

6 Y-1

ビジネスプロセス設計パターンを用いた ワークフローシステム設計開発技法の開発

大村 善秀^{*} 小林 隆^{*} 趙 廷悦^{*} 尾越 昌子^{*} 酒井 克彰^{**}^{*} (株)日立製作所 ビジネスシステム開発センタ^{**} 日立システムエンジニアリング (株)

1. はじめに

ワークフローは、業務の変化に対応して、プロセスを柔軟に変更できるというメンテナンス性の高さを特長とする[1]。しかし、現実にはワークフローシステムの保守には大きな工数がかかっている。これは、ワークフローで扱うアプリケーションプログラムが、この特長を考慮した設計になっていないために発生している。

本稿は、ワークフローのメンテナンス性の高さを活かす設計開発技法を提案するとともに、適用事例を示す。なお、本稿ではシステム化の対象となる業務を構成する処理単位を作業、作業とその実行順序をプロセスと呼ぶ。

2. ワークフローシステムの保守における問題点

一般に、ワークフローシステムを保守する上で、アプリケーションプログラムの修正に工数が大きくかかっている。この原因は、アプリケーションプログラムの仕様が、プロセスの変更を考慮せずに開発されていることにある。

ワークフローシステムの保守に関する問題点は、次の2点である。

a. 作業の追加／削除の問題

各担当者の作業に合わせて、別々に画面や処理を作り込んでいる場合、作業の統合や追加時に画面を再開発する必要がある。

例えば、現行業務で帳票の審査作業が「経理審査→資材審査」と行なわれており、各担当者ごとに専用の画面を開発しているとする。しかし、将来、2つの審査が統合される可能性もある。このとき、両方の作業ができる新しい画面を開発する必要がある。

b. 作業の実行順序変更の問題

ワークフローで閲覧する帳票のプログラムに、作業の順序性を前提としたアルゴリズムを使用している場合、順序を変更したときに影響が大きい。

例えば、aの例で、将来、経理審査の前に資材審査が行なわれる可能性もある。このとき、資材審査で経

理審査結果に応じた処理を行なっている場合、順序変更による影響を確認する必要がある。

3. ビジネスプロセス設計パターンを用いたワークフローシステム設計開発技法

前述の問題点を解決するためには、ビジネスプロセス上の固定部分と可変部分を判別し、アプリケーションプログラムをそれに合わせて開発することが必要である。

ビジネスプロセスを複数の人間の協調作業という観点からみると、その構造は対象業務に関わる規則(法, 社則), 組織(責任, 権限), 資源(人員, 設備)により変化する。

Winograd は、業務を依頼者と実行者という二者間の委託関係でモデル化した[2][3][4]。そして、ビジネスプロセスを、依頼者と実行者が本来自分が行なうべき作業を分割して第三者に委託するというように、二者間の委託関係の再帰的なネットワークで表わした。このような協調的ネットワークは、次のようなメカニズムにより種々に変化する。

例えば資材購入や勤休管理といった間接業務は、業務の目的は同一でも企業によりプロセスが異なる。第一に、金額により承認部署が変化するというような社則の違いがある。第二に、同じ総務部門でも全社で1部署の企業と各拠点ごとにある企業とでは書類回覧経路が異なる。第三に、人員数や計算機導入率により運用が異なることもある。これは、上記の規則、組織、資源というパラメータが会社ごとに異なることに起因している。

さて、システムの保守性を考える場合、上記のパラメータのうち時間的に変化しやすいものだけを考えればよい。組織のフラット化や設備の分散化、イントラネット化など、組織と資源に関する制約は時間の経過とともに変化していく。これに対して規則は時間的に変化しにくい。

組織と資源の変化に依存しないプロセスの安定な構造を導き出すために、業務を依頼者と実行者の二人だけで実施するという究極の形態を考える。これにより、協調的ネットワークは2つのノードのみとなり、組織と資源による変化はみえなくなる。この二者間の委託関係を仮定して生成するビジネスプロセスを、**設計パターン**と呼ぶ。設計パターンは、組織や資源の制約を受けない、業務の安定した基本構造を表わす。

A Method for Workflow System Design and Development Using the Concept of Business Process Design Pattern by Yoshihide OHMURA^{*}, Takashi KOBAYASHI^{*}, Jeongyul CHO^{*}, Shoko OGOSHI^{*} and Katsuaki SAKAI^{**} (Institute of Advanced Business Systems, Hitachi, Ltd. ^{*}Hitachi System Engineering, Ltd.)

例として、ある会社の資材購買のプロセスを考える。購買依頼者が注文を依頼し、経理課や資材課が依頼内容を審査する。見積が必要ならこれを行ない、取引先に対して発注を行なう。最後に納品された品物を検収して業務は終了する。金額によっては上位の部長、重役も決裁するため、審査注文依頼、審査、検収のチェックは多部門に渡る。このため、作業数は約20個ある。

このプロセスを購買依頼部門と購買実行部門の二者間だけで実施すると仮定すると、多段階の審査は発生しないため、図1に示す設計パターンを得る。これが、対象とする購買業務プロセスの基本構造である。

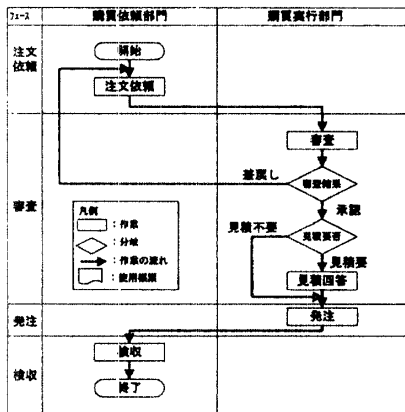


図1 購買業務の設計パターン

このように、設計パターンにより、業務で変わりにくい部分が判別できる。ワークフローシステムを開発する際は、まず対象業務の設計パターンを作成し、次に前述の制約条件の許す範囲で、新しいプロセスを設計する。次に、この設計結果にもとづいて帳票の開発を行なう。

開発にあたっては、設計パターンをもとに基本となるアプリケーションを設計する。そして、多段階の委託により発生する作業のアプリケーションは、基本アプリケーションのカスタマイズにより開発する。これにより、プロセスが変化しても、基本となる設計パターンは変わらないため、保守性の高いシステムが構築できる。

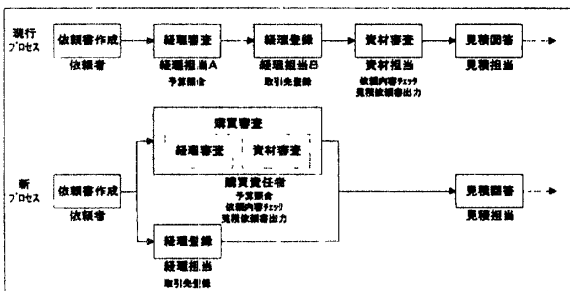


図2 購買プロセスの変更

4. 適用事例

本技法を適用した一つの事例を紹介する。図1で説明

した購買業務で、業務改善のため、経理と資材の審査業務を統合し、取引先登録と並列化した(図2)。

設計パターンによれば、経理審査と資材審査はどちらも「審査」であり、その順序、多段階委託の分割数は可変である。しかし、従来の方法ではこれが分からないため、この順序を前提として、また各々の作業専用アプリケーションを開発してしまうことが多い。

以上の前提のもとで、従来の方法では現行のアプリケーションに対して以下の修正が発生した。

①【統合の影響】経理担当者A、資材担当者が行っていた予算照合、注文内容チェックを統合する新しい画面を開発した。

②【並列化の影響】資材審査は取引先登録を前提に帳票を作り込んでいたため、見積依頼書出力のアルゴリズムを全面的に見直した。

一方、設計パターンを使って開発すると、修正は以下の作業となる。

①【統合の影響】審査一般に使用可能な帳票を開発しておき、必要に応じて予算や注文内容に関するデータを非表示にしたり分岐させている。今回の変更では両方を表示するようにカスタマイズすることで即対応可能となる。

②【並列化の影響】お互いの作業結果を前提とする設計は行なわないため、作業順序が変更されてもプログラムの作り替えは発生しない。

表1に、従来の開発と設計パターンを用いた開発の効果を整理する。

表1 設計パターンを用いた開発の効果

修正内容	従来	設計パターン利用
統合	画面帳票の再開発	基本的作業の画面帳票をカスタマイズ
並列化	変更による影響範囲の見積りと改造	順序を前提としないため修正無し

5. おわりに

本稿では、ビジネスプロセスのうち比較的变化しにくい部分と頻繁に変化する部分とを設計パターンにより明確に分離し、変化しにくい基本的作業に沿って帳票を設計することで、業務の変化に柔軟に対応できる保守性の良いワークフローシステムを設計できることを示した。

参考文献

[1]White, T., E. and Fischer, L. (Ed.), "New Tool for New Time: Workflow Paradigm, Future Strategies Inc., 1994.
 [2]Winograd, T., "Understanding Computers and Cognition" Addison-Wesley, 1986.
 [3]Medina-Mora, R., Winograd, T., et al., "The Action Workflow Approach To Workflow Management Technology" CSCW 92 Proceedings (Nov. 1992).
 [4]Winograd, T., "A Language/Action Perspective on the Design of Cooperative Work", Human-Communication Interaction 3:1, 1987/1988.