

企業モデルに基づく分散協調作業支援システムの試作

大久保 隆夫, 松塚 貴英, 金谷 延幸, 田中 ユカ, 原 裕貴, 上原 三八
(株) 富士通研究所

モデルベースによる分散協同作業支援システムCEE (Cooperative Enterprise Environment) について述べる。本論文ではオブジェクト指向に基づいた、組織・作業・文書を統合的に扱う企業モデルを提案し、そのモデルに基づいて業務を支援するためのアーキテクチャについて説明する。企業モデルを定義することによって、業務情報を必要とする複数のサーバから構成される分散システムにおいて一貫した作業支援が可能になる。次に企業モデルを利用した作業支援の例として、作業と文書の関係を利用して、先行的な作業を可能にする新しい作業管理方式について提案する。作業と文書の関係に加え、作業と制約やリソースの関係に関する情報をモデルとして定義することによって、これらの情報を利用した多岐に渡るサービスが提供できる。

A System for Cooperative Environment Based on an Enterprise Model

Takao OKUBO, Takahide, MATSUTSUKA, Nobuyuki KANAYA, YUKA TANAKA,
HIROTAKE HARA, SANYA UEHARA
Fujitsu Laboratories Ltd.

This paper proposes a system called CEE(Cooperative Enterprise Environment) for distributed and collaborative work environment. First, we propose an enterprise model which is represented by organization, task, document etc. Next we explain CEE architecture which supports business activity using the enterprise model. We present a new task management method using the CEE enterprise model, especially relationships between task and document. In our system users can define not only task-document relationships but also task-constraint and task-resource relationship using the enterprise model, then they can be served many kinds of services.

1. はじめに

近年、イントラネットを利用して文書共有やワークフローを実現するグループウェアシステムが注目を浴びている。特に、ここ数年のWWWおよびJava言語などの急速な普及によって、WWWブラウザを利用したインストール不要なクライアントが用いられるようになった。しかし、既存のこれらのツールは、複数のサーバによる分散環境などで作業支援を行うために十分なモデル化された情報を持っていない。そのためシステム導入時には、システムをその企業での運用形態に適合させるためにカスタマイズが必要になり、

大きな工数が割かれる。また、運用形態が変更されたり、新たに別のツールと連携させる場合にも、同様の問題が生じることになる。

本論文では、これらの問題を解決するために次のような手法を提案する。

1) 企業モデルの構築

James Martin の提唱する企業モデル¹⁾を拡張し、組織・作業・文書を中心とした企業モデルを提案する。モデルはオブジェクト指向に基づいており、更に利用者がモデルを容易に定義できる手段を用意する。また、業

務上必要なルールもモデル化し、ルールに基づいてサービスの起動が行えるようにする。

2)標準的な基盤の構築

モデルを業務として利用し、かつ他ツールとの連携を実現するための基盤となるアーキテクチャを構築する。基盤技術には標準もしくは標準に近い技術を採用することによって、拡張性、統合性を向上させる。

3)イントラネットを利用した情報共有

Java[®]などを利用してクライアントを記述し、WWWブラウザによってクライアントのアプリレットを取得し起動できるインストール不要のクライアント環境を実現する。

筆者らは以上のような手法に基づいて、Windows上で動作するCEE (Cooperative Enterprise Environment) システムの試作を行った。本論文では上記手法について詳述するとともにCEEシステムの構成、機能について述べる。

2. CEEのアーキテクチャ

CEEは企業モデルを管理するCEEサーバ、企業モデルを格納するリポジトリ、WWWサーバ、およびCEEクライアントで構成される。CEEのアーキテクチャを図1に示す。サーバとクライアントはLANまたはWANによるイントラネット環境を介して通信を行う。

CEEサーバはリポジトリに対して企業モデルオブジェクトの参照・更新を行い、各種のサービスを提供する。提供されるサービスには業務ルールに基づいて行われるサービスや、ユーザや他サーバからの要求に対して情報を提供するサービスなどがある。ユーザへの情報提供は、サーバが自動生成したHTMLの形式でクライアントに提供される。また、CEEサーバは必要に応じて他のアプリケーション(ワープロ、ワークフローなど)の自動起動を行う。

CEEクライアントはJavaアプリレットを利用する。利用者はWWWブラウザでCEEクライアントの置かれたURLにアクセスすることによって、クライアントをダウンロードし、起動することができる。起動されたCEEクライアントはCEEサーバと接続して処理を行う。

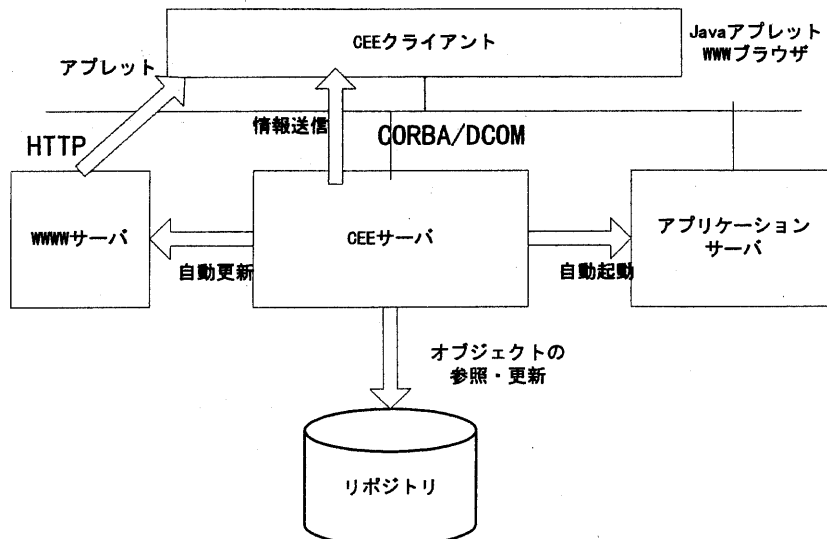


図1 CEEのアーキテクチャ

また、CEEでは他のアプリケーションとの通信や連携のために、標準的なプロトコルを採用している。WWW ブラウザ-CEE 間では HTTP, CEE のサーバとクライアント間では DCOM または CORBA³⁾ を利用する。ワークフローに対しては WAPI, ディレクトリサービスに対しては LDAP のインタフェースを用意する。

3. CEEの企業モデル

3.1 モデルの構成

企業モデルの構成要素となるものをビジネスオブジェクトと呼ぶ。James Martin の提唱する

モデルにおける主要なビジネスオブジェクトは、組織単位、所在地、業務機能、リソースとビジネスルールである。CEEではそのモデルに業務処理、作業（プロセスのインスタンス、ワークフロー、シングルタスク）を追加し拡張を行った。これによってCEEでは作業に基づく文書管理や、組織単位での作業管理など、企業活動全般に渡るモデル化、作業支援が可能になった。CEEの企業モデルを OMT⁴⁾形式で記述したものを図2に示す。

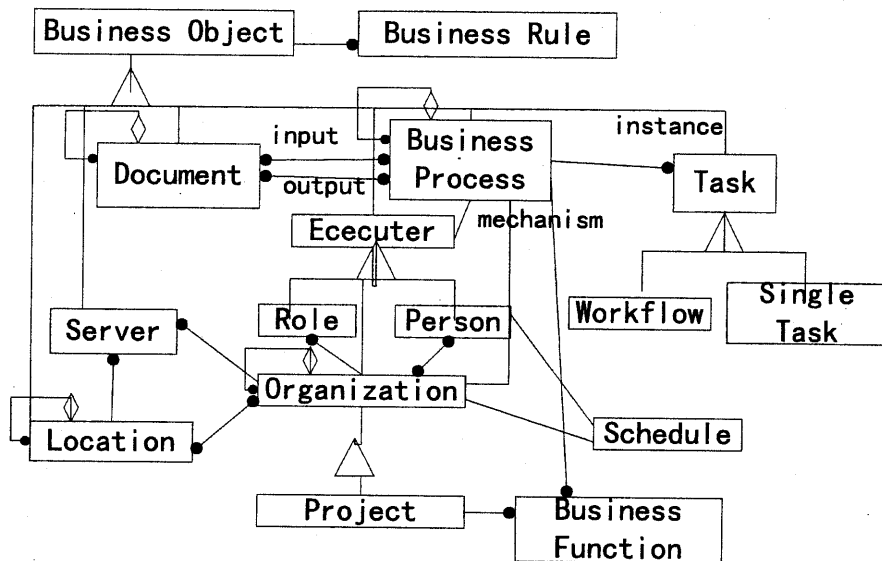


図2 CEEの企業モデル

1) Business Object (ビジネスオブジェクト)

すべてのビジネスオブジェクトの基本となるオブジェクト。ビジネスオブジェクトに共通の属性、操作が定義される。

2) Organization (組織)

企業における個々の組織を表す。特定の業務目標、期間、組織、人、作業群に対応する概念であり、これらの情報を持つ。階層的に定義可能。

3) Project (プロジェクト)

組織のうち、組織にまたがって一時的、暫定的に定義される組織のことを指す。

4) Business Function (業務機能)

特定の業務目的を持つ業務活動の集合。

5) Business Process (業務プロセス)

個々の定義された業務活動を表す。IDEF 0 に準拠し、入力および出力として文書を、メカニズムとして実行者、コントロールとして業務

ルールが関連づけられる。また、業務プロセス自身も階層的に定義可能である。

6)Task(タスク)

業務プロセスが企業内で定義されている業務であるのに対し、インスタンスとしてある個々の業務を指す。

7)Workflow (ワークフロー)

ワークフローの定義。

8)SingleTask (シングルタスク)

プロセスが手順を持つ作業であるのに対し、単独で実行される作業(例えば会議など)を指す。

9)Business Rule (業務ルール)

企業活動を制御するための条件や制約、ポリシーを表現するルール。

10)Document (文書)

企業において扱われる文書。階層構造を定義できる。文書は業務プロセスの入力または出力に指定される。

11)Executer (実行者)

業務プロセスの「メカニズム」に割り当てられる人、組織または役割に相当する作業実行者の抽象クラス。

12)Role (役割)

組織または業務において規定された役割。実際にその役割を担当する個人の集団を構成員として持つ。

13)Person (個人)

組織に属する個人を表す。氏名、電話番号などの情報を持つ。

14)Location (所在地)

住所、建物名、フロア等、所在地に関する情報を表す。

15)Server (サーバ)

CEEサーバに関する情報を表す。

16)Schedule (スケジュール)

組織および人の稼働日、および予定情報を表す。

3.2 モデルの定義と利用

企業モデルを定義することには、企業において現状の組織や業務形態を管理可能な形式として表現するとともに、業務の再利用性を向上させるという利点がある。CEEはこのため「メタモデル(クラス)」としての業務の定義手段と、個々の業務をインスタンスとして生成する手段との双方を用意する。

例として、「モデル設計」という業務を定義する場合を図3に示す。この業務は、「業務情報」という文書を入力し、「モデル設計書」を出力する。業務を担当するのは「設計者」という役割である。ここで定義されたのは特定の業務向けではなく、一般的に利用されるクラスである。例えば「モデル設計書」は、文書の実体ではなくフォーマット等を規定したテンプレートである。次に、「CEE開発」という実際のプロジェクトにおいて、「CEEモデル設計」という業務を実行する場合は、先に定義された「モデル設計」のインスタンスとして「CEEモデル設計」を生成する。生成された業務インスタンスには、具体的に担当となる個人や文書実体に対する関連が設定される。

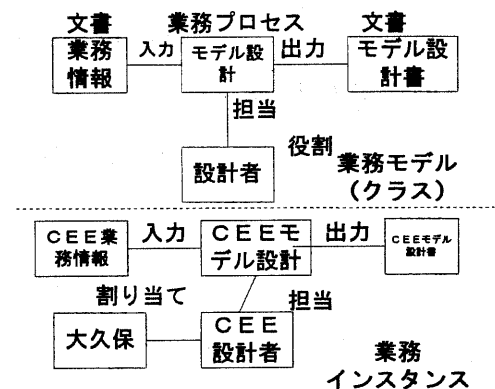
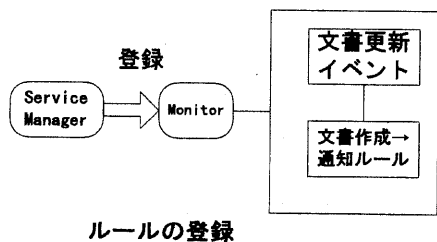


図3 業務の定義と利用

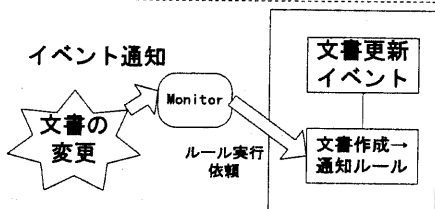
3.3 業務ルールの利用

CEEでは業務ルールもビジネスオブジェクトとして定義可能にするだけでなく、企業モデルと関連付けてルールを実行するための機構を用意する。ルールの処理部分はスクリプト言語に

よって記述している。ルールは業務の実行（情報の変更）や時刻などのイベントの生起によって駆動される。例として、公開用の文書を作成したら関係者全員に通知するというルールの実行の流れを図4に示す。Monitor と Service Manager がイベントの処理とルールおよびサービスの実行制御を行う。Service Manager がまず文書の作成完了イベントおよび対応する業務ルールを Monitor に登録する。イベントが生起されると、それを Monitor が処理し、該当するルールオブジェクトにメッセージを配送し、サービスの処理（ここでは通知）を依頼する。定期的なサービスについてのルールも同様の方法で実現可能である。



ルールの登録



ルールの実行

図4 ルールの登録と実行

4. 作業と文書の関係を用いた作業管理

従来のワークフローなどによる作業管理方法は、作業同士の関係によって管理するものであったが、成果物との関連付けがないか、密接でないために効率的な作業支援が行えなかった。CEEでは、作業と文書の間を IDEF0 を含む企業モデルによって定義することにより、従来よりも先行的な作業管理を可能にした。

IDEF は米軍が中心となって開発した業務活動のモデル化手法であり、欧米では広く用いられて

いる。IDEF0 は、業務のプロセスについて定義するための記法であるが、プロセスだけではなく業務の入力・出力・コントロール・メカニズムの4つの関係で記述することができる。業務活動についてはボックス、残りの4つについてはボックスの4辺に接する矢印によって表現される（図5参照）。業務と文書の間はボックス同士の入力と出力を結合することにより表現される。また、1つのボックス（アクティビティ）内に、IDEFの図を記述することにより階層的な業務定義が可能である。

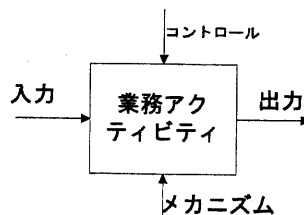


図5 IDEFの記法

従来のプロセス管理では、ワークフローなどのように作業の流れを直接作業間の関係で表現していた（図6参照）。この形式では、作業Cの作業者は作業Aおよび作業Bの完了によって、自己の作業を開始することができる。しかし、現実には作業の流れは作業の成果物と密接に関連している。先の作業Cの開始に必要な条件は、作業Aそのものの完了ではなく、作業Aの成果物である文書1の完成であることもある。また、文書が未完成の状態であっても、次工程の作業者が先行的に作業開始するには十分な条件を満たしている場合もある。このような作業に関連づけられた成果物の情報が提供されれば、従来よりもプロセス全体の効率は向上する。

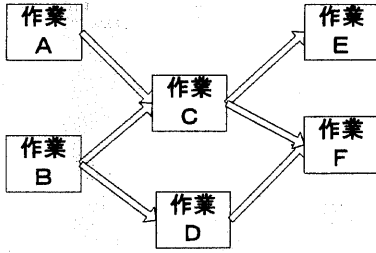


図6 作業の流れ

筆者らはCEEの企業モデルの定義手段として、IDEF0エディタを独自に作成した。エディタは利用者が入力したIDEF図をCEEシステムが入力可能な形式に変換を行う。またこのエディタは逆に、CEEの業務モデルをIDEF形式で表示するモデルブラウザとしても利用できる。

図5の作業の流れ図に成果物の関係を挿入し、IDEF形式で記述したものを図7に示す。先の例の作業Cが開始できる時点は作業A、作業Bの完了ではなく、実際には作業の成果物である成果物1、成果物2の完了であることが分かる。

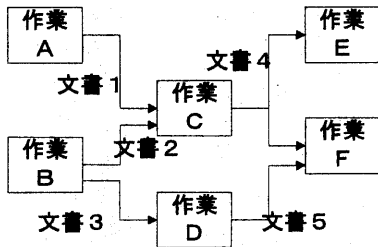


図7 IDEF0による業務表現

従って従来は作業Bが完全に完了するまで作業Cの担当者は待たなければならなかったが、図7によれば作業Bが（完了していなくても）成果物2を完成させた時点で作業を開始することができるようになる。

そこで、各文書に対し、次のような属性を定義する。

- 1) 作成状況
未着手/未完成/完成/完成後修正中
- 2) 公開属性

公開不可/公開可

2) の公開属性は、「未完成なので公開できない」「未完成だが見せても差し支えない」のように場合分けするために用いる。

文書を作成する作業者は、作成状況に応じて上記の文書情報を設定する。一方、文書を入力する作業の担当者は文書情報を参照し（可能ならば実体も参照して）、作業可能か判断する。CEEでは更に通知機能を用意し、各文書の属性をシステムが監視して、変更があればそれを入力とする作業の担当者に通知を行うようにする（図8参照）。更に詳細に効率化を図る方法としては次の2つの方法が考えられる。

- 1) 文書の利用者単位毎に文書属性を定義する

図7において、文書4の利用者（入力とする作業）には作業Eと作業Fがあるが、このように同一文書を複数の作業が入力する場合は、入力とする作業毎に文書の作成状況属性・公開属性を設定させるようにする。図7の例では作業Eには公開できるが作業Fにはまだ見せられない場合、作業Cの担当者は、文書4の作業Eに対する公開属性は「公開可」、作業Fに対する公開属性は「公開不可」とそれぞれ設定する。

- 2) 文書の構成単位毎に文書属性を定義する

方法1) では前の作業者が後に利用する作業者を考慮しなければならないという問題があった。そこで、作業単位ではなく文書の構成単位毎に属性を設定させる。図7の例で、文書4は2章で構成されているとする。作業Cの担当者は1章の属性および2章の属性を個別に設定する。たまたま文書4の1章が「公開可」となっており、作業Eは文書4の1章のみを利用する場合（作業Cの担当者は誰がどの章を利用するかは知らないと仮定する）、その時点で作業Eの担当者は作業を開始することができる。

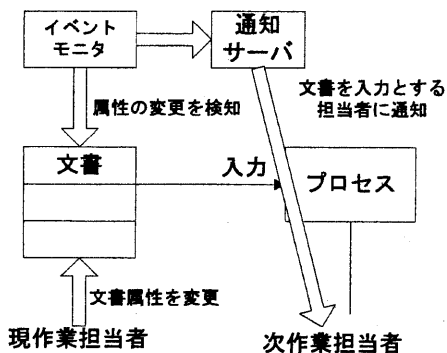


図8 システムによる変更の自動通知

5. システムの実装

プロトタイプとして実装したCEEのシステム構成を図9に示す。試作システムはサーバ・クライアント構成であり、サーバはWindows NT、クライアントはWindows NT/95上で動作する。サーバ部分はVisual C++、クライアント部分はJavaおよびJavascriptを用いてプログラミングを行った。また、リポジトリ部分(作業・文書情報および文書の实体管理)には、PentaBASEというOracleベースの文書管理システムを使用している。

試作システムが提供する主な機能を次に挙げる。

1) 作業情報の表示

個人に割り当てられた作業情報、およびプロジェクトに関する情報の参照・共有

2) 組織情報の表示

組織情報(階層構造および組織自体の情報)、および個人に関する情報の参照・共有。

個人の情報は、所属組織、担当する作業、プロジェクトのいずれからも辿ることができる。

3) プロジェクト・作業への文書登録・共有

プロジェクトや作業に関連する文書を登録し、表示・編集を行うことができる。

CEE試作システムのクライアント画面を図10に示す。利用者はWWWブラウザでCEEのURLにアクセスすることによってJavaのアプレットをダウンロードし、クライアントを自動的に起動することができる。

文書ファイルのダウンロードには、HTTPとFTPの双方による転送を用意している。HTTPで転送した場合、Netscapeのhelper機能を用いて、文書を起動するアプリケーションを選択することができる。またFTPによるダウンロードを行った場合はCEEがクライアントのレジストリ情報から必要なアプリケーションを検索して自動起動する。このように文書に対応するアプリケーションがインストールされていればその文書を見ることができるので、自由な形式の文書を扱うことができる。

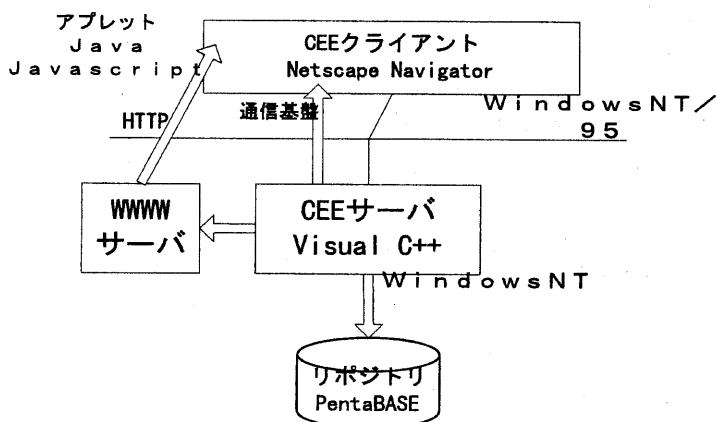


図9 CEEシステム構成

一方文書のアップロードはFTPを利用して
いる。FTPを利用したダウンロード・アップ
ロードおよびアプリケーションの自動起動は、
Netscape の plugin によって実装した。

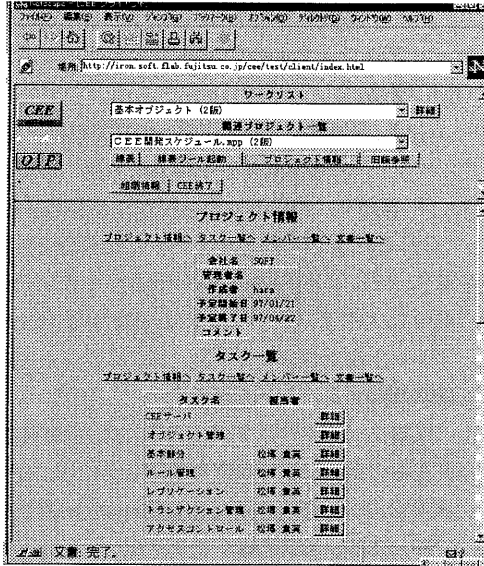


図10 CEEのクライアント画面

参考文献

- 1) Martin, J.: Information Engineering Book 1 Introduction, Prentice-Hall, 1989.
- 2) Gosling, J. and Steele, G.: The Java Language Specification, Addison-Wesley, 1996.
- 3) Mowbray, T.J. and Zahavi, R.: The Essential CORBA, Wiley & Sons, 1995.
- 4) Rumbaugh, J.: Object-Oriented Modeling and Design, Prentice Hall, 1991.
- 5) Hara, H., Matsutsuka, T., Kanaya, N., Ookubo, T. and Uehara, S.: Model-Based Architecture For Intranet, to appear in proceedings of COMPSAC 97, 1997.
- 6) 上原三八: 大規模作業支援のためのモデル化と実現アーキテクチャ, 電子情報通信学会論文誌, Vol.J80D, No.7, 1997.

6. おわりに

プロトタイプとして作成したCEEシステムは、実際に設計・開発部門など社内の数個所で試行され、情報共有や進捗管理に利用されている。

4章で挙げたプロセス管理の方式として一般的なものにはワークフローがある。CEEとワークフローの統合により、業務ルールに従ったワークフローの自動起動や、文書や組織情報と関連付けて作業者がワークフローを定義することが可能になる。

プロトタイプ版は1サーバだが、遠隔地を含む環境ではサーバおよび情報の分散が必要になる。このうち情報の分散については、LDAPなどの標準的なAPIによるディレクトリサービスが注目されている。組織などの固定的な情報については、ディレクトリサービスによる実現を検討中である。