

## XML と CORBA を用いたワークフロー管理システムの開発

韋慶傑, 堀川健一, 桑原教彰, 吉江信夫

住友電気工業株式会社 システムエレクトロニクス研究開発センター

ネットワークシステム開発グループ

大阪市此花区島屋1-1-3

TEL 06(6466)6536

FAX 06(6462)4586

E-mail wei@rcom.sei.co.jp

**要旨** XML と CORBA を用いたワークフロー管理システムを開発した。本システムでは既存システムを連携させるため通信プロトコルに CORBA を採用した。さらに、既存システムのワークフローへの組み込みをプログラムレスで行うことを可能にする為、CORBA インタフェースの入出力パラメータを XML で表現し、システム間のデータ I/O を XML の検索言語の XQL で記述する。インタフェースの情報は CORBA のインタフェースリポジトリから動的に収集され、上記の XML マッピングがワークフロー開発者用の GUI パネル上に提示される。ワークフロー開発者はインタフェースの各パラメータに XQL を記述することで既存システムの連携を定義できる。

### Development of Workflow management system utilizing XML and CORBA

WEI QingJie, HORIKAWA Kenichi, KUWAHARA Noriaki, YOSHIE Nobuo

SUMITOMO Electric Industries, Ltd.

System Electronics R & D Center

TEL +81-6-6466-6536

FAX +81-6-6462-4586

E-mail wei@rcom.sei.co.jp

**Abstract** We have developed the workflow management system utilizing XML and CORBA. CORBA is used for integrating legacy information systems. The feature of our system is XML representation of IDL interface of CORBA systems. It enables us to describe data exchange between CORBA systems without programming. CORBA system interface information is obtained dynamically from the interface repository. Then, our system generates XML mapping of this information, and presents this mapping to us graphically. Only by using XQL(XML Query Language), we can get data from CORBA systems, and set data to CORBA systems.

### 1. はじめに

ワークフロー管理とは、複数の担当者がネットワークを經由して仕事をする際に、作業を円滑に進めるために、担当者間で受け渡すドキュメントや情報の流れを管理し、自動化する事である。そして、それを実現するのがワークフロー管理システムである。

古くからワークフロー管理システムは、定型業務を支援するために、回覧される情報の電子化、情報伝達の自動化といったことなどを実現してきた。例えば紙による決済業務では、仰裁者と決裁者は一般に物理的に離れた場所で仕事をしているため、帳票を社内輸送便で送るだけで日単位の時間がかかってしまう。この帳票を電子化しイントラネットで回覧することで多大な時間コストが短縮でき、

また情報の共有化コストの短縮も実現できる。

また近年、上記のような機能に加えて、基幹システムとの連携を指向するシステムが増えている[1]。例えば社員の給与計算を行う業務では、人事情報データベース、勤怠管理データベース、出張旅費清算システムなどが矛盾なく連携しなければならない。そのように既存の基幹システムを統合して一つのワークフローとして動作させるには、(a)マンマシンインタフェースの統合、(b)システム間の通信プロトコルの統合、(c)データ形式の統合が必要である。

最近のイントラネットベースのシステムでは、マンマシンインタフェースを Web で統合することが一般的である。既存システムとワークフロー管理システム間の通信には、業界標準の CORBA[2] が使用されることが一般的である。しかし一度統合したシステムでも、そのインタフェースが変更されたり、また新しいシステムがワークフローに追加される際には CORBA を理解したエンジニアのスキルが必要となり、システムの保守コストは無視できない。一方データ形式の統合という観点からは、多くのワークフロー管理システムはリレーショナルデータベースに基づいており、各システムのデータに対応するテーブルをワークフロー管理システムのデータベースに用意する。しかし一度統合したシステムの変更は当然のようにテーブルの設計変更に伴って波及するため、その保守コストはやはり無視できない。しかしインターネット時代においては、ビジネスの形態は短期間でドラスチックに変化するため、これにシステムが迅速かつ低コストで追従できなければ、ビジネスチャンスを失うことに繋がってしまう。

ところで、電子商取引 (Web-EDI) の分野では多種多様な企業のデータを扱う必要があり、またそのデータの形式は頻繁に変更される性質を持つ。これを効率的に扱うため従来のテーブル形式に変わるデータ表現が望まれており、XML[3]がその本命として注目されている。最近では、XML データをそのままデータベース化でき、また XML データの形式の変更に柔軟に対応できる、いわゆる XML データベースが開発され一般に利用できるようになった。

筆者らは XML データベースの性質で特に、データ形式の変更に柔軟に対応できる特徴に着目し、これをワークフロー管理でのデータ統合コンポーネントとして利用した。そして、既存システムとの通信を CORBA で統合する際そのインタフェースを XML で表現し、それらシステムから取得したデータや設定するデータを、XML データとして XML データベース上で一元的に管理する仕組みを開発した。これにより既存システムに対するアクセスを、XML の検索言語である XQL で統一的に記述できる。またシステムのインタフェースの変更には、その XQL の記述を変更するだけで対応できる。

2. では筆者らが開発したシステム Dr\_Flow<sup>TM</sup> の特徴を述べる。3. では Dr\_Flow<sup>TM</sup> のワークフローモデルを説明し、4. で Dr\_Flow<sup>TM</sup> の構成の概要を述べる。5. ではその中で特に CORBA による既存システム連携のメカニズムについて説明する。6. ではまとめを述べる。

## 2. Dr\_Flow<sup>TM</sup> の特徴

Dr\_Flow<sup>TM</sup> は業務フローを管理するシステムを構築するためミドルウェアであり、以下の機能を有している。

### (1) アドミニストレータ向け機能

#### (1-1) 対象業務で使用する帳票のデータ形式などを定義する機能

業務に必要なデータ (帳票データ) や、業務の管理に必要なデータ (ワークフロー管理データ) を Web から登録して定義する。定義されたデータ形式に応じて、帳票の入力、更新、あるいは検索を実行するための HTML のフォームが自動生成される。

#### (1-2) 対象業務のワークフローを定義する機能

Dr\_Flow™の管理対象となる業務のワークフローは、(a)Webからの帳票の入力、更新、(b)メールによる通知、(c)Javaプログラムの呼び出し、(d)CORBAのオペレーション呼び出しといった部品を組み合わせることで定義される。メールによる通知機能とは、Webから帳票に必要なデータを入力したことを上位者に通知するなど用いられる。Javaプログラムの呼び出しでは利用者が定義したJavaプログラムを登録することで、例えばワークフローの一部として、JDBCを用いたリレーショナルデータベースへのデータ書き込みや参照を実行させることができる。CORBAオペレーション呼び出しでは既存システムのCORBAインタフェースに対して、ワークフロー管理システムに保持されているXMLデータの一部を入力して実行させる、あるいはCORBAインタフェースからの返り値をワークフロー管理システムにXMLデータとして保存することができる。

#### (2) 一般利用者向け機能

##### (2-1) Webからの帳票の入力、更新機能

上記の(1-1)で定義されたデータ形式に応じて自動生成されたHTMLのフォームを用いて、一般の利用者は帳票データの入力、更新が行える。

##### (2-2) Webからの帳票の検索機能

上記の(1-1)で定義されたデータ形式に応じて自動生成されたHTMLのフォームを用いて、一般の利用者は帳票データの検索が行える。

##### (2-3) スケジュール管理機能

一般利用者は、入力した帳票に対して特定の期日までに処理できない場合に、処理を依頼した他の利用者にメールでそれを通知してトレースすることが可能である。一般利用者は個別の帳票に対して、どの処理をいつまでに行って欲しいかをそれぞれに指定できる。

##### (2-4) 通知メールからのチケットの参照機能

上記の(1-2)でのメール通知機能により一般利用者に通知されるメールには、そのメールに関する帳票データがXML形式で添付される。また、その帳票にアクセスするためのURLが記述されており、それをクリックすることで帳票の入力、更新の画面を呼び出すことができる。

### 3. Dr\_Flow™のワークフローモデル

筆者らは自立分散ネットワーク管理メカニズムの開発において、複数のネットワーク管理システムを協調させて一つの処理（協調処理）として実行させるため、状態遷移に基づいた協調処理の記述言語 NetScript®を開発した[4]。Dr\_Flow™では協調タスクをワークフローに対応させ、NetScript®のモデルに基づいてワークフローを記述した。

すなわちワークフロー定義を、ワークフローの状態遷移と、状態を遷移させるイベント、そして状態遷移の際に実行されるタスクから構成する。Dr\_Flow™では、帳票データとワークフロー管理データをチケットと呼ぶ単位で管理する。チケットは個別の業務ごとに生成され、本システムのXMLデータベース中に保持される。チケットはワークフロー定義を参照し、対応するワークフローの状態をワークフロー管理データとして保持している。またDr\_Flow™では状態を遷移させるためのイベントは、チケットに保持された帳票データの特定の値を更新することで発生させる。例えばユーザの苦情管理のワークフローで帳票に状況という項目を設け、担当者が未処理、対応中、対応済みと更新していくことでイベントが発生してワークフローの状態遷移が起こり、担当者の上位者にメール通知が行われる、あるいはJavaのプログラムが実行されるなどのタスクが実行される。（図3. 1）

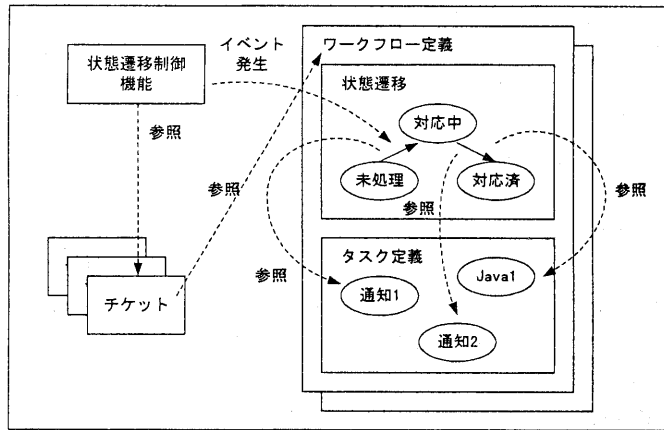


図3. 1 Dr\_Flow™のワークフローモデル

#### 4. Dr\_Flow™の構成

Dr\_Flow™の構成の概要図を図4. 1に示す。Dr\_Flow™は、利用者から入力されるチケットを蓄積し、また他のデータベースや既存システムから取得したデータをXMLデータとして統合するデータ統合サーバと、チケットに記録された項目を参照して、その値に応じてチケットの状態遷移を行うことでワークフローとして定義されたタスクを実行したり、また指定されたスケジュールに応じて定められたタスクを実行するワークフロー管理サーバからなる。

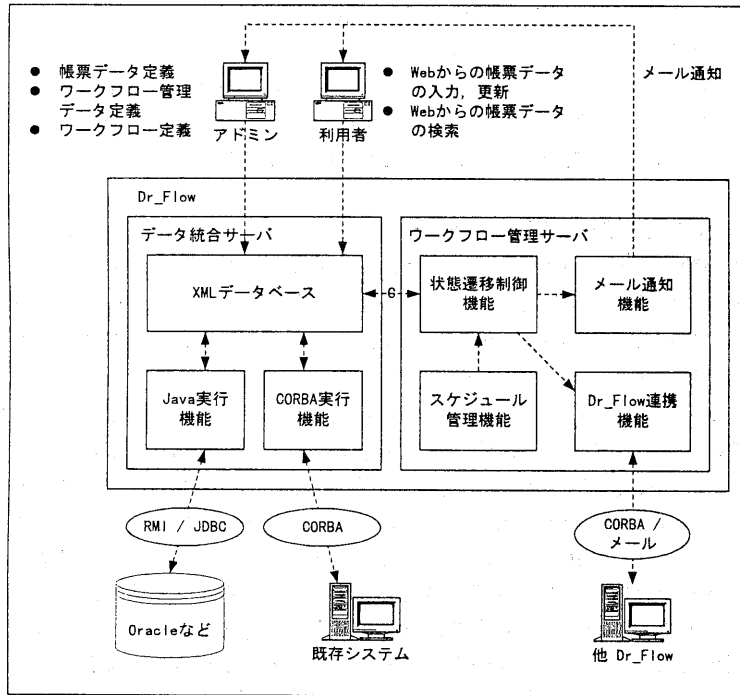


図4. 1 Dr\_Flow™の構成概要

データ統合サーバは、RMI あるいは JDBC を用いて Java のプログラムにより Oracle などのデータベースとの間でデータを取得、設定するための Java 実行機能、また CORBA を用いて既存システムとの間でデータを取得、設定するための CORBA 実行機能、そして取得したデータを保持するための XML データベースから構成される。XML データベースには図 3. 1 で示した利用者から入力、更新されるチケットやワークフロー定義のデータも保持されている。

ワークフロー管理サーバの状態遷移制御機能は、XML データベース中のチケットを参照しワークフロー定義に従って状態遷移を行いタスクを実行させる。スケジュール管理機能は、指定されたスケジュールに応じて定められたタスクを実行するよう状態遷移制御機能に要求する。メール通知機能はアドミニストレータや一般利用者にチケットの情態遷移を通知する。また Dr\_Flow 連携機能は、他の Dr\_Flow にチケットを生成したり、また他の Dr\_Flow チケットの状態遷移と XML データベース中のチケットの状態遷移を連動させることができる機能である。

## 5. CORBA による既存システムとの連携のメカニズム

### 5. 1 CORBA オペレーション呼び出しのタスク登録機能

Dr\_Flow™ で既存システムを CORBA で連携させるためには、まず既存システム間とのデータの入出力を、ワークフローで実行されるタスクとして登録する必要がある。図 5. 1 に示すように、まずワークフローを定義する GUI から CORBA ネーミングサービス(NS)に登録されている CORBA サーバを問い合わせる。次に、CORBA インタフェースリポジトリ(IFR)にその CORBA サーバのオペレーションの一覧とそれらのインタフェース情報を取得する。GUI で使用するオペレーションを選択して、オペレーションの各引数の入出力を設定すると、それに対応する XML マッピングを生成する。

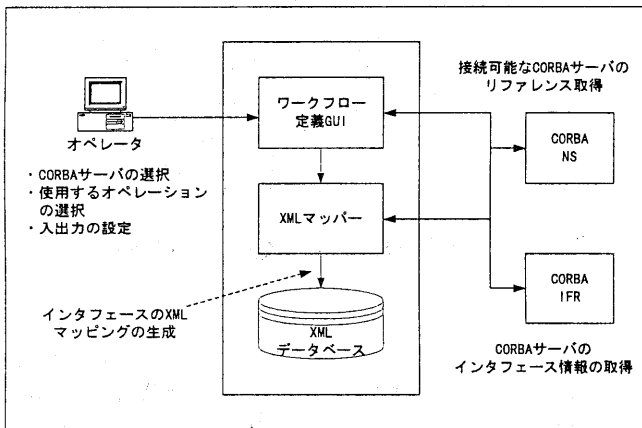


図 5. 1 CORBA オペレーション呼び出しのタスク登録機能

Dr\_Flow™ の特徴は CORBA オペレーションの入出力を XQL で指定できることにある。XQL は XML データ中の特定のタグを検索するための枠組みであり、例えば

```
<DATA><ARG @name="a"><VALUE>100</VALUE></ARG></DATA>
```

のような XML データから VALUE タグを検索する場合、

```
/DATA/ARG[@name="a"]/VALUE
```

というように、XML データのツリー構造をファイルシステムのディレクトリに見立てて、特定のノー

ドを指定することができる。また ARG タグの name のように属性値を検索条件に加えることができる。

## 5. 2 CORBA 実行機能

図5. 2に示すように、ワークフロー定義に記述された状態遷移で CORBA オペレーションの呼び出しタスクを実行するとき、ワークフロー管理サーバはデータ統合サーバにそれを要求する(1)。そしてデータ統合サーバの CORBA 実行機能がそれを実際に行う。CORBA 実行機能は XML データベース中のタスク定義の情報を参照する(2)。タスク定義の情報には CORBA サーバ名が記録されているので、CORBA サーバに接続を実行する(3)。Dr\_Flow™は ORB に Iona 社の OrbixWeb3.1c を使用している。このためサーバに接続する際には、サーバインタフェースのヘルパークラスが必要である。これを(3)の実行時に Dr\_Flow™の JavaVM にロードする(4)。次にインタフェースの情報を読み込み入力が必要なパラメータを組み立てる(5)。値は XML データから取得する(6)。ユーザ定義の struct 型が存在すれば、そのヘルパークラスの JavaVM のロードを行う(7)。また Java のリフレクションの機構を用いて、CORBA サーバのインタフェース情報からオペレーションの Method オブジェクトを生成し、それに(6)(7)で生成したパラメータを引き渡しオペレーションを実行する(8)。実行結果は XML データとして保存される(9)。

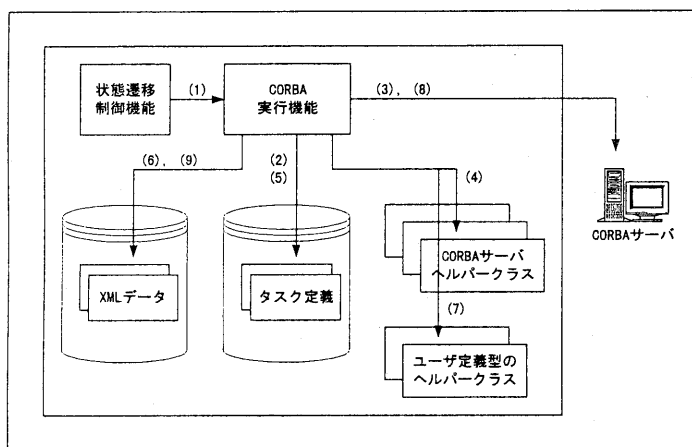


図5. 2 CORBA 実行機能の処理の概要

## 6. まとめ

XML と CORBA を利用したワークフロー管理システム Dr\_Flow™について述べた。5. で述べたように、CORBA インタフェースを XML にマッピングしてそれを実行時に解釈するため、CORBA サーバのインタフェースの変更や、ワークフロー定義の変更に対応できる。

## 参考文献

- [1]速水治夫：“ここまで来たワークフロー管理システム(1)ワークフロー入門”，情報処理学会誌 Vol.39 No.11, 通巻 405 号, p1160-1165,1998.
- [2]CORBA Domain Technology of OMG: “<http://www.omg.org/library/specindx.html#domain>”.
- [3]XML 1.0 of W3C Recommendation: “<http://www.w3c.org/TR/1998/REC-xml-19980210>”.
- [4]桑原,岡田,夏目,吉江,梅村：“ネットスクリプト(ネットワーク自律分散管理のための協調コンポーネントの開発)”，電子情報通信学会技術研究報告 Vol.97, No.492, IN97-138, OFS97-48, p1-8, 1998.