

システムの仕様変更調査における設計情報と作業プロセス間の依存関係を用いた影響分析システムの開発

福田 貴三郎^{1,a)} 村瀬 健太郎^{1,b)} 中村 覚^{2,c)} 稗方 和夫^{2,d)} 笈田 佳彰^{2,3,e)} 松本 滋^{3,f)}
岡田 伊策^{2,3,g)}

受付日 2017年7月26日, 採録日 2018年1月15日

概要: パッケージソフトや既存システムの仕様変更において、改修・カスタマイズの対象となる機能の規模の見積りや、それにとまなう作業全貌の把握には、システムの全体構成や個々の機能の詳細に関する知識が不可欠であり、豊富な経験と高度なスキルを有するベテラン SE のノウハウが必要である。また設計情報に加え、作業プロセス間の依存関係に基づく影響波及範囲の推定も求められる。本研究では、設計情報と作業プロセス間の暗黙的な依存関係を可視化し、パッケージソフトや既存システムの改修・カスタマイズによる影響波及範囲の特定を支援するシステムの開発を行った。また、実際の情報システム開発プロジェクトにおける事例を用いた評価実験により、従来の手法に比ベドキュメントベースでより網羅的に影響波及範囲を抽出できることを確認した。

キーワード: 影響分析, 仕様変更, 設計情報, プロセス依存関係, 可視化

Development of Impact Analysis System for Specification Change of System with Design Information and Process Dependency

TAKASABURO FUKUDA^{1,a)} KENTARO MURASE^{1,b)} SATORU NAKAMURA^{2,c)} KAZUO HIEKATA^{2,d)}
YOSHIAKI OIDA^{2,3,e)} SHIGERU MATSUMOTO^{3,f)} ISAAC OKADA^{2,3,g)}

Received: July 26, 2017, Accepted: January 15, 2018

Abstract: Knowledge of the overall structure of the system and the details of each function is indispensable for estimating the size of the functions and the overall work in the repair of the package software and the existing system. This requires knowledge of veteran SE with extensive experience and advanced skills. In addition, it is also required to estimate the influence spread range based on the dependency relationship between the work processes. This study proposes the system to support impact analysis for specification change of system with design information and process dependency. We confirmed the effectiveness of our proposed method by comparing with conventional method.

Keywords: impact analysis, specification change, design information, process dependency, visualization

¹ 株式会社富士通研究所
Fujitsu Laboratories LTD., Kawasaki, Kanagawa 211-8588, Japan

² 東京大学
The University of Tokyo, Kashiwa, Chiba 277-8561, Japan

³ 富士通株式会社
Fujitsu Limited, Ota, Tokyo 144-8588, Japan

a) f.takasaburo@jp.fujitsu.com

b) kmurase@jp.fujitsu.com

c) nakamura.satoru@mail.u-tokyo.ac.jp

d) hiekata@edu.k.u-tokyo.ac.jp

e) oida.yoshiaki@jp.fujitsu.com

f) matsumoto-s@jp.fujitsu.com

g) isaac-okada@jp.fujitsu.com

1. はじめに

経済産業省の調査 [1], [2] によると、日本の企業では汎用的なパッケージソフトを導入するにあたり、約 90 パーセントのケースにおいて仕様変更における改修・カスタマイズ（以後仕様変更における改修・カスタマイズを一言で「改修」と表記する）を行っている。また、システム構築を行った企業のうち、約半数以上が新規の開発ではなく、従来のシステムの再構築を行っていると報告されている。このように、現在の日本の企業におけるシステム開発では、

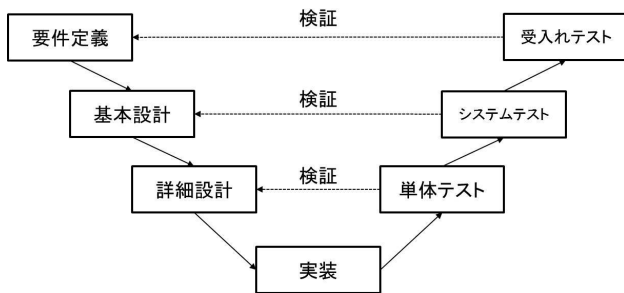


図 1 ソフトウェア開発の品質管理における
Fig. 1 Example of V-Model.

パッケージソフトや既存システムの改修を行うケースが大きな割合を占めている。

改修業務では、改修規模の把握が重要となる。一般的には、システムの設計情報に基づき、改修範囲の把握を行う。たとえば、機能要件を基に、直接的に改修が必要となる「項目*1」,「テーブル (マスタ)」「サブシステム*2」,「機能*3」などを考える。そして、各要素を改修した場合に、その影響がどれくらいの範囲で及ぶのかを考え、間接的に改修が及ぶ範囲を分析する必要がある。具体的には、ある項目名の桁数を改修する要件の場合、直接的に改修を行うのは当該項目であるが、間接的には、当該項目を持つテーブル、およびそのテーブルに入出力を行っているサブシステムや機能にまで影響が及ぶ。対象となるパッケージソフトや既存システムに精通しているベテラン SE は、間接的な影響が及ぶ範囲を、設計書を読みながら上記の項目・テーブル・サブシステム・機能の関係性を経験的に理解しているため、改修規模を正確に推定できる。しかし、設計書やソースコードを精査する時間がない場合や、経験の浅い非ベテラン SE が見積りを行う場合には、当然見落としが発生するリスクが存在する。そのため、改修の影響のある項目・テーブル・サブシステム・機能などの設計情報の関係性を効率的に精度良く把握することが求められる。

一方、改修範囲の正確な把握には、この設計情報の依存関係の利用のみでは不十分なケースが考えられる。図 1 にソフトウェア開発の品質管理における V 字モデルの例を示す。V 字モデルでは、まず要件定義から基本設計や詳細設計といった開発工程の分解がなされ、さらにこれらの開発工程に対応したテスト工程が並べられる。本稿では、このような開発工程に基づく作業単位を「作業プロセス」、作業プロセス間のつながりを「作業プロセス間の依存関係」と呼ぶ。

改修業務においては、図 2 に示すように、先述した設計情報の依存関係に加え、この作業プロセス間の依存関係に基づく影響波及を考慮する必要がある。図 2 では、設

*1 テーブルのカラムと同義。
*2 機能の集合を単位としてまとめたもの。
*3 画面・帳票・バッチなどを組み合わせて、データ処理の単位としてまとめたもの。

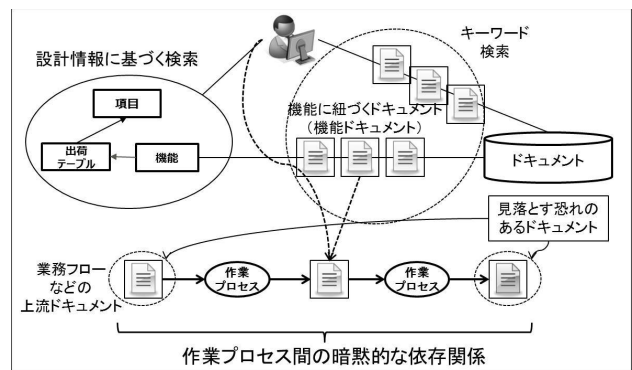


図 2 作業プロセス間の依存関係
Fig. 2 Dependencies between work processes.

計情報を起点として機能に紐づくドキュメントを「機能ドキュメント」、設計情報に直接紐付かず見落とす恐れのあるドキュメントを「上流ドキュメント」と定義している。設計情報に紐づくドキュメントとは、具体的に前述の「項目」,「テーブル」,「サブシステム」,「機能」などの設計情報を直接定義しているドキュメントを指す。そのため、設計情報を起点に機能ドキュメントを推定することは可能である。しかし、図中の下部に示すような作業プロセス間の暗黙的な依存関係に依存するドキュメントは見落とす可能性がある。たとえば開発プロセスにおいて、あるテーブルに関する改修が生じた場合には、対象プロセスで作成するテーブル定義書などの設計情報に紐づく機能ドキュメントに加え、その上流工程である要件定義プロセスで作成する『業務フロー』や『業務関連図』など、設計情報に直接紐付かない上流ドキュメントの修正が必要な場合がある。この上流工程レベルでの修正を見落としした場合、開発が進むにつれ、後から上流工程レベルでの見直しが必要であったと気づき、結果的に多大な手戻りが発生し、必要以上に開発コストがかかってしまうリスクがある。既存パッケージソフトの開発から携わっているベテラン SE であれば、パッケージソフトの開発プロセスを熟知しているため、作業プロセスに基づく依存関係の推定を経験に基づいて行うことができる。しかし、一般的に既存パッケージソフトの改修業務は、開発から携わってきたベテラン SE とは異なる SE が担当することも多々ある。このため、対象システムに馴染みのないなどの理由により、作業プロセス間の暗黙的な依存関係に基づく影響を見落とすリスクがある。そのような理由により、改修規模の正確な把握には、設計情報に基づく依存関係に加え、作業プロセス間の依存関係も含めた影響調査を支援する仕組みが求められる。

そこで本研究では、設計情報と作業プロセス間の暗黙的な依存関係を可視化し、パッケージソフトや既存システムの改修による影響波及範囲の特定を支援するシステムを開発する。また、実際の情報システム開発プロジェクトにおける改修事例を用いた評価実験により、提案するシステム

の有効性を検証する。

2. 関連研究

設計情報を活用し、機能改修などによる影響波及範囲の把握を支援する研究として、元山ら [3] はあらかじめ設計書のテンプレートを作成することによって、項目間の不整合をなくす手法を提案し、テスト工程に流出する欠陥を減少した。また近年ではソフトウェアの品質保証を目的とし、設計書、試験成績書、障害記録、ソースコードなどの各種情報間のトレーサビリティを確保することが求められており、その構築に関する研究 [4] が行われている。その他、要件定義から運用保守までのトレーサビリティの管理を支援するツール [5] など、設計者の負担を軽減するためのツールなども提供されている。これらのツールの有用性は認識されているが、これらのツールは、新規の情報システム開発プロジェクトにおいて生成される各種情報間のトレーサビリティの確保を支援することを目的としている。すなわち、トレースすべき管理項目が事前に与えられ、それに基づいて資産が体系化されるため、既存の設計書などの資産を対象とする場合、これらのツールを導入することはできない。

また、既存ソフトウェアを修正して開発するシステムや新規開発のシステムにおける保守工程において、設計書とソースコードを関連付け、仕様変更に対して影響を受ける設計書とソースコードを抽出し、仕様変更の影響波及範囲を特定する研究も行われている [6], [7]。さらに稗方ら [8] は図 3 に示すように、既存パッケージソフトの設計書に記述された設計情報を基に、マスタや機能といった設計情報単位で改修作業の影響波及範囲を推定し可視化する「項目 CRUD 検索」を開発した。この項目 CRUD 検索では、実際のプロジェクトへの適用を通じ、改修が必要となるドキュメントを網羅的に抽出できることを実証し、定量的に精度向上の効果があることを確認している。一方、彼らの手法では、設計情報に基づく表層的な依存関係でしかドキュメントの抽出および可視化を行っていない。そのため、先述したように、設計情報とは別の作業プロセス間の依存関係に基づく影響波及範囲の推定が困難であるという

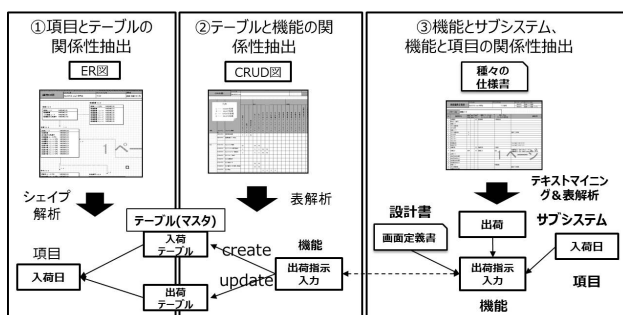


図 3 ER 図や CRUD 図を用いた設計情報の構造化 [8]

Fig. 3 Design information structured from documents.

課題を持つ。

作業プロセスの依存関係については、IT サービスマネジメントのベストプラクティスを集めたフレームワークである「ITIL (Information Technology Infrastructure Library)」や、プロジェクトマネジメント知識体系である「PMBOK (Project Management Body of Knowledge)」などのプロジェクト管理手法において標準化が進められている。また、情報システム開発会社各社は、情報システム開発プロジェクト作業の標準化を目指し、開発ナレッジを整理することで標準的なプロセスを開発方法論として精緻に体系化してきた [9], [10]。これらの標準プロセス体系には、各プロセスの入力情報および出力情報が設計書のテンプレートとともに網羅的に定義されているものが多い。そのため、このような標準プロセス体系を参考とすることにより、機能改修時における作業プロセス間の依存関係に基づく影響波及範囲の推定を支援することが期待できる。

したがって、本研究では既存の設計書などの資産を対象に、設計情報の依存関係に加え、標準プロセス体系を用いた作業プロセス間の依存関係を考慮することにより、機能改修による影響波及範囲の特定を支援する手法を提案する。

3. 提案手法

3.1 提案手法の概要

本研究では、作業プロセス間の暗黙的な依存関係を推定するためのポイントとして、標準プロセス体系において定義されている作業プロセスとドキュメントテンプレート（以下標準ドキュメント）の情報を利用する手法を提案する。本提案手法の概要を図 4 に示す。

一般的に、標準プロセス体系を用いた開発プロジェクトでは、標準プロセス体系において定義された作業プロセスを基にプロジェクトの進行を行う。また、標準プロセス体系において各作業の出力として定義されている標準ドキュメントに従い、プロジェクトの成果物としてのドキュメント（以下 PJドキュメント）の作成を行う。もし個々のプロジェクトにおいて行われた作業プロセスとその成果物であるドキュメントの情報が厳密に残っている場合は、その作業プロセスを標準プロセス体系の作業プロセスと対応付

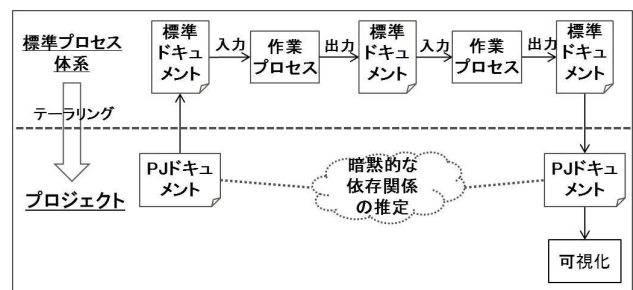


図 4 提案手法の概要

Fig. 4 Overview of the proposed method.

けることによって、個々のプロジェクトにおける作業プロセス間の暗黙的な依存関係の推定が可能になると考えられる。しかし、個々のプロジェクトにおいて、実際に行われた作業プロセスの情報が厳密にデータとして残っているケースは少なく、作業プロセスベースで対応付けを行うことは困難である場合が多い。そのため、本研究では図4で示すとおり、ドキュメント起点でプロセス間の依存関係に基づく影響波及の推定を行う。

提案する手法の全体像として、まず標準プロセス体系における、作業プロセスとその入出力となる標準ドキュメントの対応関係を事前に形式知化しておく。次に、先の入出力の標準ドキュメントと各プロジェクトで作成するPJドキュメントの対応付けを行う。これにより、標準プロセス体系において定義された作業プロセスに基づき、PJドキュメント間のプロセスに関する依存関係を得る。そして、実際の改修業務において、改修を行う機能やデータベースに紐付くドキュメントを起点に、標準プロセス体系における作業プロセスをたどっていくことにより、改修による影響波及範囲の分析を支援する。具体的には、たどった標準ドキュメントと対応関係にあるPJドキュメントおよびたどった過程における作業プロセスを、改修の影響があるドキュメントと発生する作業プロセスとして判定する。最後に出力として、これらのドキュメントと作業プロセスをリスト表示およびネットワーク表現を用いて可視化する。開発プロジェクトに携わる担当者は、改修業務の内容に合わせてこの出力結果を確認することにより、作業プロセス間の依存関係を考慮した影響波及範囲を把握することができる。

以下、上述の提案する手法の全体像における、それぞれの過程について詳述する。

3.2 標準プロセス体系の形式知化

まずソフトウェアシステム開発における標準プロセス体系の形式知化を行う。具体的には、WBS (Work Breakdown Structure) などの作業単位を用いて、ソフトウェアシステム開発の標準的な作業プロセスを定義する。各作業プロセスには、先述のV字モデルなどのウォーターフォール型モデルの開発工程のように「いつ」の情報と、各作業の順序関係、および入出力情報を定める。この入出力情報は、設計書などのドキュメント情報が成果物として定義されていることを想定する。例として、画面上の項目の設計作業を想定し、作業『画面項目の設計』を定義する場合を考える。画面項目を設計するにあたり、どのような画面レイアウトなのかを参照する情報として、『画面レイアウト設計書』を入力情報として定義する。そして具体的な作業における成果物として、画面項目の詳細情報を示す『画面項目一覧設計書』を出力情報として定義する。この場合、画面に関する改修業務が発生し、『画面レイアウト設計書』に改修を行

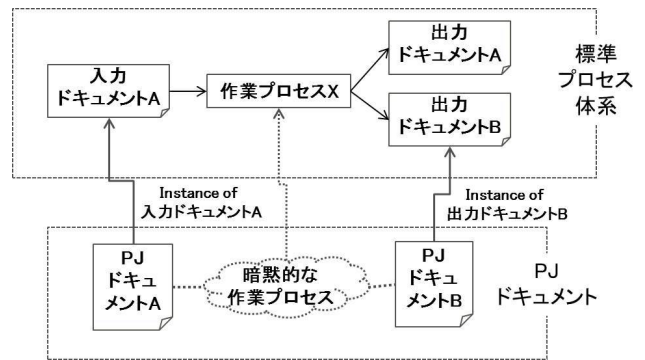


図5 PJドキュメントの関連作業プロセス推定
Fig. 5 Estimation of work process related to PJ-documents.

う場合、作業『画面項目の設計』の出力情報である『画面項目一覧設計書』にも影響が波及することを示唆している。

3.3 標準プロセス体系とPJドキュメントの対応付け

次に形式知化した標準プロセス体系とPJドキュメントの対応付けを行い、各PJドキュメントが入出力となりうる作業プロセスの推定を行う。具体的には図5に示すように、先に定義した入出力情報とPJドキュメントの対応付けを行う。ドキュメントの対応付けに関しては人手で行うことも可能であるが、ドキュメントの数が数千や数万以上のオーダで大量に存在する場合は、対応付け作業に多くの工数を要してしまう。また、パッケージソフトを拡張するたびに、増えたドキュメントに対して新たに対応付けを行わなければならないという問題もある。そのため、本研究では自動で対応付けを行う方法を提案する。方式については、対象ドメインに応じて多様な方法が考えられるが、ここではドキュメントに記述されている情報を基に対応付けの指針について述べる。

標準ドキュメントは一般的にドキュメント作成を容易にするだけでなく、情報俯瞰を容易にするためのフォーマットの統一や、使用する語彙などを標準化する役割を持っている。そのため、本研究で扱う標準ドキュメントは、上記の役割を満たしたものを想定する。しかし、標準プロセス体系を用いた開発プロジェクトでは、標準ドキュメントが個々のプロジェクトでそのままテンプレートとして使用されないケースも多々ある。具体的には、個々のプロジェクトに合わせて標準ドキュメントのテンプレートを改変するなど、テーラリングして使用されることが多い。しかし、標準ドキュメントは使用する語彙を標準化する役割を持つ以上、テーラリングされた場合においても、ドキュメント内で使用する基本的な語彙に関しては、大きく変化するケースは少ないと考えられる。そのため本研究では図6のフローに従い、ドキュメント名などの表層的な情報に加え、各ドキュメントを特徴付ける適切な特徴量を用いて、標準ドキュメントとPJドキュメントの対応付けを行う。

標準ドキュメントとPJドキュメントを対応付けするに

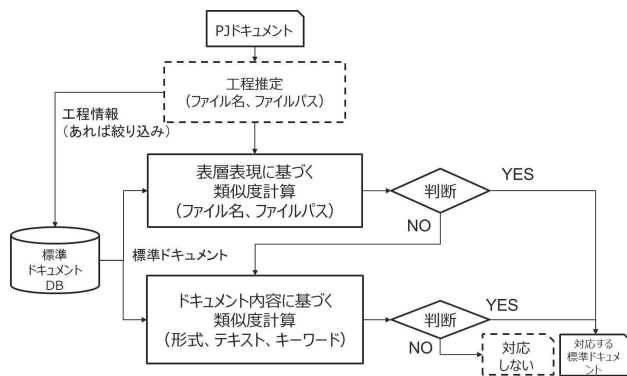


図 6 ドキュメント対応付けの処理フロー

Fig. 6 Process flow for associating documents.

表 1 標準ドキュメントごとの特徴語の例

Table 1 Example of keywords for each standard document.

標準ドキュメント名	特徴語の例
サービス処理系設計書 A	サービス, 項目, チェック, 仕様
レイアウト系設計書 B	レイアウト, パターン, 設計
エンティティの定義書 C	概念, データ, 定義

あつて、PJ ドキュメントには標準ドキュメントとは対応しないものや、複数の標準ドキュメントの情報が記述されるケースがある。そのため、各 PJ ドキュメントに対して 0 または 1 以上の標準ドキュメントを割り当てる処理を行う。処理フローとして、まず工程ごとにドキュメントの対応付けを行うため、ドキュメントごとに工程推定を行う。工程推定は、下記例で示す正規表現を用いて、ドキュメントのファイルパス情報に工程を示す表現が含まれているかどうかで判断する。本処理は、たとえば「\PackageA\01_RD 工程\02_仕入入力\C1UI1A0_画面処理機能の設計.xlsx」といったファイルパスのように、標準プロセス体系に基づくドキュメントの管理では、ディレクトリ構造やファイル名に開発工程の情報を持つことが多いという特性に基づくものである。そのため、開発組織の文化によってはドキュメント内の特定の箇所から工程情報を抽出するなど、他の処理方法で行うことも想定している。

例) RD (要件定義) 工程を抽出する正規表現

$$[^\wedge a-zA-Z]RD[^\wedge a-zA-Z][^\wedge a-zA-Z]RD[^\wedge a-zA-Z]$$

工程推定後は、工程ごとに対応するドキュメント間の表層表現における類似度判定を行う。類似度判定におけるパラメータとして、PJ ドキュメントはファイルのパス情報に出現する単語の出現情報をそのまま扱い、標準ドキュメントに関しては表 1 に示すように、それぞれ事前に特徴語を抽出し、Bag of words の形式で定義したベクトル空間モデルを使用する。各標準ドキュメントの特徴語を定義するにあつては、TF-IDF のスコアが高いキーワードを抽出し、各ドキュメントにそれぞれ手でキーワードのフィルタリングを行った。キーワードのフィルタリングを行う際には、特に各標準ドキュメントに必須で記述される定義項

目を優先して選別している。

各 PJ ドキュメントにおける表層表現を用いて類似度判定を行い、類似度が閾値以上でマッチング判定されたペアに関しては、そのまま対応付けが成されるペアとして判定する。また、判定されなかったペアに関しては引き続き PJ ドキュメント内のテキスト情報を用いた類似度判定を行う。具体的にテキスト情報として用いる特徴語として、PJ ドキュメント中における見出し語や表のカラム名における単語の出現情報を使用する。見出し語に関しては文字サイズやセルの色から判定し、カラム名に関しては文献 [11] を参考に表の罫線情報を基に抽出を行っている。このように抽出したテキスト情報を基にベクトル空間モデルに変換し、前述の事前に定義した標準ドキュメントのベクトル空間モデルと類似度計算を行い、対応付けられるか否かの判定を行う。ここでマッチング判定されたものに関しては、そのまま対応付けが成されるペアとして判定し、判定されないペアに関しては対応しないペアとして処理される。

3.4 改修範囲の推定と可視化

これまでの処理を経て、標準プロセス体系における作業の入出力情報と個々のプロジェクトの PJ ドキュメントの対応関係が構築され、標準プロセス体系に基づき個々のプロジェクトにおけるプロセス間の依存関係を推定することができる。ただし、標準プロセス体系に基づいて単純にプロセス間の依存関係をたどってしまう場合、実際の改修の影響波及範囲と比べて必要以上に網羅的になる恐れがある。本研究ではこの問題に対して、標準プロセス体系を形式知化する際に、あらかじめソフトウェア開発の工程情報などを絞り込みのための属性として形式知する方式をとった。このようにすることにより、改修業務の内容に応じて、任意の工程などの絞り込み属性に限定して、改修の影響波及範囲を推定することが可能になる。たとえば、上流工程から影響波及範囲を分析したい場合は、要件定義工程などの上流工程を含めてプロセス間の依存関係をたどって分析を行う。また、下流工程に限定して分析を行いたい場合は、詳細設計工程などの下流工程以下に限定してプロセス間の依存関係をたどって影響波及範囲の分析を行うことになる。

以上のように推定を行った改修における影響波及範囲をユーザに伝達する手段として、本研究では単純にリスト一覧で表示する方法に加え、図 7 に示す可視化 UI の開発を行った。図中の四角で示された情報が個々のプロジェクトの PJ ドキュメント (設計書) であり、図中の丸で示された情報が標準プロセス体系で定義されている作業プロセスである。各ノード間の配置は、グラフ描画アルゴリズムの 1 つである力学モデルによって位置情報を計算しており、各ノードは重ならず自動で配置される。また、工程ごとに事前におおよその位置座標を指定し、各ノードに対して属している工程ごとの前述の座標に向けて引力を働かせるこ

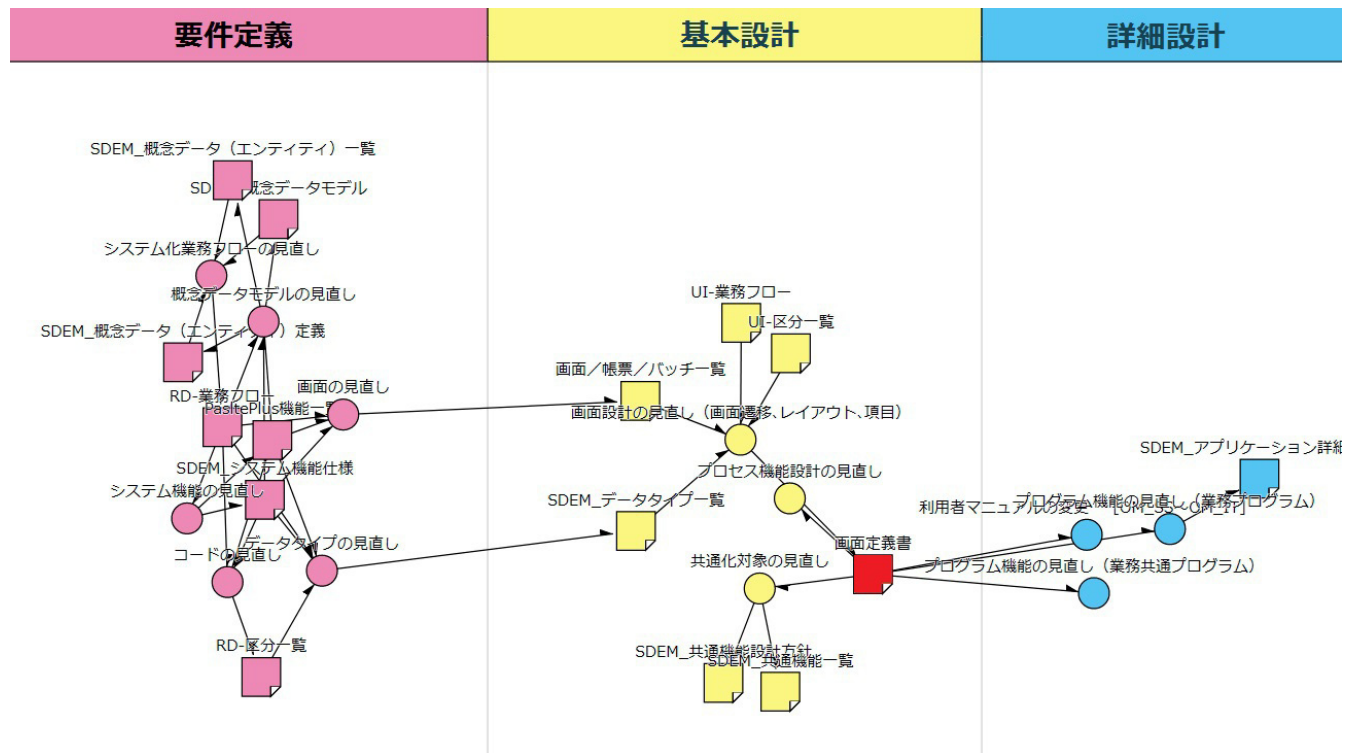


図 7 影響波及範囲の可視化の例
 Fig. 7 Visualization of the modification area.

とにより、工程別にノードが分かれて配置されるように工夫をしている。

図 7 の例は図の中央の基本設計工程の『画面定義書』を起点とした影響波及であるが、先行作業や後続作業として、要件定義工程や詳細設計工程の影響波及も読みとることができる。このようにドキュメントと作業プロセスの依存関係を可視化することにより、非ベテラン SE でも容易にプロセスの依存関係に基づいた影響波及を把握することができる。

4. 評価実験

4.1 評価実験の概要

本研究では、情報システム企業 A 社が過去に担当したプロジェクト案件を対象に、提案手法の有用性を検証する。A 社はソフトウェアシステムの開発標準として『SDEM』を採用している。SDEM とは、富士通株式会社の企画、開発、運用・保守、品質保証活動の基本的な考え方を示した標準プロセス体系である [9]。また、SDEM はソフトウェア開発における作業の国際標準である ISO/IEC12207 のベースともなっており、国際標準としても使用可能な標準プロセス体系である。ISO/IEC12207 では作業プロセスごとに必ず出力情報が含まれるように定義しており、SDEM においても、作業プロセスを表す WBS ごとに入出力情報となる標準ドキュメントのテンプレートが定義され、そのドキュメントに記述されるべき情報がテンプレート内に含まれている。また、新規開発、機能改修などの要件ごとに修

正が必要となる WBS の依存関係を定めている。これらの作業の選択と手順化により、ウォーターフォール・モデルなどの各ソフトウェア開発プロセスに適用することが可能である。本評価実験では、この SDEM に準拠したものと非準拠のものを合わせた計 4 つのパッケージソフトのドキュメントを対象に、標準プロセス体系と PJ ドキュメントの対応付けが可能かどうかを検証する。また、過去に行われた実際の見積り業務に基づく計 3 つのタスクを用いて、影響波及範囲調査として必要となるドキュメント抽出の再現性について検証し、作業プロセス間の依存関係を加味した改修業務における影響波及範囲推定の有用性の評価を行う。

4.2 SDEM における標準プロセス体系の形式知化

本ケーススタディでは、SDEM で定義されている情報を用い、図 8 に示す標準プロセス体系の形式知化を行った。

SDEM では、WBS のレベルとして 3 段階あり、レベル 1 を上位の層として、順にレベル 2、レベル 3 へと分解される。そのうちレベル 2 とレベル 3 である WBS-2 と WBS-3 では、直接的な入出力ドキュメントが定義されている。

これらを形式知化するにあたり、WBS-2 をベースに作業プロセス間の依存関係をたどれるように設計し、WBS-3 の入出力情報として定義されているドキュメントは WBS-2 に従属されるものとして知識化を行った。また、WBS-2 の情報に対し、工程情報とカテゴリの情報も付与している。

SDEM におけるカテゴリとは、情報システムを取り巻く世界を概念的に分割してモデル化したものであり、このカ

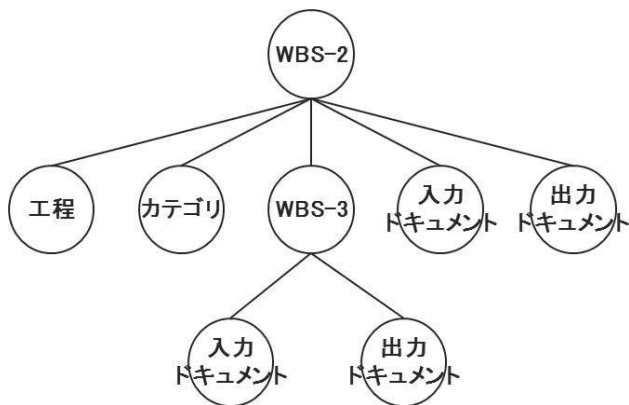


図 8 SDEM における標準プロセス体系の形式知化

Fig. 8 Formalization of standard process architecture in SDEM.

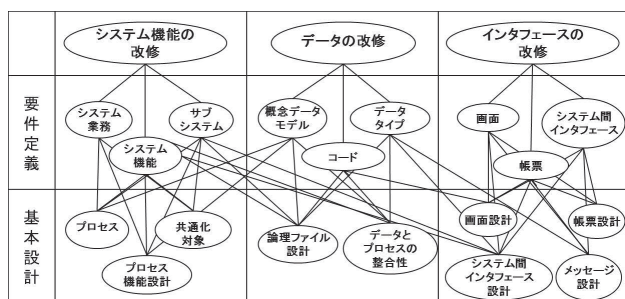


図 9 工程・カテゴリ別の標準プロセス体系の例

Fig. 9 Example of formalization of standard process architecture for each process and category.

カテゴリが示す領域をふまえ、システム開発に関わるステークホルダーは互いの役割分担において共通認識を持つことを目的としている。このカテゴリの情報も合わせて形式知化することにより、改修業務の内容に応じて、任意の工程やカテゴリに限定して改修範囲を柔軟に推定することができる。形式知化した標準プロセス体系に関して、各工程およびカテゴリ別に改修における作業の対象を図示した例を図 9 に示す。図 9 における縦軸が工程、横軸がカテゴリに該当する。たとえば、インタフェースの改修に関する基本設計工程に限定して影響波及範囲を推定したい場合は、図 9 における右下の領域に限定して作業プロセス間の依存関係をたどることになる。

本研究では具体的に形式知化を行うにあたり、794 個の作業プロセスと 85 個の標準ドキュメントの対応関係の形式知化を行った。作業プロセスの内訳は、レベル 2 の WBS が 172 個、レベル 3 の WBS が 622 個である。

4.3 標準プロセス体系と PJ ドキュメントの対応付け

4.3.1 評価データ

標準プロセス体系と PJ ドキュメントの対応付け精度の評価を行うにあたり、まず標準ドキュメントに対し、特徴語の定義を行った。特徴語の数は文書ごとで平均して 6.9 で

表 2 各パッケージソフトにおける対応付けの実験結果

Table 2 Results of association in each package.

	正解 ペア数	総出力 ペア数	出力における 正解ペア数	適合率 (%)	再現率 (%)
A	1,838	1,868	1,700	91.0	92.5
B	2,332	2,331	2,140	91.8	91.1
C	1,228	1,163	1,094	94.1	89.1
D	641	560	407	72.7	63.5

あり、全体的な特徴語の次元数は 96 であった。次にウォーターフォール・モデルの開発プロセスに基づき開発された以下の計 4 つのパッケージソフトを対象に、事前に手動で対応付けた正解データを作成し、評価を行った。類似度の計算方法については、Simpson 係数で類似度の算出を行い、表層表現に基づく類似度計算の閾値は 0.3、ドキュメント内容に基づく類似度計算の閾値は 0.8 とした。これらの閾値は経験的に設定を行っている。対応関係については、前述のとおり各 PJ ドキュメントに対して 0 または 1 以上の標準ドキュメントの対応付けがなされる。そのため、システムが出力した対応関係のペア数のうち正解となるペアの割合を適合率とし、正解データとなる対応関係のペア数に対し、システムが出力したペアがカバーしている割合を再現率として、算出を行った。なお、パッケージソフト A, B, C は SDEM に準拠してテラリングした上でパッケージソフトが作成されており、パッケージソフト D に関しては SDEM に準拠せず、独自のドキュメントフォーマットを用いて作成されている。

- パッケージソフト A：計 1,989 ドキュメント
- パッケージソフト B：計 3,172 ドキュメント
- パッケージソフト C：計 2,803 ドキュメント
- パッケージソフト D：計 318 ドキュメント

4.3.2 実験結果

各パッケージソフトにおける対応付けの適合率、再現率を表 2 に示す。

実験の結果、パッケージソフト A, B, C に関しては、おおむね 9 割前後の適合率・再現率となった。一方、パッケージソフト D に関しては、適合率・再現率ともに他のパッケージソフトと比較して低いスコアとなった。パッケージソフト D では、標準プロセス体系とは違う語彙で記述されているものが多く、そのため特に再現率が低いスコアとなった。本提案の技術は、事前に定義した標準プロセス体系に準拠したパッケージソフトに適用することを前提としており、実際に結果を見ても、標準プロセス体系に準拠していないパッケージソフト D ではスコアが低いことが分かる。この結果、事前に定義した標準プロセス体系に準拠しないパッケージソフトには適用不可であるが、標準プロセス体系に準拠したドキュメントを使用するパッケージソフトに対しては 9 割前後の適合率・再現率で適用できることを確認した。

表 3 各タスクにおけるドキュメントの適合率と再現率
Table 3 Results of precision and recall in each task.

タスク	手法	出力した 設計書数	正解 設計書数	不正解 設計書数	判断困難な 設計書数	適合率 (%)	再現率 (%)	F 値
タスク 1 正解設計書数：406	従来手法	362	358	4	0	98.9	88.2	0.932
	従来手法+提案手法	450	405	16	29	96.2	99.8	0.980
タスク 2 正解設計書数：121	従来手法	111	99	0	12	100	89.2	0.943
	従来手法+提案手法	121	109	7	5	94.0	98.2	0.961
タスク 3 正解設計書数：118	従来手法	102	94	6	2	94.0	79.7	0.863
	従来手法+提案手法	145	114	13	18	89.8	96.6	0.931

パッケージソフト A, B, C において具体的に対応付けが失敗したデータの分析として、False Positive だったデータは主に使用される語彙が類似していて区別が付きにくい標準ドキュメント間で多く発生していた。たとえば SDEM の標準ドキュメントである『画面処理仕様書』と『画面項目処理仕様書』は、画面全般の処理か画面項目に関する処理かの違いしかない。そのため、本来『画面項目処理仕様書』に対応付けるべきドキュメントが、『画面処理仕様書』にも合わせて対応付けがなされる場合が多くあった。この点に関しては、区別が付きにくい標準ドキュメント間で、区別を行うために特定の語彙に重み付けするなどの工夫を行うことで改善する余地はある。

一方、False Negative だったデータの原因は、主に図を多く含んでいるドキュメントで発生していた。たとえば、SDEM の標準ドキュメントである『ジョブネット構成図』に対応する PJ ドキュメントでは、図以外のテキストの情報量が少なく、語彙レベルで対応関係を行うことが困難であった。一般的にパッケージソフトのドキュメントでは、図の情報量が大きい割合を占めていることが多いため、今後は図を解析して必要な情報を抽出する技術を検討する必要があると考えている。

4.4 改修影響のあるドキュメントの推定

4.4.1 実験概要

作業プロセス間の依存関係を加味した改修業務における影響波及範囲の推定精度について評価を行う。評価方法として、過去に行われた実際の見積り業務に基づく前述のパッケージソフト A を使用したタスクを計 3 つ作成し、各タスクにおいて影響波及範囲調査として必要となるドキュメントが抽出できるかどうか評価を行った。評価を行うにあたり、パッケージソフト A の各ドキュメントに対し、タスクごとに影響波及範囲調査において必要となるドキュメントの事前定義を、過去に行われた当該タスクの改修結果および当該タスクを経験したベテラン SE の視点で作成を行った。ただし今回の評価では、調査において確認が必須であるとはいえないが不必要であるとも判断が付きにくいドキュメントがある程度存在していた。そのため、適合率の評価に関しては、判断が付きにくいドキュメントは除外し、

明らかに不必要であると判断されるドキュメントのみを対象に評価を行っている。

本評価では、本来であれば推定した作業プロセスの精度も評価する必要があるが、各タスクにおいて正解となる作業プロセスの定義が困難であったため、今回はドキュメントのみの評価に留めている。ただし、作業プロセスの推定とドキュメントは互いに関係し合うものであり、たとえば『業務フロー設計書』を見直すということは、『業務フロー設計』の作業プロセスを見直すことと同義であるといえる。

比較対象となる従来技術として、稗方らの提案する項目 CRUD 検索 [8] との比較を行った。文献 [8] では、影響波及範囲の調査漏れを防止し、開発における手戻りを防ぐ目的から、再現率に焦点を当て評価を行っている。本評価においても同方針で、設計情報ベースのみで検索を行う項目 CRUD 検索と、項目 CRUD 検索と提案手法を組み合わせた結果を比較することにより、再現率が向上し、調査漏れの防止を抑制できるかを検証し、本提案手法の有用性を示す。

4.4.2 タスクの例

以下に実験に使用したタスクの一例を示す。

【顧客要件】仕入先への発注で、納品形態区分(箱, 袋, たみなど)が入力されない場合に、納品形態名称が空白で表示されてしまうのを「XXX 様ご指定」にしたい。(XXX は仕入先の会社名)

【実現案】商品マスタ登録, 発注登録で、納品形態区分が入力されない場合、「9 (仕入先指定)」を初期値として登録する。その場合、納品形態名称の表示で、仕入先名称と組み合わせ「XXX 様ご指定」と固定で表示する。

【調査ポイント】商品マスタ登録画面, 商品マスタ発注登録画面を改修した場合における影響波及範囲を調査する。

4.4.3 実験結果

各タスクにおける結果を表 3 に示す。適合率は出力した正解設計書数を出力した正解設計書数と不正解設計書数の和で割って算出し、再現率は出力した正解設計書数をタスクごとの正解設計書数で割って算出している。実験の結果、項目 CRUD 検索に提案手法を加えることにより、改修範囲となるドキュメントの再現率がすべてのタスクにおいておよそ 10~15% 高いスコアとなった。項目 CRUD 検

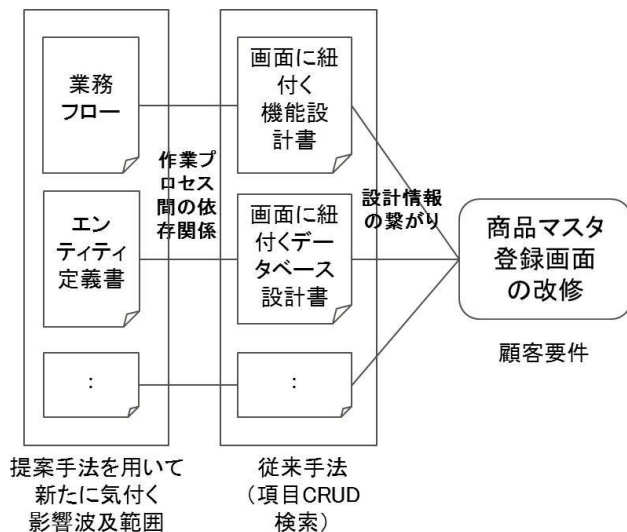


図 10 提案手法の効果

Fig. 10 Effectiveness of our proposed method.

索のみの場合、機能やテーブルなどの設計情報に紐づくドキュメントは抽出できるが、特に上流工程で作成される業務フローなどのドキュメントは抽出することができていなかった。その点で、図 10 に示すとおり、本提案手法を加えたことにより、プロセス間の依存関係に基づいてより網羅的で適切に影響波及範囲を推定し、再現率が向上することを確認した。この特に上流工程への影響波及を見落とさないということは、開発の手戻りを防ぐことを意味し、結果として見積り時の金額以上に開発コストが発生してしまうリスク回避へとつながる。さらに可視化 UI を通して、影響波及範囲の直感的な理解を促進することで、ベテラン SE と非ベテラン SE 間の属人性の問題の解消に加え、見積り業務を効率化する効果も期待できる。なお、提案手法を加えても抽出できなかったドキュメントは、主に標準プロセス体系との対応付けに失敗していたことに起因していた。

適合率に関しては、従来の項目 CRUD 検索に本提案手法を加えることにより、スコアが低下している。従来の項目 CRUD 検索と提案手法を組み合わせることにより、ユーザが確認する必要がある設計書数は増加するため、必然的に適合率のスコアは低下する。しかし、本提案手法では、可視化 UI を通して、明らかに関係のない設計書に関しては、作業プロセス単位でフィルタリングすることにより、ある程度軽減できると考えている。また、前述のとおり、本研究は影響波及範囲の調査漏れを防止し、開発における手戻りを防ぐ目的から、再現率の向上を重視している。その点で本実験結果では F 値のスコアは増加していることから、許容できる適合率の減少幅に基づき、再現率の向上を達成できたと考えられる。

5. 結論

本研究では、ソフトウェアシステムの改修における影響

波及範囲の把握支援を目的とし、開発プロジェクトにおけるプロセス間の依存関係を利用して影響波及範囲を推定し、可視化するシステムを開発した。具体的には、標準プロセス体系における作業プロセスとドキュメント間の関係性を活用し、個々のプロジェクトにおけるドキュメントを対応付けることにより、依存関係の推定を行う手法を提案した。これにより、ある改修が生じた際、設計情報に関する依存関係に加え、プロセス間の依存関係に基づく作業プロセスやドキュメントを抽出する。また、これらの依存関係を可視化するシステムを構築することにより、開発プロジェクトに見識の少ない担当者でも、改修による影響波及範囲の特定を容易とする。

実際に、提案手法および開発したシステムを複数の業務タスクに適用した結果、ドキュメントベースで 95%以上の再現率で影響波及範囲の抽出を実現した。特に、開発プロジェクトの上流工程におけるドキュメントなど、設計情報の依存関係だけでは抽出困難なドキュメントを抽出できている点で、本研究の有用性を示した。今後は提案手法を実際の業務現場に適用し、見積り業務などにおける影響波及範囲調査に与える本手法の効果の検証、および利用者のレビューに基づく可視化 UI の改良を行う。

参考文献

- [1] 経済産業省：我が国産業の強さを活かす IT 投資の在り方，経済産業省(オンライン)，入手先 (<http://www.meti.go.jp/report/downloadfiles/g70702c01j.pdf>) (参照 2017-05-22).
- [2] 経済産業省：平成 25 年情報処理実態調査報告書の概要，経済産業省(オンライン)，入手先 (<http://www.meti.go.jp/press/2014/05/20140528004/20150528004.1.pdf>) (参照 2017-05-22).
- [3] 元山 厚，松下直樹，福島健太：設計仕様の整合に基づいた設計品質の向上，プロジェクトマネジメント学会誌，Vol.14, No.5, pp.18-23 (2012).
- [4] 宇田川佳久：ウォーターフォールモデルに基づく情報システム開発における成果物の品質管理手法について，情報処理学会論文誌，Vol.49, No.2, pp.922-929 (2008).
- [5] 穴田啓樹，渡辺政彦，高田広章：一般社団法人 TERA の紹介(前編)：安心・安全・快適な社会のために全ドキュメントのトレーサビリティを目指す，SEC Journal, Vol.7, No.4, pp.183-187 (2011).
- [6] Antoniol, G., Canfora, G., Casazza, G., Lucia, A.D. and Merlo, E.: Recovering traceability links between code and documentation, IEEE Trans. Softw. Eng., Vol.28, No.10, pp.970-983 (2002).
- [7] 海谷治彦，長田 晃，原賢一郎，海尻賢二：要求変更によるソースコードへのインパクトを分析するシステムの開発と評価，電子情報通信学会論文誌 (D)，情報・システム = The IEICE Trans. Information and Systems (Japanese edition), Vol.93, No.10, pp.1822-1835 (2010).
- [8] 稗方 和夫，大和裕幸，深田直人，中村 覚，岡田伊策，齋藤 稔，笈田佳彰，渡辺郁雄，松本 滋：1305 システムの仕様変更調査における設計情報を用いた影響分析システムの開発 (OS6 知識マネジメント・情報共有)，設計工学・システム部門講演会講演論文集，Vol.2014.24, pp.1305-1-1305-10 (2014).
- [9] 室中健司，原 直朗：システム構築の標準プロセス体

系：SDEM (特集新たな時代の SI 技術), *Fujitsu*, Vol.63, No.2, pp.193-199 (2012).

- [10] 向坂太郎, 日下照英, 浅田隼人, 岡野信保: 2214 日立システム開発方法論 HIPACE のグローバル化への取り組みとオフショア開発への適用・評価 (一般セッション), プロジェクトマネジメント学会研究発表大会予稿集, Vol.2014, pp.196-200 (2014).
- [11] 駱琴, 渡邊豊英, 杉江昇: 帳票文書の構造認識のための書式構造知識の自動獲得, 電子情報通信学会論文誌 (D-II), 情報・システム, II-情報処理 = The Trans. Institute of Electronics, Information and Communication Engineers, Vol.76, No.3, pp.534-546 (1993).



福田 貴三郎

2011 年慶應義塾大学理工学部管理工学科卒業。2013 年同大学大学院理工学研究科修士課程修了。同年富士通研究所入社。現在、同研究所に勤務。2016 年東京大学共同研究員。2018 年カリフォルニア大学サンタクルーズ

校 Assistant Specialist。ソフトウェア工学, ユーザインタフェース, 対話システム等の研究に従事。



村瀬 健太郎

1997 年京都大学工学部電気系学科卒業。1999 年同大学大学院工学研究科電子通信工学専攻修士課程修了。同年富士通研究所入社。音声合成, 対話システム等の研究に従事。電子情報通信学会, ヒューマンインタフェース学会

会員。



中村 覚 (正会員)

2017 年東京大学大学院新領域創成科学研究科博士後期課程修了。博士 (環境学)。同年東京大学情報基盤センター学術情報研究部門助教。現在に至る。オントロジーを用いた知識処理に関する研究等を行っている。情報処理学

会, 人工知能学会, Alliance of Digital Humanities, 日本船舶海洋工学会等の会員。



穂方 和夫 (正会員)

1974 年 12 月 17 日生。2000 年東京大学大学院工学系研究科環境海洋工学専攻修士課程修了。同年 4 月日本アイ・ビー・エム株式会社入社。2004 年東京大学大学院工学系研究科環境海洋工学専攻助手。2007 年 4 月同助教等

を経て 2010 年 2 月より東京大学大学院新領域創成科学研究科准教授。所属学会は, 日本船舶海洋工学会, 人工知能学会, 日本機械学会, Society of Naval Architects and Marine Engineers 等。



笈田 佳彰

2010 年東京大学工学部システム創成学科卒業。2012 年同大学大学院新領域創成科学研究科修士課程修了。同年富士通株式会社入社。2015 年東京大学大学院新領域創成科学研究科博士後期課程入学。現在に至る。システムイ

ンテグレーション, プロジェクトマネジメントに関する共通技術の研究開発に従事。



松本 滋

1989 年富士通株式会社入社。フィールド SE として多くの基幹系業務システムの開発や導入プロジェクトに携わり実務経験を積む。現在は流通業向けパッケージソフトウェアの企画・開発に従事し, 主に開発における設計品質

の向上に取り組んでいる。



岡田 伊策 (学生会員)

1984 年富士通株式会社入社。現在まで同社に勤務。2011 年東京大学共同研究員。2013 年同大学大学院新領域創成科学研究科博士後期課程入学。2014 年より同大学主席共同研究員。人工知能系技術, 自然言語処理技術の社内実

践, およびシステム・ダイナミックスを用いた人工知能系技術, 自然言語処理技術適用効果評価法に関する研究に従事。人工知能学会, 電子情報通信学会, 言語処理学会, System Dynamics Society 等の会員。