

## 知識コミュニティ活動のためのコラボレーション支援アーキテクチャ

坂上 秀和<sup>†</sup> 坂口 基彦<sup>†</sup> 延藤 里奈<sup>†</sup> 石川 雄一<sup>‡</sup> 朝倉 敬喜<sup>§</sup>  
<sup>†</sup>NEC インターネットシステム研究所 <sup>‡</sup>NEC システムプラットフォーム研究所  
<sup>§</sup>NEC システム基盤ソフトウェア開発本部

従来のコラボレーション・システムの多くは、部門や企業などの固定的な組織を前提に設計されている。しかしこれらは、昨今の業務の変化が激しい事業環境においては、有効に機能しないことが問題となり、近年は、組織内で公式・非公式に形成されるコミュニティ活動に焦点をあてる試みが注目されている。本稿では、知的労働者のワークスタイルとコミュニティ活動の特徴について考察し、事業環境の変化に伴うそれらの変化について明らかにする。その上で、動的コミュニティ管理の概念をシステム・アーキテクチャに導入することにより、情報共有、セキュリティ確保などのコラボレーション支援環境実現に必要な機能モジュール群の連携・統合が可能なモデルを提案し、本モデルに基づく試作システムについて報告する。

## Collaboration System Architecture for Knowledge Communities

Hidekazu Sakagami<sup>†</sup> Motohiko Sakaguchi<sup>†</sup> Rina Nobuto<sup>†</sup>  
Yuichi Ishikawa<sup>‡</sup> Takayoshi Asakura<sup>§</sup>

<sup>†</sup>NEC Internet Systems Research Labs. <sup>‡</sup>NEC System Platform Research Labs.

<sup>§</sup>NEC System Platform Software Development Division.

Many of conventional collaboration systems are designed for fixed organizations such as sections and companies. However, such systems are not effective in recent business environment because of the incessant changes; there is greater attention on the approaches of focusing on the communities that are created formally or informally. This paper argues the workstyle and community activities of knowledge-workers and their shifts caused by the changes of business environments. We propose a collaboration system architecture for knowledge communities based on a concept of “dynamic community management”. This architecture enables to integrate a group of function modules which is necessary for realizing collaboration environments such as information sharing and security ensuring. We also discuss on the prototype system based on this architecture.

### 1. はじめに

従来のコラボレーションやナレッジマネジメントシステムは、部門や企業などの固定的な組織における業務の効率化を目的とするものが多い。これらの分業型コラボレーションは、業務内容と責任範囲が明確に規定され、変更が少なかったため、システム主導で組織内の成果物や手段を管理する視点で設計されていた。システム主導の考え方では、グループウェアを導入することで、掲示板や情報共有データベースを通じて組織全体で自由に情報や意見を交換することが期待されたが、現実には特定の人しか書き込みを行わず、ある期間を過ぎると誰もアクセスしないデータベースが散在し、情報自体も価値のあるものが少ない状態となってしまう、という失敗があった。情報を共有する活動母体が不明確であるため、情報更新・参照への動機が保てないことが大きな要因である。

これに対して、近年は、組織で活動する「人」に焦点をあて、組織内で公式・非公式に形成されるコミュニティを発見し、その活動を支援するアプローチが注目されている。システム主導ではなく、先に活動母体を認識し、その活動をシステム支援することで新たな知識を効率的に産み出し組織全体で活用する、という考え方である。中でも、Wenger[1]らが提唱した Communities of Practice (CoP: 実践コミュニティ) は、現在最も注目されている概念の一つである。彼らは、企業組織の文化人類学的な観察を通し、共通の専門スキルや、ある事業へのコミットメント(熱意や献身)によって結び付けられた人々の集まりがあり、「知識を理解する考え方の枠組み」を共有していることを明らかにした。CoP は業務に基づかないインフォーマルな自発的活動体である場合も多いが、その存在が縦割り組織間の分業における横断的な課題の解決に重要な役割を果たす事例を示している。また、荒井ら[2]は、ナレッジアセスメント

調査を通じ、企業において、将来展望を高く持ち、知識創造に貢献する知識労働者達が、異質な知の融合を求めて組織横断的なコミュニティを形成し、互恵主義的に活動していることを指摘している。

本稿は、このような知識コミュニティの重要性が高まる背景となった、企業経営スタイルと知識労働者のワークスタイルの変化について考察し、そこで求められるコラボレーション支援環境の要件を示し、それを実現するためのシステム・アーキテクチャ提案と試作システムについて報告する。

## 2. コラボレーション支援環境の要件

### 2.1. コラボレーション・スタイルの変化

1980年代までの高度成長期とその延長の時代における企業経営は、自社の資源をもとに事業規模の拡大を追及することで企業価値を高める「総合主義」戦略が中心であった。ここでは自社内で業務を効率的に完結させるために、事業部制や機能別部門などのトップダウンな組織化による意思決定・伝達と、業務の標準化や固定化(スキルワーク化)が図られた。従来型のグループウェアは、このような固定的な組織・環境での共同作業を主なターゲットとしていた。

1990年代に入り、バブル経済崩壊後の景気後退期を迎えると、各社は経営の合理化を余儀なくされた結果、自社のコア・コンピタンスに経営資源を集中させ、他は積極的にアウトソーシングを活用する「選択と集中」型の経営戦略に移行し始めた。1980年代に米GE社を復活させたJ.ウェルチ[3]の影響を受けている。近年はさらに発展形として、コア・コンピタンスを持つ企業同士がそれぞれの強みを組み合わせ、より大きな付加価値を生み出す戦略もとられている。ここでは、ビジネス環境の変化への素早い対応が求められ、部門横断、企業間プロジェクトのように、業務上の目標に応じて最適なメンバを集めて組織(知識コミュニティ)を構成し、メンバが自律的に目標実現に向けて行動する形に変化している。

このようなコラボレーション・スタイルの特徴は、以下の点で従来型スタイルと大きく異なる。

#### 1) コミュニティのオープン性

コミュニティは従来の縦割り組織などの帰属組織に縛られず形成され、部門横断的、時には企業の壁を越えた活動になる。

#### 2) メンバ構成の動的変化性

コミュニティのメンバ構成は常に一定ではない。新たに生じた課題を解決するための人材の追加や一時的な参加のほか、活動フェーズの変化によってコミュニティから外れるメンバもいる。

### 2.2. システムの課題と要件

上記のコラボレーション・スタイルの変化により、システム実現上は、以下のような課題が発生する。

#### 1) コミュニティのオープン性に起因する課題

メンバ同士が共通帰属組織を持たない場合、一般に情報の共有が困難になる。従来は共通サーバを用意して情報共有を図ることが多かったが、異なる組織間ではネットワーク・ファイアウォールなどにより共通アクセスできるサーバの設置が難しい。インターネット上に全員がアクセス可能なサーバを設置することは技術的には可能でも、設定や手続きの煩雑さのために、現実にはメーリングリストなどを用意し添付ファイルで情報を交換・共有するが多い。この場合、情報の管理は個人に依存するため、情報の更新によってどれが最新情報であるかが不明になる、などの混乱が生じやすい。

また、活動内容によっては、共有情報をコミュニティ外部に流出させてはならない場合もある。多くの企業では、組織に帰属する従業員の認証管理を行うためのディレクトリやLDAPなどを用意して情報アクセスへの安全性を確保しているが、共通帰属組織がないときは、これも困難である。

#### 2) メンバ構成の動的変化性に起因する課題

後から参加したメンバも含めてコラボレーションを円滑に行うためには、それまでに共有していた情報を素早く共有する必要があるが、添付メールなどで情報共有が行われていた場合は困難である。セキュリティ観点では、コミュニティから外れたメンバに対して今後情報が渡らないだけでなく、過去の共有情報も含めてアクセスを制御したい。Microsoft社のWindows Rights Management Services (RMS)[4]などの従来の情報保護技術を用いることで、個人単位のアクセス権限制御は技術的には可能であるが、メンバ構成の変更や情報更新の都度、権限設定のやり直しやコンテンツの再暗号化などが必要になるため、事実上は運用が不可能である。利用者の負担を最小限に抑えながら適切な制御を行うことが出来ない限り、実際の運用が徹底されなくなる危険が大きい。

これらのコミュニティ活動特有の課題を考慮すると、コラボレーション支援環境要件は以下のようになる。

1. 共通組織に属さないオープンなコミュニティメンバ同士が手軽に情報共有でき、新たな参加メンバも過去の資産を迅速に共有できること。

2. コミュニティから外れたメンバに対して過去の共有情報も含めてアクセスを禁止できるなど、内部の情報が外部に漏洩しない制御ができること。

3. 運用時にユーザの作業負担を最小限に抑え、コラボレーション自体に集中できること。

これらは「柔軟性と安全性の両立」ということができる。両者を共存させつつ、利用者に過度な作業負担をかけないことが重要である。

### 3. システム・アーキテクチャの提案

#### 3.1. アーキテクチャ概要

前記要件を満たすためには、コミュニティメンバによる情報共有、共有された情報の保護、通信時のセキュリティ確保など、様々な機能モジュールを組み合わせて提供する必要があります。また、個々のモジュールは、コミュニティの構成情報に基づいてシステムを制御することになるが、それぞれが独立に管理すると、メンバ構成が変化する度にモジュール毎の設定をやり直さなければならないなど、利用者の作業負担が大きくなってしまふ。従って、ユーザの作業負担を抑えるためには各モジュールが相互に連携することも重要である。そこで筆者らは、オープンなコミュニティにおいても、コミュニティ構成を仮想的に一元管理する「動的コミュニティ管理」の概念をシステム・アーキテクチャに導入し、これに基づいて、情報共有、セキュリティ確保などの、コラボレーション支援環境実現に必要な機能モジュール群を連携・統合させるアーキテクチャを採用することとした。図1に、アーキテクチャ概要を示す。

最上位のAP層は、利用者に提供される仮想的なコラボレーション支援環境であり、文書共有などのツール(アプリケーション)が提供され、最下位層は物

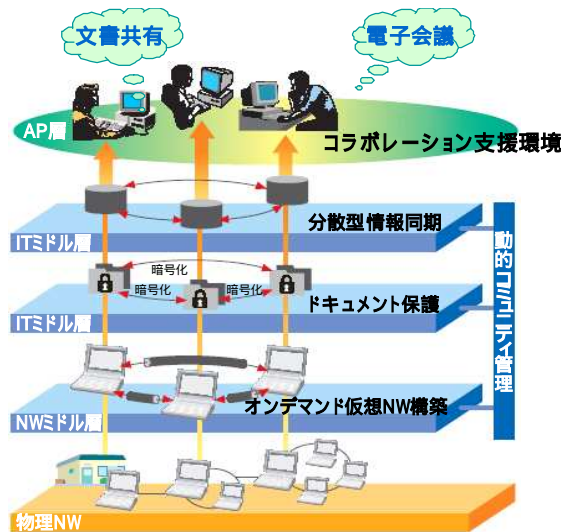


図1 アーキテクチャ概要

理ネットワークを表している。

両者の中間はミドルウェア層であり、情報共有のための分散型情報同期機構、共有情報の保護のためのドキュメント保護機構、通信時のセキュリティ確保のためのオンデマンド仮想ネットワーク構築機構の三層を含む。また、動的コミュニティ管理機構は、コミュニティ情報を一元管理するとともに、各機能モジュール群を連携させるハブとして機能する。AP層の各種アプリケーションは、これらのミドルウェア上に実装される形となる。本アーキテクチャをとることでコミュニティ情報が一元的に管理され、メンバ構成の変化などは、動的コミュニティ管理機構を通じて各機能モジュールに通知され、自動的に設定を変更させる機構を実現することが可能となり、利用者による個別の設定変更は不要となる。以下、機能モジュール毎の詳細について述べる。

#### 3.2. 分散型情報同期機構

共通帰属組織を持たないメンバ間の情報共有を行う一つのアプローチとして、ここでは特定の集中管理サーバを前提としない利用者の端末同士による情報の同期共有を採ることとし、分散型情報同期基盤DISH(Distributed Information SHaring platform)を開発した。図2にDISHの概要を示す。

情報共有を集中管理サーバ無しに実現するために、コミュニティを構成する全てのメンバ端末が、共有情報とコミュニティ参加メンバ情報(共有メンバリスト情報)を保持する構成とし、共有情報に編集を加えた場合は、編集内容を共有メンバ全員に通知する情報同期方式を採用している。共有メンバリスト情報に変更が発生した場合も、共有情報と同様に情報同期を行うことで、サーバレスでのコミュニティ管理を実現している。情報共有を行う際には、端末上で利用者がコミュニティ内の情報共有空間(Community Space: CSP)を作成し、そこに他のメンバを招待する方式としている。具体的には、招待者が自分の端末情報等が書かれたXML形式の招待状

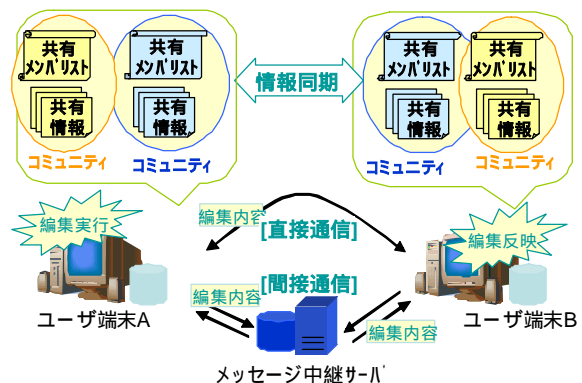


図2 分散型情報同期基盤 DISH 概要

ファイルを作成し、それを被招待者の端末上に読み込むことで招待者/被招待者の端末間で参加プロトコルが作動し、CSP への参加手続きが行われる。参加手続きが完了すると、被招待者の端末内に CSP 内で共有されていた情報が招待者端末から送り込まれ、以後は CSP 内の共有情報の変化が同期されるようになる。端末間の通信は、同一 LAN 内などの直接到達可能な範囲内では直接通信を行い、直接到達できない場合や通信相手端末が起動されていない場合などは、メッセージ中継サーバを介した間接通信(本試作では SMTP/POP を利用)を行う方式とした。

### 3.3. ドキュメント保護機構

RMS などの従来型ドキュメント保護技術の多くは、文書作成時に「誰が、どの操作を許可されているか」という権利情報を文書に埋め込んでおき、文書がアクセスされた際に利用者を認証して操作を認める機構を持つ。しかしこの方式では、コミュニティ構成の変化のたびに、権利情報を変更して文書に埋め込み直す必要がある。筆者らは、既存のドキュメント保護技術の上位にコミュニティ単位での権利管理を行う統合権利管理機構を設けることで、コミュニティ構成の変化時に権利情報を埋め込み直す必要がない方式を考案した。図 3 に本方式の概要を示す。文書作成時にコミュニティ ID を含む権利情報を埋め込み、統合権利管理サーバ内に権利情報を登録する。作成した文書は、DISH によってコミュニティ内で共有する。共有された保護対象文書をアクセスする際には、アプリケーションが統合権利サーバにライセンスを要求し、統合権利サーバが動的コミュニティ管理機構に対して要求者が当該文書を共有するコミュニティメンバであることを確認した後、RMS などの既存の DRM サーバに対してライセンスを要求し、発行されたライセンスを利用者のア

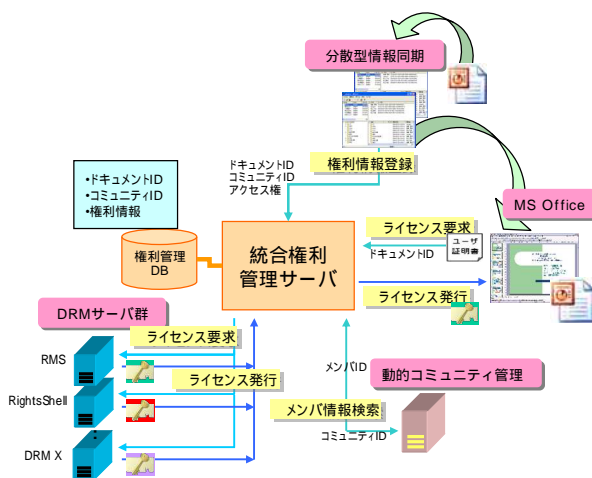


図 3 ドキュメント保護機構概要

プリケーションに渡すという流れを取る。本機構により、文書作成時にコミュニティ単位のアクセス制限情報を登録するだけで、文書アクセス発生時点でのコミュニティ構成に基づく権限制御が可能となり、後から参加したメンバのアクセスを許可したり、コミュニティから外れたメンバの操作を拒否するなどの制御が可能となる。

### 3.4. オンデマンド仮想ネットワーク

コミュニティ内の安全な通信を図るために、メンバ端末同士に閉じた仮想ネットワーク：CUG (Closed User Group)が必要な場合がある。筆者らはメンバ構成の変更に合わせて動的に CUG の再構成が可能な、オンデマンド仮想ネットワーク技術[5, 6]を、必要に応じて連携できるアーキテクチャを構成した。

オンデマンド仮想ネットワーク技術は、VPN ゲートウェイを利用せずに網内の任意のノード間で CUG の構築が可能な VPN アーキテクチャであり、CUG を管理する専用のプライベート DNS (PDNS) サーバと、端末プラグインによって、オンデマンドに任意の数の仮想ネットワーク(CUG)を構築する。図 4 にオンデマンド仮想ネットワークの概要を示す。各 CUG は仮想的なアドレス空間と名前空間を持ち、端末は PDNS サーバから払い出された仮想 IP アドレスに基づいて CUG 内の通信を行う。仮想 IP アドレスと実 IP アドレスのマッピングは PDNS サーバが管理しており、ノード間の通信は、P2P のセキュア通信(IPsec 通信)で行われる。1 台の端末が複数の CUG に参加する場合は、CUG ごとに異なる仮想 IP アドレスが割り当てられる。端末プラグインソフトウェアと PDNS サーバの協調動作により、端末は他のノードとの通信の際、DNS を用いた名前解決時に相手と同じ CUG メンバであるかどうかを識別

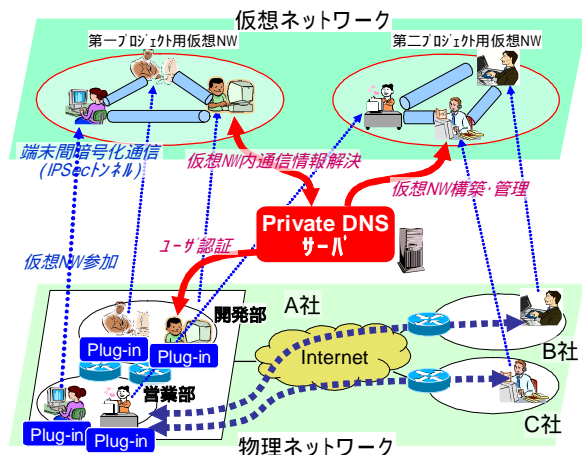


図 4 オンデマンド仮想ネットワーク



し、メンバであれば IPsec トンネルを確立しセキュアな通信を行い、非メンバならば通常の IP 通信を行うという特徴を持つ。各端末において予め設定しなければならないのは、参加する CUG のドメイン名、ID/パスワード、プライベート DNS サーバの IP アドレス設定だけであり、従来方式で必要だった IPsec パラメータ等の面倒な設定が不要となる。

本連携方式では、コミュニティ構成情報の変化が発生すると、後述する動的コミュニティ管理機構を通じて PDNS サーバに CUG の再構成等を指示し、端末上のプラグインソフトウェアが PDNS サーバの状態に同期して自動的に CUG への参加処理を行う機構としており、利用者は CUG の存在さえ意識する必要がない。

### 3.5. 動的コミュニティ管理機構

コラボレーション支援環境を構成する各モジュール間の連携が不十分だと、メンバ構成変化の度に全てのモジュールの設定変更が必要になるなど、管理の手間が増え、運用時の人為的ミスの危険も高い。筆者らはコミュニティ情報の管理を一元化して各モジュールから独立させると共に、モジュール間の連携機構を提供する動的コミュニティ管理機構 (Community Directory :CD) を考案し、Windows 2003 サーバ上に実装した(図 5)。CD は、コミュニティ情報管理、メンバの信頼性保証、システム連携ハブ、の役割を持つ。CD は、コミュニティ情報を格納するコミュニティ・データベース(CDB)、コミュニティ情報を外部操作する API を提供するコミュニティサーバ(CServer)、CD と連携するモジュール側に設置され CServer への接続機能を提供する CD 連携ライブラリ(CDL)、から構成される。CDB には、コミュニティ ID やメンバリストなどのコミュニティ情報、CD に接続するシステムの認証情報や権限情報、CD が信頼する認証サーバ情報等を保持する。動的コミュニティでは構成メンバの信頼性保証がセキュリティ上重要であるが、CD はメンバの信頼性をメンバの所属組織の認証サーバと連携して保証できる機構を持つ。具体的には、CD と信頼関係を結ぶ認証サーバで個人認証されたメンバのみ

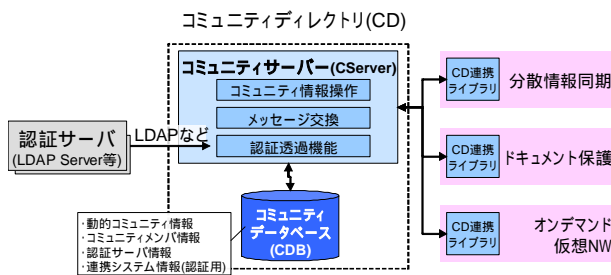


図 5 コミュニティ・ディレクトリ

がコミュニティに参加できるモデルとしている。また、システム連携ハブ機能を実現するために、CD はコミュニティの生成・削除やメンバの追加・削除が発生した際に接続中のモジュールに対し自動的に変更通知する機構と、接続モジュール間でメッセージの交換を行うための機構を提供している。これらは Web サービスとして公開されるが、CDL を利用することで WindowsOS のローカル API と同様に利用できるようにしている。

## 4. 考察

### 4.1. システム試作と運用

本アーキテクチャに基づき、筆者らは分散情報同期/ドキュメント保護/オンデマンド仮想ネットワークを統合したコラボレーション基盤を実装し、その上にファイル共有ツールを実装した(図 6)。画面左ペインに参加中のコミュニティ情報が表示され、共有するファイル一覧が右側ペインに表示される。UI は Drag&Drop 操作による外部アプリケーション (Explorer 等) とのファイルのコピーなどが可能である。本ツールから開いたファイルを修正・保存すると、変更後のファイルがコミュニティの他のメンバ端末上のツールに自動的に同期される。本ツール上で新たにコミュニティの作成やメンバ招待をすることで、オンデマンド仮想ネットワークおよびドキュメント保護機構に対してコミュニティ情報の変更が通知され、自動的に設定変更される。これらの機構により、利用者が任意に作成したオープンなコミュニティにおける情報共有を実現するとともに、運用時にユーザの作業負担をかけずに共有情報の保護および通信上のセキュリティ確保を行うことが可能となっており 2.2 で述べたコラボレーション支援環境の要件を満たすことできている。現在、部門横断の共同開発プロジェクトなどの業務に、本ツールを約 1 年間試用している。参加コミュニティ数やメンバ数、共有ファイル数・容量などは利用者ごとに異なるが、一人 20 コミュニティ、総容量で 500MB 程度

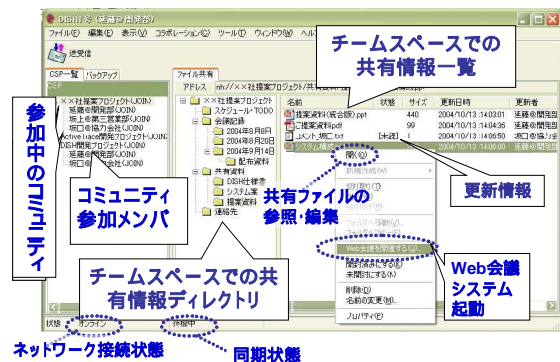


図 6 ファイル共有ツール画面例

の情報共有が運用されている。

## 4.2. アーキテクチャの拡張性

コラボレーション支援には、セキュリティ確保だけでなく、電子会議システムなど利用者の利便性を高める機能群もある。筆者らは本アーキテクチャに基づき、Web 会議システム(コミュニケーションドア[9])との連携機構を試作に盛り込んだ。

通常の Web 会議システムは、事前に会議サーバへの参加者設定、会議資料登録を行い、参加者に開催通知を email で送るなどの準備をし、予定時刻に各参加者が指定の会議室(URL)にアクセスする、という流れになる。本連携試作では、CD で管理されるコミュニティメンバを会議参加者として会議室を設定することで、煩雑な事前会議室設定を行わずに、簡単に Web 会議を開始することが可能となった。会議主催者がファイル共有ツール上で資料ファイルを選択し、メニューから「会議開催」(図 6 参照)を選択すると、連携機構が会議サーバに対して選択した資料とコミュニティ ID を含む会議設定の要求を送る。会議サーバは、CD からコミュニティのメンバリストを取得して会議参加者として設定し、会議室の URL 情報を会議主催者端末に返す。会議主催者端末は、これを、DISH を利用してコミュニティメンバに対して参加依頼の形で送信する。コミュニティの各メンバが参加依頼を受け取り、参加を許可すると、Web 会議画面を自動的に立ち上げ会議が開始される。このとき、会議主催者が選択した会議資料が、Web 会議の共有画面に表示された状態となる。

このように、本アーキテクチャは目的に応じてコラボレーション支援環境を構成する機能モジュール群を、コミュニティ情報を中心に連携・自動設定させることが可能な構成となっている。

## 4.3. 関連研究との関係

分散型の情報共有機構を提供する製品として、GrooveNetworks 社[10]やアリエルネットワーク社[11]などの製品がある。これらは完結したシステムとして組織を越えたコミュニティにおける情報共有機能を提供しているが、他の機能モジュールを連携させる際には、コミュニティ情報の変化を管理・通知するための機構を別途用意しなければならない。

また、P2P 型の汎用的なアプリケーション開発のための基盤技術として、JXTA[12]、SOBA[13]、SIONet[14]などのプラットフォームが提案されている。これらは、サーバレスの P2P 型通信によってコミュニティ内の仮想共有空間を構成し、その中で情報共有(一部、ストリーミング等も含む)などを実現する。これらは、通信層におけるフレームワーク

を規定するが、IT 層におけるコラボレーション機能モジュールをコミュニティ管理に基づいて連携させる概念は含まない。また、試作では情報共有機構を P2P 型の分散情報共有基盤で実現したが、本提案アーキテクチャは、情報共有機構を、共有ファイルサーバ利用などのクライアントサーバ型の構成に置き換えて適用することも可能である。本アーキテクチャは、コミュニティ管理機構を軸とした機能モジュール群連携のための汎用的枠組みを提供するものであり、情報共有の通信方式の違いは本質ではない。

## 5. おわりに

本稿では、企業におけるビジネススタイルと知識労働者のワークスタイルの変化について考察し、そこで営まれるコラボレーションの特徴として、コミュニティのオープン性と、メンバ構成の動的変化性を抽出した。この活動形態を支援するためのシステム要件として、動的コミュニティ管理の概念をシステム・アーキテクチャに導入することで、コラボレーション・システムを構成する機能モジュール群を連携できるアーキテクチャを提案し、試作システムを構築した。今後は同基盤上において、コミュニティ活動をより活性化させるための支援機構や、メンバ相互の刺激によって知識を創発させるためのツールなどの形態について検討を発展させていきたい。

## 参考文献

- [1] E.ウェンガーほか: “コミュニティ・オブ・プラクティス”, 翔泳社 (2002).
- [2] 荒井, 野村: “知識経営に貢献する企業内コミュニティの一考察”, 情報処理学会研究報告, 2001-GN-40, Vol.2001, No.048, pp.25-30(2001).
- [3] J.ウェルチほか: “わが経営(上・下)”, 日本経済新聞社 (2001).
- [4] マイクロソフト株式会社: “Windows Rights Management Services”, <http://www.microsoft.com/japan/windowsserver2003/technologies/rightsmgmt/default.mspx>, .
- [5] 藤田, 石川ほか: “DNS を用いたスケーラブルな VPN アーキテクチャ”, 2004 年電子情報通信学会総合大会 B-6-200, 2004 .
- [6] 藤田, 石川ほか: “大規模な動的グループ通信に適したオーバレイ網制御方式”, 信学会 NS 研究会, 2004 .
- [7] NTT 東日本: “フレッツ・グループアクセス”, <http://flets.com/groupaccess/>.
- [8] Vipalette: “CUG サービス”, <http://viplt.net/>.
- [9] NEC: “コミュニケーションドア”, <http://www.sw.nec.co.jp/middle/commdoor/>.
- [10] GrooveNetworks 社: <http://www.groove.net/>.
- [11] アリエルネットワーク: <http://www.ariel-networks.com/>.
- [12] JXTA, <http://www.jxta.org/>
- [13] SOBA フレームワーク, <http://www.soba-project.org/>
- [14] 星合ほか, 意味情報ネットワークアーキテクチャ, 電子情報通信学会論文誌, B, Vol.J84-B, 2001.