

# 協調的プロジェクト管理のための共通リソースモデルの提案

吉田裕之<sup>†1</sup> 藪田和夫<sup>†1</sup> 青山幹雄<sup>†2</sup>

本稿では、組織を越えた協調的プロジェクト管理におけるデータ交換のための共通リソースモデルを提案する。協調的プロジェクト管理の重要な課題の一つは、組織間での管理データの差異である。しかし、これまで、プロジェクト管理データモデルの提案は個別的で、共通データモデルが確立されているとは言えない。我々は、プロジェクトの管理内容は組織によらず共通性があることから、管理データ構造にも共通性があると考えた。そこで、国内の主要なITベンダ6社と大学で構成するPROMISコンソーシアムで、実際に使用しているプロジェクト管理データを分析し、共通リソースモデルを策定した。共通リソースモデルは、Scopeltem, WorkItem, Artifactの三つの管理単位を中核としてプロジェクト管理ドメインモデルを構成する。さらに、共通リソースモデルの有用性を検証するための実証実験を進めている。

## A Common Resource Model for Collaborative Project Management

HIROYUKI YOSHIDA<sup>†1</sup> KAZUO YABUTA<sup>†1</sup> MIKIO AOYAMA<sup>†2</sup>

This article proposes a common resource model for exchanging the project management data over the Web across the organizational boundaries. One of the major obstacles in the collaborative project management across the organizational boundaries is the differences and varieties of the management data between projects. However, we believe the essential data structure for the project management should be same despite the differences in the appearance of data. Therefore, we intensively analyzed the management data of real projects collected from six major IT companies of PROMIS consortium, and proposed a common resource model of the project management data. The resource model structured the project management domain model in terms of management objectives including Scopeltem, WorkItem, and Artifact. We discuss the effectiveness of the common resource model through empirical studies conducted in PROMIS consortium.

### 1. はじめに

ソフトウェア開発は、自社内で開発リソースを調達して進める「インハウス型」と、開発リソースを保有している外部の専門会社に発注する「受託型」に大別される。後者は、日本のシステム開発に典型的であり、特に大規模開発の場合は多数の組織が数段階の受発注関係が連鎖したサプライチェーンを構成する。欧米ではインハウス型が中心であったが、オフショア開発の進展で、受託型と同様な受発注が連鎖する開発組織構造も取り入れられている。

受託型開発においては、工数の予実や人員の割当てなどの開発リソースの管理は受注側に任せられる。しかし、プロジェクト全体の進捗や成果物の品質は、発注側を主体として受注側と協力しながら管理するのが基本である。このような形態のプロジェクト管理を本稿では、「協調的」プロジェクト管理と呼ぶ。

多数の組織をまたがってプロジェクト管理のデータを収集するためには、発注者と受注者との間で作業項目や成果物の内容・名称などを整合し、データモデルを統一する必要がある。しかし、これまで合意された標準のプロジェクト管理データモデルは存在せず、プロジェクトごとに異なっているため、データの変換を行う必要がある。特に、受注者が複数の発注者に管理データを提出する場合、発注者

ごとに変換する必要がある。

プロジェクト管理ツールはプロジェクトごとにカスタマイズが必要になるので普及が阻害され、カスタマイズが簡便に行えるスプレッドシートを使ったプロジェクト管理が好まれている。スプレッドシートをメールで交換しながら集約する方式は、多大な間接工数を費やすだけでなく、リアルタイム性にも欠けている。

このような課題を解決し、開発組織をまたがったリアルタイムなプロジェクト管理データ交換を実現するために、南山大学、IBM、富士通、日本電気、NTTデータ、日立、野村総研はPROMIS (PROject Management Information exchange Services)コンソーシアムを2012年5月21日に設立した[3]。PROMISコンソーシアムは、W3Cが提唱するLinked Data技術に基づき、プロジェクト管理データをWeb上で相互にリンクし、交換するための標準技術を開発している[8]。本稿では、PROMISコンソーシアムで開発した共通リソースモデルとその評価を報告する。

本稿の構成は以下の通りである。第2章は関連研究について議論する。第3章は、協調型プロジェクト管理の実管理データの分析に基づいて抽出した課題を述べる。第4章は、Linked Dataに基づくアプローチを議論する。第5章は提案する共通リソースモデルを示し、第6章はその実際のプロジェクト管理データへの適用を示す。第7章で適用による評価を議論する。最後に第8章でまとめとする。

<sup>†1</sup> 富士通株式会社  
FUJITSU LIMITED

<sup>†2</sup> 南山大学  
Nanzan University

## 2. 関連研究

プロジェクト管理の知識体系である PMBOK[13]や標準ガイドラインである ISO21500[7]は、プロジェクト管理のプロセスや用語などの標準化に寄与しているが、管理データモデルの標準化までは対象としていない[6]。一方、プロジェクト管理のデータモデル、あるいは、その基礎となるオントロジの研究がある。

Abelsらは、プロジェクトの主要な要素とその間の関係を特定し、PROMONT (PROject Management ONTology)として整理した[1]が、管理データ全体のモデルを提示していない。

Ahlemnnは、28個のプロジェクト管理ソフトウェアを分析し、プロジェクト管理の参照モデル RefMod を定義し、その中でプロジェクトマスターデータを定義している[2]。しかし、その要素は、プロジェクトの抽象化定義 Initiative と工程 LifecyclePhase のみから成り、高度に抽象化、簡潔化した結果、実際のプロジェクト管理データの定義とは隔たりがある。

Bērzišaは、プロジェクト管理の要素を合成することによりプロジェクト管理概念モデルを提案している[4]。しかし、その一般性や有効性について評価が示されていない。

特定のプロジェクト管理ツールで用いるデータのスキーマは公開されているものもある[10]が、広く共用できるような標準化には至っていない。

公共機関などのデータ公開方法として RDF (Resource Description Framework)[15]による意味関係を基礎とする Linked Data が注目されている[5, 14]。Linked Data を基礎として、要求管理、構成管理、変更管理などを行うツール連携のための標準リソースモデルが OSLC (Open Services for Lifecycle Collaboration)により提案されている[12]。しかし、プロジェクト管理については提案されていない。

## 3. 協調的プロジェクト管理の実態と課題

PROMIS コンソーシアムは、協調的プロジェクト管理における管理データのリアルタイムな交換の実現を目指している。前述のように、その主な阻害要因はプロジェクトごとに異なる管理データモデルにある。

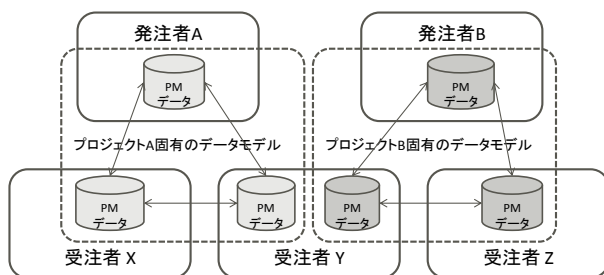


図1 プロジェクトごとに異なる管理データモデル

Figure 1 Management Data in Different Data Models

例えば図1では、A社が発注者として管理するプロジェ

クトAにX社とY社が受注者として参画し、同様にB社のプロジェクトBにY社とZ社が参画している。Y社は、プロジェクトAとBの異なる管理データモデルに個別に対応する必要があるため、プロジェクト管理ツールを導入するメリットを得にくい。結果として、プロジェクトごとに異なるデータ形式のスプレッドシートを使って、手作業で管理するケースが多くなる。図1は2階層のみを示しているが、実際の大規模プロジェクトでは受発注関係が多階層に及び、参画する組織も多数であり、プロジェクト全体の間接工数が多大なものになっている。

こうした受発注階層における実態と課題を理解するために、PROMIS コンソーシアムのメンバ6社が実際のプロジェクトで使用している管理データを持ち寄って分析した。その結果、協調的プロジェクト管理では主に「進捗管理」「品質管理」の二つの管理作業が重視されていることが分かった。また、ほとんどのデータは二次元のテーブル形式を持ち、その多くがスプレッドシートで表現されていた。

組織間の管理データの相違は、大きく二種類に分類できる。一つ目は管理対象の違いである。これは、各社の開発プロセスの定義が異なるためである。例えば、作業工程の粒度や階層構造、名称が異なり、それらによって生産される成果物も異なり、その目的、内容、名称が異なっている。

二つ目は形式の違いである。二次元テーブル形式という共通点はあるものの、行と列の定義が異なる。予定と実績を行と列のどちらに配置するか、プロジェクト全体の情報を一つのテーブルで管理するか複数に分割するか、など様々なバリエーションがある。さらに、一つの企業内でも必ずしも統一されず、部門ごと、プロジェクトごとにバリエーションがある。

テーブル形式のデータを交換する標準データ形式としてCSV(Comma-Separated Values)がある。実際、6社が発注者となってリードするプロジェクトにおいては、受注者からCSV やスプレッドシートで管理データを受け取ることが多い。このため、特に多階層の受発注構造を持つ大規模プロジェクトでは、下位層の組織がプロジェクトごとに上位層が設定した異なる形式の CSV やスプレッドシートで管理データを提出することを求められている。

このような状況で、例えば図1のプロジェクトA向けのツールを、プロジェクトB向けにカスタマイズするのは容易ではない。その理由は、第一に、物理的なテーブル形式に管理データの意味構造が束縛されているため、異なる形式どうしを対応付けることが困難だからである。第二に、プロジェクトごとに異なる開発プロセスを採用する以上、異なる管理対象物どうしを対応付けることもまた困難だからである。

二次元テーブル形式によるインターフェースに代えて、管理データの物理形式に束縛されず、任意の管理対象に対応付けが可能な、共通インターフェース策定が必要である。

## 4. アプローチ

プロジェクト管理データは、本来は、多種類の管理対象からなる多次元の情報空間を構成すると考えられる。それをプロジェクトごとに異なる方法で二次元に射影したものが現状のデータ構造であり、そのために共通構造を見出すことが難しい。本来の意味構造を適切に表現できる形式が利用できれば、共通インタフェースを設計し易いと期待できる。

Linked Data 技術は意味構造をリンクとして柔軟に定義できることから、任意の意味構造に基づいたデータ交換が共通のインタフェースで実現できる[11]。例えば、図 1 のシナリオでは、受注者 Y が Linked Data に基づくインタフェースを実装する開発ツールを採用すれば、発注者 A と B のそれぞれ異なる意味構造を持つ管理データを交換できる。

ただし、Linked Data リソースのスキーマを個別に定義すると、例えば発注者 A は受注者 Y からリソースを受け取ったあとで、その意味構造を解析してプロジェクト A のデータモデルとして理解しなければならない。この手間を避けるためには、交換するリソースに一定の制約を課し、各プロジェクトのデータモデルとの間でのスキーママッピングを容易にする必要がある。我々は、プロジェクト間のデータモデルのバリエーションを吸収できるように、適切な抽象度のドメインモデルを定め、それに基づいてリソースの標準スキーマを定義するアプローチを採用した。

## 5. 提案する共通リソースモデル

### 5.1 PROMIS モデリングフレームワーク

図 2 に示す PROMIS モデリングフレームワークを提案する。これは、共通リソースモデルの位置づけと利用方法を定義する。

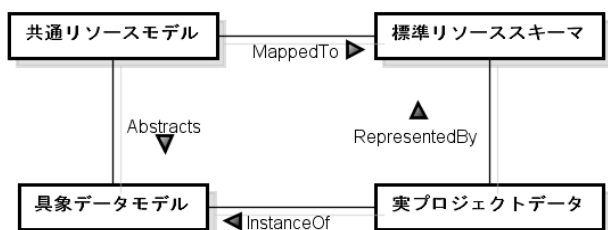


図 2 PROMIS モデリングフレームワーク

Figure 2 PROMIS Modeling Framework

「共通リソースモデル」は、協調的プロジェクト管理の抽象ドメインモデルである。このモデルに基づき、管理データを Linked Data リソースとして交換するためのスキーマを「標準リソーススキーマ」として策定した。標準リソーススキーマの各リソース定義は、共通リソースモデルの各クラスから 1 対 1 にマッピングして定義する。このように抽象ドメインモデルと特定のスキーマモデルでのスキーマ定義とを分離することにより、異なるスキーマモデルにも対応可能となる。

「具象データモデル」はプロジェクト個別のデータモデルであり、共通リソースモデルが定義する抽象クラスのサブクラス群としてプロジェクトごとに定義する。具象データモデルのインスタンスが「実プロジェクトデータ」となる。実プロジェクトデータは、共通リソースモデルの抽象クラス群のインスタンスとなるので、標準リソーススキーマに基づく Linked Data としてそのまま表現でき、リンクや交換が可能となる。

### 5.2 PROMIS 共通リソースモデル

PROMIS 共通リソースモデルは、協調的プロジェクト管理の抽象ドメインモデルである。我々は各社の管理データ構造に共通する次の三つの抽象クラスを抽出した。

#### (1) WorkItem

進捗管理データの共通点に着目し、プロジェクトを遂行する上で受注者が実施する「活動」を表現するオブジェクト群を WorkItem とした。これらは、例えば分析、設計、実装、テストのような開発工程を表すものもある。一方、設計書執筆、レビュー、指摘反映のような具体的な作業を表すものもあり、様々な粒度のオブジェクトが存在する。これらは開始と終了という二つのイベントが存在する点で共通性があり、それらの予定と実績の日付を比較することで進捗を管理する。

#### (2) Artifact

品質管理データの共通点に着目し、開発プロジェクトの成果物として作成される「具体的な実体」を表現するオブジェクト群を Artifact とした。これらは、例えばプログラムやモジュールのように計算機上で動作するものもある。仕様書、設計書、テスト結果報告書などのドキュメントもある。これらは、何らかの指標を測定して品質を評価できる点に共通性がある。品質指標の中には、一度測定するだけでなく日々測定してその変化を管理する場合もある。品質指標を Measure と呼ぶことにした。

#### (3) ScopeItem

協調的開発における発注者(顧客)と受注者間で取り交わされる開発契約に対して、開発対象ソフトウェアが提供する「価値の単位」を表現するオブジェクト群を ScopeItem とした。これらは、例えば要求、提供機能、ユースケース、画面などがある。ScopeItem は WorkItem とは異なり、開始、終了のイベントを持たない。例えば、特定の「要求」に開始、終了があるのではなく、それを遂行する分析や設計という WorkItem に開始、終了が定義される。また、ScopeItem は Artifact とは異なり、測定可能な実体を持たない。「要求」自体を品質指標で測定するのではなく、成果物である仕様書やレビュー報告書という Artifact を測定する。

顧客価値を考慮せずに WorkItem と Artifact のみを管

理することは受注者内での進捗と品質の管理に過ぎなく、協調的プロジェクト管理の対象とはならない。顧客の視点から管理することは、協調的プロジェクト管理における本質的な特性である。各 WorkItem や Artifact を ScopeItem に関連付けることで、開発における活動や成果物の価値が認識でき、進捗遅れや品質低下が顧客に及ぼす損害などの影響を判断できるようになる。ScopeItem に優先順位を設定すれば、それに関連する WorkItem や Artifact の重要度も評価でき、プロジェクトの遂行に問題が起きた場合に適切に対処できる。

現時点における PROMIS 共通リソースモデルを図 3 に示す。ManagedItem は、ScopeItem, WorkItem, Artifact の三種の管理単位を抽象化したスーパークラスである。Change は ManagedItem の履歴を管理するために導入したクラスである。Issue は、To Do やバックログなど、ManagedItem に関わる課題を表すクラスであり、今後、課題管理に利用することを想定している。WorkItem には、TimeSpan の計画値と実績値、および責任者である Person が関連する。Artifact には、日々計測される適切な Measure が関連する。

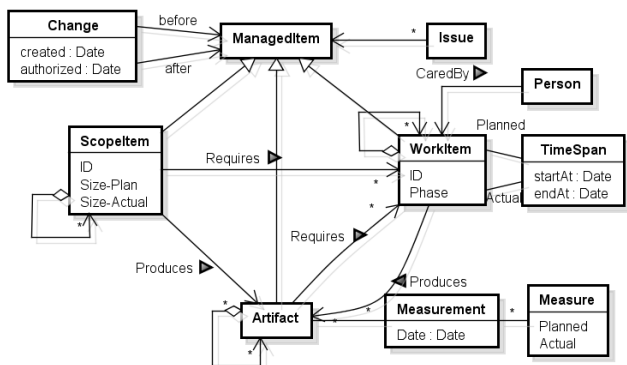


図 3 PROMIS プロジェクト管理共通リソースモデル  
Figure 3 PROMIS Common Resource Model

## 6. 共通リソースモデルの適用

### 6.1 進捗管理の実データへの適用

各開発プロジェクトは、PROMIS 共通リソースモデルを継承し、それぞれの固有の具象データモデルを定義できる。

表 1 プロジェクト A の進捗管理テーブル  
Table 1 Project A's Progress Management Table

機能	サブ機能	分析		設計		実装		
		開始	終了	開始	終了	開始	終了	
A	A1	予定	6/4	6/11	6/12	6/19	6/20	6/27
		実績	6/4	6/10	6/11	6/19	6/20	6/26
	A2	予定	6/4	6/11	6/12	6/19	6/20	6/25
		実績	6/4	6/12	6/13	6/20	6/21	6/26
B	B1	予定	6/4	6/11	6/12	6/19	6/20	6/27
		実績	6/4	6/12	6/13	6/20	6/21	6/28
	B2	予定	6/4	6/11	6/12	6/19	6/20	6/26
		実績	6/5	6/11	6/12	6/19	6/20	6/25

表 1 は、PROMIS コンソーシアムのあるメンバ企業で使っている典型的な進捗管理テーブルである。管理単位は SubFunction(サブ機能)であり、Function(機能)を分割したものである。各 SubFunction は分析、設計、実装工程を通じて開発される。表 1 は管理テーブルの本質的構造のみを示し、実際の管理テーブルはより多くの列から成り、Function-SubFunction 構造は数レベルの木構造を持つ。

表 1 に相当する具象データモデルの例を図 4 に示す。

Function と SubFunction は ScopeItem のサブクラスであり、Analysis, Design, Coding は WorkItem のサブクラスである。Function は 1 つ以上の SubFunction に分解され、さらに三種類の WorkItem に分解される。この具象データモデルにより表 1 のすべての管理データは、PROMIS 共通リソースモデルが定める抽象クラスである ScopeItem, WorkItem, TimeSpan のインスタンスとして表現できる。

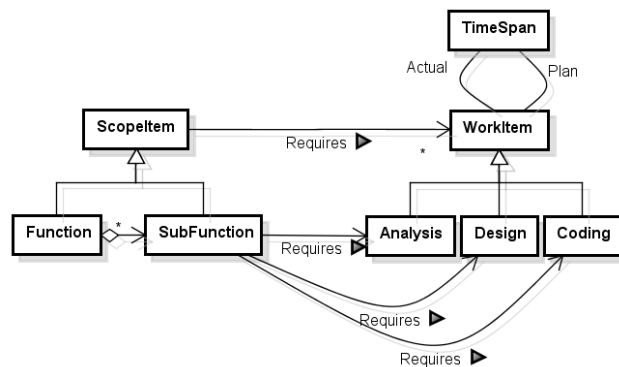


図 4 プロジェクト A の進捗管理具象データモデル  
Figure 4 Concrete Data Model of Progress Management

### 6.2 品質管理の実データへの適用

表 2 は、PROMIS コンソーシアムの別のメンバ企業で使っている典型的な品質管理テーブルである。ここでは管理単位は Module(ソースモジュール)である。Module は Requirement(要求)で分類されている。各 Module は、行数、テストケース数、障害数などの Measure で品質を測定する。

表 2 プロジェクト B の品質管理テーブル  
Table 2 Project B's Quality Management Table

要求	モジュール	行数		テストケース数		障害数	
		予定	実績	予定	実績	予定	実績
R1	M1-1	2,000	2,130	60	62	5	5
	M1-2	1,500	1,450	45	43	3	2
R2	M2-1	2,000	1,980	60	65	5	4
	M2-2	1,000	950	30	35	2	2
	M2-3	1,200	1,100	38	35	2	0

表 2 に相当する具象データモデルを図 5 に示す。

Requirement は ScopeItem のサブクラスであり、Module は Artifact のサブクラスである。Measure のサブクラスは LOC(行数), #TC(テストケース数), #Defect(障害数)の三種類がある。Requirement は、それを実現するために開発する

Module に分解される。この具象データモデルにより表 2 のすべてのプロジェクト管理データは、PROMIS 共通リソースモデルが定める抽象クラスである Scopeltem, Artifact, Measure のインスタンスとして表現できることになる。

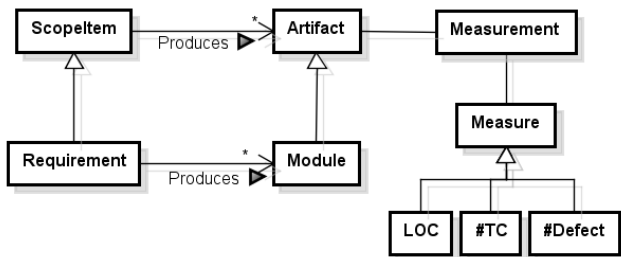


図 5 プロジェクト B の品質管理具象データモデル  
Figure 5 Concrete Data Model of Quality Management

### 6.3 実データから Liked Data へのマッピング

標準リソーススキーマの各リソース定義は、共通リソースモデルの各クラスから 1対1にマッピングして定義する。例えば、共通リソースモデルの WorkItem クラスに対して、RDF スキーマを使う標準リソーススキーマでは WorkItem リソースを表 3 のように定義する。「dcterms」は Dublin Core 共通語彙の名前空間であり、「promis\_pm」は PROMIS コンソーシアムが定めるプロジェクト管理リソースの名前空間を表すプレフィックスである。

表 3 WorkItem のリソース定義  
Table 3 Resource Definition of WorkItem

属性		
名前	型	内容
dcterms:title	XMLLiteral 型	タイトル文字列
dcterms:identifier	String 型	識別子
dcterms:type	String 型	具象サブクラス名
promis_pm:plannedStartAt	Date 型	開始計画日
promis_pm:plannedEndAt	Date 型	終了計画日
promis_pm:actualStartAt	Date 型	開始実績日
promis_pm:actualEndAt	Date 型	終了実績日
… (以下省略)		
関連		
名前	型	関連先
promis_pm:requiredBy	Resource 型	この WorkItem を必要とする Scopeltem または Artifact, または親の WorkItem
promis_pm:consistsOf	Resource 型	子の WorkItem
promis_pm:caredBy	Resource 型	この WorkItem の責任者である Person
… (以下省略)		

実際のプロジェクトデータは、プロジェクトごとの具象データモデルのインスタンスであり、共通リソースモデルの抽象クラスのインスタンスになるので、標準リソーススキーマに基づくリソース表現に自然にマッピングできる。例えば、表 1 のサブ機能 A1 の「分析」作業は WorkItem のサブクラスのインスタンスとなるので、このリソース定義

によって図 6 に示す Linked Data として表現される。dcterms:type の指定により、このリソースは WorkItem クラスの具象サブクラスである「Analysis」のインスタンスであることが示される。

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:dcterms=http://purl.org/dc/elements/1.1/
  xmlns:promis_pm="http://promis.jp/ns/pm/">
  <promis_pm:WorkItem rdf:about=
    "http://localhost:8080/oslc_rest/services/projectA/travel1/WI1">
    <dcterms:title>分析</dcterms:title>
    <dcterms:identifier>A1-Analysis</dcterms:identifier>
    <dcterms:type>Analysis</dcterms:type>
    <promis_pm:plannedStartAt>2012-06-04T00:00:00JST
      </promis_pm:plannedStartAt>
    <promis_pm:plannedEndAt>2012-06-11T00:00:00JST
      </promis_pm:plannedEndAt>
    <promis_pm:actualStartAt>2012-06-04T00:00:00JST
      </promis_pm:actualStartAt>
    <promis_pm:actualEndAt>2012-06-10T00:00:00JST
      </promis_pm:actualEndAt>
    <promis_pm:requiredBy rdf:resource=
      "http://localhost:8080/oslc_rest/services/projectA/travel1/SFunc1"/>
    … 一部省略 …
  </promis_pm:WorkItem>
</rdf:RDF>
```

図 6 実データから Linked Data へのマッピング例  
Figure 6 Sample Mapping from Actual Data to Liked Data

## 7. 評価

### 7.1 提案した共通リソースモデルの抽象化能力

共通リソースモデルと具象データモデルを分離して定義した結果、実際のプロジェクト管理データは共通リソースモデルが提供するクラスのインスタンスとみなせるようになる。それらは RDF リソースとして表現できることから、プロジェクト管理ツールはこれらのリソース表現を相互にリンク、交換できるようになる。表 1 と表 2 の例はそれぞれ異なるデータモデルを使っているが、同一ツールをほぼカスタマイズせずに利用できる。実際のプロジェクト管理データが異なっても、Linked Data としてその構造を構成できるので、データへのアクセスが標準化できる。

### 7.2 2次元テーブル構造へのマッピングの課題と対処

図 3 に示すように、Scopeltem, WorkItem, Artifact の三つの情報要素は三次元の情報空間を構成する。スプレッドシートのようなテーブル形式は二次元の情報空間しか表現できないので、三次元情報空間から二次元情報空間への射影が必要となる[9]。ここで三次元情報空間の要素は、図 3 で表されるように関連を持つことから、その射影では関連とその意味を保存する必要がある。実際のプロジェクト管理では、図 3 に示すモデルに基づくことなく、アドホックにプロジェクト管理データのテーブル形式を規定していることが多いため、テーブル間でデータ定義に一貫性が保証されず、バリエーションが生じることとなる。

本稿で提案した共通リソースモデルは、プロジェクト管

理の共通参照モデルの役割を果たすことができる。この参照モデルを基準として、二つの適用例で示したように、各プロジェクトにおける管理データがどのように射影されているか評価できる。この点からも、共通リソースモデルの役割は重要であるといえる。

さらに、図 1 に示すシナリオでは、多くの受注者が発注者指定の形式のスプレッドシートを利用している。このような状況から、標準リソーススキーマによるデータ交換へ容易に移行できるように、我々はスプレッドシート用アダプタを試作した。このアダプタを利用すれば、プロジェクトは具象モデルへのマッピング定義を記述するだけで、既存の管理データを容易にリソース表現に変換できる。

### 7.3 実例による提案リソースモデルの妥当性検証

当初分析した管理データに加え、メンバ企業それぞれの複数の実プロジェクトで、具象モデルの有効性を検証した。ある管理データは、一つのテーブルで進捗と品質の双方を管理していた。このテーブルの各行はソースコードのような *Artifact* を表し、それを進捗管理にも使っている。共通リソースモデルでは、進捗管理の単位は *WorkItem* であり、*Artifact* と *TimeSpan* に関連を設けていない。このケースでは、各 *Artifact* を生産するための暗黙の *WorkItem* が 1 対 1 に存在すると考えることができる。これにより、*Artifact* を単位とする品質管理と *WorkItem* を単位とする進捗管理が、一つのテーブルにマージされているとみなせる。

逆に、*WorkItem* を品質管理の単位としている実データも存在した。共通リソースモデルでは、*WorkItem* は *Measure* とは関連を持たない。この例でも、各 *WorkItem* によって生産される暗黙の *Artifact* が存在するとみなせる。

テーブルのレイアウト上のバリエーション、用語の違い、管理作業の粒度の違いなどは、具象データモデル策定を阻害するものではなかった。管理単位ごとに一枚の表、つまり単票形式を使っているプロジェクトもあった。このレイアウトでは管理単位の名称やその分類を 1~2 行目付近に記述し、それ以降に進捗データや品質データを記述する。このようなレイアウトであっても、共通リソースモデルに基づく具象データモデルを容易に策定できた。実際に、PROMIS コンソーシアムのメンバ企業による全てのプロジェクト管理データに対し共通リソースモデルに基づき具象データモデルが適切に定義できることを確認した。

このようなプロジェクト管理データの共通リソースモデルの定義と検証は、これまでになかった成果である。

## 8. まとめ

本稿は、プロジェクトや組織の境界を越えてプロジェクト管理データを交換するための、PROMIS モデリングフレームワークと共通リソースモデルを提案し、その妥当性を議論した。モデリングフレームワークは、プロジェクト管

理のドメインモデルとそれを利用するプロジェクトごとのデータモデルを分離し、プロジェクト管理データの共通性を実現する。共通リソースモデルは、プロジェクト管理の抽象ドメインモデルであり、これを継承することで各プロジェクト固有の具象データモデルを定義でき、管理データをリンク可能なリソース表現にマッピングできる。

共通リソースモデルは PROMIS コンソーシアムのメンバ企業が持ち寄った実際の管理データの詳細分析から策定し、さらに複数のプロジェクトで有効性を評価した。プロジェクト管理データが、*ScopeItem*, *WorkItem*, *Artifact* で抽象できることを提案した。これらの管理単位は三次元の情報空間を構成し、スプレッドシートのような管理テーブルはこの情報空間の二次元射影として扱えることを示した。

筆者らの知る限り、一定の範囲で適用可能性をプロジェクト管理データの共通リソースモデルの定義は他にない。

今後、標準リソーススキーマに基づくプロトタイプを OSLC Core 上に実装し、実証実験により提案共通リソースモデルの有用性を検証する予定である。さらに、共通リソースモデルの国際標準化も予定している。

**謝辞** 本稿の検討に協力頂いた PROMIS コンソーシアムのメンバ各位に感謝する。

## 参考文献

1. Abels, S., et al.: PROMONT – A Project Management Ontology as a Reference for Virtual Project Organizations, Proc. OTM Workshop 2006, LNCS Vol. 4277, Springer, pp. 813-823 (2006).
2. Ahlemann, F.: Towards a Conceptual Reference Model for Project Management Information Systems, Int'l J. of Project Management, Vol. 27, No. 1, pp. 19-30 (2009).
3. Aoyama, M., et al.: PROMIS: A Next-Generation Project Management Data Exchange Architecture, Proc. ProMAC 2012, pp. 493-500 (Oct. 2012).
4. Bērziša, S.: Towards an XML Schema for Configuration of Project Management Information Systems: Conceptual Modeling, Proc. ADBIS 2009, LNCS Vol. 5968, pp. 229-237 (2010).
5. Bizer, C., et al.: Linked Data - The Story So Far, Int'l J. on Semantic Web and Information Systems, Vol. 5, No. 3, pp. 1-22 (2009) [荻野達也(訳), Linked Data の仕組み, 情報処理, Vol. 52, No. 3, pp. 284-292 (Mar. 2011)].
6. Dippelreiter, B., et al.: Semantic Technologies for the Project Management Life Cycle Improvement, Proc. 6<sup>th</sup> Int'l Workshop on Semantic Business Process Management, LNCS Vol. 7117, Springer (2011).
7. ISO, ISO 21500:2012, Guidance on Project Management (2012).
8. Kamimura, T., Ohsawa, K., Speicher, S., and Wakao, M.: Applying OSLC in PROMIS initiative, Proc. ProMAC 2012, pp. 408-514 (Oct. 2012).
9. Lenzerini, M.: Data Integration: A Theoretical Perspective, Proc. ACM PODS '02, pp. 233-246 (Jun. 2002).
10. Microsoft, Introduction to Project XML Data, [http://msdn.microsoft.com/en-us/library/office/bb968652\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/office/bb968652(v=office.12).aspx).
11. Mulwad, V., et al.: Using Linked Data to Interpret Tables, Proc. COLD 2010, pp. 1-12 (Nov. 2010) <http://people.aifb.kit.edu/aha/2010/cold/>.
12. OSLC: Open Services for Lifecycle Collaboration, <http://open-services.net/>.
13. PMI: A Guide to the Project Management Body of Knowledge, 4<sup>th</sup> ed., PMI (2008).
14. W3C: Linked Data, <http://www.w3.org/standards/semanticweb/data>.
15. W3C: RDF (Resource Description Framework), <http://www.w3.org/RDF/>.