

業務プロセスのパターン化による 大規模業務システム開発の効率化とその評価

岡村 美和^{†1†2} 菊島 靖弘^{†2} 青山 幹雄^{†2}

本稿では、業務システムの要求分析において、業務プロセスをユーザが扱うデータタイプと操作に着目してパターン化を行い、業務プロセスの再利用性を高める開発方法を提案する。提案方法により、パターンを使用した業務プロセスを体系立てて設計できる。提案方法を業務形態や業務システム運用の異なる複数の組織が利用する日本政府の大規模人事システムの開発へ適用した。この結果、371の業務プロセスを15パターンで実現し、多様なユーザ要求の統一とソフトウェアの再利用性を高めたことで、開発規模を従来のシステムより約7割削減した。

A Development Methodology Based on Business Process Patterns and its Application to the Development of e-Government System

MIWA OKAMURA^{†1†2} YASUHIRO KIKUSHIMA^{†2} MIKIO AOYAMA^{†2}

We propose a development methodology for business information systems by reusing the business process patterns. The methodology enables to identify business process patterns at the requirements engineering by focusing on the data type and its operations by users along with the business processes. We applied the proposed methodology to the development of human resources management systems for the Japanese government to be used by multiple different ministries and agencies operated under different rules. The proposed methodology enabled to implement 371 business processes with 15 business process patterns, and reduce software to be developed by 70%.

1. はじめに

ネットワークの高速化やサーバの処理性能向上を背景として、情報システムの開発・運用コスト削減を目的としたシステム統合が進められている。日本政府においても、2004年に複数の府省や地方拠点において個別に開発・運用していたシステムを統合し、情報システム経費のコスト削減に取り組むことを決定した。

システム統合においては、システムを利用するユーザの拡大や他システムとの連携インタフェース等の増加により、ユーザ要求が多様化し、システムが複雑化する。それにより、システムの開発と保守コストが増大するリスクが高い。

コスト増大を軽減するためには、業務要求分析においてシステムを利用して実現する業務プロセスを整理し、システム要求を合理的かつ簡潔な構造にすることが重要である。

本稿では、要求分析で業務プロセスをパターン化することでシステム開発を効率化する開発方法を提案し、日本政府向けの大規模業務システム開発へ適用した効果を示す。

2. 適用事例

2.1 対象システム

適用事例は、府省ごとに個別の人事・給与等業務に関するシステムを統合し開発された、30府省庁が共同で利用する府省共通人事・給与等業務情報システム(以下、人給シ

テムと略記)である⁶⁾。業務を効率化するために5つの他システムと情報連携し、約30万人の国家公務員が利用する大規模業務システムである。

2014年4月時点で12府省が稼働しており、その他の府省も順次稼働を予定している。

人給システムは、人事、給与、共済、勤務、宿舎等の各サブシステムで構成されており、本稿ではそのうち人事サブシステムへの適用事例について述べる。本開発方法は、人給システムの中核的な情報であり、各サブシステムで利用される職員情報を管理する人事サブシステムの開発において考案した。その後、他のサブシステムへ展開している。

2.2 研究課題とその背景

わが国では、人事院規則をはじめとする諸制度で細部にわたる業務、および人事運用の規定が定められており、各府省は、それに基づき人事給与等業務を行っている。各府省の業務は諸制度による規定に基づいているが、勤務形態などの違いにより、各府省の業務の詳細は異なる。

例えば、多くの職員は標準的な業務スケジュールに沿って業務を行うが、海上保安庁の一部職員はその時間の多くを船上で過ごす。そのため、システムの利用可能時期が限定され、短期間に一括して処理を実施しなければならない。また、多くの職員は平日日中の勤務だが、空港などのように24時間365日の勤務形態もあり、勤務形態に応じて支給される手当が異なるなどの特徴がある。

このような多様な業務要求を吸収し、システム統合によるコスト削減を実現するためには、システムへの要求を体

^{†1} 富士通株式会社
Fujitsu Limited

^{†2} 南山大学
Nanzan University

系立てて整理し、実装に反映させる必要があった⁷⁾。

3. アプローチ

多様な業務要求を吸収し、合理的かつ簡潔な構造を持つシステム要求を作成するため、要求分析において業務プロセスをパターン化する開発方法を提案する。

開発を効率化するために、実装段階においてフレームワークを活用し、実装規模を削減している。このフレームワークが担う共通部分を増やし、個別開発を削減するためには、設計時にパターン化ではなく、要求分析において業務のパターン化が必要である。特に、業務システムは様々な形態があり、それらをパターン化するには、業務要求を分析し、パターン化の方針を定めることが重要である。

あわせて、本開発方法の大規模業務システム開発における有用性を評価する。

4. 業務パターンに基づく開発方法論

業務要求分析と業務プロセスのパターン化に基づく開発方法論を説明し、具体的な実施手順について述べる。

4.1 業務パターンに基づく開発プロセス

提案する開発方法のプロセスを図1に示す。

本開発方法においては、業務要求仕様書に定義された業務フロー図を元にパターン分析を行い、業務フローパターンを抽出する。さらに、業務フローパターンを詳細化し、その構成要素を抽出して業務プロセスのパターン化を行い、業務プロセスパターンを定義する。最後に、業務プロセスパターンを組み合わせることで業務プロセスを設計する。

4.2 業務パターンの構造

業務パターンの構造を図2に示す。

業務パターンは、業務フローパターンと業務プロセスパターンから成る。

業務フローとは、ユーザが業務要求として、業務ごとにシステムを利用して行う業務の流れを業務フロー図に示したものである。例えば、提出された人事異動案から人事異動情報を作成し、確定する。業務フローパターンとは、業務フロー図におけるユーザのシステムの利用の流れが同じものをパターンとして抽出したものである。

業務プロセスパターンとは、業務プロセスを構成するデータタイプと操作からなり、業務におけるシステムの操作手順をパターン化したものである。これは、オブジェクト指向の考えに沿っているが、ユーザによる業務レベルで、かつ、要求分析でパターン化を行う点に特徴がある。

業務プロセスは、業務を遂行する手続きの最小単位である。例えば、届出申請や業務情報の修正がある。業務プロセスは、データとそれを操作するアクションで構成する。

4.3 業務要求定義

多様な業務要求を漏れなく抽出するため、各府省のユーザが参画して業務要求を定義した。業務要求定義において

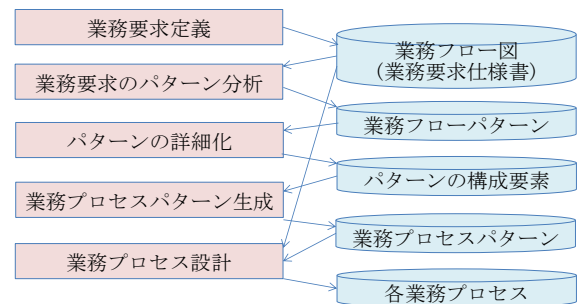


図1 業務プロセスパターン化に基づく
業務システム開発プロセス

Figure 1 Development Process of Business Information Systems Based on Business Process Patterns

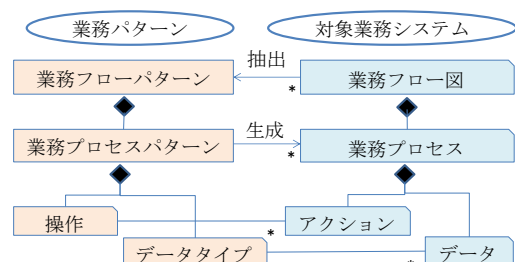


図2 業務パターンの構造

Figure 2 Structures of Business Process Patterns

作成した成果物は、下記のとおりである。

- (1) 業務一覧表
- (2) 業務スケジュール
- (3) 業務フロー図
- (4) 業務要求定義表

本開発方法論では、業務フロー図を拡張し、図3に示すように、誰(Who)が、何(What)を、に加えて、どのように(How)業務を行うかを表現可能にした。これを本稿では拡張業務フロー図と呼ぶ。

従来の業務フロー図は、誰(Who)が何(What)を処理するかは記載されていたが、どのように(How)手続きするかまでは記載されていなかった。そこで、「受領」や「登録」といったあいまいな表現を排除し、「入力/修正」や「登録(IMPORT)」、「移出(EXPORT)」、「入力(入力チェック)」、「生成」など、ユーザが行う手続きの内容とシステムの振る舞いを具体的に表現した。また、人手による手続きとシステムが担う手続きを明確にし、システムの出力を業務でどう利用するかを明確にした。

この結果、拡張業務フロー図において、「登録」とは、情報を画面から入力するか、CSV ファイルを用いて一括入力するか、「受領」とは、手作業か、ワークフローによる情報の回送か、情報の一括出力かが判別可能になる。帳票については、定型帳票か、再加工可能な帳票かが明確になる。

業務形態の異なるユーザが、拡張業務フロー図に対し、業務スケジュールや業務遂行の条件が記載された他の業務要求仕様書とあわせて、業務実現可能性の観点でレビューした。これにより、異なる業務の運用形態における業務要

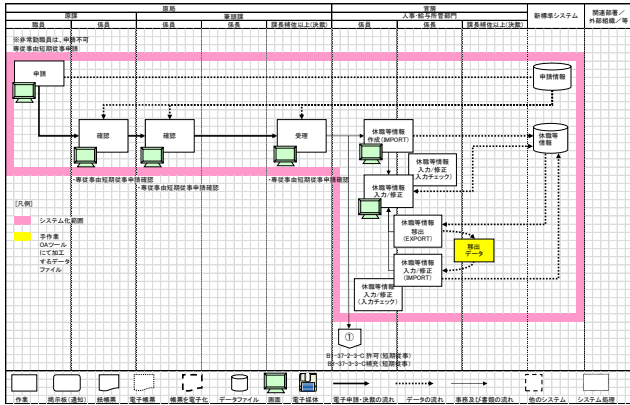


図3 拡張業務フロー図の例

Figure 3 An Example of Extended Business Flow Diagram
 求を漏れなく抽出した。あわせて、業務システムの中で情報システムが実現する範囲を明確にし、業務における入出力を決定した。

4.4 業務フローのパターン化

業務要求仕様書の拡張業務フロー図を用いて、業務フローのパターン化を行った。その目的は、業務要求からシステム化の範囲を抽出し、俯瞰することで、システムの再利用可能性を査定することにある。それに基づいて、業務フローを再利用するためのパターン化の方針を定めた。

複数の府省のユーザにより作成された拡張業務フロー図は多様な組織に跨るものであったが、一定の規則性を持つことがわかった。そこで、拡張業務フロー図から、誰と何を抽象化し、業務フローパターンとして抽出した。

その結果、業務フロー図は表1に示す10のパターンとその組み合わせで構成できることがわかった。業務フローパターンの例を図4に示す。

業務フローのパターン化分析により、次の3点が明らかになった。

(1) 主要パターンの存在

パターン No.1~No.6 の出現率は全体の 80%以上を占めていることがわかった。

(2) 業務フローパターンから業務のパターン化の可能性

業務フローパターンを詳細化すると、共通の業務が複数の業務フローパターンにおいて実行されていることが分かった。これによって、業務プロセス単位でのパターン化とその再利用が可能となることがわかった。

(3) 中核的データの存在

パターン No.1 が出現する 19 の業務フロー図のうち、16 においては、最初に職員情報から業務で使用するデータを抽出し、生成するという共通性のあることがわかった。これは、業務における中核的なデータとして職員情報が位置づけられることを示している。

(4) パターンの共通化

パターン No.1 と No.2~No.3, パターン No.1~No.2 と No.5~No.6, あるいはパターン No.4 と No.8 のように業務

表1 業務フローパターン一覧

Table 1 A List of Business Flow Patterns

パターン	概要	出現数	出現率
1	入出力処理	19	26%
2	画面入力処理	4	6%
3	エクスポート処理	30	42%
4	帳票出力処理	3	4%
5	職員情報メンテナンス	3	4%
6	職員情報一括入力	2	3%
7	ワークフロー	4	6%
8	一括更新処理	4	6%
9	データ配信	2	3%
10	他システム連携	1	1%
合計		72	100%

フローパターン間に次のような関係があることがわかった。

1) 包含関係

パターン No.1 はパターン No.2 とパターン No.3 を含んでいる。また、パターン No.4 はパターン No.8 を含んでいる。

2) 手続きの共通性

パターン No.5~No.6 はパターン No.1~No.2 の対象のデータを置き換えることで実現できる。

この分析結果に基づき、業務フローパターンを中心に業務フロー図を詳細化し、再度組み合わせることで、汎用的な業務プロセスのパターンを生成する方針とした。

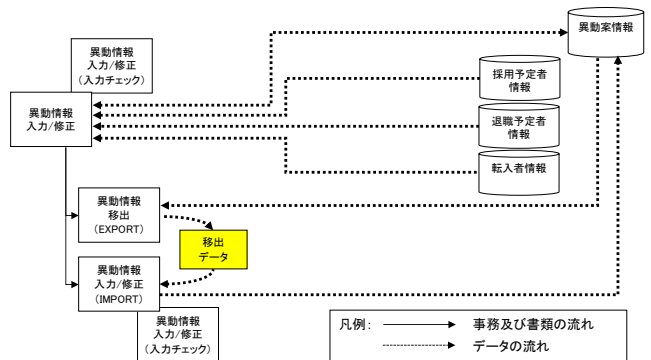


図4 業務フローパターン例 (入出力処理)

Figure 4 An Example of Business Flow Pattern Diagram

4.5 業務プロセスパターンの構成要素の抽出

業務プロセスパターンを構成する要素をデータタイプとその操作の対として抽出し、その関係を分析した。

(1) 業務フローパターンから操作を抽出

業務フローパターンごとに、ユーザが何を対象にどうアクションするか観点で手続きを詳細化し、そこから操作を抽出した。その結果を表2に示す。

(2) 業務フロー図からデータタイプを抽出

対象システムのデータを管理特性によって分類したものをデータタイプとした。対象システムのデータタイプは、表3に示すとおり、4つに分類できた。

業務プロセスで扱うデータの特徴を分析するためデータタイプの関連をそのライフサイクルに沿って分析し、図5に示すデータ関連図として定義した。

4つのデータタイプの中で職員情報が中核的なデータ

イブである。業務情報と届出情報は人給システムの中核である職員情報を参照して作成され、処理の確定により、職員情報が更新される。また、職員情報は、他のサブシステムからも参照される。コード情報は変動が少なく、再利用性の高いデータをコード化したものであり、全サブシステムから参照される。

表 2 業務プロセスパターンの操作一覧

Table 2 A List of Operations in Business Process Patterns

操作	操作の定義
情報生成処理	情報を抽出し、別の情報を生成する
検索処理	画面で表示するための情報を検索する
メンテナンス処理	画面入力での情報の登録・更新・削除を行う
エクスポート処理	画面から情報を一括エクスポートする
インポート処理	画面から情報を一括入力する
入力チェック処理	入力時のチェックを行う
帳票生成処理	定型帳票を生成する
ワークフロー	情報を回送する
バッチ処理	計算等の自動処理によるデータ更新
データダウンロード	画面からデータをダウンロードする

表 3 データタイプ一覧

Table 3 A List of Data Type

データタイプ	データタイプの定義
職員情報	職員の属性を管理する情報 氏名・階層・異動履歴など職員の属性が階層で管理される
業務情報	職員情報を作成、更新する上で必要な業務処理の情報 採用や退職、異動など業務イベントごとに管理される
届出情報	届出申請の内容を管理する情報 (基本情報の届出などワークフローで管理される)
コード情報	複数の業務から参照される一元的に管理すべき情報 組織コードや発令文コードなど履歴と共に管理される

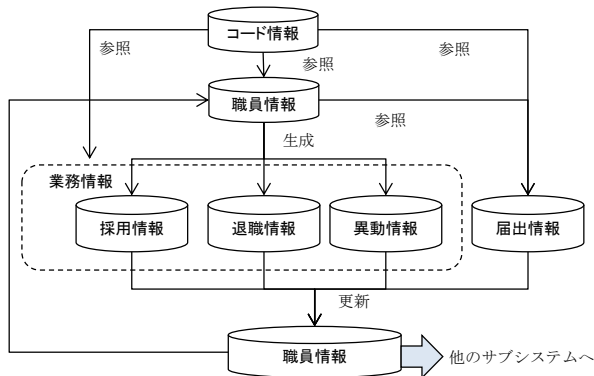


図 5 データ関連図

Figure 5 Data Relationship Diagram

4.6 業務プロセスのパターン化

業務プロセスパターンの操作とデータタイプは、図 2 に示すように、業務プロセスのアクションとデータを抽象化したものである。表 2 と表 3 からデータタイプごとに業務において求められる操作を組み合わせることで業務プロセスパターンを生成した。業務プロセスパターンには、複数の操作を含み、操作の順序も含めてパターンとして定義する。

(1) 業務プロセスのパターン化の方法

業務プロセスのパターン化の方法を、業務情報に関するパターンを例として説明する。

図 6 は業務情報に関する 6 つの業務プロセスパターンを示

す。業務情報作成パターンは、検索処理と業務グループ生成処理の 2 つの操作で構成され、職員情報から業務情報を生成する。業務情報メンテナンス等、5 つの業務情報を利用する業務プロセスパターンは、業務情報の参照や更新の操作で構成される。

業務プロセスパターンから、画面レイアウトイメージを作成し、各パターンの具体的な操作手順を詳細化した。業務プロセスパターンは、操作の順序を含むため、業務プロ

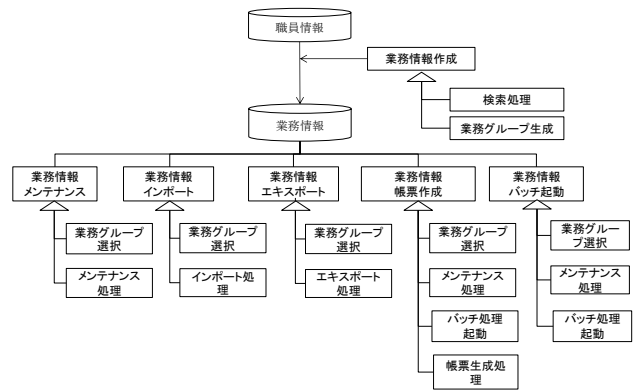


図 6 業務情報に関する業務プロセスパターン

Figure 6 Business Process Patterns with Business Data

セスパターンを介して、画面遷移と画面レイアウトに対するアクションを組み込むことができた。

生成した業務プロセスパターンを表 4 に示す。

(2) 業務プロセスのパターン化の基準

業務プロセスパターンは単一の業務手続きが完了できる単位である。パターン化にはパターンの粒度が重要である。パターンが大きすぎると柔軟性が低下し、小さすぎると共通要素が減少する。本事例では、パターン化の単位を最小業務手続きとした。最小業務手続きとは、ユーザが業務を行う上で、それ以上分割できない業務目的を完了できる手続きの単位である。例えば、ユーザが必要とする情報を画面から選択し、出力する。この手続きは画面から選択するだけでは、完了せず、これ以上分割できない。

最小業務手単位で業務プロセスパターンを定義することにより、それを組み合わせることで業務が実現できる。さらに、業務プロセスパターンの組み合わせによって、多様な業務要求が実現可能となる。

表 4 業務プロセスパターン一覧

Table 4 A List of Business Process Patterns

データタイプ	業務プロセスパターン	内容
業務情報	業務情報作成	業務処理単位のグループを作成する
	業務情報メンテナンス	グループを選んで業務情報を更新する
	業務情報エクスポート	グループを選んで業務情報を一括出力する
	業務情報インポート	グループを選んで業務情報を一括更新する
	業務情報帳票作成	グループを選んで業務情報を帳票出力する
	業務情報バッチ起動	グループを選んでバッチで業務情報を更新する
職員情報	職員情報メンテナンス	職員情報を選んで職員の属性を更新する
	職員情報エクスポート	職員情報を一括出力する
	職員情報インポート	職員情報を一括更新する
	職員情報帳票作成	職員を選んで職員の属性を印刷する
届出情報	届出申請	変更届出情報を入力し、管理者が承認する
	認定簿出力	届出情報を元に法定帳票を出力する
コード情報	マスタ情報エクスポート	マスタを選んでマスタ情報を出力する
	マスタ情報インポート	マスタを選んでマスタ情報を更新する
	マスタ情報処理起動	マスタを選んでバッチで一括処理する

(3) 業務プロセスパターンの改良

一つの業務プロセスパターンを追加することにより、他の複数のパターンを改良できる。

図6に示した業務プロセスパターンにおいては、最初に行う共通の操作として、業務情報生成処理があった。これを共通パターンとして抽出し、業務情報作成パターンとする。それにより、業務情報作成以外のパターンでは、業務情報生成処理に代わり、業務グループ選択とできる。これにより、パターン間で同一の業務情報を利用可能にし、パターンの組み合わせを容易にした。

4.7 業務プロセスの設計と業務プロセスパターンの検証

業務フロー図と業務プロセスパターンを元に、業務システムを開発するために、業務とシステムの流れを図7に示す3層構造で設計する方法をとった。この3層構造は、実際の業務の階層と対応している。

最上位の1層目はサブシステム内における業務とデータの流れを示すシステムフローである。2層目は、業務を遂行するための手続きの流れを示す処理フローである。3層目は処理に必要な機能の流れを示す機能フローである。この3層は、業務の階層と対応しているので、業務システムのメニュー階層と対応づけることができる。

次に、業務プロセスパターンを用いた設計手順を示す。この設計手順を用いて、パターンにより業務が実現可能かを実装前に検証した。

(1) システムフロー(1層目)

システムフローは、業務フロー図を元に、サブシステム内の業務体系を整理し、作成した。図5のデータ関連図を元に、業務とデータの流れを整理し、他サブシステムや他システムと授受するデータを明確にした。

(2) 処理フロー(2層目)

処理フローは、システムフローと業務フロー図を元に、業務プロセスパターンを組み合わせで作成した。処理フローにおいては、業務の入出力と実行条件を明確にした。業務処理間の媒体がシステムから出力したデータか、手作業かを明確にし、業務プロセスパターンの組み合わせで業務が実現可能かを確認した。

(3) 機能フロー(3層目)

機能フローは、業務の最小手続単位である単一の業務プロセスパターンに対応する。処理フローと業務プロセスパターンを元に、業務プロセス内の画面遷移と機能の流れを作成した。機能フローの画面遷移はパターンによって構成されているため、入力チェックやデータ更新など、パターンでは対応できない機能を付加した。それにより、パターンで実現できる部分とできない部分を明確にした。

5. 評価

5.1 開発・保守コストの削減効果

本開発方法の適用効果を従来型開発と比較し、評価した

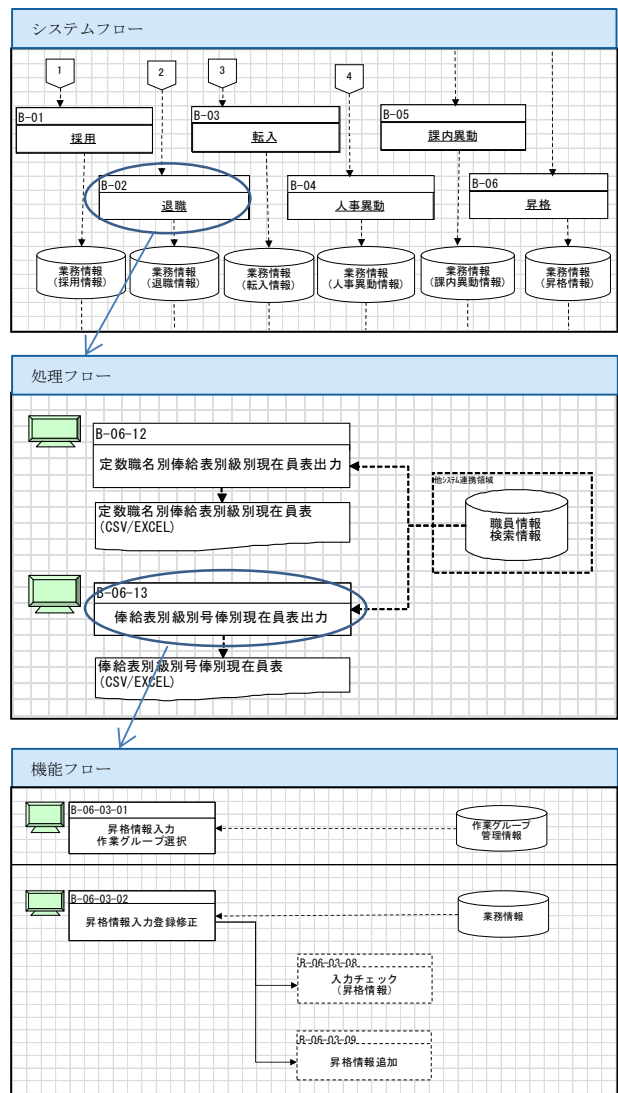


図7 業務プロセスの設計書の体系図

Figure 7 Systematic View of the Business Process Diagram

結果を図8に示す。

(1) 開発規模の削減

従来のシステムと比較し、画面数、帳票数は約4倍と増加した。これは、従来の個別最適な画面構成から、汎用的な画面構成へ変更したことと、業務の柔軟性を高めるため、様々なデータを画面からインポート、エクスポート可能にしたことによるものである。

画面数と帳票数は増加したが、業務プロセスをパターン化した結果、386の業務プロセスのうち、371の業務プロセス

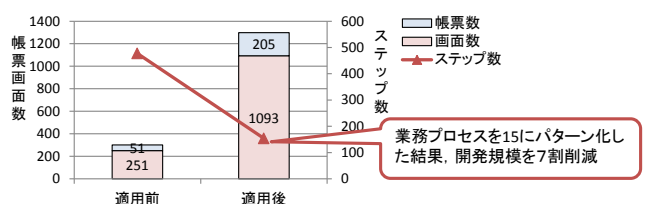


図8 パターン適用比較

Figure 8 Comparison of Business Process Patterns Application

スを 15 パターンで実現した。残り 15 の業務プロセスはパターンの適用外となった。適用外の業務プロセスは、人事異動のシミュレーションを行う業務であり、業務固有の操作性が要求されるものである。

業務プロセスパターンは画面遷移と画面レイアウトを固定するため、業務ごとの画面開発が不要となった。さらに、業務プロセスの設計においてパターンで実現できる部分とできない部分を明確化したことにより、実装における共通部分を増やし、開発規模を約 7 割削減した。

(2) 業務プロセスパターン別の適用率

業務プロセスパターン別の業務プロセスへの適用率を図 9 に示す。図 9 は業務プロセスパターンの適用率が高い順に並べたパレート図である。折れ線は累積適用率を表す。

この図から、主要 5 パターンが業務プロセスの約 65% に適用されていることが分かる。この 5 パターンは帳票作成やエクスポート、インポートに関するものであった。これは、出力形式や入力形式が異なる業務プロセスをパターンで吸収できた効果である。主要な出力形式や入力形式には、CSV, Text, Excel 等がある。

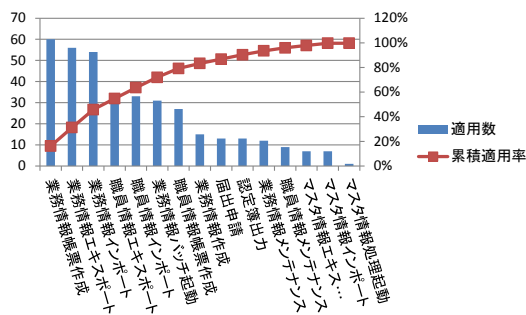


図 9 業務プロセスパターンの適用数と累積適用率

Figure 9 Pareto Diagram of Business Process Patterns Reuse

6. 考察

6.1 パターン化による課題の解決について

本提案方法は、パターンを業務の最小単位で作成することにより、パターンの組み合わせで業務プロセスを作成できるため、パターンの適用を高めることができた。その結果、開発規模削減に効果があることが実証できた。

6.2 関連研究との比較

オブジェクト指向分析によりソフトウェアの再利用性を高める設計ノウハウとして、「オブジェクト指向における再利用のためのデザインパターン」や「アナリシスパターン」がある¹⁾²⁾³⁾⁴⁾。これらは、よく知られている一般的なビジネス領域における問題に対する解決策を示している。

アナリシスパターンは、分析において、メタクラスと振る舞いに着目した問題解決方法を示している。しかし、パターンの例は在庫管理と会計などに留まり、かつ、ユーザによる業務遂行の視点からのパターン化は扱われておらず、パターンを用いた業務設計の方法までは提案していない。

本提案方法では、パターンを用いて業務プロセスを設計する具体的な方法を示している点で実践での有用性が高いといえる。

また、業務プロセスのモデリングに関する研究⁵⁾では、モデリングツールを提供し、類似業務の業務プロセスパターンを用いて業務分析を行うことで設計期間を短縮する方法が提案されている。本提案方法論では、業務要求分析を元にパターン化に取り組むことで、業務要求を踏まえたパターンの生成を可能としたことにより、パターンの適用率は先行研究と比較して、高い結果を得ることができた。

7. まとめ

人給システムの開発で実践した業務プロセスのパターン化に基づく業務システム開発方法論は、開発規模の大幅な削減を実現した。その主たる要因は、ユーザによる業務要求定義の明確化と、それを分析してパターン化し、システム要求を合理的かつ簡素な構造にできたことにある。

この開発方法は他サブシステムへパターンが展開できたことから、特定の業務特性によらず適用可能である。

業務システムには様々な特徴や形態があり、パターン化のアプローチも様々である。重要なのは、業務要求分析において再利用可能性を検討し、パターン化の方針を定めることである。

しかし、フレームワークの活用による開発規模の削減には定常的に取り組む一方、業務要求分析におけるパターン化は必ずしも実施されているわけではない。本稿により、業務要求分析におけるパターン化が開発規模削減に効果があることが実証できた。

今後は、他のサブシステムにおけるパターンの適用評価と、パターンの適用による保守効率の向上の評価を行う。

謝辞 本研究においてご協力いただいた人事院と伊藤忠テクノソリューションズ株式会社の関係各位に感謝する。

参考文献

- 1) M. Fowler, Analysis Patterns: Reusable Object Models, Addison-Wesley, 1996.
- 2) E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
- 3) P. Hruby, Model-Driven Design Using Business Patterns, Springer, 2006.
- 4) R. E. Johnson, 中村 宏明, 中山 裕子, 吉田 和樹, パターンとフレームワーク, 共立出版, 1999.
- 5) R. Yamamoto, K. Yamamoto, K. Ohashi, and J. Inomata, Development of a Business Process Modeling Methodology and a Tool for Sharing Business Processes, Proc. of APSEC '05, Dec. 2005, pp. 679-686.
- 6) 人事院, 「人事・給与等業務・システム最適化計画」, 各府省情報化統括責任者(CIO)連絡会議決定, 2004年(平成16年)2月27日.
- 7) 片山 昭, 菊島 靖弘, 野村 典文, 吉田 宏明, 岡村 美和, 人事・給与システム: 大規模シェアード型府省共通業務情報システムの開発, 情報処理学会デジタルプラクティス, Vol. 4, No. 2, Apr. 2013, pp. 141-151.