

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

NEC における実証実験報告書

第 1 版

2013 年 10 月 22 日

日本電気株式会社

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	目的.....	4
2	実証準備.....	5
2.1	プロジェクト管理帳票の COMMON RESOURCE MODEL への変換方式.....	5
2.2	PMO と各プロジェクトとの通信方式.....	7
2.3	実証シナリオ.....	8
2.4	評価対象.....	8
3	実証結果.....	9
3.1	評価結果.....	9
3.2	その他の評価.....	9
4	考察.....	10
5	今後の予定.....	11
6	まとめ.....	12

1 目的

NECの実証実験は、(1)PMOが管理方法の異なるプロジェクトを容易に監視対象として追加できること、(2)プロジェクトリスク発見の遅延を短縮できることを評価目的とした。

従来のプロジェクト管理では、プロジェクト管理ツールごとに取り扱うデータ項目が異なり、PMOがリアルタイムに監視できないプロジェクトが残されていた。そこで、PROMCODE仕様に基づき、データ変換および交換を行うことで、より多くのプロジェクトの状態をリアルタイムに監視できることを期待している。実証実験では、(1)社内PMOの利用するプロジェクト監視システム（プロジェクト管理基盤）と各組織で利用しているプロジェクト管理ツール間でのデータ変換の正確さ、および連携の容易さ（図1）、(2)プロジェクト監視システムとプロジェクト管理ツール間のデータ連携のリアルタイム性（図2）を確認した。

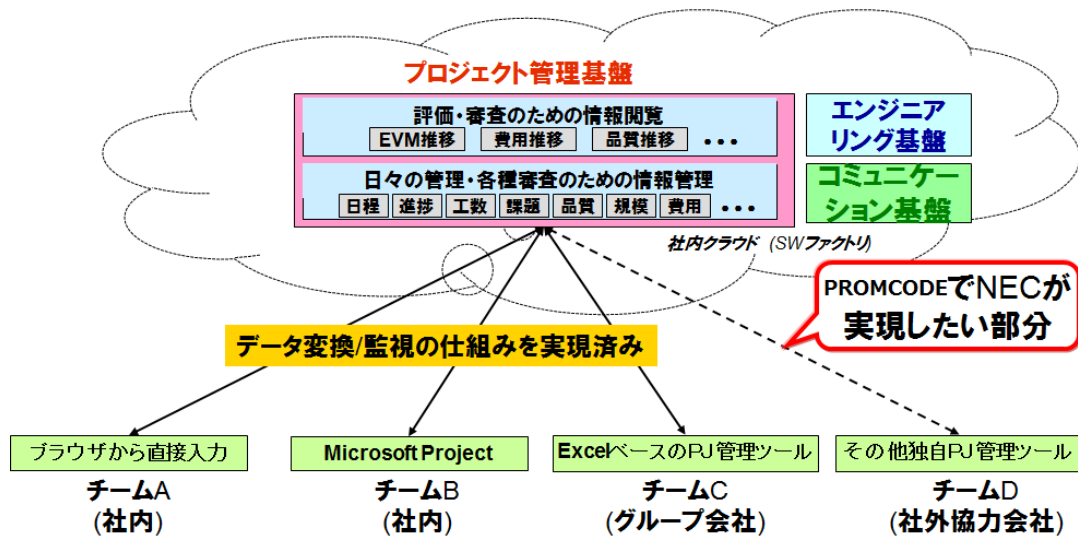


図1 社内PMOと管理方法の異なるプロジェクトの可視化

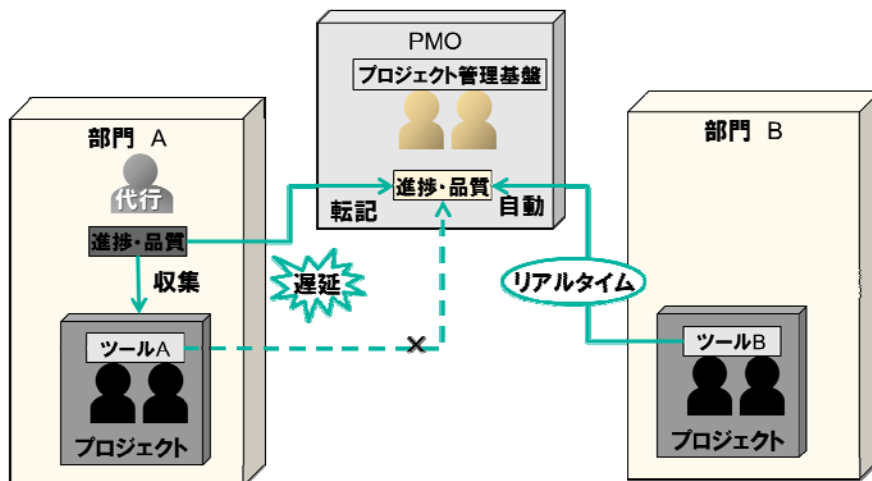


図2 プロジェクト進捗のリアルタイム監視

2 実証準備

2.1 プロジェクト管理帳票の Common Resource Model への変換方式

図 3 に示すように、PMO が必要とするプロジェクト管理帳票 (Excel) の項目を、PROMCODE の Common Resource Model のオブジェクトに変換するため、Common Resource Model で規定されている WorkItem, UseCase (ScopeItem), Artifact, Measurement, KPI の具象クラスを定義した (図 4)。

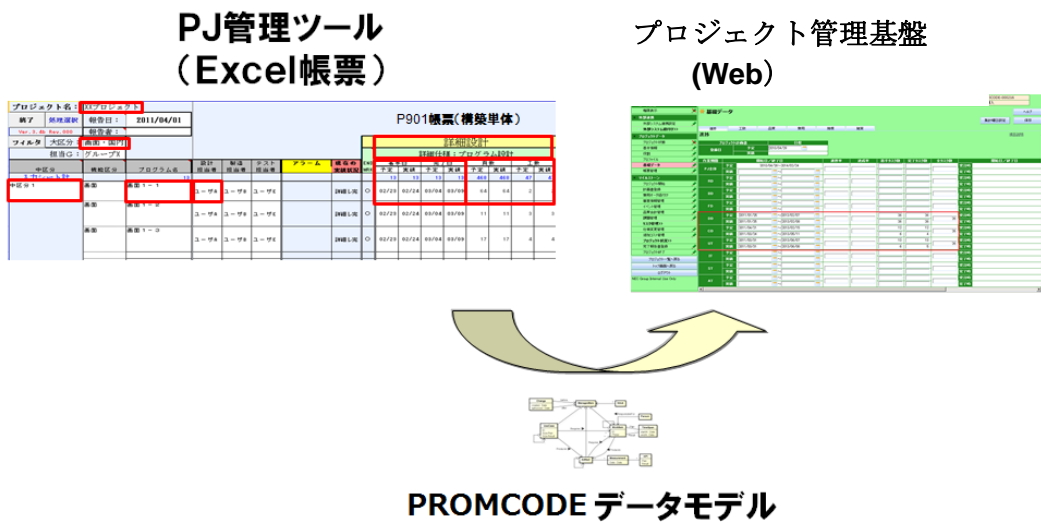
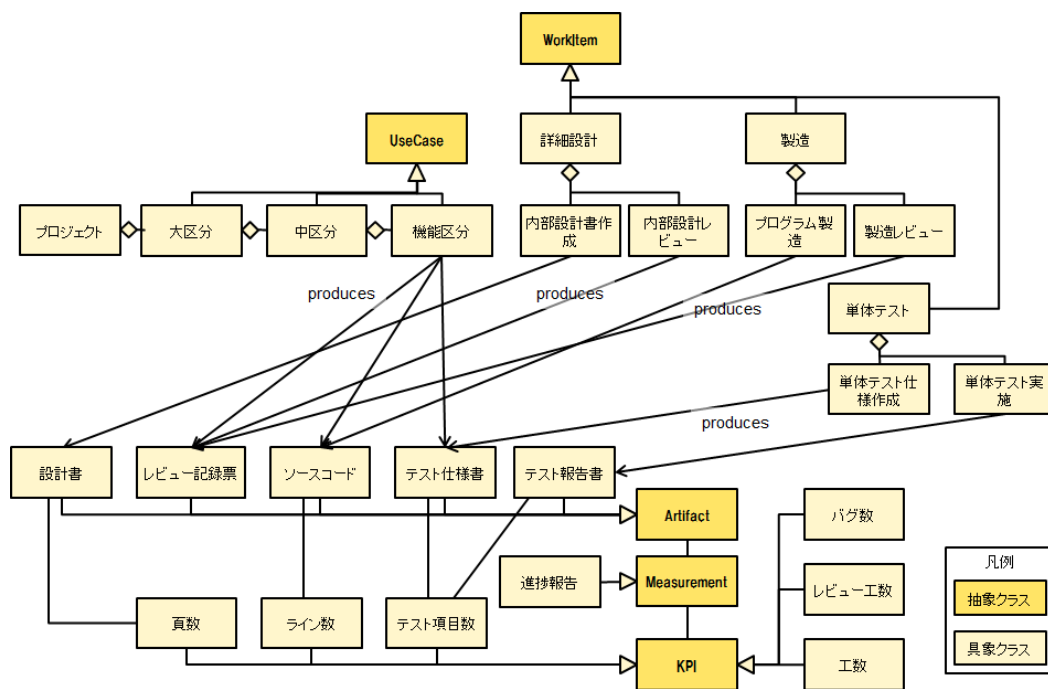


図 3 PROMCODE Common Resource Model への変換



この具象クラス定義をもとにプロジェクト帳票項目のCommon Resource Model へのマッピングを行った。図5にマッピングの詳細を示す。また、表1, 2にそのルール例を示す。

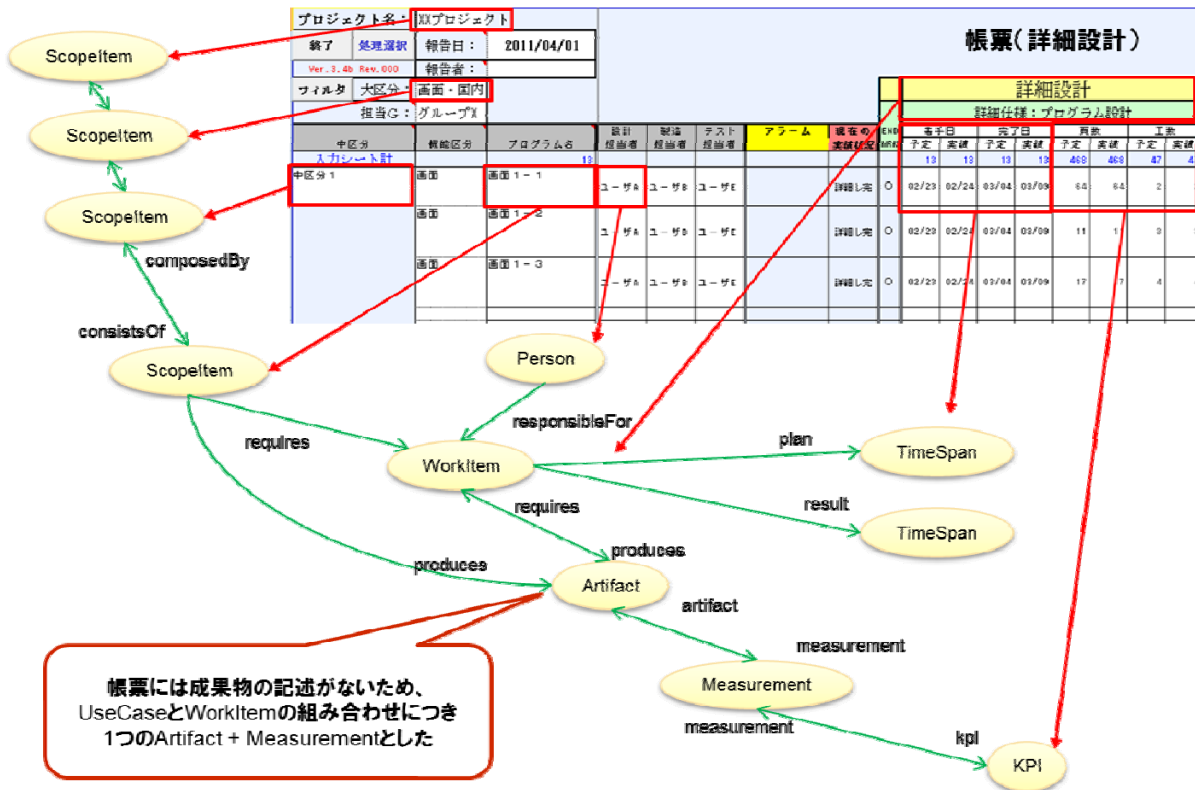


図5 Common Resource Model へのマッピング

表1 ScopesItem の生成ルール

項目	割り当てる値	意味
Name	SIL0	生成時に参照する簡易名称
Type	promcode_pm:ScopesItem	生成するクラスはScopesItem
Line	Sheet(DomainPresentation),0,0	DomainPresentationシートのセル(0,0)の値があればScopesItemクラスを生成
Uri	SIL0	RDFで関連づけに利用するURI

表2 ScopesItem の属性割り当てルール

項目	割り当てる値	意味
Name	Dcterms:title	Title属性にセル値を適用
Type	String	セル値はString型
Column	B	B列の値が属性に
Reference	なし	通常は利用しない

2.2 PMO と各プロジェクトとの通信方式

社内 PMO と各プロジェクトのデータを連携させる仕組みを図 6 に示す。ここで、Excel 用アダプタはプロジェクト管理ツールのデータを Common Resource Model のオブジェクト群に変換し、データ変換プロキシーは収集した Common Resource Model のオブジェクト群を社内 PMO のプロジェクト監視ツール（プロジェクト管理基盤）が解釈できるようにデータ変換する機構となっている。

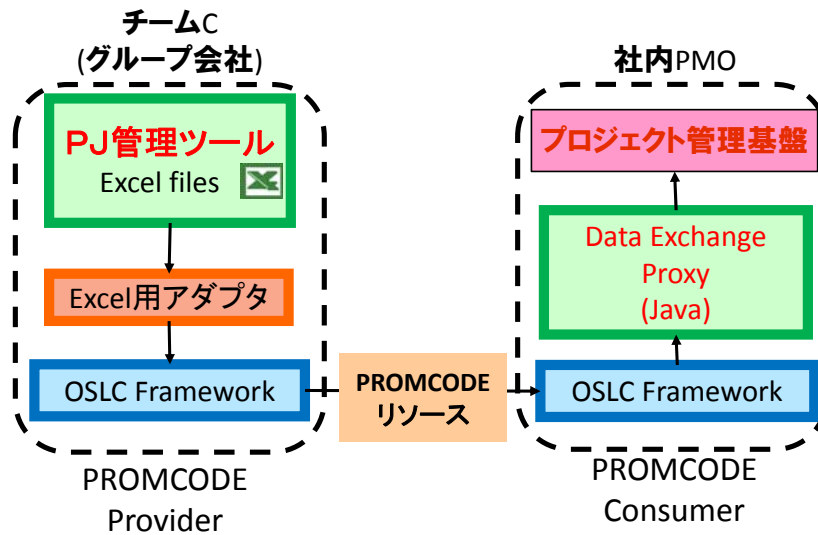


図 6 データ交換の仕組み

上記に示したデータ交換の実装として、HTTP プロトコルによる Pull 型の通信ができるように、HTTP サーバをデータ変換プロキシーと、各プロジェクト管理ツール・帳票の間に配置している（図 7）。これによって、PMO の指示による即時のデータ交換や定期的なデータ交換が可能になっている。

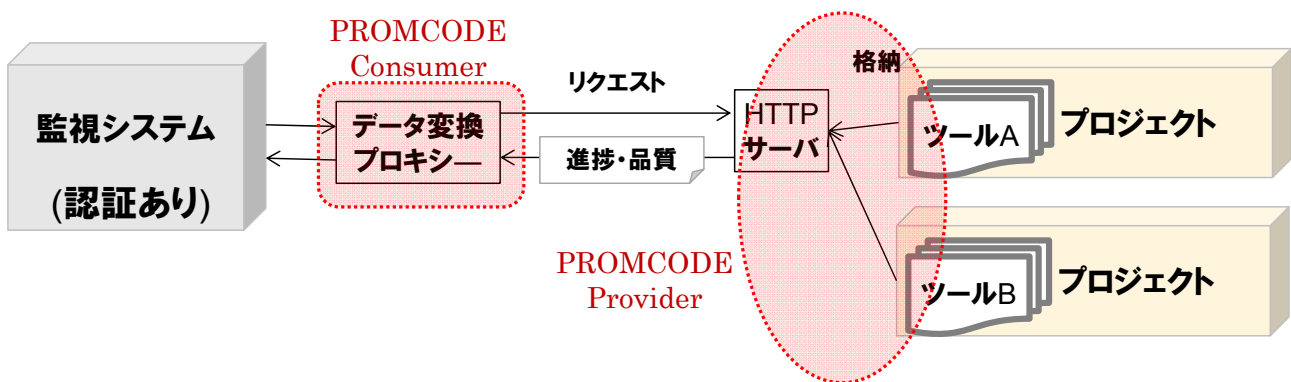


図 7 データ交換の実装モデル

なお、本実証実験では、プロキシーを JSP で実装した。プロキシーは、Common Resource Model (RDF 形式) を解釈し、PMO のプロジェクト監視システムが必要とする粒度の集計情報を生成する。

2.3 実証シナリオ

実際の利用場面として、次の実証シナリオを想定した。

- (1) プロキシの管理者は、ツール A から Common Resource Model を生成するルールを定義し、HTTP サーバへ登録する。
- (2) プロジェクトマネージャは、予め公開された HTTP サーバ上の共有ディレクトリに、進捗・品質情報を記載したツール A をそのまま、格納する。
- (3) PMO は、監視システム上で、定期的に各プロジェクトの進捗・品質情報を監視する。ツール A の記載内容と比較して、正しくプロジェクト情報が反映されていることを確認する。

2.4 評価対象

PMO が利用しているプロジェクト監視システム、Excel をベースにしたプロジェクト管理ツールを実証実験の対象とした。Excel 帳票は、開発フェーズごとに分かれており、以下に示す進捗・品質情報が記載されている。

- ・進捗：各フェーズの期間（予定/実績）、各フェーズのタスク数(予定/実績)
- ・工数：各フェーズの作業工数(予定/実績)
- ・品質：各フェーズの検出バグ数(予定/実績)
- ・規模：設計フェーズでは页数(予定/実績)、実装フェーズではライン数(予定/実績)、テストフェーズではテスト項目数(予定/実績)
- ・変更要求：顧客からの要望数(実績)
- ・課題：プロジェクトで発生したレベル別課題数(実績)

また、対象工程は、要件定義～総合テストまでとした。

3 実証結果

3.1 評価結果

2012年12月から2013年9月までの期間、前述の実証システムを構築し実証評価を行い、次の結果が得られた。

- (1) Excelをベースとしたツールや帳票であれば、PROMCODE協議会が提供するフレームワークを用いて、プロジェクトが利用するツールが異なっても、従来通りPMOの監視が可能になることが確認できた。ただし、Information Rights Managementによるアクセス制限がかかっている場合、Excel上に記載されたデータ読み取りには認証が必要になり、対応不可である。
- (2) プロキシは週1回定期的にデータ交換するモード、PMOの指示でデータ交換するモードを備えているため、PMOが必要とするタイミングでリアルタイムに最新のプロジェクト情報を収集できることが確認できた。

3.2 その他の評価

- (1) HTTPサーバに格納する、ツールA向けのマッピングルールは容易に記載できるか否かを評価した。結果、特殊な書き方であるため、PROMCODEを良く理解している人のみ、マッピングルールを定義可能であることが判った。ただし、1日程度で記載ルールの習得が可能と推察する。例えば、ツールAがバージョンアップした場合、ルールの書き直しが発生するが、バージョンアップ頻度は年に1回程度であり、問題ないと思われる。
- (2) セキュリティの観点で問題ないかを評価した。結果、監視システムに認証を設けているため、収集したデータは正当な人が登録するのと同等のセキュリティレベルであることが判った。ただし、HTTPサーバで、PROMCODE形式のプロジェクト情報が公開されてしまうため、HTTPサーバの認証機構を用いて、正当なプロキシからのリクエストであることを判別する仕組みが必要となる。
- (3) 監視システムに代行でデータ登録する人の工数削減につながるかを評価した。結果、PROMCODEのデータ交換の仕組みにより、定期的な収集、転記作業が自動化されるため、手作業で行っていた登録作業を削減できることが判った。

4 考察

事前準備を行えば、部門ごとに独自の帳票（Excel 形式）で管理していたプロジェクト情報は、監視システムに容易に登録できるようになる。類似プロジェクトを比較することで、早期のリスク発見につながることを期待できることが判った。

一方、OSLC、PROMCODE の適用において、不明な点も明らかになった。Change モデルは、プロジェクト管理データの変更履歴を表現するリソースモデルである。今回の実証実験では、Excel 用アダプタが自動的に Change モデルを生成するわけではないため、各社で独自に活用することになる。活用方法が、標準的な使い方に違反していないか、検証と確認が必要となることが見込まれる。

5 今後の予定

実ツールを用いた実証実験を継続して行い、実ツールでの利用、実データ量を想定した実験を進め、認証方法含め管理業務上の運用課題を明らかにする。その後、NEC 内クラウドのプロジェクト管理基盤にデータが登録できるよう関連機能の強化を行い実験範囲を拡大し、管理業務上の課題をクリアした場合、NEC グループ内での本番適用を順次行う予定である。

6 まとめ

Common Resource Model は、プロジェクト管理における本質的な情報を表現しつつ、拡張性も高く設計されている。NEC では、社内のプロジェクト管理ツールやシステム間の連携は、結合度が高い方式で実現してきた。Common Resource Model を活用すれば、ツールやシステム間の結合度が下がり、より多くのツール・システム間で、リアルタイムな連携が実現できることを期待できる。