

## ソフトウェアプロセスのグラフモデル表現に基づく プロジェクト事例共有システムの開発

粕淵 健郎<sup>†</sup> 飯田 元<sup>††</sup> 松本 健一<sup>†</sup> 田中 康<sup>‡†</sup>

<sup>†</sup> 奈良先端科学技術大学院大学情報科学研究科 〒630-0192 けいはんな学研都市

<sup>††</sup> 奈良先端科学技術大学院大学情報科学センター 〒630-0192 けいはんな学研都市

<sup>‡</sup> ソニー(株) ネットワーク&ソフトウェアテクノロジーセンター 〒108-6201 東京都品川区港南 2-15-3

E-mail: <sup>†</sup> {takeo-ka, matumoto}@is.aist-nara.ac.jp, <sup>††</sup> iida@itc.aist-nara.ac.jp, <sup>‡</sup> Yasushi.Tanaka@jp.sony.com

あらまし ソフトウェア開発における過去のプロジェクト事例を組織内で共有し、必要なときに適切な事例を検索して参照できる手法を提案する。本手法では、作業成果物などをプロジェクトのソフトウェアプロセス定義(PDSP)と関連付けられたグラフモデルとして記録し、それをプロジェクト事例として事例データベースに登録する。PDSPを開発時のコンテキストを表現する手段として利用することで、ユーザはコンテキストの類似した事例を参照することができるようになる。本稿ではこの手法におけるプロジェクト事例のデータ形式および検索手法、ならびに事例共有システムのプロトタイプ的设计について述べ、その有効性を考察する。

キーワード ソフトウェアプロセス, 知識共有, 情報検索, データベース

## Development of Project-Case Sharing System based on Graph Model Expression of Software Process

Takeo KASUBUCHI<sup>†</sup> Hajimu IIDA<sup>††</sup> Ken'ichi MATUMOTO<sup>†</sup> and Yasushi TANAKA<sup>‡†</sup>

{ <sup>†</sup> Graduate School of Information Science, <sup>††</sup> Information Technology Center }, Nara Institute of Science and  
Technology Kansai Science City, 630-0192 Japan

<sup>‡</sup> Sony Corporation, Network & Software Technology Center 2-15-3 Kounan, Shinagawa-ku, Tokyo, 108-6201 Japan

E-mail: <sup>†</sup> {takeo-ka, matumoto}@is.aist-nara.ac.jp, <sup>††</sup> iida@itc.aist-nara.ac.jp, <sup>‡</sup> Yasushi.Tanaka@jp.sony.com

**Abstract** In this paper, we propose a design of the process knowledge sharing system for software development organization that the developers can accumulate the cases of past projects and can reuse them. In our system, set of process elements generated in a project, such as software process component and software work product, constitute the software process as the graph model representation. All process elements and graph models are stored in a database, and the system offers the feature of graph model based search for past process data having similar context of execution. By using this system, developers can efficiently retrieve past successful process data that can be easily adopted to the current project situation for reuse.

**Keyword** Software Process, Knowledge Sharing, Information Retrieval, Database

### 1. はじめに

ソフトウェアを開発している多くの組織において、開発プロジェクトに関する事例やノウハウを組織情報として資産化し、共有することの重要性が高まっている。例えば CMM(Capability Maturity Model)[4]では、組織のプロセス資産を使って過去のベストプラクティスを反復したり、組織標準のプロセスを資産化したりすることを、開発成熟度の高い組織の要件として掲げている。また、ナレッジマネジメントの技術を使って組織が蓄積した経験を検索できるデータベースを構築する試みも行われている[2]。

通常、情報検索を行うことのできるデータベースシステムでは、「どのような情報を検索キーとして利用できるか」、がそのシステムの有効性を大きく左右する。最もオーソドックスな検索キーとしては、“対象に含まれる単語”が挙げられるが、それだけでは機能として不十分な場合も多く、他の様々な情報を検索キーに含められるシステムが開発されてきている。

ソフトウェア開発における組織資産を対象として検索する場合には、それが開発プロセスにおけるどのようなコンテキストにおいて利用されるべきものなのかが重要であると考えられる。ソフトウェア開発の事例やノウハウを参照する際に、その情報がソフトウェア開発の

どのような状況で作られたのかによってその価値は異なる。可能であれば、現在の自分が置かれている状況と類似したコンテキストで行われた事例を発見できることが望ましい。

そこで本研究では、ソフトウェア開発のコンテキストに即した事例を有効に参照することのできる「ソフトウェアプロセス事例共有システム」を提案する。ここで提案するシステムは、コンテキストを表現する手段として、ソフトウェアプロセス記述を利用することである。

以降、まず第2節において関連研究を挙げて本研究の位置付けを示す。第3節で本研究におけるプロジェクト事例のフォーマットを示し、第4節で、このフォーマットを用いて類似したコンテキストのプロジェクト事例を発見できる類似検索の手法を説明する。第5節では、提案した事例検索手法を持つ情報共有システムのプロトタイプ構成について述べる。第6節で考察を行い、最後に本稿のまとめと今後の研究課題について述べる。

## 2. 関連研究

本研究に関連する研究として、大きく2種類のカテゴリの研究を紹介する。ひとつはワークフローに連携した組織の情報共有システムについて、もうひとつはグラフモデルというデータ構造化手法とそのデータの類似度計算に関する研究についてである。

### 2.1. 組織情報の共有システム

ソフトウェア開発に限らず、組織やグループで行われるワークプロセスやビジネスプロセスで発生するノウハウ等の情報を体系化して共有するシステムについては数多く研究されている。近年の研究では、情報をワークフローと関連付けて蓄積することで、情報検索の有効性を高めようとするものがある。単純なデータベースシステムでは、情報はプロセスの流れからは切り離されており、その情報の提供者がどのようなコンテキストでそれを提供したのかを知ることができなかった。またその情報に関連する背景情報が存在したとしても、それに適した検索語を指定できなければ有効な利用は行えない。

敷田らは、ノウハウを業務のアクティビティに関連付けて蓄積する、フローに連携した組織情報共有システムを提案している[5]。類似アクティビティを実行するユーザが、それに関連したノウハウを検索することができる。

しかし、これらのワークフロー連携情報共有システムをソフトウェア開発プロセスに適用することを考えると、ソフトウェアプロセスと一般的なワークフローとは、活動とその関連を明記するものという点では類似しているが、ソフトウェアプロセスはより複雑で、様々なバリエーションが発生する。このため、これらのシステム

をもってそのままソフトウェアプロセスの再利用に結びつけるのは難しい。

### 2.2. グラフモデルに基づく類似度検索

そこで我々は、より柔軟に(つまり様々なバリエーションを考慮して)事例検索を行う手法としてグラフモデルに着目した。富田らは、類似文書検索のための手法としてグラフモデルの類似性検索による効率的なデータアクセスを提案している[6]。グラフモデルとは文書検索モデルの一種で、検索キーおよび検索対象文書を、単語の関連に関するグラフとしてモデル化し、これらのグラフ間の類似度計算によってグラフ間の類似度を判定するというものである。本研究では、プロセスモデル記述をプロセス要素間の関連に関するグラフモデルであると考え、この類似性検索システムを応用する。

## 3. プロジェクト事例蓄積データとしてのソフトウェアプロセス記述

### 3.1. ソフトウェアプロセスモデルの概要

ソフトウェアプロセスモデルとは、ソフトウェア開発過程の形式的な記述法のことである。ソフトウェアプロセスモデルとしてはさまざまなものが提唱されており、近年ではオブジェクト指向プロセスモデリングについての研究が行われている[3]。

我々の研究グループでは、オブジェクト指向プロセスモデルの研究の一環として、コンポーネントベースのプロセス記述手法を策定している。これは、プロセスの典型的な部分タスクを再利用可能なレベルに抽象化したものをプロセスコンポーネントとして定義しておき、プロセスモデリングはこのプロセスコンポーネントを利用することにより行うものである。本研究におけるソフトウェアプロセスモデルの記述は、このコンポーネントベースのプロセスモデルを元に設計している。

以下に、我々が想定するソフトウェア開発プロジェクトの典型的な流れと、その過程で生成されるソフトウェアプロセス記述の状態を示す。プロジェクト開始時にはプロジェクトの責任者がそのプロジェクトにおけるソフトウェアプロセスモデルを計画し、PDSP(Project's Defined Software Process)[4]を定義する。PDSPには、プロジェクトで使用するプロセスコンポーネントを実行可能なレベルに実体化したもの(プロセスコンポーネントインスタンス、第3.2節で詳述)と作業成果物、およびそれらの相互関連が定義される。以後、プロセスコンポーネントや作業成果物をまとめてプロセス要素と呼ぶ。その後PDSPからプロジェクト計画が作られ、PDSPとプロジェクト計画に基づいて開発が進行する。そして各作業が実行されるにつれ、成果物の実体が生成されていく。このとき、元々PDSPに定義されているプロセス要素の

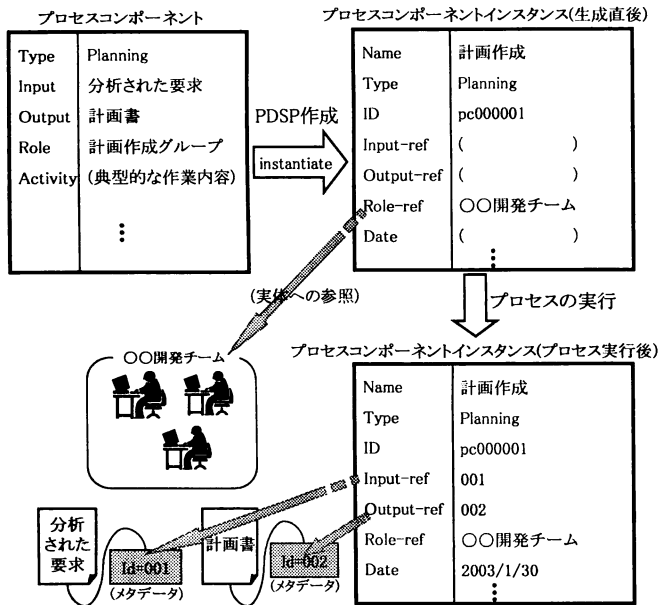


図 1 プロセスコンポーネントとインスタンス

記述に対し、実際の作業結果や作成された成果物へのリンクが追加される。このようにしてプロセス要素の実体に PDSP リンクを付与したものが、プロジェクト事例としてプロジェクト事例 DB に登録される。こうして登録されたプロジェクト事例に含まれるプロセス要素は、他のプロジェクトにおけるプロセス定義段階や実行段階において再利用される。

### 3.2. プロセス要素の定義

#### ● プロセスコンポーネントとそのインスタンス

プロセスコンポーネントはプロセスの典型的な部分タスクを再利用可能なレベルに抽象化したもので、図 1 に示すような値を含んでいる。PDSP 作成時には、このコンポーネントに対し名前や一意な ID、担当者名などの情報を加えることでインスタンス化し、PDSP の記述に含めて利用する。これをプロセスコンポーネントインスタンスと呼ぶ。生成直後のプロセスコンポーネントインスタンスにはまだ成果物の実体への参照リンクは持たないが、プロセスが実行されるに従って参照リンクが加えられていく。

#### ● ソフトウェア作業成果物

ソフトウェア作業成果物とは「ソフトウェアプロセス活動において作成される全てのもの」[4]を指し、ソフトウェアプログラム、開発計画書、会議の議事録などがこれにあたる。プロセスコンポーネントと違い、ソフトウェア作業成果物の実体はデータ形式が統一されていないため、ここでは統一したフォーマットのメタデータを用意し、このメタデータに背景情報および実体へ参照を記述することで検索を可能にする。

エレメント名	識別子	限定子	説明	備考
タイトル	Title		資源に与えられた名前	
資源識別子	Identifier		識別のためのIDやURN	
作成者	Creator		作者の名前	
		Manager	管理者	IDによる参照
		Executor	実施者	IDによる参照
主題及びキーワード	Subject		キーワード	
内容記述	Description		文書の概要	
公開者	Publisher	Organization	組織名	IDによる参照
		ProjectName	プロジェクト名	IDによる参照
寄与者	Contributor		リソースの内容に貢献している人・団体	
日付	Date	Created	作成日	
		Start	実行開始日	日付
		Finish	実行終了日	日付
		Valid	有効期日(期間)	
		Available	利用可能日(期間)	
		Issued	正式発行日	
		Modified	更新日	
資源タイプ	Type	KindOf	プロセス要素の種類	プロセスコンポーネントor成果物
		implementOf	抽象データへの参照	コンポーネントの種類or成果物の種類
形式	Format		フォーマット	
資源識別子	Identifier		識別のためのIDやURN	
出处	Source		このメタデータの参照元	
言語	Language		言語	
関連	Relation	IsVersionOf	リソースのバージョン	
		HasVersion	非参照リソースを翻案として持つ	
		IsReplacedBy	置き換えられたリソース	
		Replaces	置き換えるリソース	
		IsRequiredBy	要求されるプロセスコンポーネントインスタンス	自身は成果物
		Requires	要求する作業成果物	自身はコンポーネントインスタンス
		IsGeneratedBy	生成元のプロセスコンポーネントインスタンス	自身は成果物
		Generates	生成する作業成果物	自身はコンポーネントインスタンス
		Next	次の作業に当たるプロセスコンポーネントインスタンス	自身はコンポーネントインスタンス
		Previous	前の作業に当たるプロセスコンポーネントインスタンス	自身はコンポーネントインスタンス
References	作成時に参照したプロセス要素			
IsReferencedBy	作成時に参照されたプロセス要素			

図 2 プロセス要素のメタデータセット

### 3.3. プロセス要素に付随するメタデータ

3.1.節で挙げたプロセス要素に対し、互いの関連やその他の背景情報を記述し、情報検索を行うことができるようにするためには、それぞれに統一したフォーマットのメタデータを用意する必要がある。そこで、本研究において必要なメタデータフォーマットを定義する。ここで取り扱うメタデータは、ネットワーク上の情報資源発見のためのメタデータセットである Dublin Core[1]に準拠し、必要に応じて限定子を追加し拡張したものである。図 2 にメタデータセットの一覧を示す。図中の網掛けエントリが今回拡張を行ったメタデータを表している。

このメタデータのうち、本研究における検索で主に利用されるエレメントは“Relation”で、プロセス要素の関連を詳細に記述するために限定子の拡張を行っている。図 3 に、このメタデータ記述によって記述できるプロセス要素間のリンクの例を挙げる。成果物“要求”はタスク“要求分析”の入力として必要とされるため、その間には Requires と IsRequiredBy というメタデータによってリンクが定義される。“要求分析”の出力として成果物“分析さ

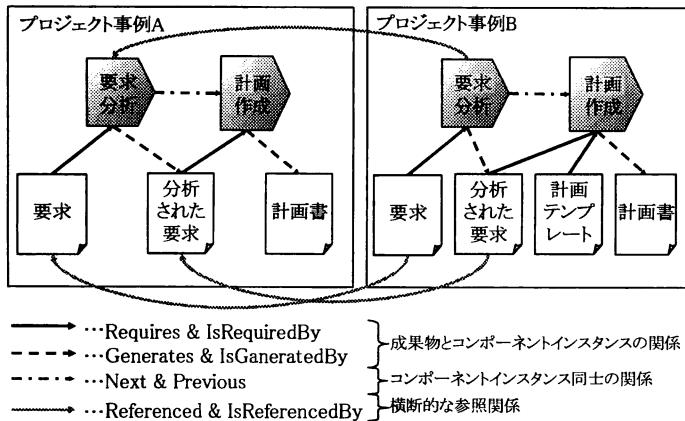


図 3 メタデータによるプロセス要素間のリンク

れた要求”が生成されるため、Generates と IsGeneratedBy によってリンクが定義される。“要求分析”と“計画作成”はタスクの前後関係にあたり、Next と Previous によってリンクが貼られる。また、プロジェクト事例 B の“要求分析”が、プロジェクト事例 A の“要求分析”を参考にして作られたものであれば、Referenced と IsReferencedBy によるリンクとしてプロジェクトを横断して参照関係が記述される。

#### 4. プロジェクト事例の類似検索手法

第3節に示したプロセス記述を持つプロジェクト事例に対して、2.2.節で紹介したグラフモデルの類似性検索という手法により、プロセス要素間の関連によってグラフ化されるプロジェクト事例の類似プロセス要素を検索する。

##### 4.1. グラフモデルによる類似検索

ここでは、本研究におけるグラフモデルの類似性検索手法の概念を簡単な例を挙げて説明する。

例えば、図4に示すようなプロセス記述を持つプロジェクト事例A,B が登録されている場合に、あるプロジェクトの PDSP 作成者(以降ユーザと呼ぶ)が“システム設計”および“実装”に関連する部分のプロセス計画を行う際に、他のプロジェクトの事例を参照しようとしたとする。このとき、我々の提案する事例検索システムに対し、ユーザは“システム設計”と“実装”というプロセスコンポーネントで互いにリンクを持ったものを検索語として与えることが出来る(図4の検索語 X)。

まず、プロジェクト事例 A と検索語 X の類似度を考える。事例 A 内には、検索語 X に含まれる2つのプロセス要素が存在し、それらは直接のリンクを持っている。つまり検索語 X は事例 A の部分グラフとなっている。このときシステムは、事例 A と検索語とは高い類似度を持つと判断し、検索結果として出力する。ユーザは事例 A のコンポーネントの内容や、関連する成果物などを参照し、自分の PDSP 作成の参考にする。

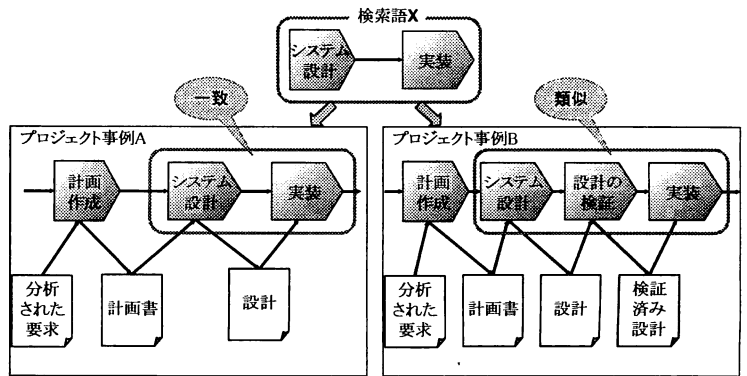


図 4 類似検索の例

一方、プロジェクト事例 B と検索語 X の類似度を考える。事例 B 内には検索語 X に含まれる2つのプロセス要素が存在するが、それらは直接のリンクは持たない。このときシステムは、事例 B と検索語 X とは低い類似度をもつと判断し、検索結果として出力する。ユーザは事例 B の記述により、“設計の検証”というコンポーネントを使う事例を知り、元々考えていたコンポーネントの組み合わせ以外の手法を検討することが出来る。

このような類似度による検索手法を適用することで、検索語と完全には一致しない結果も検索結果に含めることができ、また類似度という値により事例の順位付けを行うことができるようになる。結果、開発のコンテキストの類似した事例を漏らさずに、かつコンテキストも含めて参照することが可能になると考えられる。

##### 4.2. 類似度の計算

PDSP をグラフモデルとみなして類似検索を行うには、図4のようなプロセス要素間の関連グラフに対し、プロセス要素の重要度と関連度という値を定義し、それに基づいて類似度の計算を行う。以下、順を追って説明する。

###### プロセス要素の重要度

プロセス要素の重要度は、蓄積された全ての PDSP インスタンスにおける、同じプロセス要素の参照頻度を、正規化した値とする。プロセス要素の他のプロセス要素への参照は、メタデータの IsReferencedBy 限定子に記載されており、この個数により求められる。

いま、プロセス要素  $x$  の IsReferencedBy 限定子の記述個数を  $r_x$ 、登録されているすべての PDSP に記述されているプロセス要素の中で最大の値を  $r_{max}$  としたとき、プロセス要素  $x$  の重要度  $I_x$  は次のようになる。

$$I_x = r_x / r_{max} \quad (\text{このとき } 0 \leq I_x \leq 1)$$

また、検索語のプロセス要素の重要度は 1 とする。

###### プロセス要素間の関連度

プロセス要素  $x, y$  間の関連度は、プロセス要素間の関連グラフにおける要素間のリンクの最短パスの逆数で

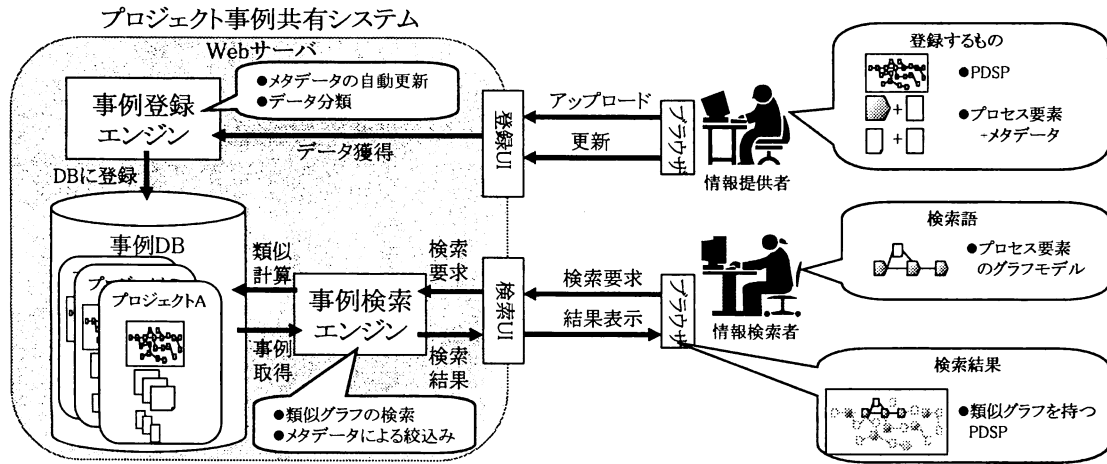


図 5 プロジェクト事例共有システムの構成

与える。ここでは、IsRequiredBy と Requires, IsGeneratedBy と Generates, Next と Previous すべてのリンクを対象とし、その中で最短のパスを用いる。x, y 間の最短パスを  $p_{xy}$  とすると、x と y の関連度  $R_{xy}$  は次のようになる。

$$R_{xy} = 1/p_{xy} \quad (p_{xy} \neq 0 \text{ のとき})$$

$$R_{xy} = 0 \quad (p_{xy} = 0 \text{ のとき})$$

### 類似度計算

重要度・関連度を用いて、対象とする 2 つのグラフ間の類似度を以下のように定義する。

#### i. 重要度についての重み付け

グラフ  $g_0, g_1$  に共通する(同じ種類が存在する),  $m$  個のプロセス要素群の重要度をベクトル  $I_{g_0}, I_{g_1}$  で表現する。

$$I_{g_0} = (I_{1g_0}, \dots, I_{ig_0}, \dots, I_{mg_0})$$

$$I_{g_1} = (I_{1g_1}, \dots, I_{ig_1}, \dots, I_{mg_1})$$

このとき、グラフ  $g_0, g_1$  における重要度についての重み付け  $f_I(g_0, g_1)$  という値を定義する。

$$f_I(g_0, g_1) = \sum_{i=1}^m (I_{ig_0} \cdot I_{ig_1})$$

#### ii. 関連度についての重み付け

グラフ  $g_0, g_1$  に共通するプロセス要素群の、それぞれの要素間の関連度をベクトル  $R_{g_0}, R_{g_1}$  で表現する。

$$R_{g_0} = (R_{12g_0}, \dots, R_{ijg_0}, \dots, R_{(m-1)mg_0})$$

$$R_{g_1} = (R_{12g_1}, \dots, R_{ijg_1}, \dots, R_{(m-1)mg_1})$$

このとき、グラフ  $g_0, g_1$  における関連度についての重み付け  $f_R(g_0, g_1)$  という値を定義する。

$$f_R(g_0, g_1) = \sum_{\substack{i,j=1 \\ i \neq j}}^m (R_{ijg_0} \cdot R_{ijg_1})$$

#### iii. 類似度の計算

グラフ  $g_0, g_1$  間の類似度  $S(g_0, g_1)$  は、全 PDSP 中の  $f_I(g_i, g_j), f_R(g_i, g_j)$  の最大値を  $f_I(max), f_R(max)$  とすると、ここで定数  $p$  は、検索結果の順位付けを行う上で、重

$$S_{(g_0, g_1)} = p \times (f_I(g_0, g_1) / f_I(max)) + (1-p) \times (f_R(g_0, g_1) / f_R(max))$$

( $p$  は  $0 \leq p \leq 1$  の定数)

要度と関連度の値にどれだけ比重を置くかを決定するための定数である。  $p$  が大きくなれば、よく参照されている要素を多く持つプロセス記述に対し高い類似度が与えられる。一方、  $p$  が小さくなれば、検索語との形状が似たプロセス記述に対し高い類似度が与えられる。  $p$  の値は検索を行うユーザが自由に設定できる。

## 5. プロトタイプシステムの設計

本研究では、第 3, 4 節に述べた PDSP 記述と類似検索の機能を利用する、プロジェクト事例共有システムの設計を行った。以下にその概要を示す。

### 5.1. システムの構成

プロジェクト事例共有システムの構成を図 5 に示す。システムはプロジェクト事例 DB, 事例登録する事例登録エンジン, 類似検索機能を提供するための事例検索エンジン, およびそれぞれの UI 提供部で構成される。システムは Web サーバ上に構築され、Web アプリケーションとしてサービスを提供する。ユーザは Web ブラウザを介してデータの登録・検索を行う。

### 5.2. 事例登録エンジン

このシステムでは、ソフトウェア開発者が情報提供者となり、プロジェクト事例の内容が更新されるたびにそれを共有システムに登録することを想定している。

- プロジェクト開始時には、プロジェクトマネージャが PDSP を生成し、それを共有システムに登録する。
- 開発実行時には、PDSP 中のそれぞれのプロセスコンポーネントインスタンスの作業担当者が、該当作業の入出力として関連しているソフトウェア作業成果物をアップロードする。このとき、PDSP 上の対応するプロセス要素や、関連するプロセス要素をメタデータ

として同時に登録する。メタデータの入力登録 UI により提供されるフォームで行える。事例登録エンジンでは、メタデータの“Date”など、自動更新可能なエレメントを更新する。また、“Relation”など、他のプロセス要素への参照があった場合は被参照要素のメタデータも更新し、データの整合性を保持する。

- プロセス要素に何らかの変更を加えたい場合には、新規アップロード時と同様の手段で行う。このとき、旧バージョンの保持/廃棄を選択可能で、その選択に応じて、システムが“Date.Modified”や“Relation.Is VersionOf”などのデータを更新する。

### 5.3. 事例検索部

事例登録エンジンにより登録されたプロジェクト事例は、事例検索エンジンにより検索参照する。

- 利用者は、プロセス要素で構築されるグラフモデルを検索語として事例検索を行うことが出来る。グラフモデルの入力は、PDSP から検索に利用したいプロセス要素の種類と、それらの接続関係を指定することで行う。これは、ユーザ自身がそのとき関わっているプロジェクトのプロセス要素の種類とつながりを指定することで、このシステムの目的である開発のコンテキストを検索語として表現する意味を持つ。重要度と関連度のどちらを重視するかの割合(4.2.節で述べた定数  $p$ )を指定することも出来る。
- 情報検索エンジンは検索語を受け取り、類似検索を行う。検索語に類似したグラフを部分グラフとして持つプロジェクトのPDSPを結果として出力する。
- 検索 UI により、グラフの類似度の高い順に結果を出力する。結果としては、PDSP への参照、関連するプロセス要素の実体及びメタデータへの参照などが含まれる。そして、ユーザはこの検索結果を次の検索語とする。再検索が可能で、検索語を拡張した類似検索を容易に行うことができる。

### 5.4. システム利用例

システムの利用例としては以下のようなシナリオが考えられる。

- **プロジェクトのプロセス記述作成支援**  
プロジェクトマネージャ等がプロセス記述を行う際に、過去のプロジェクトにおけるプロセス要素とその構成を参考にプロセスの組み立てを行う。
- **タスク手順や成果物の参照**  
事例を作業実行・成果物作成の手引きとして用いる。
- **組織標準プロセスの定義支援**  
事例を比較し、ベストプラクティスを定式化したり、よく参照されているプロセス要素を元にテンプレートを作成したりする。

## 6. 考察とまとめ

本研究では、ソフトウェア開発の際して適切な事例を提供することを目的とし、ソフトウェアプロセス記述と、タスクや作業成果物の実体を関連付けて蓄積するプロジェクト事例システムを提案した。このシステムを用いれば、それぞれのデータを独立して保存しておくだけでは把握することのできないプロセス要素間の関連やその背景情報を効果的に参照し、再利用することができる。システムを利用するユーザは、単語やプロセス要素そのものを検索キーとして、容易に類似したプロセス要素を検索することができると思われる。

今後は、このシステムを実際のソフトウェア開発現場に適用し、類似度による検索の有効性を定量的・定性的に評価する必要があるが、今回設計した段階では、開発現場での運用に耐えるための課題を抱えている。まず、現在の設計ではデータ登録時にユーザが多くのメタデータを手動で入力する必要があり、負担が大きい。データの自動入力機能を拡充してより簡単化することや、ユーザがデータ入力することを促進させるようなシステムを導入することが必要になる。また、グラフモデルを可視化し、プロセス要素の関連構造をグラフィカルに閲覧する機能を導入し、より直感的なデータ検索・参照を行えるようにするべきである。今後の研究でこれらの課題に対応していく予定である。

## 謝 辞

本研究を進めるにあたり、奈良先端科学技術大学院大学情報科学研究科の中村匡秀先生にメタデータの設計等について貴重なご意見を賜りました。深く御礼申し上げます。

## 文 献

- [1] DCMI, Dublin Core Metadata Element Set, Version 1.1: Reference Description, DCMI Recommendation, 1999.
- [2] 平井千秋, 工藤裕, 降旗由香里, “ナレッジマネジメントのソフトウェア開発への適用,” 人工知能学会誌 16 巻 1 号(2001 年 1 月), pp.59-63, (社)人工知能学会, 2001.
- [3] OMG, “Software Process Engineering Metamodel Specification (SPEM),” An Adopted Specification of the Object Management Group, formal/02-11-14, November, 2002.
- [4] M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber, Capability Maturity Model for Software Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24, 1993.
- [5] 敷田幹文, 門脇千恵, 国藤進, “フローに連携した組織内インフォーマル情報共有手法の提案,” 情報処理学会論文誌, Vol.41, No.10, pp.2731-2741, Oct.2000.
- [6] 富田準二, 竹野浩, 菊井玄一郎, 林良彦, 池田哲夫, “グラフモデルの提案とテキスト検索システムへの適用による評価,” 情報処理学会論文誌: データベース, Vol.43, No.SIG2(TOD13), pp.94-107, Mar.2002.