事例ベースを用いたやわらかいビデオ会議システムの構成

武田　敦志†，菅沼　拓夫†，木下　哲男††，白鳥　則郎†
†東北大学電気通信研究所 / 情報科学研究科
††東北大学情報シナジーセンター / 情報科学研究科

本稿では，環境の変化や利用者の要求に適応するやわらかいビデオ会議システムにおいて，その知識処理機構に事例ベース推論を適用した，事例ベース型ビデオ会議システムについて述べる．従来のやわらかいビデオ会議システムはシステム設計時に想定されていない環境や利用者要求に対しての対応能力が低いという問題があった．そこで，本稿ではシステム設計時に想定されていない環境や利用者にも適応できるやわらかいビデオ会議システムを提案し，その設計と実装について述べる．また，プロトタイプシステムによる実験により，本システムの有効性を検証する．

Composition of Flexible Videoconference System using Case-base

Atushi Takeda†, Takuo Suganuma†, Tetsuo Kinoshita†† and Norio Shiratori†
†Research Institute of Electrical Communication /
Graduate School of Information Sciences, Tohoku University
††Information Synergy Center /
Graduate School of Information Sciences, Tohoku University

In this paper, we propose the Case-base type Videoconference System which is applied the case-based reasoning to the knowledge processing mechanism in Flexible Videoconference System, to adapt to environmental changes or demands of users. The conventional Flexible Videoconference System had the problem that the adaptation capability to the environment and the user requirement, which are not assumed at the time of system design, was limited. So, the new Flexible Videoconference System which can also adapt to the environment and the demands of users which are not assumed is proposed, and the design and implementation is described. Moreover, the validation of this system is verified by the experiments using the prototype system.

１．はじめに

近年，IP ネットワークをはじめとするネットワーク通信の普及に伴い，ネットワーク環境やネットワーク利用者の多様化・複雑化が進んでいる．これらの多様性，複雑性に柔軟に対応するには，従来のソフトウェア技術では環境の変化に適応する能力が欠如していることや，利用者の要求するサービスが提供できないことが障害となっている．そこで，我々は動的ネットワーキングプロジェクトにおいて，上述の問題を解決するための仕組みとして動的ネットワーキングアーキテクチャを提案している[1].

動的ネットワーキングアーキテクチャの目標は以下の通りである．

(T1)環境（ネットワーク・計算機資源）の変化に適応したネットワークサービスの提供

(T2)利用者要求に適応したネットワークサービスの提供

これらの目標を達成するために，動的ネットワー

キングアーキテクチャでは図１に示す３層構造を採用している.

従来のネットワーキングアーキテクチャは，TCP/IP をはじめとする Logical Network Layer (LNL)の機能を Application Layer (APL)に位置するアプリケーションが直接利用する構造になっていた.

これに対し，動的ネットワーキングアーキテクチャでは，LNL と APL の間に新しく Flexible Network Layer (FNL)を導入する．FNL は APL や LNL の状況に応じて動作特性を変化させることにより，環境の変化や利用者の要求に適応できるようになっている．また，FNL はやわらかいネットワーク[2]の研究成果に基づき，マルチエージェントシステムを用いることにより(T1) (T2)を実現する.

FNL 内部の各機能については現在研究が進められている[3]が本研究では, FNL の機能のうちミドルウェアサービスユニット(MSU)に関する機構に焦点を当てる．MSU は APL に対し直接的にマルチメディア通信サービスを提供する上位層インタフェースである．MSU では，様々なマルチメディアデータの処理機能群の組み合わせとパラメータの調整によってマルチメディア通信サービスを提供する.

MSU の機能要件は以下のとおりである.

(R1) 環境の変化に適応したミドルウェアサービスの提供

(R2) 利用者要求に適応したミドルウェアサービスの提供

(R1)(R2)を満たす MSU の実装例としてやわらかいビデオ会議システム (FVCS) [4]が設計・開発されてきた．しかし，従来の FVCS は，FVCS 設計時に想定されていない環境や利用者に対しての適応能力が低いという問題をもっていた(P1).

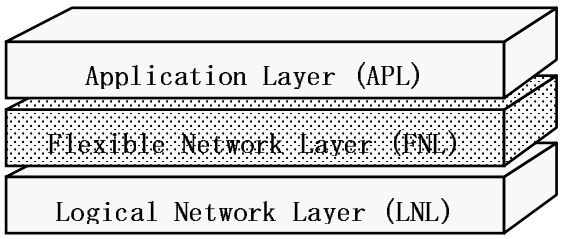

図１ 動的ネットワーキングアーキテクチャ

そこで，本稿では，従来の FVCS の問題点である(P1)を解決する新しい FVCS として，事例ベース推論(CBR)を知識処理機構に導入した Case-based FVCS (C-FVCS)を提案し, その設計と実装について述べる.

２．やわらかいビデオ会議システム

２．１． FVCS の概要

FVCS(Flexible Videoconference System) は MSU の機能要件である(R1)(R2)を満たすことを目的としたシステムである．このシステムの特徴は，環境の変化や利用者の要求に適応するためにマルチエージェントシステムを利用していることにある.

図２に FVCS の構成を示す．FVCS はネットワーク資源や計算機資源などの環境に関する情報を収集し，この情報に基づいて通信相手のビデオ会議管理エージェントとサービス品質について交渉を行う．この交渉の結果に基づいて各メディアの QoS パラメータを調整することにより，環境の変化や利用者の要求に適応したサービスを提供している.

このシステムでサービス品質を決定する役割を負っているのはビデオ会議管理エージェントである．ビデオ会議管理エージェントの以下の特性が FVCS 全体の動作特性を決定付ける.

(1) 知識（収集した情報の解析方法）

(2) 協調方式（通信相手との交渉方法）

ただし，FVCS は動作中に協調方式を変更することにより状況に応じた交渉を行うことができる[5]．この協調方式の変更はビデオ会議管理エージェント内部に格納されている知識により判断されるため，最終的にはビデオ会議管理エージェント内部に格納されている知識が FVCS 全体の動作特性を決定することとなる.

２．２． FVCS の問題点

FVCS の動作特性はビデオ会議管理エージェント

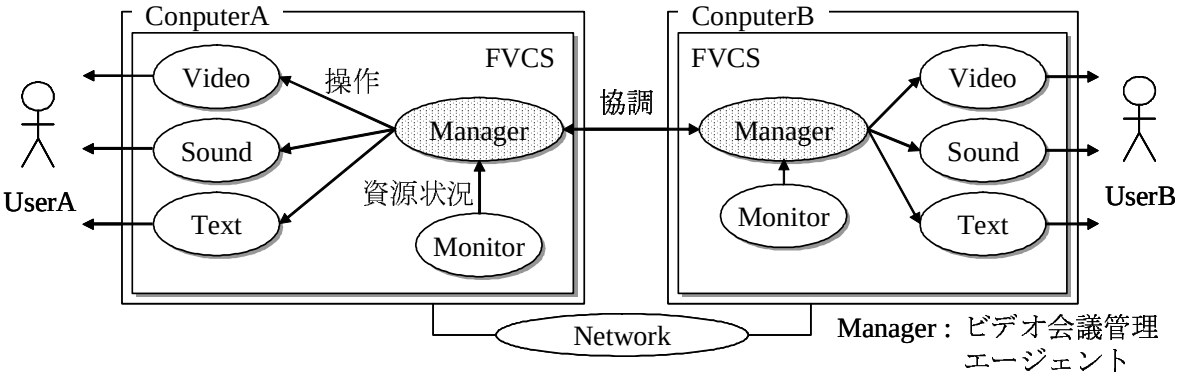

図２　やわらかいビデオ会議システムの概略

内に格納されている知識によって決定されるが，従来のFVCSでは,この知識が静的に記述されていた．ここで，静的に記述されるとは，一度記述された知識が動作中にFVCSによって更新されることがないことを表す．すなわち，この知識はシステム設計時に決定され，それ以降は変更されない．

このため，FVCS の動作特性はシステム設計時に決定され，環境や利用者が設計当初の想定から大幅に変化しても変わることはない．このことに起因し従来の FVCS では以下の問題があった．

(P1)　システム設計時に想定されていない環境や利用者に対しての適応能力が低い

具体的には，従来の FVCS では想定外の状況の場合，環境の変化に対する対応が遅れる，利用者要求の解析が不正確であるなどの問題が発生した．

３．C-FVCS：事例ベースを用いた FVCS

３．１．　概要

従来の FVCS の問題点であった(P1)を解決するためのシステムとして，事例ベース推論[6]を知識処理機構に導入した FVCS を提案する．このシステムを C-FVCS と呼ぶ．

従来の FVCS における問題(P1)は，FVCS のサービス品質を決定するための知識が静的に記述されていることに起因した．よって，C-FVCS では知識を動的に生成・変更するシステムを知識処理機構に導入する．ここで動的に生成・変更するとは，C-FVCS の動作中に知識を生成し変更することを指す．

C-FVCS では，知識の生成・変更するための機構として，事例ベース推論(CBR)を用いる．すなわち，実際に発生した状況とそのときの解決法を事例として蓄積し，事例ベース推論を用いることによりこれらの事例を活用する．ここでの事例の蓄積が，知識の動的な生成・変更にあたる．

知識の動的な生成・変更を可能にすることにより，システム設計時に想定されていない環境や利用者に対しても動作中にシステムの動作特性を変化させることにより適応能力を維持することが可能となる

３．２．　C-FVCS の動作

３．２．１．妥協レベルの導入

C-FVCS はネットワーク資源や計算機資源に関する情報に基づき通信相手の C-FVCS と交渉を行い，利用者に提供すべきサービス品質を決定する．ただし，ネットワーク資源や計算機資源の情報をそのまま交渉に用いることはできない．なぜなら，計算機資源の情報や提供するサービスのパラメータは，環境により種類や数値の意味が異なるからである．よって，環境や利用者に関する情報を抽象化した，C-FVCS の交渉のための単位として「妥協レベル」を導入する．妥協レベルとは，QoS 劣化をどれだけ許容するかを表現した知識であり，交渉を行う C-FVCS 間において統一された単位で表現される．

３．２．２．　サービス品質決定までの動作

まず，ネットワーク資源や計算機資源などの環境に関する情報，及び利用者についての情報を収集する．この収集された情報を基に通信相手の C-FVCS

との交渉に用いる協調戦略を決定する．

次に，決定された協調戦略に従って，通信相手のC-FVCSと妥協レベル調整のための交渉を行う．この交渉の結果として妥協レベルが決定される．

最後に，決定された妥協レベルに従って利用者に提供すべきサービス品質を決定する．

以上の動作により，環境や利用者に関する情報から利用者に提供すべきサービス品質を決定する．

３．２．３． 知識処理機構の役割

３．２．２にてサービス品質を決定する動作を述べたが，この動作に必要な知識を以下に挙げる．

(K1) 環境や利用者に関する情報から，用いるべき協調戦略を導出する知識

(K2) 妥協レベルから，利用者に提供すべきサービス品質を導出する知識

C-FVCSでは(K1)(K2)の知識は事例として蓄積され，この事例は事例ベース推論によって活用される．

事例データの詳細については３．３にて述べる．

３．２．４． 事例の蓄積

まず，(K1)に関する事例の蓄積手順について述べる．

(1) 利用者の指示等から，利用者が要求しているサービス品質を解析

(2) (1)で解析したサービス品質の実現を目標としている協調戦略を選択

(3) 環境や利用者の情報と，(2)で選択した協調戦略を関連付けた事例を、ライブラリに登録

以上の手順により(K1)に関する事例データを蓄積する．

次に，(K2)に関する事例の蓄積手順について述べる．

(1) 利用者の指示等から，利用者が望んでいるサービス品質を解析する

(2) 直前の交渉によって決定された妥協レベルと，サービス品質を関連付けた事例を，ライブラリに登録する

以上の手順により，(K2)に関する事例データを蓄積する．

３．３． 事例データ

３．３．１． 環境や利用者に関する情報 ⇔ 協調戦略

環境や利用者に関する情報から，そのとき用いるべき協調戦略を導出するために用いる事例データについて述べる．この事例データを活用することにより知識(K1)を実現している．

Cases = {Case}*

Case = <Condition, Resolve>

Condition = <EnvInfo, UserInfo>

Resolve = <CSAgentName>

EnvInfo = <CpuLoadInfo, NwBandInfo, ...>

UserInfo = <Service, UserAction, ...>

ServiceInfo = <VideoSize, SoundBps, ...>

事例の条件(Condition)は環境情報と利用者情報である．この条件に関連付けられている解決法(Resolve)はCSAgentとなっている．CSAgentとは任意の協調戦略を持つエージェントである．通常，CSAgentは複数存在し，それぞれが異なる協調戦略をもつ．C-FVCSでは複数のCSAgentの中から一つのCSAgentを選択することにより協調戦略を決定する．よって，事例データの中には与えられた条件に合ったCSAgentの情報が格納されている．

３．３．２． 妥協レベル ⇔ サービス品質

妥協レベルから利用者に提供すべきサービス品質を導出するために用いる事例データについて述べる．この事例データを活用することにより知識(K2)を実現している．

Cases = {Case}*

Case = <Condition, Resolve>

Condition = <CompromiseLevel>

Resolve = <Service>

Service = <VideoSize, SoundBps, ...>

事例の条件(Condition)はC-FVCS間における交渉結果である妥協レベルである．そして，そのときの解決法(Resolve)として，利用者に提供すべきサー

ビスが関連付けられている.

４．実験，評価

４．１．　実装

C-FVCS の実験，及び評価を行うため，C-FVCS のプロトタイプを実装した. 実装指針を以下に示す.

(1) 環境に関する情報は, CPU 負荷率により表現する.
(2) 利用者に関する情報は，提供しているサービスの妥協レベルにより表現する.
(3) 妥協レベルは１１段階のスカラー値とする.
(4) メディアデータとして動画メディアを扱う.
(5) 協調戦略は，「調整の迅速さ」「資源の占有率」の異なる戦略を１０種類用意する.

以上の実装指針に従い, C-FVCS を Java, 及びエージェントフレームワークである DASH フレームワークを用いて実装した.

４．２．　実験環境

Ethernet で接続された計算機 ComputerA 及び ComputerB それぞれにおいて C-FVCS を動作させ，メディアデータを送受信させる．このとき ComputerA には別のプログラム App1 を動作させる. すなわち，ComputerA の CPU 資源は C-FVCS と App1 によって共有され，App1 が計算を開始すると C-FVCS の使用できる CPU 資源は減少する.

以上の環境にて次の実験を行った

(E1)　CPU 資源と妥協レベルの変化を観測
(E2)　CPU 資源と提供される動画パラメータの変化を観測

４．３．　評価

４．３．１．実験(E1)の考察と評価

CPU 資源の変動と妥協レベルの変化の観測を行った結果を図３に示す．妥協レベルは「どれだけサービスの劣化を妥協できるか」という値なので，妥協レベルが高いほど実際に提供されるサービスは低い.

図３の[Learning]で囲まれた部分は利用者からの指示がある部分である．ただし，[Learning]以外の部分では一切の指示を行っていない.

図３の[Learning]以降の部分は指示を行っていないにも関わらず，[Learning]内と似た状況になった場合，[Learning]内で指示された変化と同様の変化を行っていることが観測される. これは, [Learning]内で行われた指示を事例として登録，再び似た状況に遭遇すると蓄積された事例を活用して利用者が望んでいる協調戦略を導出し，それを実行しているためである.

この実験結果より C-FVCS は協調戦略決の特性をシステム動作中に変化させられることがわかる．これは，C-FVCS の環境や利用者に対する適応性がシステム動作中に変化することを表し，動作開始時に想定されていない環境や利用者に対しても適応能力を維持できることを示す.

４．３．２．　実験(E2)の考察と評価

CPU 資源の変動と利用者に提供されるサービスのパラメータ変化を観測した結果を図４に示す．サービスのパラメータは画像サイズとフレームレートについて観測しているが，それぞれ１１段階で変化をし，値が高くなるほど実際に提供されるサービスの品質は高い.

図４(a)の部分において，C-FVCS に対して画像サイズよりもフレームレートを優先するようにとの指示を行った．この指示以降に提供されるサービスは常にフレームレートを画像サイズよりも優先した結果になっている．これは，C-FVCS が指示を事例と

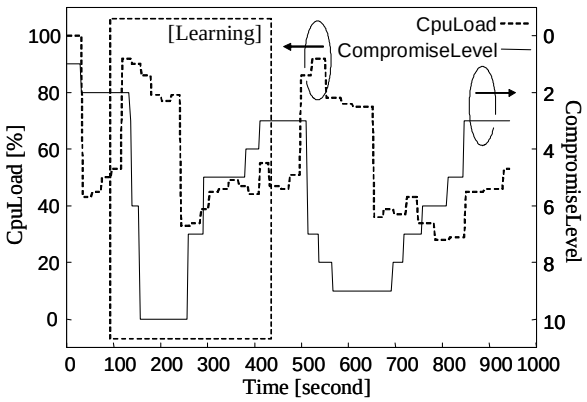

図３　CPU 資源と妥協レベルの変化

して登録し，この事例を活用することにより利用者の要求を最大限に満足させるパラメータの調整を行っているためである．

この実験結果よりC-FVCSは利用者からの指示を事例として登録し，その事例を活用することにより，サービスのパラメータを調整する知識を動的に変更し，パラメータ調整の特性を動作中に変更させることができることがわかる．これは，C-FVCSの利用者に対する適応性が動作中に変化することを表し，動作開始時に想定されていない利用者に対しても適応能力を維持できることを示す．

５．結論

本稿では動的ネットワーキングアーキテクチャにおけるミドルウェアサービスユニットとしての実装例であるやわらかいビデオ会議システム(FVCS)について述べた．また，従来のFVCSが抱えていた問題，すなわち，システム設計時に想定されていない環境や利用者に対しての適応能力が低いことを指摘した．また，この問題点は知識が静的に記述されていることに起因することを述べた．

次いで，従来のFVCSの問題点(P1)を解決するために知識処理機構に動的に知識を生成・変更するシステムを導入する必要性を述べた．

知識の生成・変更するシステムとして事例ベースを用いたシステムを採用し，事例ベースを用いたFVCS(C-FVCS)を提案し，その動作原理，及び知識となる事例データのモデルについて述べた．最後にC-FVCSの実装，及び実験を行い，その結果を，システム設計時に想定されていない環境や利用者に対しても，動作中にシステムの動作特性を変化させることにより，適応能力を維持できることを確認した．

今後の課題としては，例外的な事例に対する対処方法や，複数の事例データにある関連性の活用方法の考案が挙げられる．

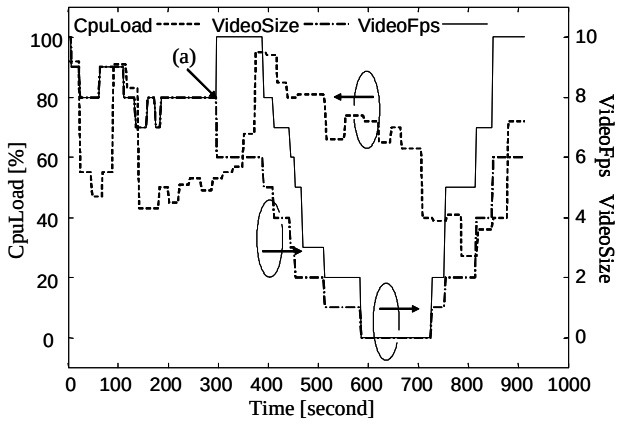

図４ CPU資源とサービスパラメータの変化

[参考文献]

[1] Susumu Konno, Gen Kitagata, Takuo Suganuma, Tetsuo Kinoshita, Kenji Sugawara and Norio Shiratori, “Dynamic Networking: Architecture and Prototype Systems”, International Conference on Computational Intelligence and Multimedia Applications 2001(ICCIMA2001),pp.93-97,2001.

[2] N. Shiratori and K. Sugawara, “Developing of a Next Generation Network Based on Knowledge-based Design Methodology”, IEICE Technical Report, AI93-28, Jul., 1993.

[3] Gen Kitagata, Takuo Suganuma and Norio Shiratori, “Application-oriented Flow Control in Dynamic Networking Architecture”, ICOIN-16(2002)

[4] 菅沼 拓夫, 藤田 茂, 菅原 研次, 木下 哲男, 白鳥 則郎, “マルチエージェントに基づくやわらかいビデオ会議システムの設計と実装”,情報処理学会論文誌, Vol.38, No.6 (1997)

[5] 李 成竺, 唐橋 拓史, 菅沼 拓夫, 木下 哲男, 白鳥 則郎, “やわらかいビデオ会議システムにおけるエージェント領域知識の構成と評価”,電子情報通信学会論文誌 B Vol.J83-B, No.2 pp.195-206 (2000)

[6] Kolodner, J. and Riesbech, C., “Case-Based Reasoning”,IJCAI-89 Tutorial-MA2 (1989)