

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

日立製作所における実証実験報告書

第 1 版

2013 年 10 月 22 日

株式会社日立製作所

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	日立製作所における実証実験.....	エラー! ブックマークが定義されていません。
1.1	目的	4
1.2	実証実験の内容.....	6
1.2.1	実証実験のユースケースとシナリオ.....	6
1.2.2	実証実験に適用したデータ	6
1.2.3	プロジェクトモデルの設計.....	7
1.3	実証実験の環境.....	8
1.3.1	実証実験の環境.....	8
1.3.2	開発ソフトウェア.....	9
1.4	実証実験の結果.....	10
1.5	実証実験の評価.....	12
1.5.1	期待効果.....	12
1.5.2	課題.....	12
1.6	まとめ	13

1 目的

日立製作所における実証実験の目的は、次の2つである。

- (1) PROMCODE 仕様(PROMCODE ドメインモデルと OSLC 基盤), PROMCODE アダプタソフトウェアの実プロジェクトへの適用性を評価すること。
- (2) 上記適用の際の課題を認識すること。

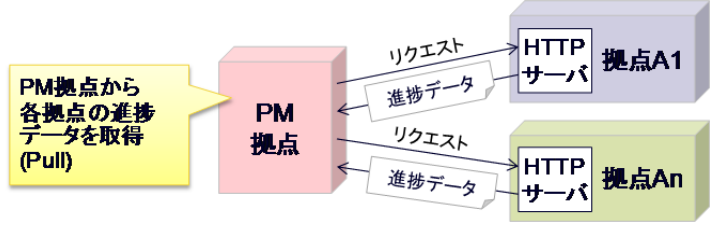
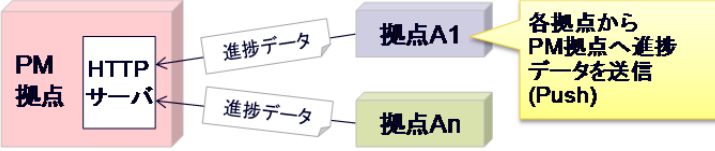
上記実験目的に沿い、実験における評価・検証項目を、表 1 の通り設定した。

表 1 実験における評価・検証項目

#	評価・検証項目	評価・検証内容
1	技術的観点 進捗管理データ への変換	各拠点で管理している作業進捗データを、プロジェクトマネージャ(以下、PM)の求める形式の進捗管理データに変換できるか？
2	拠点間でのデータ授受	拠点間で作業進捗データを滞りなく受け渡しできるか？
3	導入難易度の観点 開発現場のツール導入難易度	各拠点で、ツール導入のインフラ要件(ネットワークや稼動マシン等)を容易に満たせるか？ また、各拠点で、ツールのセットアップを容易に行えるか？
4	交換データのマッピング定義の難易度	拠点個別の作業進捗データを変換するためのマッピング仕様を容易に設計できるか？ また、同仕様にもとづくマッピングの定義を容易に行えるか？
5	ツールサポートの難易度	ツール実装、各種サポート等の運用の難易度は？

また、PMの開発拠点(以下、PM拠点)と、それ以外の開発拠点との間のデータの授受は、表2に示す2種類の方法が考えられるため、双方のパターンについて、実験を行うこととした。

表2 データ授受パターン

パターン名	説明
パターン A (Pull 型)	PM 以外の拠点の組織規模が大きく、HTTP サーバの設置とインターネット公開が可能な場合。 
パターン B (Push 型)	PM 以外の拠点の組織規模が小さく、HTTP サーバ設置とインターネット公開が困難な場合。 

2 実証実験の内容

2.1 実証実験のユースケースとシナリオ

本実験で想定したシナリオは、PROMCODE の「シナリオ 1」である。

パターン A(Pull 型)、パターン B(Push 型)それぞれに対し、次のユースケースシナリオを想定した。

(1) パターン A(Pull 型)のユースケースシナリオ：

- 1) (準備)：プロジェクト開始時、PM 以外の各拠点では、以下の準備を行っておく。
 - a) PM 拠点から随時 HTTP アクセス可能なサーバマシンを用意し、PROMCODE 準拠の作業進捗データを応答で返すサーバソフト(3 節で詳述)をセットアップ、稼働させる。
 - b) 併せて、その拠点で使用する独自形式の作業進捗データを PROMCODE 準拠形式の作業進捗データに変換する「マッピング定義(3 節で詳述)」を作成しておく。
- 2) 各拠点のリーダーは、自拠点の作業進捗を更新する都度、上記 1)の HTTP サーバマシン所定ディレクトリに、最新版の作業進捗データを配置する。
- 3) PM は、所定のツール(3 節で詳述)を用い、各拠点の HTTP サーバから最新の作業進捗データを取得し、定型書式の進捗管理データに変換された状態で進捗状況を確認する。

(2) パターン B(Push 型)のユースケースシナリオ：

- 1) (準備)：プロジェクト開始時に、各拠点で以下の準備を行っておく。
 - a) PM 拠点：
各拠点から随時 HTTP アクセス可能なサーバマシンを用意し、HTTP で受信した PROMCODE 準拠作業進捗データを、定型書式の進捗管理データに変換出力するサーバソフト(3 節で詳述)をセットアップ、稼働させる。
 - b) 各拠点：
その拠点で使用する独自形式の作業進捗データを PROMCODE 準拠の作業進捗データに変換する「マッピング定義」を作成しておく。
- 2) 各拠点のリーダーは、自拠点の進捗情報を更新する都度、作業進捗データを PROMCODE 形式に変換し送信するツール(3 節で詳述)を用いて、PM 拠点に進捗データを送信する。
※本シナリオでは、送信の際に進捗情報を PROMCODE 形式に自動変換し送信することを想定しているが、今回の実験では、該当する処理はパターン A で検証済みであるため、あらかじめ手作業で PROMCODE 形式に変換した進捗データを送信することで、擬似的に本場面を再現し実験した。
- 3) PM は、上記 1)のサーバソフトで定型書式に変換された進捗管理データを閲覧し、各拠点の進捗状況を確認する。

2.2 実証実験に適用したデータ

実証実験に適用したデータは、産業系プロジェクト、金融系プロジェクト、各々、実際の開発現場の進捗管理データである。双方のデータともに、進捗管理の対象工程は、プログラミング～単体テストである。

2.3プロジェクトモデルの設計

今回の実証実験に適用したデータにおいて、PM の求める形式の進捗管理データで表現されている情報項目を元に、PROMCODE ドメインモデルのクラス群にサブクラスを追加し、これらのサブクラスの属性として、必要となる情報項目を追加した。

追加したサブクラス群および情報項目群(サブクラスの属性)の抜粋を、図 1 に示す。

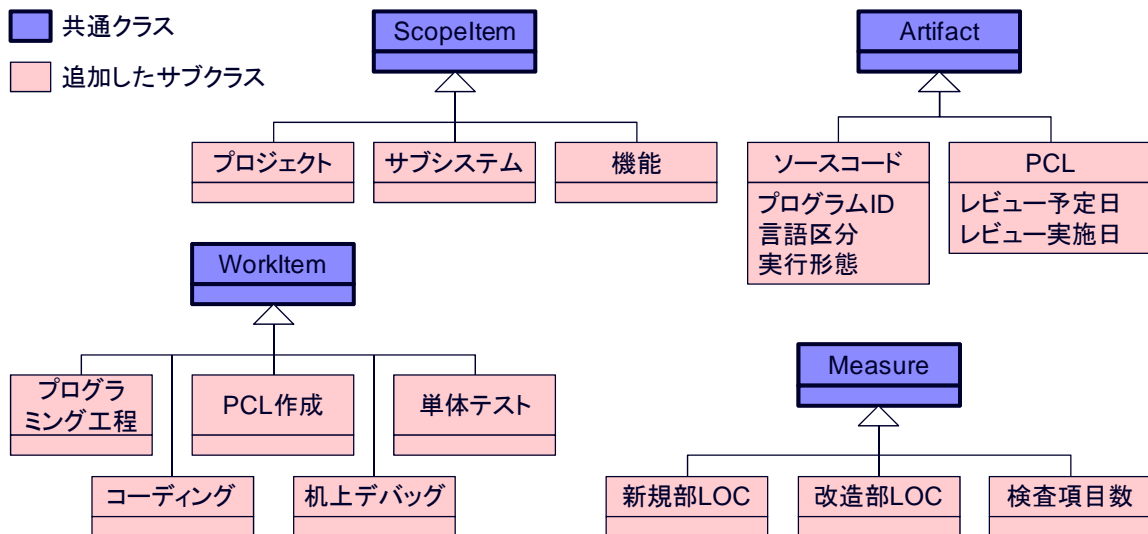


図 1 追加したサブクラス群および属性 (抜粋)

3 実証実験の環境

3.1 実証実験の環境

パターン A(Pull 型), パターン B(Push 型)の実証実験環境を, それぞれ図 2, 図 3 に示す.

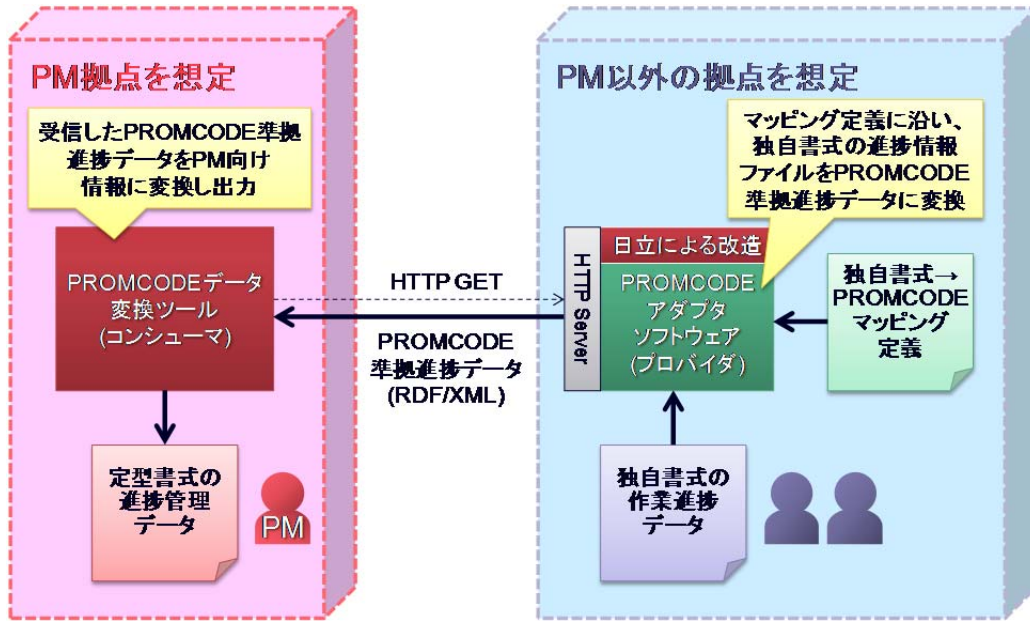


図 2 パターン A(Pull 型) 実証実験環境

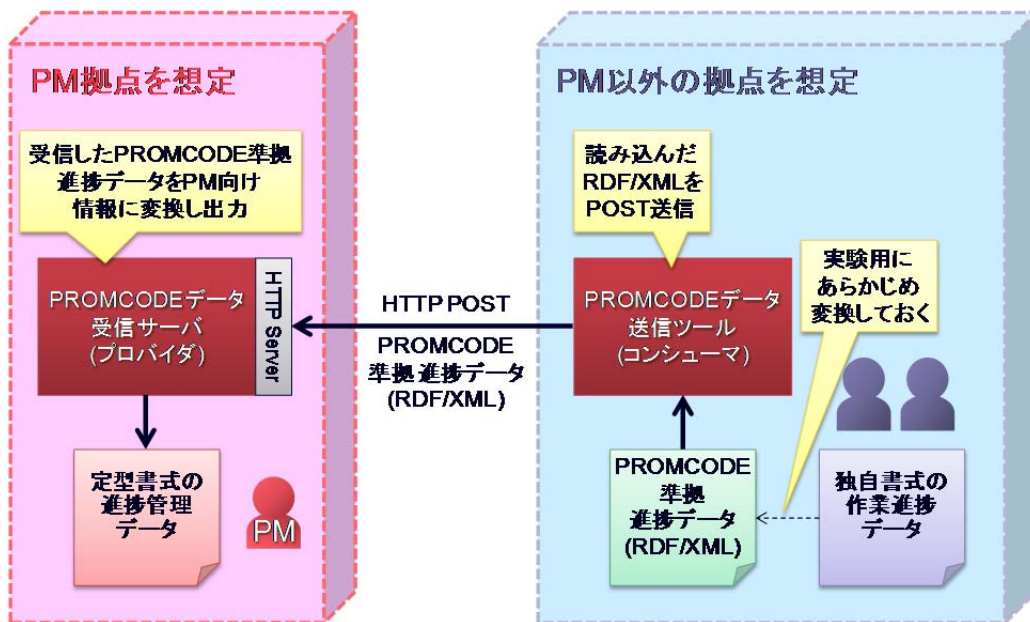


図 3 パターン B(Push 型) 実証実験環境

3.2 開発ソフトウェア

本実験における開発ソフトウェアの一覧を表 3, 表 4 示す.

表 3 パターン A(Pull 型) 実験の開発ソフトウェア一覧

#	開発ソフトウェア	開発言語	機能説明
1	PROMCODE アダプ タソフトウェア 改造	Java	<p>PROMCODE アダプタソフトウェアに対し, 本実験用に次の機能追加を行った.</p> <p>(1)本実験用にカスタマイズした PROMCODE モデルの RDF/XML 表現向け XML ネームスペースを追加.</p> <p>(2)PROMCODE 準拠データへの変換時に固定値を出力するマッピング定義機能(例: 工程名, 等)</p> <p>(3)変換前の作業進捗データのセル値として, 数式が記入されていた場合に, 数式の演算結果を得る機能(例: テスト項目件数値が, 正常条件数, 異常条件数, 境界条件数の合計で算出されるよう, 数式で設定されている場合, 等)</p> <p>(4)マッピング定義において, セル参照値の数式演算を定義する機能(例: 変換前の作業進捗データに記載されている机上デバッグ時解決件数と単体テスト時解決件数からバグ解決件数を合計にて算出, 等)</p>
2	PROMCODE データ 変換ツール	Excel® マクロ (VBA)	<p>OSLC コンシューマとして動作し, OSLC プロバイダ(本実験においては, 上記#1 の PROMCODE アダプタソフトウェア)から PROMCODE 準拠進捗データを取得して, PM 側で想定している定型書式の進捗管理データに変換出力する.</p>

表 4 パターン B(Push 型)実験の開発ソフトウェア一覧

#	開発ソフトウェア	開発言語	機能説明
1	PROMCODE データ 送信ツール	Excel® マクロ (VBA)	<p>OSLC コンシューマとして動作し, あらかじめ用意した, PROMCODE 準拠形式に変換済みの RDF/XML 準拠進捗データを, OSLC プロバイダ(本実験においては, 下記#2 の「PROMCODE データ受信サーバ」)に送信する.</p>
2	PROMCODE データ 受信サーバ	Java	<p>OSLC プロバイダとして動作し, OSLC コンシューマ(本実験においては, 上記#1 の「PROMCODE データ送信ツール」)より送信された PROMCODE 準拠進捗データを受信し, PM 側で想定している定型書式の進捗管理データに変換出力する.</p>

4 実証実験の結果

1 節で述べた、実証実験の評価・検証項目別の結果を表 5 に示す。

技術的観点では、パターン A(Pull 型)、パターン B(Push 型)とも、データの変換と HTTP を介したデータの受け渡しは可能であることを確認できた。

導入難易度の観点では、実験期間中の作業から、ツールカスタマイズはマッピング定義のみならば作業工数は 4 時間程度と小さいが、前提として若干の習熟が必要である事が判った。導入初期は、現場への支援者のサポートが必要である。併せて、ネットワーク等の設備を事前に考慮する必要がある。

表 5 実験結果

#	評価・検証内容		結果
1	技術的 観点	進捗管理 データへ の変換	各拠点で管理している 作業進捗データを、PM の求める形式の進捗管 理データに変換できる か？ ○
2		拠点間 でのデータ 授受	各拠点で作業進捗デー タを滞りなく受け渡し できるか？ ○
3	導入 難易度 の観点	開発現場 のツール 導入難易 度	各拠点で、ツール導入の インフラ要件(ネットワ ークや稼動マシン等)を 容易に満たせるか？ また、各拠点で、ツール のセットアップを容易 に行えるか？ △
4		交換デー タのマッ ピング定 義の難易 度	拠点個別の作業進捗デー タを変換するための マッピング仕様を容易 に設計できるか？ また、同仕様にもとづく マッピングの定義を容 易に行えるか？ △
5	ツールサ ポートの 難易度	ツール実装、各種サポー ト等の運用の難易度 は？ △	開発拠点の独自書式の作業進捗データから PROMCODE 準拠進捗データへの変換は、 PROMCODE アダプタソフトウェアへの若干の機能 追加(表 3 項番 1)で可能となった。 PROMCODE 準拠進捗データから PM の求める進捗管 理データへの変換は、個別のプログラム開発 (パター ン A(Pull 型)の場合：表 3 項番 2、パターン B(Push 型)の場合：表 4 項番 2) で可能となった。 HTTP で直接通信のできる拠点間であれば、データの 授受を滞りなく行える見込みが得られた。 顧客先等、HTTP 直結不可の場合は、PROMCODE の データ交換方式(RESTful Web)をそのまま適用するこ とはできない。この場合は、例えばデータのメール 受け渡しを併用する等の工夫が必要となる。 PM 拠点または PM 以外の拠点、少なくともいずれか で HTTP サーバマシンの用意、セットアップ(~数人 日)とインターネットへの公開(イントラ/専用線直結 でない場合)が必要(表 6)。 マッピング作業に慣れた専門の作業者を置く運用が 可能であれば、現状のマッピング定義で通常時の運 用は可能(必要作業工数：表 7)。 ただし、PROMCODE アダプタソフトウェアの提供機 能範囲内でマッピングが行えなかった場合の対応を 考慮要。
			HTTP(S)によるデータ授受を想定したセキュリティ 対策が必要。 また、本番環境向けのツール実装は、現状、個別の 開発が必要。

表 6 ツール導入に必要な環境と作業

パターン名	PM 拠点	PM 以外の拠点
パターン A (Pull 型)	クライアントツール(Excel®マクロ等)の導入(HTTP 接続要)	HTTP サーバ機の確保, セットアップ (~数人日), 公開要
パターン B (Push 型)	HTTP サーバ機の確保, セットアップ (~数人日), 公開要	クライアントツール(Excel®マクロ等)の導入(HTTP 接続要)

表 7 マッピング作業に必要な作業工数

マッピング作業	1 プロジェクトあたりのマッピング工数	説明
開発現場作業	~数人日程度 (大部分は学習コスト)	現状のマッピング定義は, OSLC の生のリソースタイプやプロパティへのマッピングであり, その理解がまず必要. 加えて, 組織向けにカスタマイズされたプロジェクトモデルの理解が必要.
習熟した専門 作業	~4 時間程度	慣れれば難しくはなく, 作業量も小さい.

5 実証実験の評価

5.1 期待効果

形式の異なる進捗情報の変換を自動化する際に、各拠点の作業進捗データ→PROMCODE データ、PROMCODE データ→PM 向け進捗管理データ、の双方で、再利用可能な共通技術(ツールやライブラリ等)が利用できるようなれば、自動化のツール等を短期間・低コストで実現できるようになること。

5.2 課題

PROMCODE の弊社内への導入に向けた課題を挙げる。

- (1) 適用の際、特に社内での施策立ち上げ時点におけるサポート部門等での、稼働環境確保、カスタマイズ対応が必要であると考える。
- (2) 今回実証実験で検証した範囲のほか、弊社内で利用されている他のツールの進捗管理データ、品質管理情報も、収集する対象として継続して検討していく。

6 まとめ

今回の実証実験により、次の事が判った。

- (1) PROMCODE モデルを介したプロジェクト進捗情報の変換は、今回選定した2つのプロジェクト事例において、可能だった。
- (2) 開発現場への導入の際にクリアを要すると思われる主な課題：
(PROMCODE のスコープ外と思われる課題を含む)
 - 1) HTTP 直結不可の環境への導入方法
 - 2) マッピング定義専任者を前提とした運用，または開発現場が自らマッピング定義できる水準への平易化
 - 3) PROMCODE アダプタソフトウェアの機能追加を要する場合の考慮

プロジェクト現場からの進捗など管理情報を効率的に収集するための施策として、PROMCODE での検討成果を、弊社社内ツール等に活用していく計画である。