

## 多地点協同作業支援環境 GTWS の機器制御方式

木下 成顕\* 石崎 健史\*\* 宮本 宜則\*

\* (株) 日立製作所 中央研究所 \*\* (同) システム開発研究所

マルチメディア情報を活用した電子会議システムである多地点協同作業支援環境 GTWS を開発した。本システムでは、利用者が容易に協同作業空間を認識できること、複数の協同作業空間を表現できること、会議参加者の映像・音声情報を活用することを重視して設計を行った。協同作業空間を表現するため仮想会議室ウィンドウを設計した。複数の会議空間の切り替えは、ウィンドウのフォーカスと連動した動画・音声制御により実現している。本稿では、GTWS のシステム構成の概略を示し、ソフトウェア構成の設計コンセプトについて述べた後、映像、音声等のマルチメディア機器を統一的に扱うためのクライアントサーバ型の制御方式について報告する。

## An Architecture for Equipments Control of Group Tele-Working System: GTWS

Shigeaki KINOSHITA\* Takeshi ISHIZAKI\*\* Takanori MIYAMOTO\*

\*Central Research Laboratory, Hitachi, Ltd.

({kinosita, miyamoto}@crl.hitachi.co.jp)

1-280 Higashi-Koigakubo, Kokubunji-shi, Tokyo 185, Japan\

\*\*Systems Development Laboratory, Hitachi, Ltd.

(ishizaki@sdl.hitachi.co.jp)

292 Yoshida-chou, Totsuka-ku, Yokohama-shi, Kanagawa 244, Japan

We developed the Group Tele-Working system(GTWS), which enabled distributed participants to collaborate on their work stations using multimedia information. Multiple conference spaces are represented by the virtual meeting room windows and the participant can switch from one conference space to another by selecting the meeting room window. The voice and video are controlled with the change of the meeting rooms. This paper describes the system configuration overview and software architecture of GTWS and presents the concept and the mechanism to support multimedia equipment control scheme.

## 1. はじめに

これまでコンピュータは個人作業の効率化をめざしてその機能を伸ばしてきた。しかし、見方を変えるとその個人の作業はより大きなグループの中の一つの作業ととらえることができる。こうした視点から、コンピュータとネットワークを利用して、個人作業だけでなくグループによる共同作業、協調作業を支援するシステムが望まれている。グループワークの形態、方法はさまざまに支援するシステムに要求される機能的な要件もさまざまである[1][2][3]。

こうした背景の下にわれわれは、複数のワークステーション(WS)をネットワークで接続し参加者がインタラクティブに相互作用を行うリアルタイム系のシステムとして、映像・音声を利用したユーザインタフェースを提供する多地点協同作業支援環境GTWS(Group Tele-Working System)の研究・開発を進めている[4][5][6]。

GTWSは主に開発担当者や研究者のインフォーマルな打ち合わせを支援することを目的としている。従って、在席での利用形態を想定しており他者からの割り込み呼び出しといったダイナミックな変化にも対応できる設計となっている。ユーザインタフェースは、多地点の参加者が論理的に一つの協同作業空間に属していることを容易に理解できるよう、また、割り込み呼び出しといった複数の協同作業空間を容易に識別できるように、空間を表現した仮想会議室ウィンドウによるインタフェースを実現している。この仮想会議室ウィンドウは、利用者が直感的に理解しやすいインタフェースを目指し各種のオブジェクトを直接操作することで会議機能を利用できるビジュアルシェルの機能を実現している。

一方、こうしたシステムを実現するためには、映像・音声を扱うための各種の機器をソフトウェアに

よって制御できなければならない。この時、機器ごとに制御用ライブラリを用意するのは、仕様の異なる機器を混在して用いたり、異なるシステムスベック(例えば映像を省くなど)に対応する時など、アプリケーション(AP)の書き換えが発生し効率が悪い。そこで、GTWSにおいてはAP側から見て各種の機器を統一的に扱えるようなアーキテクチャの設計を行った。

本稿では、まずGTWSの構成とその主な機能について述べる。次にGTWSソフトウェア構成の基本的な設計方針について述べ、各種の機器を制御するAVサーバの処理方式について報告する。最後に、適用結果について検討する。

## 2. 多地点協同作業支援環境 GTWS

### 2.1 システム構成

図1にGTWSのシステム構成を示す。GTWSは、WSユーザに映像・音声・テキスト・静止画を用いたインフォーマルな協同作業支援環境を提供する。各ノードは、WS本体と映像・音声コーデックを備えた多地点会議ユニットAVU(Audio Video Unit)と、カメラ・マイク・スピーカといった付属機器で1システムを構成する。ネットワークにはFDDIを用い最大16地点が同時に接続可能である。

多地点会議ユニットには、映像・音声の符号化装置と多地点対応の復号化装置およびネットワークインタフェースと制御用プロセッサが備えられており、これに付属機器を接続する。多地点から受信した映像は、NTSC一画面を接続ノード数に応じて4または16分割して合成して出力する。WSの画面上にはビデオ表示ボードによって表示する。多地点からの音声は、それぞれを任意の大きさに加算しスピーカより再生することができる。

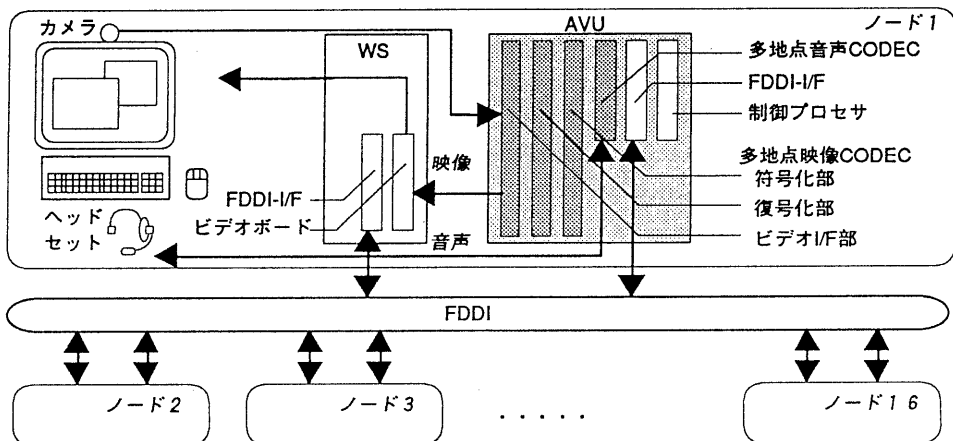


図1：システム構成

## 2.2 ユーザインタフェース

GTWSのユーザインタフェースは、多地点の協同作業の参加者が共有の作業空間を容易に認識できること、また、操作方法が直感的に分かりやすいことを目指して設計を行なった。さらに、インフォーマルな利用形態を想定し、複数の協同作業空間が同時に開かれることを念頭において設計を行なった。

### 2.2.1 仮想の部屋

GTWSでは、協同作業空間と個人の作業空間を明確に区別して表現することが容易で、マウスによる直接操作が実現できるということから仮想の部屋をメタファとして採用し、X Window System上に協同作業空間を表わすGUIを実装した。この仮想の部屋をHMR (Hyper Meeting Room)と呼ぶ。図2に実行画面例を示す。



図2：実行画面例

HMRでは会議室に備えられている机やホワイトボードとともに参加者の画像アイコンも表示され、多地点の参加者が同一の協同作業空間にいることが表現される。複数の協同作業空間を持つ時は、別のHMRが開かれ容易に空間の識別が可能である。

### 2.2.2 直接操作

HMRは協同作業に必要な機能をマウスで直接操作することによって実行できるビジュアルシェルとして機能する。この直接操作できる対象を総称して直接操作オブジェクト (DMO: Direct Manipulation Object)と呼ぶ。HMRにはOHPやホワイトボード、机、電話、共有資料が置かれ、これらは全てDMOである。DMOは各機能に応じたメソッドを持っている。メソッドはポップアップメニューから選択し起動できる。またダブルクリックによるデフォルトメソッドの起動やオブジェクトの重ね合わせによるメソッドの起動もできる。

例えば、図3に示すように、資料オブジェクトをOHPオブジェクトに重ねると「共有実行」メソッ

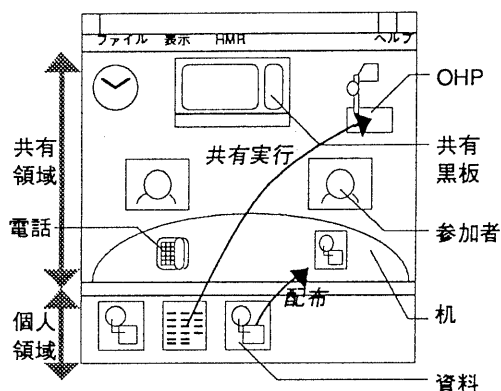


図3：DMOの直接操作例

ドが実行され、この資料を作ったAPが自動的に起動しAPのウィンドウが参加者全員に表示される。また、HMRは共有領域と個人領域に分けられ、個人と共有の資料を区別しており、個人領域から共有領域の机オブジェクト上に資料オブジェクトをのせると「配付」メソッドが実行され、参加者に資料が配られる。

### 2.2.3 協同作業空間切替え機能

協同作業の空間の認識をより明確にするため、ウィンドウ操作と連動して音声・映像の制御を行う。画面上のウィンドウを作業の種類から分類すると、

- (1) HMR ウィンドウ
- (2) 協同作業に使われるウィンドウ
- (3) 個人作業で使われるウィンドウ
- (4) 他の協同作業に関連するウィンドウ

となる。GTWSでは、(1)、(2)のように一つの協同作業空間に関連するウィンドウをグループ化して管理することにより、協同作業空間の認識/識別を容易にさせる制御を行う。具体的には、

- ・協同作業関連ウィンドウにフォーカスがある時には、その協同作業に参加している相手の音声を出しし自音声も送出する。
- ・他の協同作業関連ウィンドウや個人作業ウィンドウにフォーカスが移動した時は、移動前の協同作業空間の音声出力を停止し、自音声の送出も停止する。

という処理を行う。この時、映像も同様の処理を行う。これにより、会議参加中に別の参加者と別の会議を始めたり、会議中に他の参加者から割り込み呼びだしを受けたりした時にも、ユーザが作業を施したいウィンドウに付随する映像・音声に自動的に切り替わるので、映像音声空間と協同作業空間が一致し協同作業空間の識別がより容易になる。

### 3. ソフトウェア構成

#### 3.1 基本設計

GTWSではカメラやマイク、映像・音声コーデックといった各種の機器をソフトウェアで制御する必要がある。また、多地点の仮想会議室がお互いに接続制御したり各種の会議情報をやりとりしたりといった制御も必要である。そこで前者の機器制御を垂直方向の制御系、後者の会議制御を水平方向の制御系としてとらえ、各制御系にプロトコルを設けて制御を規定した。このプロトコルをそれぞれAVプロトコル、HMRプロトコルと呼んでいる。また、APの共有実行にはこれまで開発してきた電子対話システム ASSOCIA [7][8][9]を用いる。

#### 3.2 水平・垂直の制御系

図4にソフトウェア構成を示す。

HMRは、HMRui(会議室ユーザインタフェース)とHMRd(会議室サーバ)の二つからなる。各HMRuiはHMRdを介して多地点接続され会議情報のやり取りを行う。HMRdは、HMRuiから各種の要求を受け付け、必要な情報を他のHMRuiに通知するイベント分配制御と、参加者情報や前述したウィンドウのフォーカス状態に応じた制御に用いるウィンドウグループ情報管理などを行う。これによってネットワークに対する水平方向の制御を実現している。HMRuiとHMRdとの間で送受信されるメッセージ構成は、HMRプロトコルとして規定した。表1にHMRプロトコルの一覧を示す。

表1：HMRプロトコル

名称	内容
Join	参加要求
Leave	退席要求
InvokeHMR	HMRuiの起動要求
KillHMR	HMRuiの起動取消
ChangePersonalInfo	参加者情報の変更
ChangeHMRInfo	会議情報の変更
QueryPersonalInfo	参加者情報の問い合わせ
QueryHMRInfo	会議情報の問い合わせ
InvokeBoard	共有黒板を起動
ChangeFocusStatus	フォーカス状態通知
AddWindow	ウィンドウグループへ登録
DeleteWindow	ウィンドウグループから削除
RejectNotify	参加拒否通知
NewPersonNotify	新参加者通知
PersonLeaveNotify	参加者退席通知
FocusChangeNotify	フォーカス状態変更通知
AddWindowNotify	ウィンドウ登録通知
DeleteWindowNotify	ウィンドウ削除通知

一方、垂直方向の制御には、X Window Systemの構成を踏襲したAVサーバと呼ぶ機器制御サーバによって実現している。AVサーバへはAVlibと呼ぶ専用のライブラリでアクセスする。AVサーバはクライアントから接続され、リクエストを受信して各種の機器を制御する。AVlibとAVサーバ間で送受信されるメッセージ構成は、AVプロトコルとして規定した。表2にAVプロトコルの一覧を示す。

ここで、例えば参加者を呼び出す時には、自HMRuiから相手ホストに現在HMRdが実行されているホスト名などの情報を持ったInvokeHMRリク

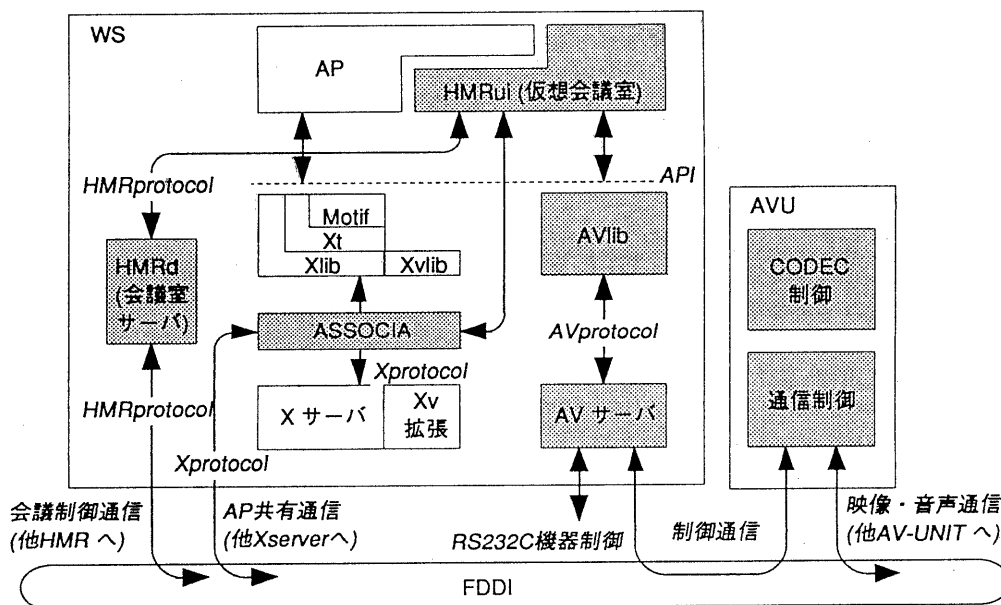


図4：ソフトウェア構成

表 2：AVプロトコル

名称	内容
QueryEquipments	利用可能機器問い合わせ
CreateControlObject	機器制御オブジェクト生成
DestroyControlObject	機器制御オブジェクト破棄
ChangeControlAttributes	属性変更
ControlEquipment	制御コマンド送出
GetControlAttributes	属性問い合わせ
ControlCreateNotify	機器制御オブジェクト生成通知
ControlDestroyNotify	機器制御オブジェクト廃棄通知
ControlChangeNotify	機器制御オブジェクト状態通知
AckNotify	機器応答通知
CompletionNotify	機器動作完了通知
NoAck	機器の無応答通知
BadIDChoice	機器 ID の不正通知
BadControl	機器制御オブジェクトの不正通知
BadValue	各種値の不正通知
BadAlloc	リソース枯渇通知

エストを送る。呼び出されたホストが参加要求を許可すると、そのホスト上に HMRui が起動され、この HMRui から HMRd に参加者名、ホスト名、実行ディレクトリ名などの情報を持った Join リクエストが送られる。これを受信した HMRd が既参加者の HMRui に NewPersonNotify イベントを送り、各 HMRui が参加者情報を取得する。次に各 HMRui は取得した参加者情報を元に、例えば映像を受信するために送信元ホスト、チャンネル番号、解像度などを指定し CreateControlObject リクエストを自 AV サーバへ送る。そして HMRui は取得した機器制御オブジェクトへ ControlEquipment リクエストで受信開始メッセージを発行する。このような手順で、水平・垂直の制御を実現している。

## 4. AVサーバ

### 4.1 仮想機器

各種の機器を取り扱う時に、機器制御の統一性や他の機器への拡張性を考慮したアーキテクチャが重要である。そこで、必要な機器を予め仮想化して扱う。AP はこの仮想化された機器を制御するスタイルとした。仮想化された機器と実際の機器は、AV サーバ が結び付け管理制御する。仮想機器の機能と実機器の機能の違いは可能な限り AV サーバ が吸収する。従って、実機器を変更しても AP からは同様の仮想機器インタフェースが実現される。

仮想機器はその機能で定義している。従って、一つの実機器に仮想機器が複数対応することもある。例えば、映像コーデックの場合、仮想映像符号化器と仮想映像復号化器に分割し定義される。仮想機器には仮想機器 ID が割り当てられており、AP はこの仮想機器 ID で機器を区別する。

仮想機器は AV サーバ の起動時に内部の管理テーブルに登録される。そのノードでどういった機器が接続されているかという情報は、接続している実機器のスペックと仮想機器の種類を記述した機器構成ファイルによって予め設定しておく。AV サーバ はこのファイルから仮想機器の情報を取得する。

### 4.2 機器制御オブジェクト

上記の仮想機器には仮想機器の状態や属性を保持するインタフェースを対応させ使用する。このインタフェースを機器制御オブジェクトと呼んでいる。一つの仮想機器に機器制御オブジェクトが複数対応することもある。例えば多地点コーデックの場合、仮想復号化器には同時に受信可能なチャンネル数と同数の機器制御オブジェクトが対応付けられる。図 5 に実機器からの対応関係を各レイヤで示す。

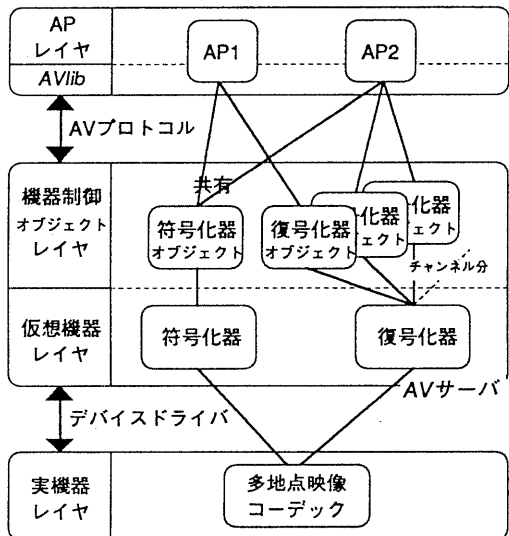


図 5：機器制御レイヤ

機器制御を行うためには、最初に機器制御オブジェクトを生成する。その際、機器は仮想機器 ID で指定する。機器制御オブジェクトの生成後は、機器制御オブジェクトにメッセージを送ることで制御が行われる。メッセージには、"PLAY", "STOP", "PAUSE" の三つがある。メッセージを送る前に該当の機器制御オブジェクトに必要な属性を設定しておく。例えば音声を受信する場合は、音声復号化器の仮想機器 ID を指定して生成した機器制御オブジェクトに、送信元のアドレス、品質、音量といった属性を設定し、"PLAY" メッセージを送ればよい。AV サーバ は機器制御オブジェクトに設定された属性とメッセージを解釈し、実際の機器のドライバレベルの制御コマンドを生成しデバイスドライバへ発行する。

### 4.3 機器共有制御

各ノードには AVサーバが一つ存在し、複数のクライアントのリクエストにサービスを提供する。AVサーバはクライアントごとに利用機器を管理しており、クライアントは HMRui だけでなく、その他の AP も AVlib を利用しサービスを利用できる。

一方、複数の協同作業空間を同時に実行する場合、機器によっては複数の HMRui で機器を共有する必要がでてくる。例えば映像の符号化器である。この時、各 HMRui が自映像を送信するため機器制御オブジェクトを生成するが、後から起動された HMRui はこの機器制御オブジェクトを取得できず映像を送ることができない。これに対処するため、仮想機器で共有可能なものは対応する機器制御オブジェクトが複数クライアントからのアクセスを受け付けるよう対応した。仮想機器の共有可/不可は 4.1 節で述べた機器構成ファイルで指定し必要に応じて切り替えられるようにした。

AVサーバは、機器制御オブジェクトの生成リクエストを受け取ると、リクエストに指定されている仮想機器 ID から共有可能かどうかを判断する。共有可能な場合、既に生成されている他の機器制御オブジェクトを調べ、リクエストに指定された仮想機器 ID で、かつ、指定されている属性と一致するものを探す。存在していたら、その機器制御オブジェクトに使用クライアントを追加し、クライアントにそのオブジェクトを通知する。存在しなければ新規にオブジェクトを生成する。生成したオブジェクトは利用クライアントが全て無くなった時点で破棄される。

### 5. 適用・考察

以上述べたアーキテクチャを C言語 で実装した。HMRui は約 12 K ステップで実装され、その中で AVlib で機器制御を行っている部分は約 0.3 K ステップであり全体の 2.5 % に収まった。AVサーバと AVlib は合計で約 11 K ステップとなっている。

本機器制御方式により、AP 側のコード量は大変少なく抑えられた。これは種々の機器のドライバを隠蔽し、機器制御オブジェクトへの属性設定とメッセージで仮想機器を制御する本方式の特長である。また多地点接続の AP 開発には、本方式のように制御系を水平・垂直に分けて考え、それぞれにプロトコルを規定することは必須であり、今回その一方式を実現することができた。

更に、本方式では複数の AP での機器共有をサポートしており、在席でのインフォーマルな打ち合わせの状況で起こり得る、第三者からの割り込み呼

び出しといった複数の協同作業空間を同時に実行する環境を構築することができた。

また AVlib による機器制御 API を GTWS 以外の AP 開発に利用する適用実験を行なった結果、ビデオサーバ<sup>[10]</sup>へのアクセス AP の試作や簡易テレビ電話の試作においても AP からの機器制御の扱いやすさが示された。これら他の AP と HMRui を、AVサーバによるクライアント毎の個別管理と機器の共有制御によって同時に実行できることも示された。

しかしながら、仮想化することにより機器固有の特殊機能を使うインタフェースが制限されてしまったことも事実である。今後、こうした固有機能への切り口を仮想機器の考え方と融合しつつ取り入れられるような検討が必要であろう。

### 6. おわりに

多地点協同作業支援環境 GTWS の概要を述べ、これを支えるソフトウェアアーキテクチャについて機器制御方式を中心に報告し、他 AP への適用も含めて本方式の特長を検討し報告した。

### 参考文献

- [1] 石井 裕著：グループウェアのデザイン，共立出版，1994.
- [2] 阪田 史郎著：グループウェアの実現技術，ソフト・リサーチ・センター，1992.
- [3] 松下 温編著：グループウェア入門，オーム社，1991.
- [4] 石崎 健史ほか：多地点間協同作業支援環境 GTWS におけるユーザインタフェース，情報処理学会第 47 回全国大会，2E-4，1993.
- [5] 木下 成顕ほか：高速 LAN を用いた多地点協同作業支援環境：GTWS，1993 年電子情報通信学会秋季大会，D-138，1993.
- [6] 宮本 宜則ほか：高速 LAN を用いた多地点映像通信システム，電子情報通信学会 [通信方式]，CS94-6，pp.39-45，1994.
- [7] 中山 良幸ほか：多者間電子対話システム ASSOCIA，情報処理学会論文誌，Vol.32，No.9，pp.1190-1199，1991.
- [8] 石崎 健史ほか：多者間電子対話システム ASSOCIA におけるアプリケーションプログラム実行方式の拡張，情報処理学会第 44 回全国大会，4M-1，1992.
- [9] 北原 千穂ほか：電子対話システム ASSOCIA における対話モニタ機能の提案，情報処理学会第 44 回全国大会，4M-2，1992.
- [10] 浅井 光男ほか：映像メディアにおける映像格納方式の検討，信学技報，IE94-13，1994-05.