

オブジェクト指向技術を用いたワークフローシステムの開発

6 A E - 3

児玉 晴彦 村田 孝志 伊賀 靖徳 清水 俊一郎 清 和由 西川 大一 小林 巧
株式会社情報技術コンソーシアム

1.はじめに

オブジェクト指向技術の実践的な評価を行うためのプロトタイプとして、ワークフローシステムを開発した。

2.ワークフローシステムの概要

ワークフロー^[1]は仕事の流れを定義し、定義に従って仕事を遂行する仕組みである。仕事の単位であるアクティビティ^[2]を直列または並列に接続したものをプロセス^[3]と呼ぶ。プロセスにはプロセスを含むことができる。プロセスから呼び出されるプロセスをサブプロセス^[4]と呼ぶ。アクティビティは担当者(Participant^[5])に割り当てられる。アクティビティで参照・更新される文書などの情報をフローデータ^[6]と呼ぶ。フローデータは条件判断が必要な箇所でも参照される。

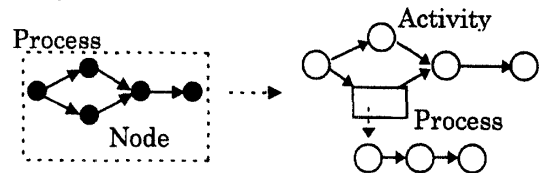
ワークフローシステムはプロセスの定義と管理を行い(定義機能)、定義に従ってプロセスを実行し(実行機能)、またその状態を監視する(監視機能)システムである。ワークフローシステムは担当者に作業指示し、担当者はフローデータを参照・更新する。ワークフローシステムと担当者が相互に作用しながらワークフローが遂行される。

3.ワークフローシステムのモデル

3.1 プロセスクラス/アクティビティクラス

Composite パターン^[7]を利用してモデル化を行った。プロセスを有向グラフと捉え、そのノードに対

応する Node クラスを導出した。プロセスとアクティビティに対応して導出したプロセスクラスとアクティビティクラスを、Node クラスを具象化したものとして定義した。プロセスクラスは Node クラスのコンテナとなる。Node オブジェクトを再帰的に組み合わせ、より複雑なプロセスを表現することができる。



3.2 プロセスの制御

プロセス定義^[8]からプロセスインスタンス^[9]が生成される。プロセスインスタンスはプロセスクラスのインスタンスに対応する。プロセスインスタンスはその下位の実行単位を Node という抽象度で認識し、それらを定義に従って実行する。

3.3 Node クラス

フレームワーク^[10]は協調関係にあるクラスの集合である。フレームワークを構成するクラスの中で、拡張が可能な部分をホットスポット^[11]と呼ぶ。反対に、拡張されない部分をフローズンスポット^[12](固定箇所)と呼ぶ。

ワークフローの自動化支援という観点から Node クラスをホットスポットとした。アクティビティは人間によって処理される Node であるが、機械的に処理できる箇所は可能な限り自動化したいというシステム要求があると予測し、この部分に柔軟性を持たせておくことにした。

3.2 節で述べた理由により、新しい Node が追加されてもプロセス制御のアルゴリズムは変わらない。この部分はフローズンスポットである。

フローズンスポットとホットスポットは Template Method パターン^[13]と Factory Method パターン^[14]を利用して実現する。

“Development of Workflow System using Object-Oriented Technology”,
H.Kodama, T.Murata, Y.Iga, S.Shimizu, K.Sei,
D.Nishikawa, T.Kobayashi,
Information Technology Consortium

3.4 担当者クラス/フローデータクラス

現実世界の担当者に対応する担当者クラスとフローデータに対応するフローデータクラスを導出した。アクティビティクラスの実行メソッドは担当者オブジェクトにアクティビティを割り当てるよう定義されている。担当者オブジェクトは割り当てられたアクティビティのリストを管理する。担当者オブジェクトはフローデータオブジェクトへの参照を保持する。フローデータオブジェクトはその属性の参照・更新のためのメソッドを持つ。

担当者クラスを抽象クラスとする個人クラス、組織クラス、役割クラスをモデル化した。

4. ワークフローシステムの実装

4.1 開発・実行環境

OMG CORBA^[4]に準じた分散オブジェクト実行環境で動作するようワークフローシステムを実装した。

4.2 ソフトウェア構成

定義機能、実行機能、監視機能を実現する分散オブジェクトサーバ4本(これらを合わせてワークフローサーバと呼ぶ)とそれらを利用するためのユーザインタフェース(UI)より構成される。

◆ 定義機能: 定義管理サーバ

◆ 実行機能: ワークフローエンジン^[1]、ワークリストハンドラ^[1]

◆ 監視機能: 監視サーバ

◆ UI: ワークフロークライアント

定義管理サーバはプロセス定義を管理するサーバである。

実行機能の2つのサーバはWfMC^[1]で定義されているソフトウェアコンポーネントに対応する。ワークフローエンジンはプロセス定義を解釈することによってプロセスインスタンスを生成し、その実行を制御するサーバである。ワークリストハンドラは担当者ごとの仕事リスト^[1](作業すべき仕事の一覧)を管理するサーバである。ワークフローエンジンはアクティビティを実行する際、それに割り当てられた担当者の仕事リストにアクティビティを追加す

る。

監視サーバは個々のプロセスの実行状態とワークフローシステム全体の稼動状況を監視するサーバである。

ワークフロークライアントの定義機能はプロセスを視覚的に定義させ、それを定義管理サーバに登録することを支援する。実行機能はプロセスインスタンスの作成・実行と仕事(アクティビティ)処理のためのインタフェースを提供する。監視機能は監視サーバが管理する情報を参照するためのインタフェースを提供する。

4.3 システムの拡張性

3.3節で述べたようにNodeはワークフローシステムのホットスポットである。NodeとそのファクトリであるNodeFactoryのインタフェースがIDL^[4]で定義されている。Nodeを拡張しようとするソフトウェア開発者は、これらを継承した分散オブジェクトが動作する新しいサーバを実装し、ワークフローシステムに追加する。システムの拡張を支援するためのC++とJavaのクラスライブラリが存在する。

メール送信のような汎用的なNodeや業務固有の処理を実装したNodeを実装し追加することによって、プロセスの一部を自動化することができる。

参考文献

- [1] WfMC (1996): *Workflow Management Coalition Terminology & Glossary*
- [2] Gamma, E. et al (1995): *Design Patterns*, Addison-Wesley
- [3] Pree, W. (1995): *Design Patterns for Object Oriented Software Development*, Addison-Wesley
- [4] OMG (1991): *The Common Object Request Broker: Architecture and Specification*

この研究は、情報処理振興事業協会(IPA)の先進的情報処理技術の開発促進事業で実施されている「オブジェクト指向によるソフトウェア生産技術開発」プロジェクトの一環として実施している。