDate,promcode_pm:plannedEndDate,promcode_pm:actualStartDate,promcode_pm:actualEndDate
Start Row	2
Columns	A,B,C,D,E,F,G

図 8 Excel クライアントの設定例

設定が必要な個所は以下のとおりです。

Requirement Server URL : PROMCODE プロバイダ (サーバ) の URL を指定します。

Where : 読み込む際の絞り込み条件を記載します (任意)

Select : 読み込みたい PROMCODE リソースのプロパティ名を列挙記載します。

Start Row : データを読み込む Excel シートの開始行位置を記載します。

Columns : Select で指定したプロパティ値を読み込む Excel シート上のカラム位置を列挙記載します。

この設定を行った後、ツールバーの「Download(OSLC)」ボタンを押すと、プロバイダからデータを取得できます。取得したデータは WorkItems シート上に読み込まれます。



図 9 PROMCODE コンシューマ実行例

8.1.3 読み込んだデータの加工・整形

読み込んだデータは、前述のとおり WorkItems シート上に一覧形式で読み込まれます。

あとは、Excel の機能であるワークシート関数や VBA マクロを用いて、望む帳票形式に加工・整形すれば完了です。

また、他のシステムやデータベースへそのデータを送りこむ場合は、任意のプログラム言語でそのためのアプリケーションを実装します。この部分は、PROMCODE 技術とは非依存のものになります。

9 (参考)Push モードの実装

参考として、3章の図2に沿った、Push モードの実装方法を示します。

9.1 PROMCODE プロバイダ

Excel 用アダプタを設計します。Excel 用アダプタは、①PROMCODE 共通リソースオブジェクトに変換するマッパー、②PROMCODE 共通リソースオブジェクトから XML 文書の生成部、③XML 文書から HTTP メッセージを作り、HTTP リクエストを送信するメッセージング部から成ります。

- ① マッパーは、6章で得たマッピングルールと共通リソースモデルに沿って、Excel 帳票を読み出し、PROMCODE 共通リソースオブジェクトを構築します。
- ② XML 文書生成部は、PROMCODE 共通リソースオブジェクトから、OSLC 仕様に従って RDF モデルを生成し、XML 文書へのシリアライゼーションを行います。
- ③ メッセージング部は、HTTP メッセージのボディに XML 文書を設定し、HTTP POST リクエストを PROMCODE コンシューマへ送信します。

以上の設計に従って、Java プログラムを実装します。

9.2 PROMCODE コンシューマ

プロキシサーバを、①PROMCODE プロバイダとの通信機構と、②社内プロジェクト管理ツールへのコネクタで構成します。

- ① PROMCODE プロバイダとの通信機構は、HTTP サーバとして設計します。HTTP POST レスポンスを PROMCODE データとして保持します。
- ② 社内プロジェクト管理ツールへのコネクタは、①の PROMCODE データに、語彙変換フィルターを当て、次に、社内プロジェクト管理ツールの入力 API を呼び出します。

以上の設計に従って、Java プログラムを実装します。