

プロセス支援システム SOFTPIE の開発

3H-2

上原忠弘*, 森偉作**, 中山裕子*, 山本里枝子*, 吉田裕之*
 (株) 富士通研究所*, 富士通 (株) **

1.はじめに

我々は、非定型で並行に行なわれる共同開発作業であるソフトウェア開発にワークフローシステムを適用する技術を開発し[1], その技術を基にしたプロセス支援システム SOFTPIE(SOFTware Process Improvement Environment)を開発した。

本稿では、オブジェクト指向開発方法論を適用して行なったプロセス支援システム SOFTPIE の開発について報告する。

2.SOFTPIE の特徴

SOFTPIE は仕様書やプログラムソースといったドキュメント (プロダクト) の開発状況という視点で作業計画 (プロジェクト) の進捗を管理することを特徴としている。通常ソフトウェア開発では作業の工程よりもドキュメントの完成度で開発作業の進捗を管理することが多い。そこで、SOFTPIE では複数のドキュメントの作業状況を取りまとめる開発ドキュメント管理機構を導入し、作業計画に関連するドキュメントの開発状況の管理を支援する。

SOFTPIE は、このドキュメントベースの管理を行うことから、開発作業全体の順序を規定する必要がない。それぞれのドキュメントは全く独立に開発作業を開始したり、修正作業を行ったりすることができる。

SOFTPIE はドキュメント間に依存関係、参照関係を設定することにより、ドキュメント開発の流れを緩やかに規定することができる。あるドキュメントが開発完了または修正完了になると、それに依存するドキュメントの開発、修正を提言する機能を有する。また、管理者はその提言を無視して計画を進めることもできる。

SOFTPIE は個々のドキュメントの開発に関してワークフローを適用する。それぞれのワークフローは並行に実施されて、開発ドキュメント管理機構に

より制御される。

3.分析

分析では上で挙げられた SOFTPIE の特徴を反映したモデルを作成する。分析時のオブジェクトモデル図の簡略版を図1に示す。

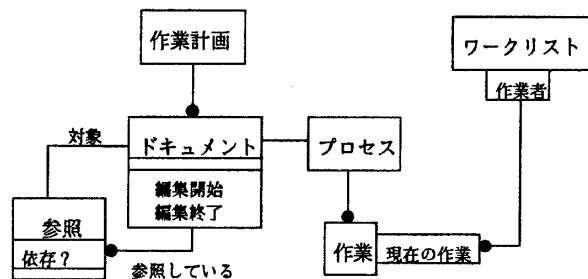


図1 分析時のオブジェクトモデル図

「作業計画」は複数の「ドキュメント」を有する。この「作業計画」がドキュメントの作業状況を取りまとめる。

「ドキュメント」はそのドキュメントの開発手順を示す「プロセス」と一対一に対応づけられる。これにより個々のドキュメントの開発を独立に進める事ができる。

「プロセス」はいくつかの開発「作業」で構成されており、現在行うべき作業が担当者の「ワークリスト」に対応づけられる。

「ドキュメント」は複数の「参照」を持つことができる。「参照」は属性「依存?」をもつ。もし、「依存?」が真ならば、「参照」に「対象」で関連づけられている「ドキュメント」は元のドキュメントに依存されているドキュメントである。「作業計画」は、各ドキュメント毎に依存しているドキュメントの作業状況を調べ作業を行うべきドキュメントの提言を行う。

SOFTPIE は複数人数での共同作業を支援するために、ドキュメント情報を共有する必要があり、情報の編集における排他制御が必要となる。そこで「ドキュメント」に「編集開始」「編集終了」の操作を付加する。

また、このようなシステムでは情報の永続性を保証する必要がある。

4. 設計

設計では他システム、インフラの利用などを考慮して分析モデルを詳細化する。

まず、SOFTPIE のシステム構成について述べる。

SOFTPIE のワークフローエンジンに当社製品である TeamWARE Flow[2]を用いる。TeamWARE Flow はクライアント・サーバ型のシステム上でワークフローを管理、自動化する。

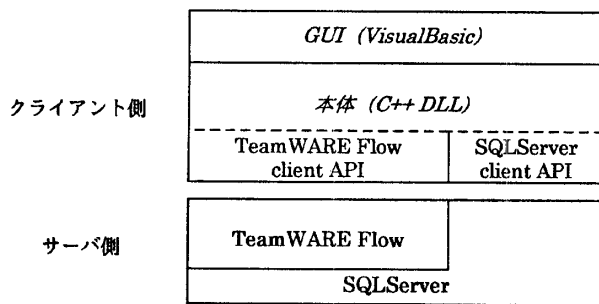


図2 システム構成

よってSOFTPIEはクライアントサーバ型のシステムである。サーバ側には TeamWARE Flow サーバと SQLServer を用いてそれぞれワークフロー情報とドキュメント管理情報を管理する。クライアント側は TeamWARE Flow クライアント API, SQLServer クライアント API を用いて SOFTPIE クライアントを構築する。SOFTPIE クライアントは C++ の本体と VisualBasic の GUI 部からなる。

つぎに、このシステム構成を考慮して、分析時のオブジェクトモデル図を詳細化した設計時のオブジェクトモデル図の簡略版を図3に示す。

新たに2系統のクラスが追加されている(太線のクラス)。一つは TeamWARE Flow から提供されるクラス (RC**) である。これらを分析により得られたクラスに対応づけ、提供される機能を利用するように設計する。ドキュメント情報の編集における排他制御については、TeamWARE Flow の各プロセス単位での排他制御機能を利用するため、「ドキュメント」にあった操作「編集開始」「編集終了」を「プロセス」に移し、点線内のクラスのオブジェクトを変更する時には「プロセス」のこれらの操作を実行してから行うよう設計した。これによりドキュメン

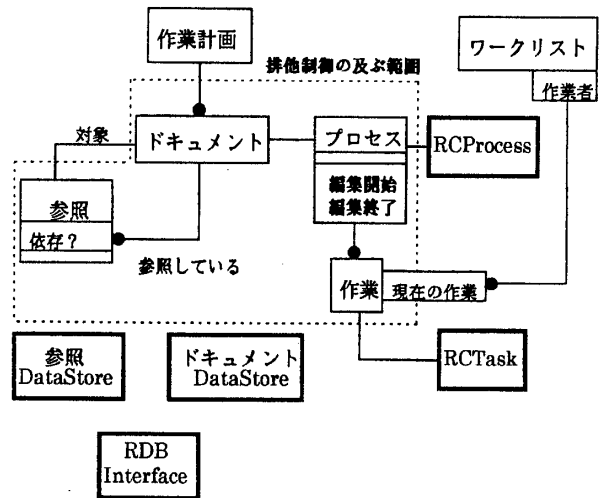


図3 設計時のオブジェクトモデル図

ト情報の編集に関する排他制御が実現される。

もう一つは永続性を保証するために追加されたクラス[3]である。永続記憶として RDB を用いるが、RDB との通信を行うクラス (RDBInterface) と特定のクラスのオブジェクトと RDB 内の情報との変換を行うクラス (**DataStore) を導入する。これにより RDB 固有の記述を隠蔽することができる。

5. まとめ

プロセス支援システム SOFTPIE の実現にあたり、分析においてシステムの特徴を反映したモデルを作成し、設計において使用するインフラを考慮してモデルを詳細化した。

SOFTPIE の実装に際し、38 のクラス、約 200 の関数、約 150 の VisualBasic への API を作成した。うち、設計時に新たに加えられたクラス数は 17 である。また TeamWARE Flow client API のクラスを 10 クラス利用した。

今後の課題として、ソフトウェア開発における SOFTPIE の有用性の検証および検証方法の検討が挙げられる。

参考文献

- [1] 山本他: プロダクトベース・プロセス支援システムの提案と実現、95-SE-105, 1995,
- [2] 富士通(株): TeamWARE Flow 使用手引書, 1995
- [3] 山本他: オブジェクト指向における RDBMS の利用, オブジェクト指向'96 シンポジウム, 1996