

やわらかいビデオ会議システムの協調プロトコルの拡張と評価

唐橋拓史[†] 勝倉真[†] 菅沼拓夫[†]
菅原研次^{††} 木下哲男[†] 白鳥則郎[†]

本論文では、やわらかいビデオ会議システムにおける QoS の自律的調整機能を実現するための協調プロトコルの拡張を提案し、実装・実験を通じてその有効性を示す。やわらかいビデオ会議システムの自律的調整機能とは、ビデオ会議中の利用者要求の変化やシステム資源状況の変化に対し、自らの機能や性能を調整することで対処するための仕組みである。従来の協調プロトコルは、エージェント間での緊密な協調動作を行うためには不十分であり、システム資源の有効な活用が十分に行えなかった。そこで、従来の協調プロトコルに対し、(1) 協調状態の導入、(2) パフォーマティブの拡張、(3) 妥協レベルの導入の3点からなる拡張を施し、ADIPS フレームワークに基づき実装および実験評価を行った。その結果、エージェント間でのより高度な協調動作が可能となり、従来よりもシステムのやわらかさが向上したことが確認された。

Extension and Evaluation of Cooperation Protocol for Flexible Videoconference System

TAKUJI KARAHASHI,[†] MAKOTO KATSUKURA,[†] TAKUO SUGANUMA,[†]
KENJI SUGAWARA,^{††} TETSUO KINOSHITA[†] and NORIO SHIRATORI[†]

In this paper, we propose the extension of cooperation protocol which realizes the autonomous control of QoS and show its effectiveness with implementation and experiments of our flexible videoconference system. The autonomous QoS control in the flexible videoconference system means the framework of adjusting its functions/abilities by itself to cope with the changes of user requirements and system resource situations. But, the original cooperation protocol used by the flexible videoconference system is not enough to perform cooperation closely to use the limited resources effectively. In our proposal, the cooperation protocol is extended in the following points: (1) introducing a cooperation states, (2) extension of performatives, (3) introducing compromise levels. We embedded the proposed protocol to the flexible videoconference system based on the ADIPS framework and experimented the behavior of the system. According to the experiments on this prototype system, we confirmed that the flexibility of the system is improved.

1. はじめに

専門知識を持たない一般利用者が、家庭やオフィス等から、小型コンピュータやインターネット/LANなどの十分な計算機・ネットワーク資源が利用できない環境において、従来のデスクトップビデオ会議システム^{1),2)}を利用する場合、その起動やパラメータの調整などは、一般に容易ではない。

本論文では、利用者要求の多様性やネットワーク環

境の多様性にあわせて、機能や性能を自律的に変更可能な、「やわらかいシステム」³⁾の概念を、従来のビデオ会議システムに追加した「やわらかいビデオ会議システム」⁴⁾の設計と実装を行っている。

論文4)においては、やわらかいビデオ会議システムの3つの機能—(G1) 要求に基づくビデオ会議環境の自律的構成機能、(G2) ビデオ会議セッション中における QoS の自律的調整機能、(G3) ビデオ会議セッション中における機能変更・追加のための自律的再構成機能—を定義し、これらの機能を実現するために、エージェント指向コンピューティングの概念に基づいたエージェント型やわらかいビデオ会議システムの設計および実装を行った。同システムの3つの機能の中の1つであるビデオ会議セッション中における QoS の自律的調整機能を実現するために、論文4)では、エー

[†] 東北大学電気通信研究所
Research Institute of Electrical Communication,
Tohoku University

^{††} 千葉工業大学情報工学科
Department of Computer Science, Chiba Institute of
Technology

エージェント間で利用者要求や資源情報を交換することにより自律的にパラメータの調整を行うためのプロトコルである協調プロトコルを導入した。しかしながら、情報交換および動作要求のためのごく単純な機構しか提供していないため、これらのプロトコルでは、エージェント間の協調により、すべての資源を有効活用しつつ妥当な QoS を提供することはできないという限界があった。

本論文では、この問題を解決するために、論文4)で提案されたやわらかいビデオ会議システムの協調プロトコルに対して、最適な QoS の提供を指向した機能拡張を提案し、実装評価を通してその有効性を示す。

本論文の構成は、2章で従来のやわらかいビデオ会議システムの概要を要約する。3章で既存の協調プロトコルにおける問題点を明確化し、その拡張プロトコルを提案する。4章で提案プロトコルの実装評価を行い、その有効性を示す。

2. やわらかいビデオ会議システムと協調プロトコル

2.1 やわらかいビデオ会議システムの概念と機能⁴⁾

やわらかいビデオ会議システムでは、利用可能な計算資源が必ずしも十分でなく、しかも事前に資源を予約することのできないネットワーク環境で利用されるビデオ会議システムに対して、「やわらかさ」³⁾を組み込むことにより、柔軟性のあるシステムが実現される。これにより、動作環境にあわせて利用者要求を最大限充足するために必要な知識の習得および設定作業などにより生ずる利用者負担の軽減が可能となる。

上記の目的を達成するために、やわらかいビデオ会

議システムでは、やわらかさを提供する以下の3つの目標機能を導入し、その有効性を確認している。

- (G1) 要求に基づくビデオ会議環境の自律的構成機能
- (G2) ビデオ会議セッション中における QoS の自律的調整機能
- (G3) ビデオ会議セッション中における機能変更・追加のための自律的再構成機能

2.2 協調プロトコル⁴⁾

前節で述べたやわらかいビデオ会議システムの3つの目標機能のうち、(G2)の機能を実現するためのエージェント間プロトコルを協調プロトコルと呼ぶ。協調プロトコルは、(G1)の機能により自律的にビデオ会議環境が構成された後に、生成されたインスタンスエージェントが行う協調動作を規定している。このプロトコルにより、エージェント群が、利用者要求の変化やシステム環境の変化に対して協調的に対応する。表1に、論文4)で提案された協調プロトコルで用いられるメッセージの種別、すなわちパフォーマティブを示す。

これらの通信プリミティブを用いたエージェント間協調動作の例を図1に示す。この例では、利用者Aからの、ビデオ映像の動きの滑らかさを改善してほしい

表1 従来の協調プロトコルのメッセージ種別

Table 1 Performatives for original cooperation protocol.

パフォーマティブ名	意味
RequestAction	S (sender) は R (recipient) に対し動作要求を行う
Acceptance	S は R からの動作要求を受諾する
Refusal	S は R からの動作要求を拒絶する
Request	S は R に情報要求を行う
Information	S は R からの情報要求に対し情報を返す
Report	S は R に情報を渡す

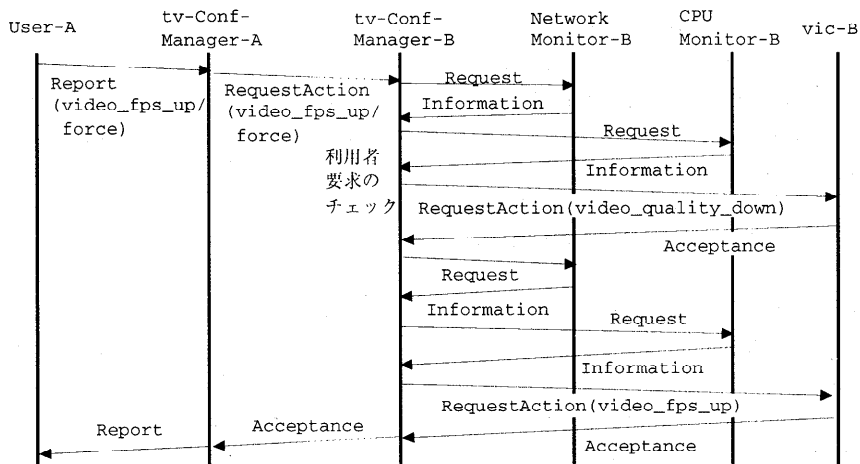


図1 協調プロトコルのシーケンスチャート

Fig. 1 Message sequence of cooperation protocol.

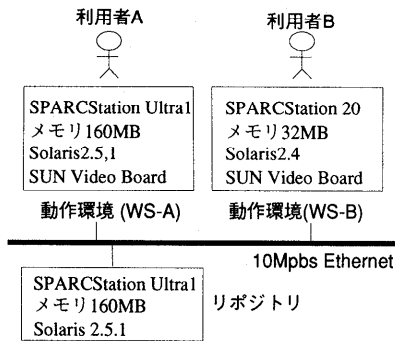


図2 実験環境
Fig. 2 Experiment environment.

という要求に対するエージェント間の協調動作を示している。具体的には、ネットワーク資源およびCPU資源にそれほど余裕がないことを tv-Conf-Manager-B エージェントが認識し、利用者 A の利用者要求でビデオ映像の画質に対する優先度が低いことから、vic-B エージェントに対し画質を下げる要求を行い、資源の開放を行ってから、滑らかさを上げる要求を vic-B に送っている。このように、協調プロトコルを用いることにより、利用者要求と資源状況を考慮した QoS 制御が行われる。

2.3 協調プロトコルの実験

2.2 節で述べた協調プロトコルを図2の環境下で実装し、各種資源が動的に変化する状況のもとで、実験を行った。

(1) 資源変動に対するシステムの対応

システムの動作中に、WS-B 側の CPU に対して負荷を与え、その動作を観察した。ここで、利用者 A の動画像に対する優先度は、動きの滑らかさが最も高く、次いで画質、解像度の順となっている。このような状況で WS-B の CPU に対して負荷を与えたとき、利用者 A に対して提供される動画像の QoS の変化を、動画像の秒間フレーム数、画像の質、解像度を測定することにより調べた。この結果を図3に示す。グラフにおいて、x 軸は時間経過 (秒)、y 軸は、各パラメータ値について、以下のそれぞれの値を 100% としたときの比率を示している。以降のグラフについても同様である。

動きの滑らかさ (Smoothness) : 10-fps

画質 (Quality) : 10-level

解像度 (Resolution) : 3-level

CPU 負荷 (CPU LOAD) : 100%

CPU に負荷が加えられると、まずシステムは a 点において画像の解像度を決定するパラメータ値を下げている。次いで、b 点において画質を決めるパラメータ

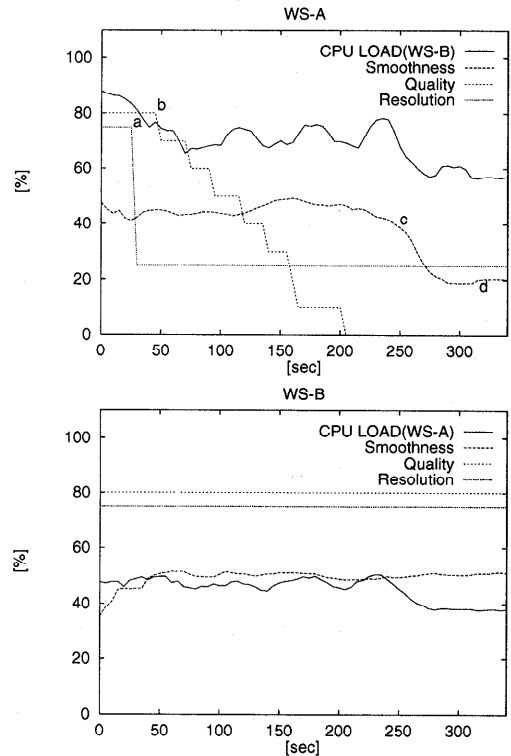


図3 実験結果 1
Fig. 3 Result of experiment 1.

値を下げていく。これにともない、CPU にかかる負荷が減少し始めている。画質のパラメータ値を最低値まで低下させた後、c 点より、優先度の最も高い、動きの滑らかさを決定するパラメータ値までも下げ始める。その後、動きの滑らかさを決定するパラメータ値が初期値の半分以下になったところで、パラメータ値の変動は停止している (d 点)。これは、この段階でエージェントの協調による処理が限界に達し、エージェント組織の再構成の処理に移行したことを示す。

以上から、優先度によって表現された利用者要求に応じて、パラメータ値を段階的に下げることにより、QoS の動的制御が行われる。しかし、WS-B 単独では、エージェント間協調による対処が d 点で限界に達し、CPU 資源状況の回復が行えなかったことが示されている。

(2) 利用者要求の変化に対するシステムの対応

システムの動作中に、利用者から QoS の変更要求を出し続け、その動作を観察した。ここで、利用者 A の動画像に対する優先度は動きの滑らかさが最も高く、次いで画質、解像度の順となっている。

システムを起動後、WS-A において、利用者 A から動画像プロセスの動きの滑らかさを上げる要求を出し

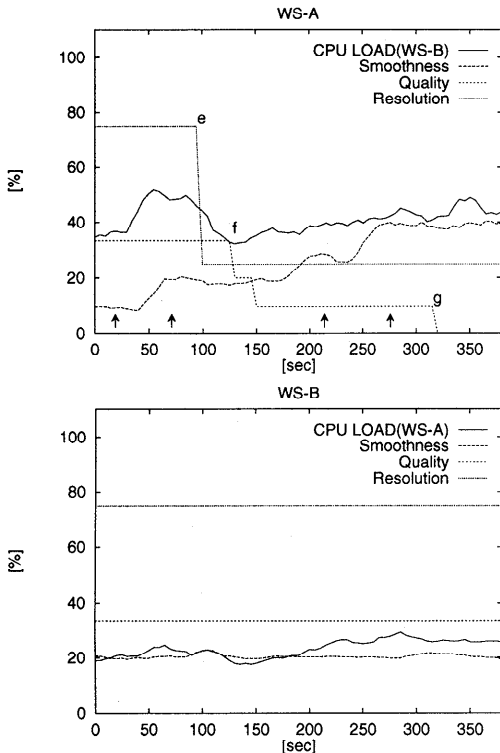


図4 実験結果2

Fig. 4 Result of experiment 2.

続けたときの動画像プロセス制御の様子を観察した。この結果を図4に示す。同図y軸上の、'↑'で示される各時点で、利用者Aから動きの滑らかさを上げる要求が発行されている。

CPUの負荷が高いときに利用者から要求を受けると、CPUの負荷を軽減するため、まずe点において、優先度の最も低い画像の解像度を下げる。次いで、f点において画質を決めるパラメータ値を下げ始める。これにともないCPUの負荷が軽減され、動きの滑らかさが改善される。さらに利用者Aが要求を出し続けると、先と同様にg点にて画質のパラメータ値を下げ始めるが、画質のパラメータ値を最低まで下げると、それ以上はパラメータ値が変動しなくなる。これは、この時点で協調によるQoS調整が限界に達し、エージェント組織再構成の処理に移行したためである。

以上から、利用者要求に応じたQoS制御が自律的に行われるが、4回目の要求発行の段階で、要求を実現する処理が不可能となったことが示されている。

3. 協調プロトコルの拡張

3.1 協調プロトコルの問題点

実験の結果、従来の協調プロトコルには、以下の2

点の問題点(P1), (P2)があることが判明した。

(P1) 利用者要求/資源状況報告の競合

従来の協調プロトコルでは、RequestActionを発行し協調のための動作を開始したエージェントに対してメッセージが送られたときには、プロトコル処理手順の整合性を保証するために、それらのメッセージを破棄する処理を行っていた。したがって、協調動作中にCPU監視エージェントCPUMonitorや利用者エージェントUserから、協調のための動作に有用な情報が送られてきた場合、すなわち協調動作と利用者要求、資源状況報告の間に競合が発生した場合に、それらを考慮した処理を行うことができない。

(P2) 協調の能力が低い

従来の協調プロトコルでは、QoS調整のためのパラメータ制御を、WS-A側あるいはWS-B側のどちらかの計算機上のエージェント組織内だけで行ってきた。これは、ビデオ会議管理エージェントtv-Conf-Manager-{A, B}間で、十分な協調を行うためのプロトコルが整備されていなかったことが原因である。

3.2 協調プロトコルの拡張

3.1節で述べた問題点を解決するために、論文4)で述べた従来の協調プロトコルに対し、(1)協調状態の導入、(2)パフォーマンスの拡張、(3)妥協レベルの導入の3点について拡張を施す。

(1) 協調状態の導入

協調動作を行っているエージェントが共通に持つ状態として、協調状態COOPを導入する。

$$\text{COOP} ::= \{\text{coop-id, status, goal, start-time, deadline, requester}\}$$

ここで、coop-idは協調状態の識別子、statusは現在の協調の状況、goalは協調を行う目的、start-timeは協調の開始時間、deadlineは協調によって問題解決を行う制限時間、requesterは協調開始を要求したエージェント名をそれぞれ表す。

この協調状態は、協調動作を開始する際の、エージェント間での協調開始のためのメッセージ交換によって発生し、goalに記述された目的が達成された場合、もしくは何らかの理由で協調動作を中止する必要がある場合に解除される。

協調状態の導入により、以下の利点が生ずる。

(a) 問題(P1)に対して、エージェントが協調中でありその協調の目的が明示されるため、協調動作中に発生するさらなる利用者要求や資源状況報告の発生に対し、適切な処理を起動することが可能となる。

(b) エージェントは、協調中のそれぞれの振舞いを、現在の協調状態を考慮しながら決定することが可能と

表2 拡張協調プロトコルのメッセージ種別 (1)
Table 2 Performatives for extended cooperation protocol (1).

パフォーマンスタイプ名	意味
Re-RequestAction	S (sender) は R (recipient) に対し動作要求を再度行う
Refusal-But	S は R の動作要求を条件付きで拒絶する
C-RequestAction	S は R に対し動作逆要求を行う
C-Re-RequestAction	S は R に対し動作逆要求を再度行う
C-Acceptance	S は R からの動作逆要求を受諾する
C-Refusal	S は R からの動作逆要求を拒絶する
C-Refusal-But	S は R からの動作逆要求を条件付きで拒絶する

表3 拡張協調プロトコルのメッセージ種別 (2)
Table 3 Performatives for extended cooperation protocol (2).

パフォーマンスタイプ名	意味
Make-Coop	S は R に対し協調開始を要求する
Acceptance-Make-Coop	S は R の協調開始要求を受理する
Refusal-Make-Coop	S は R の協調開始要求を拒絶する
Close-Coop	S は R に対し協調終了を要求する
Change-Coop	S は R に対し協調状態変更を要求する
Acceptance-Change-Coop	S は R の協調状態変更を受理する
Refusal-Change-Coop	S は R の協調状態変更を拒絶する

なり、問題 (P2) におけるエージェントの動作記述の詳細化に貢献することが期待できる。

(2) パフォーマンスタイプの拡張

表1に示されている従来の協調プロトコルにおけるRequestAction系のパフォーマンスタイプを拡張し、表2および表3に示すパフォーマンスタイプを新たに定義する。表2で示すパフォーマンスタイプは、協調状態中にあるエージェント間で交換されるメッセージであり、また表3に示すパフォーマンスタイプは、協調状態の設定/変更などのメタレベルの協調用のパフォーマンスタイプである。

これらのパフォーマンスタイプの追加によって、従来よりも高度で密な協調を行えることが期待でき、問題 (P2) の解決が見込める。

(3) 妥協レベルの導入

各エージェントの協調動作を制御するために、4段階の妥協レベルを導入する。妥協レベルは、それぞれ

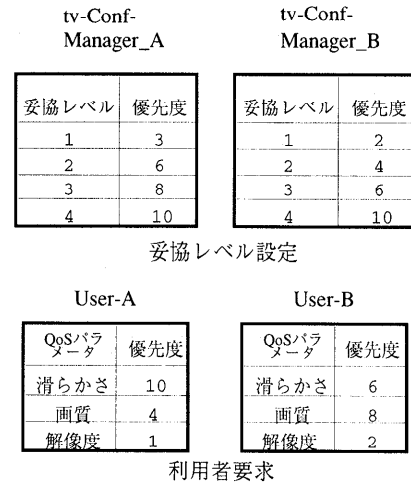


図5 妥協レベルと利用者要求の状態
Fig. 5 Status of compromise level and user requirements.

のエージェントが、協調動作において、どの程度妥協して要求に応えるかを定義するものであり、協調状態にあるエージェントは必ずどこかの妥協レベルにおいて協調を行うものとする。協調の進む段階で、協調するエージェントどうしが妥協レベルを徐々に上げながら、その妥協レベルの示す条件の範囲内で最適解を導く動作を行うことで、互いのエージェントに最適解を導き出す。

妥協レベルの変更は、表2に示されたパフォーマンスタイプの中の、Re-RequestAction, Refusal-But, C-Re-RequestAction, C-Refusal-Butの受信によって行われる可能性がある。

なお、妥協レベルはエージェント記述者によって記述され、やわらかいビデオ会議システムにおける妥協レベルは、その値として、ビデオ会議のQoSパラメータに対する優先度の値を記述する。

妥協レベルによって、各エージェントごとに柔軟に協調のための戦略を記述できるようになるため、問題 (P2) の問題の解決に貢献することが期待できる。

3.3 拡張協調プロトコルに基づくエージェント動作

本節では、3.2節で提案した拡張協調プロトコルに基づくエージェント間協調動作を述べる。この動作におけるtv-Conf-Managerの妥協レベル設定、および利用者要求の優先度は、図5に示したとおりである。

(A) システム資源状況の変化に対するQoSの自律的調整動作

図6に、拡張協調プロトコルを利用した、システム資源状況の変化に対する処理の流れを示す。

(1) 資源状況変化の検知

CPU-Monitor-Bが資源変化の許容範囲からの逸脱

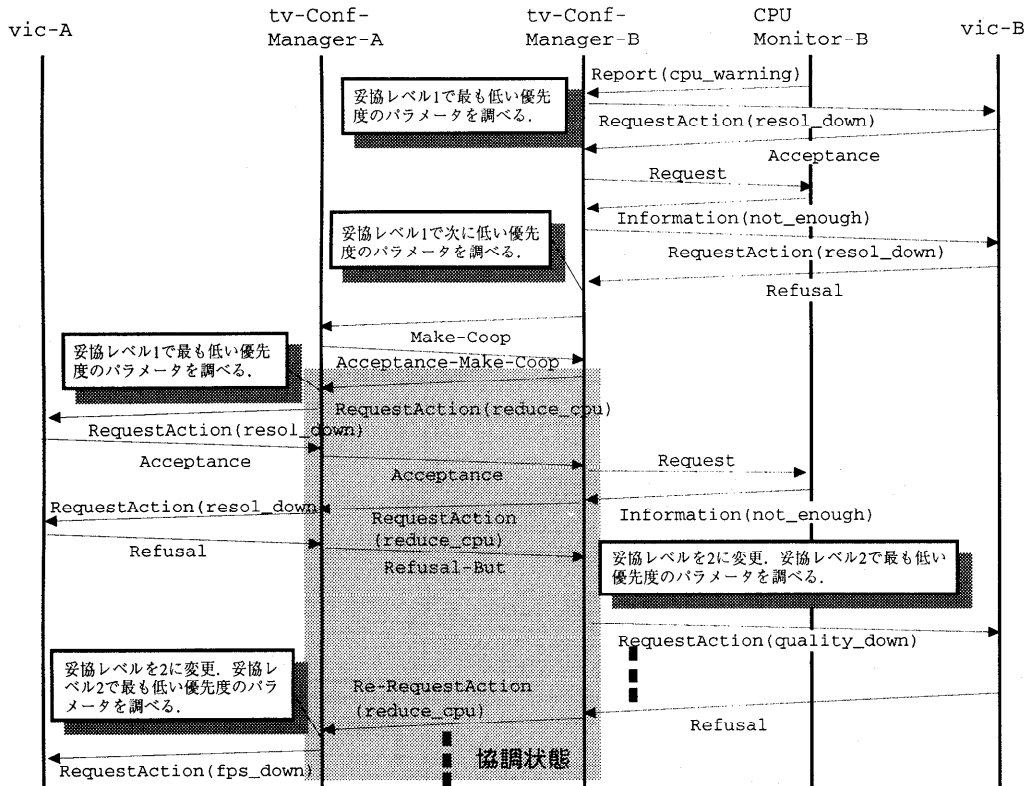


図6 拡張協調プロトコルによるCPU資源の不足に対する対応
 Fig. 6 Message sequence chart of extended cooperation protocol to cope with the lack of CPU resource.

を検知し、その旨を tv-Conf- Manager-B に Report により報告する。

(2) 障害発生側での妥協レベル1での対処

tv-Conf-Manager-B は、妥協レベル1での優先度が2であることから、User-A の優先度が2以下のQoSパラメータが存在するか調べる。この場合、解像度の優先度が1であり、妥協レベル1の範囲内であるため、まず vic-B に対し解像度を下げするための要求 RequestAction (resol-down) を発行し、CPU 資源の開放を試みる。開放を行った後に CPU 資源の状況を再度調査し、まだ資源が十分開放されていない場合は、さらに vic-B に対し、パラメータ値変更要求を発行する。

(3) 協調状態の締結

(2) の処理を繰り返し、優先度が妥協レベル1の範囲内にあるすべてのパラメータ値を下げ切っても CPU 資源状況が改善されない場合は、Make-Coop により、tv-Conf-Manager-A との間で協調状態を形成する。

(4) 相手側に対する要求と、相手側での妥協レベル1での対処

協調状態に入った後、tv-Conf-Manager-A に対し、CPU 資源を開放するための処理依頼を行う。これに対し、tv-Conf-Manager-A では、妥協レベル1の優先度が3であることから、User-B の優先度が3以下のQoSパラメータの各値を、順次下げていく。

(5) 障害発生側の妥協レベルの変更

相手側で現在の妥協レベルの範囲内の優先度を持つすべてのパラメータ値を下げ終わってもなお CPU 資源の開放が図れない場合、tv-Conf-Manager-A は、tv-Conf-Manager-B に対し、「これ以上現在の妥協レベル内では対処ができないが、そちら側が妥協レベルを上げればこちら側でも妥協レベルを上げる用意がある」旨を伝えるため、Refusal-But を発行する。これを受けた tv-Conf-Manager-B は、妥協レベルを1つ上げ、その範囲内で目的を達するための動作を行う。

(6) 相手側の妥協レベルの変更

(5) の処理によっても CPU 資源の開放が図れない場合、tv-Conf-Manager-B は、tv-Conf-Manager-A に対し、「こちらは現在の妥協レベルの範囲内で可能な対処をすべて行ったが、まだ目的が達成されていない。

要求するための逆要求 C-RequestAction を発行する。

(4) 要求発生側での妥協レベル 1 での対応

C-RequestAction を受け取った tv-Conf-Manager-A は、妥協レベル 1 の範囲内に優先度のあるパラメータ値を順次下げよう、vic-A に対し要求する。該当するパラメータ値をすべて下げ終わっても CPU 資源の回復が行えない場合、C-Refusal-But を発行する。

(5) 相手側の妥協レベルの変更

C-Refusal-But を受けた tv-Conf-Manager-B は、妥協レベルを 1 つ上げ、その範囲内で目的を達するための動作を行う。それでも CPU 資源の回復が行えない場合、C-Re-RequestAction により逆提案を再度行う。

(6) 要求発生側での妥協レベルの変更

C-Re-RequestAction を受けた tv-Conf-Manager-A は、妥協レベルを 1 つ上げ、その範囲内で目的を達するための動作を行う。該当妥協レベルでの対処をすべて終了しても資源が回復しない場合は、C-Refusal-But を発行する。

(7) 協調状態の解除

(5), (6) の処理を繰り返し、その間に CPU 資源の開放に成功した場合は、当初の目的である滑らかさ改善の要求を vic エージェントに対し発行する。滑らかさが改善されるたびに、利用者に対し、要求を充足しているかの確認を行い、充足された場合、Close-Coop により協調状態を解除する。また、改善を要求されたパラメータ以外のすべてのパラメータ値を下げ切っても CPU 資源状態が改善されない場合、協調状態が解除され、組織再構成の処理へと移行する。

4. 実験と評価

4.1 拡張協調プロトコルの実装

3 章で提案した協調プロトコルに基づくやわらかいビデオ会議システムを 2 章と同一の環境下 (図 2) で ADIPS フレームワーク⁵⁾を用い実装した。拡張協調プロトコルは、エージェントの通信機構 CM に実装するのではなく、知識ベースモジュール DK 内に Tcl/Tk⁶⁾により試験的に実装した。

拡張プロトコルのメッセージ処理部とエージェントの動作を決定する知識処理部をあわせた tv-Conf-Manager エージェントの記述量は、約 1,450 step であった。従来の協調プロトコルを実装したエージェントの記述量は、約 900 step であり、約 550 step の増加となった。

4.2 拡張協調プロトコルに基づく動作実験と評価

2 章と同様に、資源変動および利用者要求の変化に

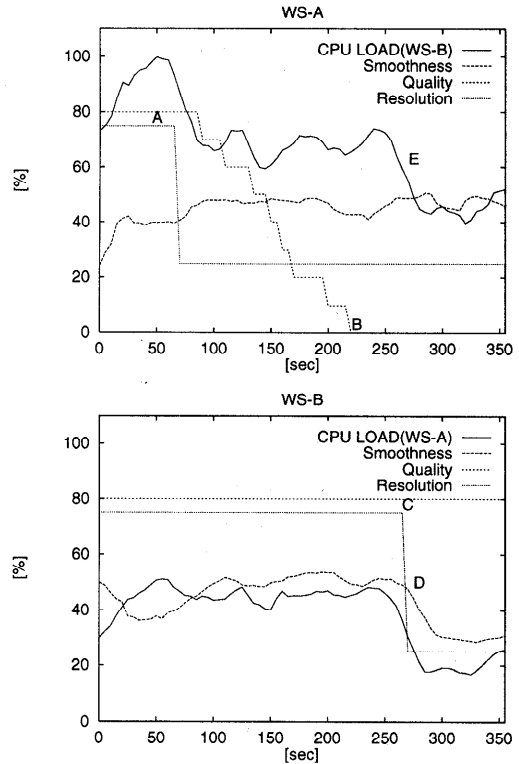


図 8 実験結果 3

Fig. 8 Result of experiment 3.

対する、拡張協調プロトコルを実装したやわらかいビデオ会議システムの対応を観察した。この結果に基づき、本プロトコルが従来のプロトコルに比べ、よりやわらかさを実現できることを示す。

(1) 資源変動に関する実験とやわらかさの評価

2 章の実験と同様にやわらかいビデオ会議システムの動作中に、一方のワークステーションの CPU に負荷を与え、システムの動作を観察した。ここで、WS-A の利用者の動画像に対する優先度は、動きの滑らかさが最も高く、次いで画質、解像度となっている。また、WS-B の利用者の動画像に対する優先度は、画質が最も高く、次いで動きの滑らかさ、解像度の順となっている。このような状況で WS-B の CPU に対して負荷を与えたとき、動画像プロセスの制御の様子を観察した。この結果を図 8 に示す。

CPU に負荷が与えられると、WS-A の画像の質のパラメータ値が最低になるまでは、従来のやわらかいビデオ会議システムと同様のパラメータ制御を行っているのが図 8 から分かる (A 点-B 点)。その後、C 点において WS-B 側の解像度のパラメータ値を下げていく。さらに D 点において WS-B 側の動きの滑らかさのパラメータ値を下げ始め、これにともない、WS-B

のCPUにかかる負荷が減少している (E点). この後パラメータ値の変動は停止している. これは, この時点でCPUにかかる負荷が十分に解放され, 組織再構成に移行することなく目的が達成されたからである. しかも, 利用者Aが最優先した動きの滑らかさ, および利用者Bが最優先した画質をまったく低下させることなく, CPU資源の回復に成功している.

従来のプロトコルでは, 一方の動作環境において資源の低下が発生した場合, もう一方の動作環境でのパラメータ調整を行えなかったため, 比較的軽い負荷であっても限界点に達して組織再構成に移行してしまうという問題点があった. しかし, 実験結果から, 協調プロトコルの拡張により, エージェント間で資源の開放を目的としたより緊密な協調動作が可能となったため, 同程度の負荷に対して, 組織再構成を行わなくとも, 現在の組織で対応が可能となることが示された.

このようにやわらかさの向上が実現できたのは, 新たに導入されたパフォーマンスと妥協レベルにより, tv-Conf-Managerが, 協調状態の中で, より緊密な協調動作が行えるようになったことによる.

(2) 利用者要求の変化に関する実験とやわらかさの評価

2章の実験と同様にやわらかいビデオ会議システムの動作中に一方の動作環境の利用者からパラメータの変更要求を送り続けたときのシステムの動作を観察した. ここで, WS-Aの利用者の動画像に対する優先度は, 動きの滑らかさが最も高く, 次いで画質, 解像度の順となっている. また, WS-Bの利用者の動画像に対する優先度は, 画質が最も高く, 次いで動きの滑らかさ, 解像度の順となっている.

このような状況でWS-Aの利用者が, 動きの滑らかさのパラメータ値の上昇を要求としてシステムに与えた際の動画像プロセスの制御の様子を観察した. この結果を図9に示す.

WS-Aの利用者Aからの要求が送られると, WS-Aの利用者の優先度が低いパラメータ値が最も低くなるまでは従来のやわらかいビデオ会議システムと同様のパラメータ制御を行っているのが分かる (F点-G点). 続けて要求が送られたときにCPUの負荷が高い状態にあると, H点においてWS-Bの画像の解像度のパラメータ値が下げられ, I点においてWS-Bの動きの滑らかさのパラメータ値が下げられ始める. これによりCPU負荷が抑えられたまま, WS-Aの動きの滑らかさのパラメータ値を上昇させることに成功している. さらにWS-Aの利用者から要求が送られると, CPU資源が不足し, J点でエージェントの組織再構成へ移

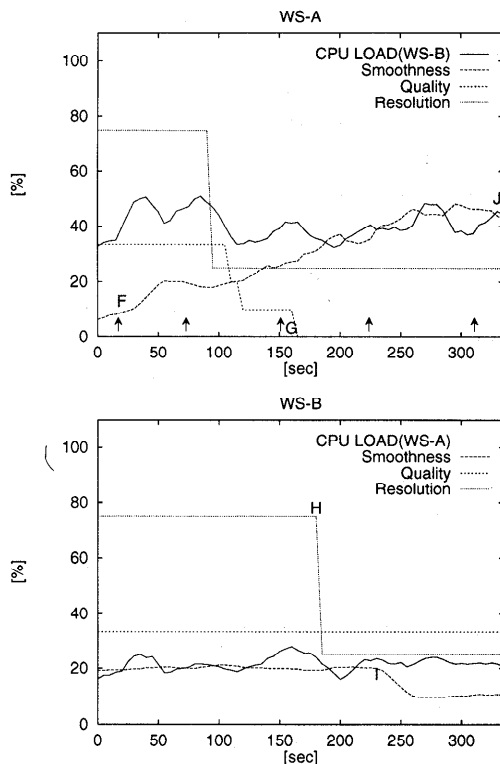


図9 実験結果4

Fig. 9 Result of experiment 4.

行している. 図4に示した実験結果2では, 4回目の利用者要求発行時に組織再構成に移行していたが, 本実験結果では, 4回目では組織再構成に移行せず, 5回目の要求発行まで, サービスが維持されている.

従来のプロトコルでは, 一方の動作環境の利用者からの要求に対して, 要求を発した利用者へのQoS決定する送信側でのパラメータ調整しか行えなかったため, 比較的早い段階で要求充足の限界点に達して組織再構成に移行してしまい, 充足可能な要求の幅が狭いという問題点があった. これに対して, 協調プロトコルの拡張により, エージェント間で要求充足を目的としたより緊密な協調動作が可能となったため, より広範な要求に対して, 組織再構成を行うことなく, サービス提供が継続できることが示された.

こうしたやわらかさの向上が可能となったのは, 新たに導入されたパフォーマンスと妥協レベルに基づく協調戦略の切換え機能によるもので, 利用者要求の変化に対してtv-Conf-Managerがきめ細かな協調動作を行えるようになったためである.

5. まとめ

従来のやわらかいビデオ会議システムの協調プロト

コルにおける問題を実験を通じ明らかにした。この問題を解決するために、(1) 協調状態の導入、(2) パフォーマティブの拡張、(3) 妥協レベルの導入の3点について協調プロトコルの拡張を行った。

拡張協調プロトコルを、ADIPS フレームワーク上に実装し、従来の方式との比較実験を行った。その結果、利用者要求の変化および計算機資源の変動に対する対応の幅が広がったことが確認され、システムのやわらかさの向上が実現できたことが検証された。

今後の課題として、協調状態や妥協レベルの記述能力を拡張することにより、協調中のエージェントの協調動作プランニングをより高度化することが残されている。

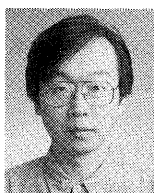
謝辞 本研究の一部はIPAの創造的ソフトウェア事業の補助を受けている。

参考文献

- 1) McCanne, S. and Jacobson, V.: vic: A Flexible Framework for Packet Video, *Proc. ACM Multimedia '95* (1995).
- 2) Turletti, T. and Huitema, C.: Videoconferencing on the Internet, *IEEE/ACM Trans. Networking*, Vol.4, No.3, pp.340-351 (1996).
- 3) Shiratori, N., Sugawara, K., Kinoshita, T. and Chakraborty, G.: Flexible Networks: Basic Concepts and Architecture, *IEICE Trans. Commun.*, Vol.E77-B, No.11, pp.1287-1294 (1994).
- 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, pp.1214-1224 (1997).
- 5) 藤田 茂, 菅原研次, 木下哲男, 白鳥則郎: 分散処理システムのエージェント指向アーキテクチャ, *情報処理学会論文誌*, Vol.37, No.5, pp.840-852 (1996).
- 6) Ousterhout, J.K.: *Tcl and the Tk Toolkit*, Addison-Wesley (1994).

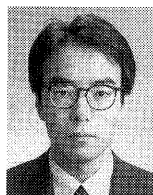
(平成9年5月15日受付)

(平成9年9月10日採録)



唐橋 拓史

1970年生。1996年東北大学大学院情報科学研究科博士前期課程修了。現在、同大学大学院同研究科博士後期課程在学中。利用者要求獲得、やわらかいネットワークに興味を持つ。



勝倉 真 (正会員)

1972年生。1997年東北大学大学院情報科学研究科博士前期課程修了。現在、日立東北ソフトウェア(株)に勤務。やわらかいネットワーク、エージェント指向プログラミングに興味を持つ。



菅沼 拓夫 (正会員)

1966年生。1997年千葉工業大学大学院博士後期課程修了。現在東北大学助手(電気通信研究所)。工学博士。やわらかいネットワーク、エージェントプログラミングに興味を持つ。



菅原 研次 (正会員)

1950年生。1980年東北大学大学院博士課程中退。同年千葉工業大学助手。現在同大学情報工学科教授。工学博士。分散人工知能、CAIに興味を持つ。1992年日本工業教育協会功績賞。1994年本会山下記念賞受賞。



木下 哲男 (正会員)

1953年生。1979年東北大学大学院修士課程修了。同年沖電気工業(株)入社。1996年東北大学助教授(電気通信研究所)。工学博士。知識表現、知識型設計支援システムなどの研究開発に従事。本会平成元年度研究賞受賞。



白鳥 則郎 (正会員)

1946年生。1977年東北大学大学院博士課程修了。1984年同大学助教授(電気通信研究所)。1990年同大学教授(工学部情報工学科)。1993年同大学教授(電気通信研究所)。情報通信システム、ソフトウェア開発環境、ヒューマンインタフェースの研究に従事。1993年本会マルチメディアと分散処理研究会主査。本会25周年記念論文賞受賞。