

Title	マルチメディアネットワークシステムにおける資源管理エージェントの構築
Author(s)	倉岡, 貴志
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1354
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士

修士論文

マルチメディアネットワークシステム における資源管理エージェントの構築

指導教官 丹 康雄 助教授

北陸先端科学技術大学院大学
情報科学研究科システム学専攻

倉岡 貴志

1999年2月15日

要旨

近年、IEEE1394 などの標準インターフェースを搭載する家電製品の登場によって、家電製品間を相互接続するホームネットワーク環境が整ってきている。家電製品の相互接続を提供するためのミドルウェアとして Jini や Havi が開発されている。こうした家電製品を企業、学校などの LAN で利用することで、高品質で低コストのマルチメディアネットワーク環境を実現できる。しかし、現在のネットワークにおいて家電製品を対象に含めた資源管理システムがなく、マルチメディアネットワーク環境を構築しにくいという問題がある。

この問題点を解決するための一手法として、本研究では JAIST VideoLAN における資源管理機構として資源管理エージェントを提案した。本研究で提案する資源管理エージェントはネットワーク上において静的であり、ただひとつだけ存在する。資源管理エージェントは資源情報を集中管理し、資源情報を元にユーザへサービスを提供するためのエージェント・サーバである。

目次

1	はじめに	1
2	資源管理	3
3	JAIST VideoLAN	6
3.1	概要	6
3.2	システム構成	7
3.2.1	ATM ネットワーク	7
3.2.2	DV 機器	10
3.2.3	ターミナルシステム	11
3.2.4	資源管理エージェント	11
3.2.5	GUI 端末	11
3.2.6	中間ノード	12
4	資源管理エージェント	13
4.1	管理対象	13
4.2	管理方法	14
4.2.1	分散管理	14
4.2.2	集中管理	15
4.3	資源管理エージェントのアーキテクチャ	15
4.3.1	資源管理エージェントの構成	16
4.4	資源管理エージェントプロセスの設計	16
4.4.1	メッセージ通信	16
4.4.2	資源管理エージェントとユーザ間のメッセージ	17
4.4.3	ターミナルシステム間のメッセージ	18
4.4.4	中間ノード間のメッセージ設計	18

5	資源管理データベース	23
5.1	データモデル	23
5.1.1	リレーショナルデータモデル	24
5.1.2	階層データモデル	24
5.1.3	ネットワークデータモデル	25
5.2	データモデリング	26
5.2.1	概念モデル設計	26
5.2.2	論理モデル設計	29
5.3	JAIST VideoLAN の概念モデル設計	29
5.4	JAIST VideoLAN の論理モデル設計	30
5.4.1	リレーショナルデータモデルのデータ構造	30
5.4.2	リレーショナルデータモデルにおける整合制約	31
5.4.3	論理モデルの導出	32
6	ユーザ・インターフェース	36
6.1	要求定義	36
6.2	設計	37
7	実装と動作実験	40
7.1	実装システム	40
7.1.1	SQL daemon における実装	40
7.1.2	Resource Management Agent daemon の実装	41
7.1.3	HTTP daemon における実装	41
7.1.4	動作実験	42
8	考察	51
8.1	SQLdaemon	51
8.2	Resource management agent daemon	52
8.3	HTTP daemon	53
8.4	資源管理エージェント	53
9	おわりに	56

第 1 章

はじめに

近年、情報産業界の技術進歩により、従来単独であった家電製品は、家庭内ネットワークという新しい体系に変化している。そのため、デジタルビデオカメラを中心とした多くの家電製品はパーソナルコンピュータなどに接続するために、IEEE1394 などの標準インターフェースを持つようになってきている。このインターフェースを通して、家電製品はデータ転送や制御コマンドを受けることが可能となっており、パーソナルコンピュータ側から家電製品を制御することが容易に行える。そして、これら家電製品がネットワークの機能を有するパーソナルコンピュータやセットトップボックスに接続されることで、新しい利用形態のマルチメディアネットワークシステムが実現可能である。

次世代のネットワークではパーソナルコンピュータや家電製品を基点とする家庭内ネットワークが考えられている。この家庭内ネットワークとはビデオ、音声といったマルチメディアデータや制御コマンドをやり取りする、マルチメディアネットワークシステムである。こうしたネットワークの充実を図るためにはネットワークと接続される機器間を相互に取り持つミドルウェアが必要となる。この要求を満たすために、Jini や Havi などの技術研究が行われており、家庭内ネットワークの基盤が整ってきている。

こうした家電製品が家庭内ネットワークに取り込まれるように、企業、学校などの LAN に家電製品が取り込まれることの恩恵は大きいものである。家電製品は非常に取り扱いが簡単で、しかも高品質なデータ処理が可能であるため、ビデオ会議、テレビ電話などのアプリケーションにはその能力を発揮する。

しかし、家電製品は限られた機能しか持たず、拡張性も乏しいため、ネットワークで扱うには以下の問題点がある。

- 限られたネットワークプロトコルしか持っていないため、ネットワーク接続機能が十分でない

- 家電製品のような新しい資源に対応した資源管理システムがない

これらの問題点を解決する手段として、別の機器が機能を提供する事が挙げられる。前者の問題点は、ネットワーク接続機能を提供する機器を用いることで解決できる。提供する機器としては、パーソナルコンピュータ、セットトップボックスが考えられる。後者の問題点を解決するには、多様な資源に対応可能な資源管理システムを構築することで解決できる。そこで、本研究では後者の問題を解決する資源管理手法を提案する。

第 2 章

資源管理

近年のネットワークは端末やユーザの増加、複数のプロトコル体系、マルチベンダーのアプリケーションやコンポーネントなどにより、ネットワーク構成がより複雑になってきている。複雑化するネットワークを把握するため、ネットワーク上に広く分散する各種資源（コンピュータやプリンタ、ユーザやプロジェクトなど）に関する情報を管理するデータベースが必要になる。こうしたデータベースは以下に示すような利点を与える。

- ネットワーク管理者の作業負荷や管理コストを軽減する
- ネットワークアプリケーションを支援する
- アプリケーション開発の効率や柔軟性を高める
- セキュリティサービスが可能

資源管理を実現する方法として、分散管理と集中管理の 2 つに分けることができる。

集中管理は図 2.1 に示すように、あるデバイスが一つのデータベースに資源情報を管理する。アプリケーションは資源管理機構に問い合わせを行い、資源管理機構はアプリケーションに情報を提供する。幅広い資源情報を管理することで、多くのアプリケーションから利用可能となる。資源情報にアクセス属性を持たせることで、セキュリティを厳しくすることも可能である。しかし、要求に対応する管理機構が一つだけの場合、高い負荷がかかる。また、故障が生じた際にはシステム全体がダウンする。このため冗長な構成をとるなどの対策が必要である。

分散管理は図 2.2 に示すように、各担当のデバイスがそれぞれのデータベースに資源情報を管理する形態である。各担当のデバイスが必要な資源情報を持っているため無駄な処理が生じない。各自が情報管理しているので、故障が生じても被害は少ないため耐故障性

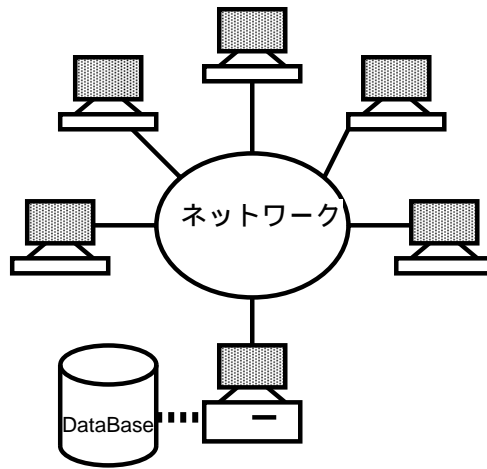


図 2.1: 集中管理のシステム構成

に優れている。しかし、資源情報が更新された場合に情報の一貫性がとれないという問題が生じる。また、担当していない資源情報が必要な場合の検索機構も必要になる。

集中管理と分散管理の比較を表 2.1 に示す。

資源情報を分散管理するデータベースの例として DNS(DomainNameSystem) や Jini の Lookup Service が挙げられる。DNS は TCP/IP ネットワークのネームサービスの仕組みで、ドメイン名を IP アドレスに変換するサービスである。DNS サーバは世界中に分散配置されており、担当のドメイン情報を管理している。

Jini では Lookup サーバと呼ばれる管理機構がサービスを管理する。サービスの名前や提供者の名前やアドレス、提供方法などの詳細情報を属性として管理し、これらの情報をアプリケーションに柔軟に提供する。この Lookup サーバがネットワーク上に分散配置され、各担当領域のサービスを管理し、サービス情報をアプリケーションへ提供する。

資源情報を集中管理するデータベースの例として NIS(NetworkInformation Service) が挙げられる。NIS は UNIX のネットワーク管理機能の一つで、管理と有効利用に必要な情報を保持するディレクトリサービスである。LAN におけるホスト名やユーザ名およびパスワードなどを集中管理し、共有することで、多くのアプリケーションを支援している。

資源管理システムは多くのネットワークで以前から用いられており、特別なアプリケーションのための専用データベースとして扱われていた。現在では、ネットワークやアプリケーションの多様化により資源管理システムはより汎用的なデータベースとして求められている。こうした資源管理機構を実現するシステムの一つとして、より高機能なディレクトリサービスが開発されている。

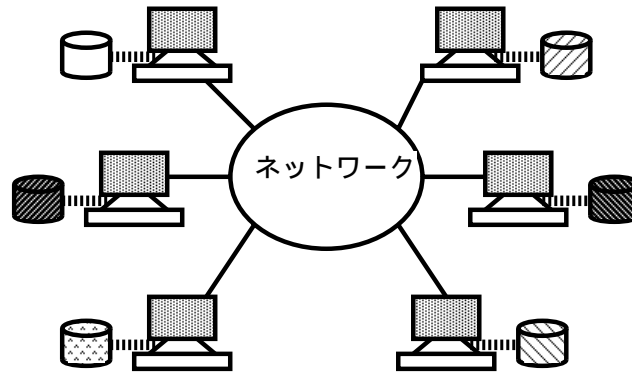


図 2.2: 分散管理のシステム構成

	集中管理	分散管理
情報の一貫性	一元管理なため一貫性が常に保てる	情報が分散するので一貫性の保持が困難
経済性	高機能な管理機構が要求されるが一つで済むため低コスト	メモリ資源や機能性が全ての管理機構に要求されるためコストがかかる
拡張性	小規模システムでは一元管理なため容易に拡張可能、大規模システムにおいては拡張が困難	拡張性は高いが管理機構間における情報の整合性を保つことが必要
信頼性	管理機構の故障でシステム全てがダウンする	一つの管理機構が故障しても動作が可能
管理性	単一の機器管理で容易	複数の機器を管理するため複雑

表 2.1: 集中管理と分散管理の比較

第 3 章

JAIST VideoLAN

本研究では設計、実装および検証の対象として、JAIST VideoLAN システムを用いる。そのため、本章では JAIST VideoLAN システムについて説明する。

3.1 概要

JAIST VideoLAN システムは、一般消費者が容易に操作できる家電製品とネットワーク化された計算機環境を融合したネットワークシステムである。このシステムは以下のアプリケーションを持っている。

- DV データによる 1 対 1、1 対多の双方向ビデオ会議
- リアルタイム MPEG2 エンコーダサーバを用いたストレージ
- スペースコラボレーションシステムを利用した遠隔会議
- Non-linear Video Editor Server を用いたビデオ編集

Non-linear Video Editor・サーバやスペースコラボレーションシステムをユーザが使うには、計算機の知識を必要とするが、1 対 1、1 対多のビデオ会議は、ターミナルシステムと呼ばれるセットトップボックスに IEEE1394 ケーブルを使って家電製品を接続し、GUI での簡単な操作だけで利用することができる。そのため、一般消費者にとって、計算機を意識せず容易に扱えるシステムである。IEEE1394 インターフェースは、現在 DV 機器を中心とした AV 家電製品に多く実装されている。

また、現在ではビデオ会議においてのさまざまなサービスを提供する中間ノードの構築および、Video on Demand サーバの利用も検討中である。

3.2 システム構成

JAIST VideoLAN システムは以下に示す 5 つのコンポーネントにより構成されている。また、同研究室の木村が研究中の中間ノードと呼ばれるデバイスが設計中であり、中間ノードを新しいコンポーネントとして組み込む予定である。

- IEEE1394 接続可能で AV/C(コントロール) コマンドを受け付けることができる DV 機器
- DV 機器などを接続する IEEE1394 インターフェースと LAN に接続される ATM インターフェースを持ち、その二つをブリッジする機能を持つターミナルシステム
- ターミナルシステム間を接続する ATM ネットワーク
- 資源情報データベースを持ち、ターミナルシステムからの要求に応答する機能を持つ資源管理エージェント
- ユーザが資源管理エージェントをアクセスするための WWW ブラウザが動作可能な端末

JAIST VideoLAN システムの概略図を図 3.1 に示す。

3.2.1 ATM ネットワーク

JAIST VideoLAN では、ターミナルシステム、資源管理エージェント間をつなぐネットワークに ATM を用いている。

ATM

ATM(非同期転送モード) はセルと呼ばれる 53 バイトの小さなパケットをスイッチングするネットワークテクノロジーで、あらゆる種類の情報を非常に高速に転送することができる。現在、ATM は LAN および、LAN 間をつなぐ WAN などのバックボーンで利用されている。ATM は以下の特徴を持っている。

- コネクション指向
ATM はエンドツーエンド間で VCC(Virtual Channel Connection) と呼ばれる仮想の通信チャネルを設定し、そのチャネルを用いて通信するコネクション指向の通信である。仮想チャネルを複数束ねたものを VP(Virtual Path) といい、仮想チャネルを設定する手順をシグナリングと呼ぶ。

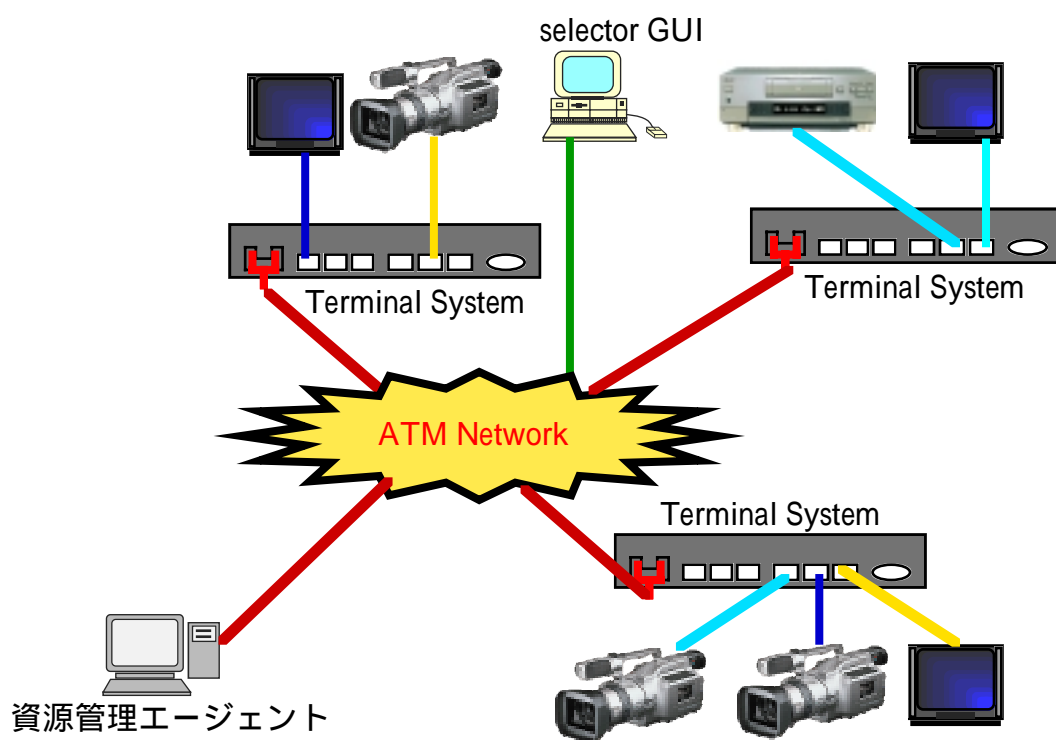


図 3.1: JAIST VideoLAN System

- セル・スイッチング

ATM はアプリケーションからのデータを 5 バイトのヘッダ部と 48 バイトのデータ部の合計 53 バイトのセルに分割して転送する。システムは 5 バイトの固定ヘッダを処理するだけでよいため、効率のよいスイッチング・ロジックをハードウェアで実現できる。そのため、ギガビットクラスまでの広帯域なデータ転送が可能である。

- QoS(サービス品質)

ATM はコネクションに対して QoS を保証する仕組みを持っており、QoS パラメータを設定することによって、単純なバルク・ファイル転送だけでなく、音声やビデオなどのリアルタイム性が要求されるデータ転送にも対応できる。ITU-T と ATM フォーラムでは、サービス・クラスに基づいて以下の 5 つの QoS カテゴリを定義している。

1. 未指定 QoS(QoS クラス 0)

最善努力型の配信サービスをサポートする。どの QoS クラスも必要としないときに使われる。

2. クラス A(QoS クラス 1)

固定ビット・レート、サーキット・エミュレーション

3. クラス B(QoS クラス 2)

可変ビット・レート、オーディオおよびビデオ

4. クラス C(QoS クラス 3)

コネクション型データ

5. クラス D(QoS クラス 4)

コネクションレス型データ

- サービス・カテゴリ

ATM はサービス・カテゴリをもっており、バースト性を持つトラフィックを扱う LAN 接続だけでなく、固定速度の専用線のようなコネクションも扱うことができる。トラフィックに合わせた柔軟なデータ転送を行うことのできるため、マルチメディア通信にも使用可能である。ATM は次の 4 つのサービス・カテゴリが定義されている。

1. CBR(固定ビット・レート)

2. VBR(可変ビット・レート)

3. UBR(未指定ビット・レート)
4. ABR(アベイラブルビット・レート)

JAIST VideoLAN では、一つの DV 接続回線でおよそ 30Mbps の CBR コネクションを張る。

3.2.2 DV 機器

JAIST VideoLAN におけるエンド ツーエンドになる機器は、ターミナルシステムに接続する DV 機器である。IEEE1394 インターフェースを持つ DV 機器は、現在、ビデオカメラ、ビデオレコーダ、メディアコンバータなどが製品化されており、デジタルビデオストリームの処理機能を持っている。また、AV/C コマンドと呼ばれる機器制御プロトコルを持っており、これを使うことで IEEE1394 ケーブルで接続されている機器を制御することもできる。

IEEE1394

IEEE1394 は FireWire や i.link といった商標名に採用されている国際標準のシリアルバス規格であり、以下のような特徴を持っている。

- 高速なデータ転送

IEEE1394 は高速シリアル転送で、現在は 100Mbps、200Mbps、400Mbps が規定されている。さらに、800Mbps やギガビットクラスのデータ転送も 1394.b でサポートされている。

- リアルタイム性の保証

IEEE1394 はアイソクロナス転送と呼ばれるデータ転送方式を採用している。この転送方式はリアルタイム性を保証してくれるため、音声や動画といったリアルタイム性の要求が強いデータ転送にも対応できる。

- Hot Plug In

電源を入れたままケーブルの抜き差しが可能であるため、新しい装置の追加や取り外しにおいて電源をオフにする必要がなく、操作が容易で安全である。

- Plug & Play

新しい装置を追加したり取り外す場合に、自動的に ID などが割り振られるため、ID

の設定をする必要がない。接続するだけで使用可能となるため、ユーザには扱いが簡単である。

- 柔軟なトポロジー

SCSIのようにデイジーチェーン接続に固定されておらず、ループ接続以外のさまざまなトポロジーの接続が可能である。接続の制約が厳しくないため、ユーザは扱いが簡単である。

3.2.3 ターミナルシステム

ターミナルシステムは IEEE1394 インターフェースと ATM インターフェースを備えたセットトップボックスである。このターミナルシステムは、ターミナルシステム同士をつなぐ ATM ネットワーク上で、VC(Virtual Circuit) と呼ばれるコネクションを張り、IEEE1394 と ATM 間において、DV データをブリッジングする機能を提供する。また、接続されている DV 機器の情報やターミナルシステム自身の情報を資源管理エージェントに報告する。

3.2.4 資源管理エージェント

資源管理エージェントは ATM インターフェースと Ethernet インターフェースを備えており、ターミナルシステムからの資源情報やユーザの情報などを管理している。これらの情報はユーザへサービスを提供する際に利用される。資源管理エージェントの詳細は次章で説明する。

3.2.5 GUI 端末

ユーザが JAIST VideoLAN でのサービスを受けるために、特別なソフトウェアは必要なく、WWW ブラウザを用いてサービスを受けることができる。WWW ブラウザが動作可能な端末であり、JAIST VideoLAN 上の資源管理エージェントにアクセスできる環境で、登録されているユーザであれば、そのユーザにサービスを提供することができる。WWW ブラウザはフリーで提供されており、現在のほとんどの PC には標準で付与されている。

3.2.6 中間ノード

JAIST VideoLAN において中間ノードはエンドツーエンドのコネクションの中間に介在し、様々なサービスを提供するデバイスで、現在設計中である。提供するサービスとして以下のものが検討されている。

- VC つなぎ合わせ

これは ATM のシグナリングにおける識別子の問題に対応するサービスである。シグナリングの際にノードを識別するビットに限りがあるために生じる問題である。つまり、ATM ではポイントツーマルチポイントのコネクションを張る際に、識別子で表現できるノードの数以上は同一のコネクションを張ることができないのである。そこで中間ノードがマルチキャストサーバの役割をして、論理的に一つのコネクションで限られたノード以上のポイントツーマルチポイントのコネクションを提供するサービスである。

- 画面分割サービス

複数の入力ストリームをまとめて一つの出力ストリームにするサービスである。複数の動画像が一つに統合されるため、臨場感のあるビデオ会議、2画面放送などが提供できる。

- DV-MPEG エンコード

DV データをリアルタイムで MPEG フォーマットにエンコードするサービスである。DV データを転送するために帯域が約 30Mbps のコネクションを張る必要がある。帯域が制限されているためにコネクションを張れない場合に、このサービスは有効になる。また、今後は MPEG だけでなくさまざまなストリームへのエンコードを検討中である。

第 4 章

資源管理エージェント

本研究では JAIST VideoLAN 内に接続されている全ての機器を資源とし、その資源の情報を管理するための資源管理エージェントを構築する。

本章では、はじめに JAIST VideoLAN 内の管理対象と管理方法について検討を述べ、資源管理エージェントの設計について述べる。

4.1 管理対象

ここでは、本研究で構築する資源管理エージェントの管理対象について検討する。資源管理エージェントは JAIST VideoLAN の資源を管理するため、JAIST VideoLAN の構成要素を管理対象として検討する。以下に、JAIST VideoLAN システムの構成要素を挙げる。

1. ATM スイッチ
2. ターミナルシステム
3. DV 機器
4. 中間ノード
5. 資源管理エージェント

これらは JAIST VideoLAN システムを構成するため、管理対象として考えられる。LAN や WAN がつながり合う大規模な ATM ネットワークを考えた場合には、ATM スイッチは各 LAN および WAN 内で管理されるからである。また、ATM スイッチ自身で管理機

構を持っているため、資源管理エージェントに管理してもらう必要性もない。これらの理由から、ATM スイッチは管理対象としない。

JAIST VideoLAN の構成要素の他に管理対象としてユーザとサービスが考えられる。一般にネットワークシステムは利用するユーザを管理している。JAIST VideoLAN においても同様にユーザを管理し、JAIST VideoLAN へのアクセスを管理することでセキュリティを高くすることができる。

JAIST VideoLAN はユーザへ複数のサービスを提供する。提供するサービスを管理することによって、アクセス制限や課金の概念を導入することが可能となる。

これらの検討によりユーザ、サービスを管理することは JAIST VideoLAN の機能性を高めるために必要であると考えた。本研究では 2 - 5 の構成要素およびユーザとサービスを資源管理エージェントの管理対象とする。

4.2 管理方法

本研究では、JAIST VideoLAN 上で集中管理する資源管理エージェントの有効性を管理方法について検討した。

ネットワーク上における資源情報の管理方法については「分散管理」「集中管理」に分類できる。ここでは、それぞれの管理方法が JAIST VideoLAN 上で実現される場合の検討について述べる。

4.2.1 分散管理

分散管理とは資源情報の共有を図らず、形式を統一せずにネットワーク上の機器それぞれが自由に資源情報を管理する方法である。はじめに、JAIST VideoLAN 上の DV 機器が資源情報を分散管理する場合を検討する。DV 機器による分散管理において最も問題となるのは、メモリ資源がなく、その拡張もできない点である。現時点の DV 機器では分散管理を行うことは現実的ではない。

次に、DV 機器が接続されるターミナルシステムが分散管理を行う場合について検討する。ターミナルシステムは高機能でメモリ資源も搭載しているため分散管理を実現できる。しかし、ターミナルシステムに資源情報を登録する必要があり、それぞれのターミナルシステムで登録処理をするため、集中管理に比べ無駄な処理が生じる。また、この登録処理を手動でやることも問題である。ターミナルシステム間で動的に資源情報を登録する処理機能を加えて、手動の登録処理をなくすこともできるが、ターミナルシステムの数が増加

するに伴い、通信トラフィックが増加する。この通信トラフィックは集中管理する場合に比べると、遥かに多くなる。最も問題となるのは、資源に依存する情報のコンシステンシの問題である。分散管理では、ある資源情報が変更された場合に、コンシステンシをとるまでに遅延が生じる。このため、誤った資源情報を使う恐れがある。また、ターミナルシステムに分散管理機構を加えることは、故障率の増加、ターミナルシステム内の CPU 資源の浪費などの問題も懸念される。

4.2.2 集中管理

集中管理はネットワーク上の、ある一つのデバイスが全ての資源情報を集中管理する方法である。この管理方法は資源情報の形式を統一したフレームワークを提供し、そのフレームワーク上で資源情報を共有する。そのため、資源情報の検索、更新、削除などが容易に行える。さらに、集中管理においては分散管理の問題点であるストレージ機能の要求は解消され、メモリ資源を節約できる。また、情報が更新される場合には、一回の更新だけで処理がすむため、コンシステンシをとる必要がなく、分散管理における余分な処理も生じない。

これらの検討により、本研究では JAIST VideoLAN の資源情報を集中管理する資源管理エージェントを構築する。

4.3 資源管理エージェントのアーキテクチャ

本研究で構築する資源管理エージェントはネットワーク上において静的であり、ただひとつだけ存在する。資源管理エージェントが行う作業は大きく三つである。それらの作業概要を以下に述べる。

- ユーザからの指示に合わせて、データベースの情報を参照し、ターミナルシステムまたは中間ノードに要求を出すこと
- 接続されているターミナルシステムおよび中間ノードからの資源情報を受け取りデータベースに格納すること
- JAIST VideoLAN のサービスを受けるための GUI をユーザへ提供すること

4.3.1 資源管理エージェントの構成

資源管理エージェントは上述したそれぞれの作業を担当する三つのプロセスから構成される。

1. RMA Daemon

資源管理エージェント本体のプロセスであり、ユーザ、ターミナルシステムおよび中間ノードとメッセージ通信し、メッセージによって要求された処理を行うエージェント・サーバ

2. SQL Daemon

資源情報を管理するためのデータベースを提供し、他のプロセスからの資源情報の問い合わせに対して応答する SQL データベース・サーバ

3. HTTP Daemon

WWW ブラウザへコンテンツを提供する JavaServlet モジュールが組み込まれた WWW サーバ

4.4 資源管理エージェントプロセスの設計

資源管理エージェントプロセスはユーザやターミナルシステム、中間ノードとメッセージ通信を行い、メッセージに応じてターミナルシステムや中間ノードの制御も行う。このプロセスは、DCS(Device Control Server)を拡張した構成にする。DCSは従来のJAIST VideoLANで動作しており、ユーザやターミナルシステムと基本的なメッセージ通信機能およびメッセージに応じたターミナルシステムの制御機能を備えている。資源管理エージェントはDCSの機能に加えて、中間ノードとのメッセージ通信機能を備える。そのため、本研究では資源管理エージェントと中間ノード間のメッセージを設計する。

ここでは、はじめにメッセージ通信の概要について述べ、ユーザと資源管理エージェント間およびターミナルシステムと資源管理エージェント間のメッセージについて説明し、本研究で設計した資源管理エージェント中間ノード間のメッセージについて述べる。

4.4.1 メッセージ通信

ここでは、メッセージ通信の概要について述べる。メッセージ通信を行うために指定したTCPポートにメッセージを送信する。メッセージの形式はメッセージ名とその引数から構成される。メッセージと引数および引数の間は空白で区切られてなければならな

い。図 5.1 にメッセージ形式を示す。このメッセージ形式で資源管理エージェントプロセス間の通信を行う。

メッセージ名 引数 1 引数 2 引数 3 . . . 引数 N

図 4.1: メッセージ形式

4.4.2 資源管理エージェントとユーザ間のメッセージ

ユーザが資源管理エージェントへサービスを要求するためにメッセージを用いて要求する。ここではユーザのメッセージとその機能について説明する。

一般ユーザが資源管理エージェントと直接メッセージ通信をするのは稀であり、直接メッセージ通信をするのは管理者またはプログラマだけである。一般ユーザは GUI を使って操作するため、GUI がユーザの要求を解釈し資源管理エージェントにメッセージを送信する。表 4.1 にユーザから送信されるメッセージとその機能の一覧を示す。

メッセージ	意味
GETCONF	VideoLAN 内のデバイス情報の一覧を得る
GETCONNECTION	DV 機器についてのコネクション情報を得る
CONNECT	ユーザから要求される送信側と受信側の DV 機器間のコネクションを設定
DISCONNECT	ユーザから要求される送信側と受信側の DV 機器間のコネクションを解放
DEBUG	デバッグレベルの再設定
PANIC	全ての TS にリセット命令を出し、データを再収集
VERSION	バージョン情報を提供
VACUUM	データベースの履歴を消去

表 4.1: ユーザ用メッセージ

4.4.3 ターミナルシステム間のメッセージ

資源管理エージェントとターミナルシステム間のメッセージおよびその機能について説明する。資源管理エージェントはユーザからのメッセージを受け、必要があればターミナルシステムにメッセージを送信する。資源管理エージェントがターミナルシステムに送信するメッセージとその機能の一覧を表 4.2 に示す。

ターミナルシステムが資源管理エージェントに送信するメッセージおよびその機能の一覧を表 4.3 に示す。

メッセージ	機能
CONNECTSEND	DV 機器間のコネクション設定を要求
DISCONNECTSEND	DV 機器間のコネクションの解放を要求
GETCONNECTRESULT	コネクション設定の結果を要求
GETNODES	TS に接続されている DV 機器の情報を要求
OVERLAY	P-MP のコネクションに対して参加要求
PANIC	TS にシステムのリセットを要求
WRITEQUADLET	DV 機器に対して制御命令の送信を要求
CONNECTIN	中間ノードを介して 2 つの TS 間のコネクション確立を要求
DISCONNECTIN	確立されている TS 間のコネクション解放を要求

表 4.2: 資源管理エージェント用メッセージ

メッセージ	機能
REGISTERBRIDGE	RMA に TS の登録を要求
TSCNOTIFY	TS の更新を通知

表 4.3: ターミナルシステム用メッセージ

4.4.4 中間ノード間のメッセージ設計

中間ノードはターミナルシステム間に介在し、従来の DV 接続サービスに付加価値を与える。中間ノードが JAIST VideoLAN に組み込まれた場合には、以下のサービスが加

わる予定である。

- VC つなぎ合わせサービス
- 画面分割サービス
- DV-MPEG 変換サービス

これらのサービスを実現するためには中間ノード、資源管理エージェント、ターミナルシステムの三者間のメッセージを設計する必要がある。

本研究では資源管理エージェントと中間ノード間のメッセージ設計について述べる。

中間ノードの登録

資源管理エージェントに中間ノードの情報を管理してもらうために、中間ノード登録メッセージが必要であるため、REGISTERIN メッセージを設計した。このメッセージはターミナルシステムの登録メッセージである REGISTERBRIDGE とほぼ同様に以下の情報をメッセージの引数として加える。

- 中間ノードの名前
- IP アドレス
- ATM アドレス
- 中間ノードが提供するサービス

このメッセージは中間ノード内の初期化処理を終えた後に資源管理エージェントへ送信される。また、常に中間ノードは資源管理エージェントのネットワークアドレスを保持している。中間ノードの情報登録動作フローを図??に示す。

メンテナンスなどのために中間ノードを停止させる場合に、登録情報を削除する必要があるために DELETEIN メッセージがある。DELETEIN メッセージには中間ノードの名前が付随して送られる。

中間ノードへの接続要求

資源管理エージェントが中間ノードを経由した接続を要求するためのメッセージが必要であるため、CONNECTIN メッセージを設計した。このメッセージは中間ノードにターミナルシステム間の接続を代行してもらうため、動作フローが異なる。

中間ノードを経由しないDV接続では、送信元となるターミナルシステムにCONNECTメッセージを送る。この時に受信先のターミナルシステムの情報が付随する。中間ノードを経由しないDV接続の動作フローを図4.3に示す。この動作フローでは資源管理エージェントが接続要求を出した後に、送信側と受信側に接続確認を出す。この接続確認に接続状態の情報要求も含まれているためである。

中間ノードを経由する場合には、中間ノードにCONNECTINメッセージを送るこの時にCONNECTメッセージに付随する情報に加え、要求するサービスIDを加える。メッセージを受けた中間ノードは、送信元と受信先のターミナルシステムにコネクションを張る。中間ノードはコネクション間をブリッジングする際に、サービスモジュールが要求されたサービスの処理を行う。両者のコネクションを確立させた後に、資源管理エージェントにINTrueメッセージを送る。INTrueメッセージにコネクションの情報も付随する。中間ノードを経由した接続の動作フローを図4.4に示す。中間ノードを経由した接続を解放するためのメッセージとして、DISCONNECTINメッセージがあり、付随する情報はCONNECTINメッセージと同じである。

本研究で設計した中間ノード間のメッセージと機能の一覧を表4.4に示す。

宛先	メッセージ	機能
RMA	REGISTERIN	RMAへ中間ノードの情報を登録
	DELETEIN	RMAへ登録してある中間ノードの情報の削除
TS	CONNECTIN	中間ノードを介して2つのTS間のコネクション確立を要求
	DISCONNECTIN	中間ノードを介して確立されているTS間のコネクション解放を要求

表 4.4: 資源管理エージェントと中間ノード間のメッセージ

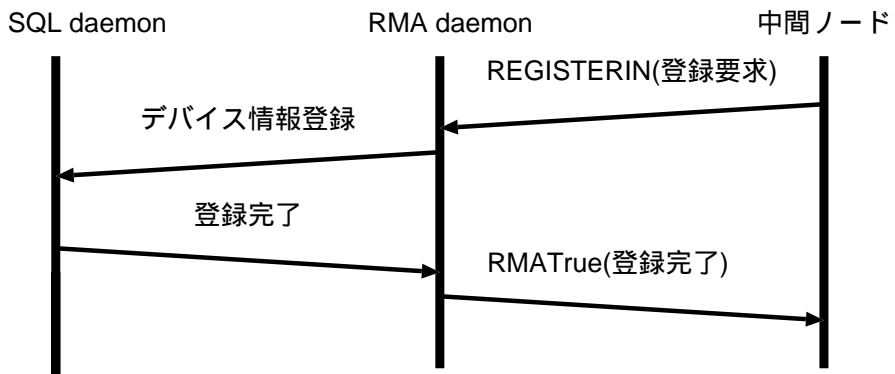


図 4.2: 中間ノードの情報登録動作フロー

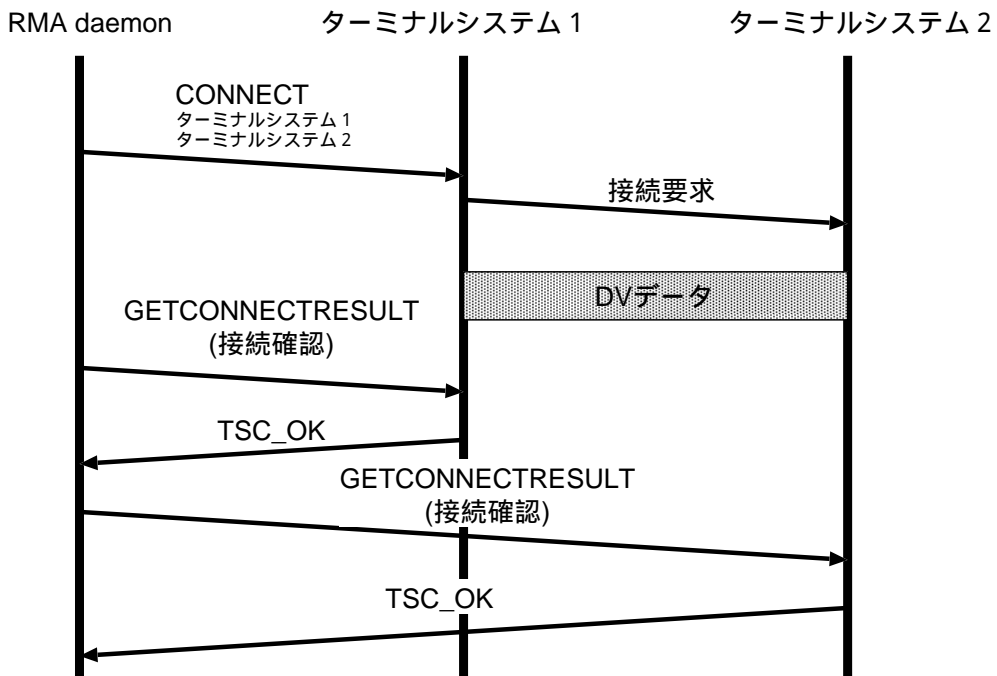


図 4.3: DV 接続動作フロー

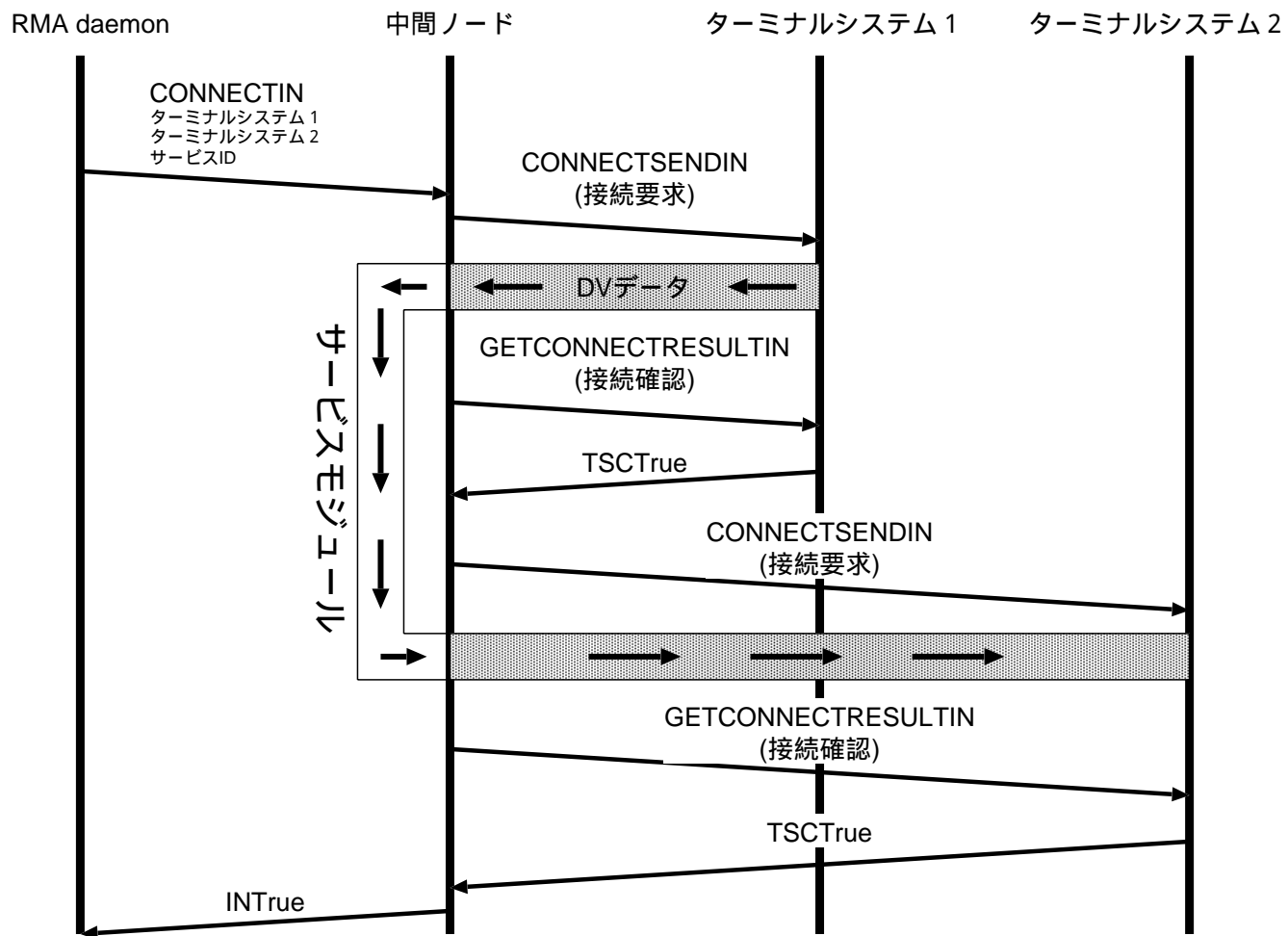


図 4.4: 中間ノード経由の接続動作フロー

第 5 章

資源管理データベース

データベースを構築するための一般的な手順に従い、はじめにデータモデルを決定する。データモデルの決定を行ったあとにデータモデリングの一連の過程に沿って論理モデルを設計する。本章では JAIST VideoLAN の資源情報を管理するデータベースの設計について述べる。

5.1 データモデル

データモデルとはデータベース中のデータとそれに対する操作を規定する枠組みであり、具体的には以下の三つの要素からなる。

1. データ構造を記述する上での規約を与える
2. データベースが正しく実世界の情報を表す上で満たさなくてはならない種々の整合性制約を表現する上での仕組みを与える
3. 規約に基づいて構造化されたデータに対してどのような検索、更新などの操作が可能かというデータ操作の体系を提供する

データモデルは古くから研究されており、これまでに数多くのデータモデルが提案されている。データモデルによって、その位置づけや機能には違いがあるが、データベースの利用においてデータモデルの主要な役割は次の二点である。

DBMS が提供するインターフェース 現在、多くの DBMS(データベース管理システム)が存在しており、これら DBMS はある一つのデータモデルをサポートしている。DBMS

がインターフェースとしてデータモデルをサポートすることによって、物理的データ格納形態や内部データ検索手順などの詳細とは独立な論理的レベルのデータ記述とその操作が可能になる。

実世界のモデル化のツール データベース設計では、対象とする実世界の複雑な情報構造や各種アプリケーションの要求を調査分析し、データベース化すべき情報を取捨選択し適切に構造化していくことが必要である、このように実世界を抽象化して論理的なデータ構造を構築することをデータモデリングと呼び、データモデルは対象の実世界の情報構造や意味をできるだけ自然に表現するための枠組みを与える。

DBMS がサポートする三つの代表的なデータモデルについて概説する。本研究では JAIST VideoLAN の資源情報を管理するデータベース構築において、現在最も主流となっているリレーショナルデータモデルを用いる。

5.1.1 リレーショナルデータモデル

リレーショナルデータモデルは、1970 年に Codd により提案されたもので、単純性、数学的基盤の明確さ、ネットワークデータモデルや階層データモデルと比べた物理的レベルからの独立性の高さなどを特徴とする。

リレーショナルデータモデルによる表現の例を図 5.1 に示す。リレーショナルデータモデルではデータの構造を表形式で表現し、それぞれの表をリレーション (関係) と呼ぶ。このリレーションの各列は属性、各行はタプル (組) と呼ばれる。リレーショナルデータモデルではデータベースをリレーションの集まりでモデル化する。リレーショナルデータベースのインスタンスはタプルの集まりによって表現される。

リレーショナルデータモデルにおけるデータ操作を規定した形式的体系としては、リレーショナル代数とリレーショナル論理がある。リレーショナル代数は、リレーショナルの集合に対して、集合演算 (和、積、差) を含む基本的操作を行う代数演算子を提供する。リレーショナル論理は一階述語論理にその基礎を置いている。リレーショナルデータモデルにおいてデータ操作は、リレーショナル代数を用いる方が一般的である。

5.1.2 階層データモデル

階層データモデルは 1968 年に IBM 社が開発した IMS (Information Management System) という DBMS に用いられており、ポインタやチェーン、多重リンクなどのコンピュータのファイル編成法が背後にあって提案されたデータモデルである。階層データモデルで

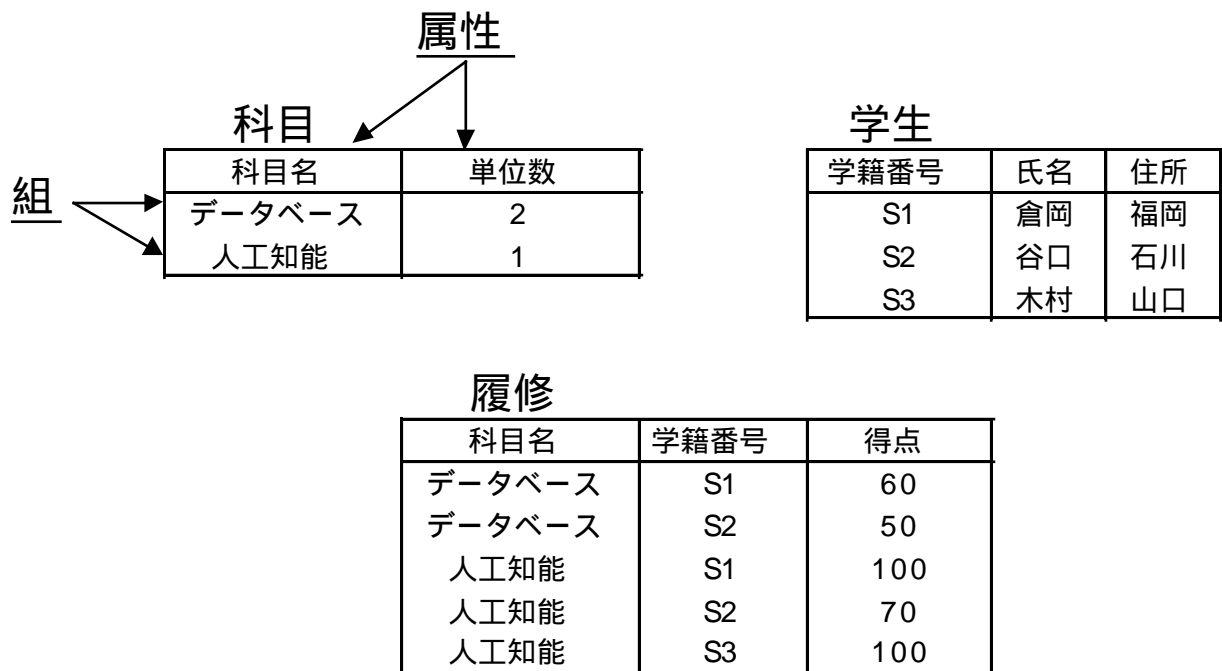


図 5.1: リレーショナルデータモデル例

はデータ構造をレコードと呼ばれる一連のデータ値の並びで表現する。このレコードをツリーを構成する節点とする。節点と節点の間で親子関係を定義し、ツリー構造で対象の実世界を表現するため、単純で理解しやすいデータベースを構築することができる。しかし、複雑なデータ構造の表現には向いておらず、多対多の関係を表現することが困難になる。図 5.2に階層データモデルによる表現の例を示す。

5.1.3 ネットワークデータモデル

ネットワークデータモデルは階層データ構造と共に、ファイルシステムにおけるデータ管理を高度化するという初期の DBMS 発展の歴史の中で体系化されたモデルである。ネットワークデータモデルは CODASYL(The Conference on Data Systems Languages) のデータベース作業班である DBTG(Data Base Task Group) が 1971 年に出した提案が元である。

ネットワークモデルも階層構造と同様で、データ構造をレコードと呼ばれる一連のデータ値の並びで表現するが、階層構造の他に複数の上位レコードを許すネットワーク構造が許されている。複数の親子関係が許されているため、親子関係を識別するために名前が付

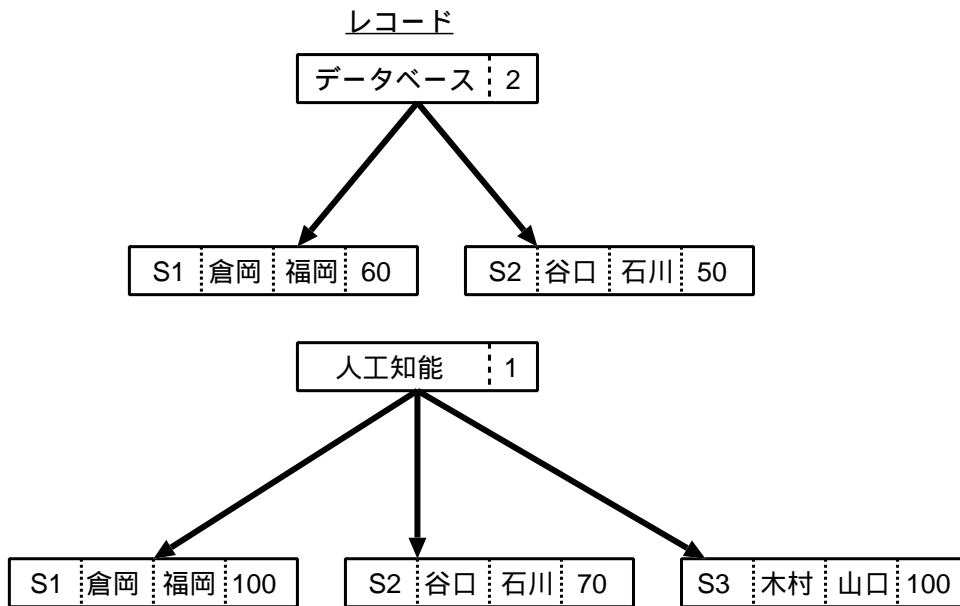


図 5.2: 階層データモデル例

けられる。図 5.3n にネットワークデータモデルによる表現の例を示す。

5.2 データモデリング

DBMS で管理可能なデータベースを構築するためには、DBMS がサポートするデータモデルによるデータ記述を行う必要がある。対象の実世界から直接データモデルのデータ記述を導出することは、記述表現上の大きなギャップを伴う。また、DBMS がサポートするデータモデルの制約にとらわれない、データの自然な表現を持っていることも有用である。これらの理由から、データベース設計の過程は一般に図 5.4 に示すように二つの段階を経て行われる。それぞれの設計過程について述べる。

5.2.1 概念モデル設計

第一段階として、DBMS がサポートするデータモデルとは独立に対象となる実世界のデータ構造がどのようなものかを、ある記号系を用いて記述する過程である。この過程は図 5.4 における記号化に相当する。実世界からデータ構造を認識するためにアクセプタが必要であり、アクセプタはデータベースデザイナーや測定器、特徴抽出プログラムなど様々

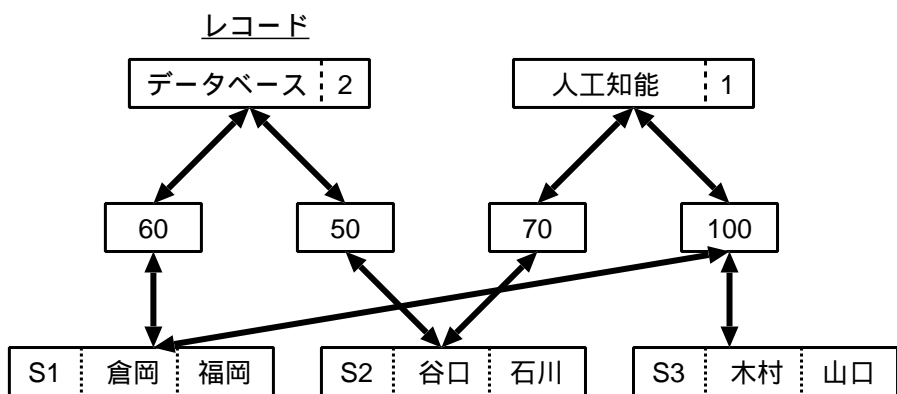


図 5.3: ネットワークデータモデル例

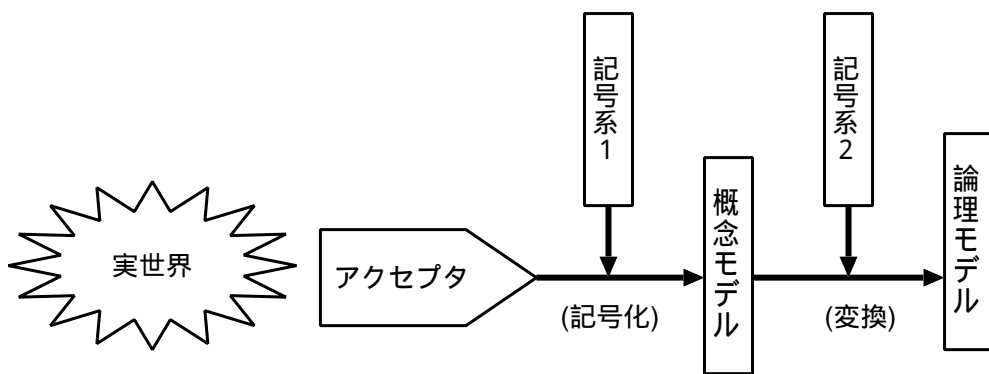


図 5.4: データモデリング

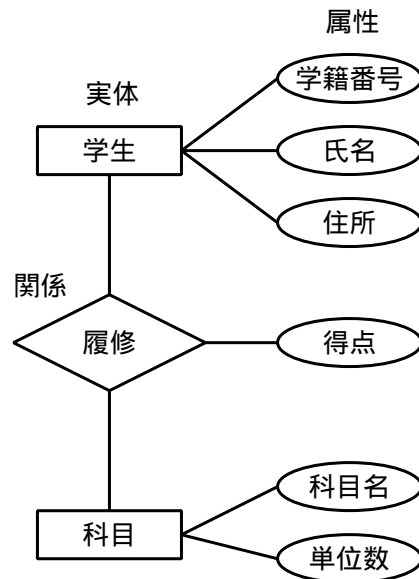


図 5.5: 実体－関連図の例

なものがある。この段階で必要な記号系 1 は、実世界のデータ構造を表現できるものであれば何でもよく、現在では実体－関連モデルが一般的である。

一般にはこの第一段階で得られる実世界のモデルは概念モデルと呼ばれる。

実体－関連モデル 実体－関連図は概念モデルの設計にしばしば利用されるデータモデルで、1976 年に Chen により提案され、その後多くの研究者によりさまざまな拡張がなされている。

実体－関連モデルは、実世界の対象を実体と関連という二つの概念を用いてモデル化する。実体は実世界をモデル化しようとしたときに、対象を包括的に述べたものであり、同一種類の実体の集まりを実体集合と呼ぶ。実体の持つ各種性質は属性によって表現され、実体を表現するための属性の集合は実体集合に対応して決定される。

関連は二つ以上の実体同士の相互関係をモデル化したものである。実体集合の関連を関連集合と呼ぶ。属性は関連集合に対しても付加することができる。

図 5.5 に実体－関連モデルを使った実世界の記述例を示す。実体－関連モデルの記述において実体集合を四角形、関連集合を菱形、属性を楕円で表現する。また、実体集合および関連集合とその属性の結びつきは線で表現する。

実体－関連モデルは、概念モデルをより詳細に記述するための制約も備えている。

5.2.2 論理モデル設計

第二段階では、概念設計で導出した概念モデルを DBMS が管理可能な表現にするためのモデル変換を行う。この段階を経て変換した結果は論理モデルと呼ばれる。図 5.4 において記号系 2 に相当するのがデータモデルである。

5.3 JAIST VideoLAN の概念モデル設計

JAIST VideoLAN をモデル化するため、はじめに実世界の対象を決定する。実世界の対象を、JAIST VideoLAN の構成を元に検討し、次に資源管理エージェントの視点から検討する。

JAIST VideoLAN の構成からの検討 JAIST VideoLAN を利用するのはユーザであり、JAIST VideoLAN はユーザにサービスを提供する関係である。サービスを実現するためには JAIST VideoLAN 内のデバイスを動作させる必要がある。そこで、次のような関係ができあがる。「JAIST VideoLAN はユーザにサービスを提供し、そのサービスを担当するのはデバイスである。」この関係を元に、実体－関連図を記述する。記述した実体－関連図を図 5.6 に示す。

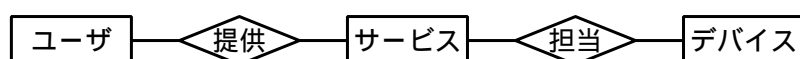


図 5.6: JAIST VideoLAN の構成を元にした実体－関連図

資源管理エージェントの視点からの検討 JAIST VideoLAN の構成を検討したモデルを資源管理エージェントの視点から検討する。図 5.6 において実体であるデバイスが資源管理エージェントから見て抽象度が高くなっている。JAIST VideoLAN に接続されるデバイスは、ターミナルシステム、中間ノード、資源管理エージェントなどのホストや DV 機器などが挙げられる。これらのデバイスは、識別するための名前を形式的に定められておらず、それぞれの機能も異なるため抽象度を下げる必要がある。ターミナルシステム、中間ノードは高機能で一意に示す名前を持つことができるため、DV 機器とは違った性質を持っている。DV 機器はターミナルシステムに接続される機器であり、利用形態が異な

る。また、DV 機器は各機器固有の値を持つノードユニーク ID を持っているため、一意に識別可能である。そのため、DV 機器をデバイスから分類し、次の関係を作る。「DV 機器はデバイスに接続される」

DV 機器は再生、停止などの機器制御コマンドの送信や受信の機能を持っている。この機器制御コマンドは数多く、DV 機器によって異なる。DV 機器それぞれの機能情報を資源管理エージェントが管理することでユーザに機能情報を提供できる。そのため次の関係を作る。「DV 機器はそれぞれの機能を持っている」

ユーザからの要求によって資源管理エージェントは DV 機器間に接続を張る。そのため、接続の情報を保持する必要があるため、次の関係を作る。「DV 機器間は送信元から配信先への接続を張る」

これらの関係と実体に管理する属性の記述を加えて概念モデルを設計する。また、関連集合と実体集合の間の対応関係を示すための制約も加えている。図 5.7 に本研究で設計した JAIST VideoLAN の概念モデルを示す。

5.4 JAIST VideoLAN の論理モデル設計

この節では、概念モデルからリレーショナルデータベーススキーマを導出する。リレーショナルデータベーススキーマとはリレーションスキーマの集合と関連する整合性制約の集合を規定するもので、リレーショナルデータモデルによる論理モデルである。

リレーショナルデータベーススキーマを規定するために、リレーショナルデータモデルにおけるデータ構造と整合性制約について説明する。

5.4.1 リレーショナルデータモデルのデータ構造

リレーショナルデータモデルにおける基本データ構造はリレーションであり、リレーションの概念は数学的な n 項関係の概念に基づいている。

一般に、集合 S_1, \dots, S_n が与えられたとき、集合 $\{(X_1, \dots, X_n) \mid X_1 \in S_1 \wedge \dots \wedge X_n \in S_n\}$ を S_1, \dots, S_n の直積集合と呼び、 $S_1 \times \dots \times S_n$ で表す。その部分集合を S_1, \dots, S_n 上の n 項関係と呼ぶ。

リレーショナルデータモデルにおけるリレーションの構造は、リレーションスキーマに $R(A_1, \dots, A_n)$ よって記述され、 R はリレーション名、 A_1, \dots, A_n は属性と呼ばれる。一つのリレーションスキーマの中では、属性は一意的でなければならない。属性の個数 n はリレーションスキーマの次数と呼ばれる。各属性 A_i にはその取り得る値の集合 D_i が付

随し、各 D_i はドメイン (定義域) と呼ばれる。

リレーションスキーマ $R(A_1, \dots, A_n)$ が与えられたとき、 $D_1 \times \dots \times D_n$ の有限部分集合 $r \subseteq D_1 \times \dots \times D_n$ はインスタンス、 r の各要素はタプルと呼ばれる。

5.4.2 リレーショナルデータモデルにおける整合制約

データモデルはデータの整合性を表現する仕組みを提供する。リレーショナルデータモデルの主な整合制約を以下に挙げる。

1. 第一正規形制約

ドメインはモデリング上分解不可能な単純値のみの集合を対象とする制約であり、具体的にはドメインがドメインの直積集合、巾集合になってはならないということである。

2. ドメイン制約

インスタンス $R(A_1, \dots, A_n)$ 中のタプルの各成分はそれぞれ A_1, \dots, A_n のドメイン要素でなければならない制約である。ドメインの要素が実世界において取り得ない値を取ってはならないということである。

3. キー制約

リレーションスキーマ $R(A_1, \dots, A_n)$ が与えられたとき、そのいかなるインスタンスにおいても、二つ以上のタプルが (空値となる場合を除く) 同一の属性値を持つことがないような属性、属性の集合は超キーと呼ばれる。超キーが極小であるとき、キーまたは候補キーと呼ばれる。超キーが極小であるとは、属性、属性の集合のうち一つでも欠けると、一意に識別できなくなる事を意味している。候補キーのうち、データ管理上最も適当でありかつその属性値が空値になり得ないものを一つ選択して主キーとする。

キー制約とはリレーションスキーマに付随して主キーと全ての候補キーが指定されたとき、そのいかなるインスタンスも主キーと候補キーに関する上記の条件を満たさなければならないという制約である。

4. 参照整合性制約

リレーションスキーマ $R_1(\dots, FK, \dots)$ と $R_2(\dots, RK, \dots)$ が与えられ、 PK は R_2 の主キーであり、 FK のドメインと PK のドメインは一致するものとする。このとき、 R_1 と R_2 のいかなるインスタンス r_1 および r_2 においても、次の関係が成り立つ場合、 FK を外部キーであるという。

[r_1 中の任意のタプルが持つ FK の値は、それが空値である場合を除き、 r_2 中に存在するあるタプルが持つ PK の値でなければならない]

参照整合性制約とは、リレーションスキーマに付随して外部キーが指定されたとき、そのいかなるインスタンスも外部キーに関する上記の条件を満たさなければならないことである。

5. 従属性

リレーショナルデータモデルでは、キー制約や参照整合性制約に加えて、実世界の各種整合性制約を記述するための手段として、従属性の概念が用いられる。その最も基本的なものが関数従属性である。

リレーションスキーマ $R(\dots, X, \dots, Y, \dots)$ が与えられたとき、その任意のインスタンス中の任意の二つのタプルに関して、もしその X の値が等しいならば、 Y の値も必ず等しいという制約が成り立つとき、関数従属性 $X \rightarrow Y$ が成立するという。

関数従属性以外の従属性としては、多値従属性、結合従属性などの従属性があり、これらの従属性はデータ更新時の振舞いに優れたリレーショナルデータベースを設計する上で必要となる。

5.4.3 論理モデルの導出

概念モデルから Theory らの手法をベースにした導出法を用いて、リレーショナルデータベーススキーマを導出する。導出手順を以下に示す。

1. 実体集合の扱い

(a) 通常実体集合

通常実体集合 E に対してリレーション R を定義し、 E の全ての属性を R の属性とする。また、 E の主キーを R の主キーとする。

(b) 弱実体集合

実体集合 E' を識別上のオーナとする弱実体集合 E に対してはリレーション R を定義し、 E の全ての属性を R の属性とする。さらに、 E' に対応するリレーションの主キーを R の属性に加え、これと E の部分キーの組合わせを R の主キーとする。

(c) 汎化階層の扱い

実体集合 E に関する汎化階層があり、 E の上位の実体集合 E' が存在する場合

には、 E' に対応するリレーションの主キーを E に対応するリレーション R の属性に加えて、これを R の主キーとする。

2. 関連集合の扱い

関連集合により関係付けられる二つの実体集合に対応するリレーションを R_1, R_2 とする。

(a) 1 対 1 対応の場合

R_1 あるいは R_2 のいずれかにもう一方のリレーションの主キーおよび (もしあれば) 関連集合自身の属性を付加する。

(b) 1 対 N 対応の場合

R_1 および R_2 がそれぞれ 1 側および N 側の実体集合に対応したリレーションであるとする。このときに、 R_2 に R_1 の主キーおよび (もしあれば) 関連集合自身の属性を付加する。

(c) N 対 M 対応の場合

R_1 の主キー、 R_2 の主キー、および (もしあれば) 関連集合自身の属性からなる新たなリレーションを定義する。通常、 R_1 の主キーと R_2 の主キーの組合わせをこのリレーションの主キーとすることができる。

上述した導出手順を JAIST VideoLAN の概念モデルに適用する。

1. 通常実体集合に関する規則より以下のリレーションを導出する。

ユーザ (ユーザ ID, ユーザ名, フルネーム, パスワード, アドレス, サービス ID, コメント)

サービス (サービス ID, サービス名, 担当機器, GUI, コメント)

デバイス (ホスト名, デバイス名, 所属, ネットワークアドレス 1,
ネットワークアドレス 2, 設置場所, サービス ID)

DV 機器 (ノードユニーク ID デバイス名, 入出力, シリアル番号)

2. 関連集合に関する規則における 1 対 N 対応の場合により、DV 機器のリレーションスキーマを以下のように変更する。

DV 機器 (ノードユニーク ID デバイス名, 入出力, シリアル番号, ホスト名)

DV 機器 (ノードユニーク ID デバイス名, 入出力, シリアル番号, ホスト名, 機器名)

3. 関連集合に関する規則における N 対 M 対応の場合により、以下のリレーションを導出する。

提供 (ユーザ ID, サービス ID)

担当 (サービス ID, ホスト名)

コネクション (送信元ノードユニーク ID, 配信先ノードユニーク ID, コネクション番号,
帯域, コンテンツ, 送信元情報 1, 送信元情報 2, 送信元情報 3,
配信先情報 1, 配信先情報 2, 配信先情報 3)

以上のリレーションスキーマからなるリレーショナルデータベーススキーマが導出される。

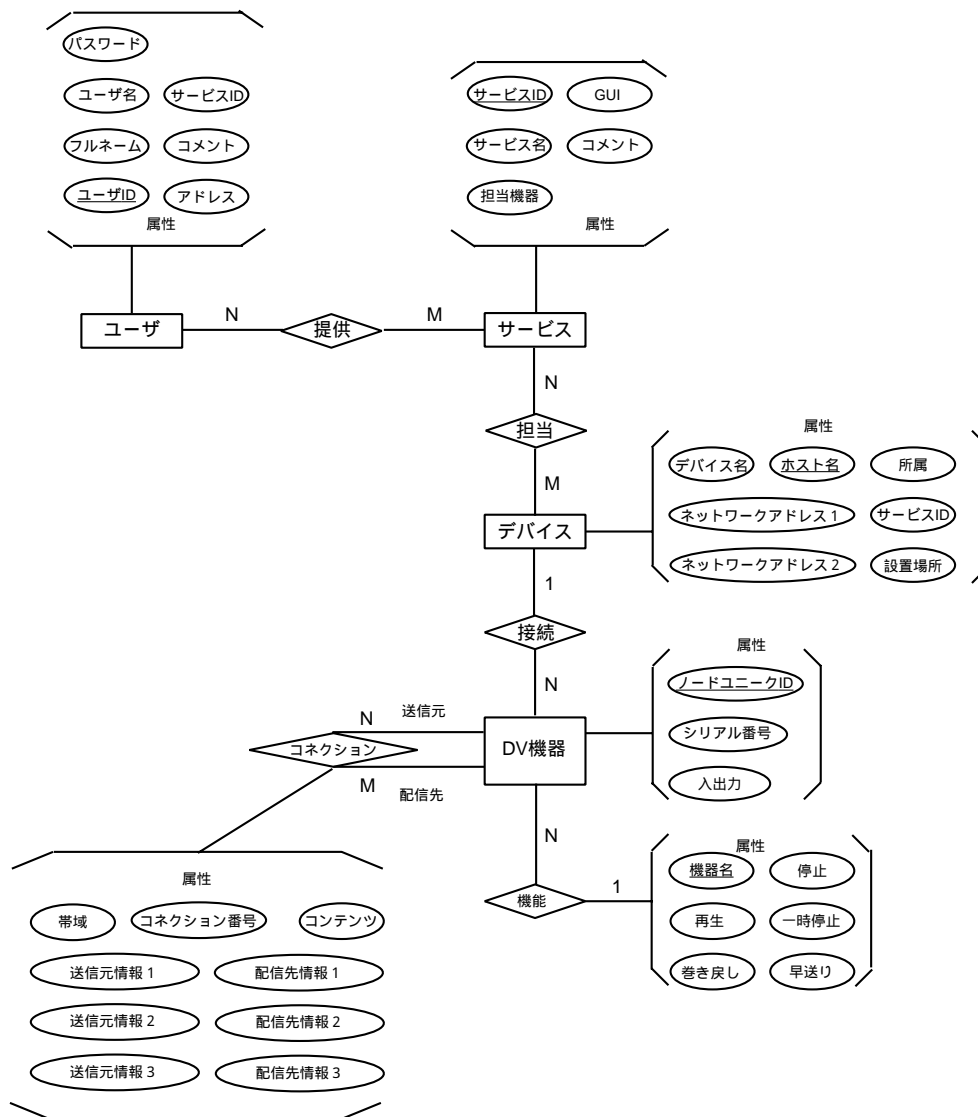


図 5.7: 実体一関連図

第 6 章

ユーザ・インターフェース

多くのプラットフォームで動作ができる GUI を構築することは、利便性および柔軟性の高いシステムを構成する要因となる。しかし、バージョンアップやパッチあてなどの更新作業が全てのプラットフォームで必要となり、高い保守性と人的コストが要求される。

この問題を解決する手段として WWW サーバを使って WWW ブラウザに GUI 機能を持つコンテンツを提供する方法がある。この方法は、WWW サーバ側の管理だけですむために管理作業や保守作業が低コストで容易に行える。また、CGI や JAVA テクノロジーを用いることで高機能な GUI を作成することが可能になっている。

本章では、WWW サーバが JAIST VideoLAN の GUI を提供するシステムの設計について述べる。

6.1 要求定義

ここでは、要求される GUI の機能を提供するサービスから検討する。本研究では以下のサービスを行う GUI を提案する。

- ユーザ認証サービス
ユーザにユーザ名とパスワードを要求し、認証を行う。ユーザ名とパスワードを確認後、ユーザの情報を調べ、サービス選択 GUI を提供する。
- ユーザ管理サービス
ユーザ情報の登録、削除、変更を行うサービスで、管理者用である。
- DV 機器接続サービス
JAIST VideoLAN 内でアクティブな DV 機器の一覧を表示し、それらの接続を提供

するサービスである。

- ネットワーク情報サービス

現在のネットワーク構成情報、DV 機器間の接続およびマルチキャストグループの情報を提供するサービスで、管理者用である。

これら GUI を構築するには、ユーザの要求を処理するプログラムをサーバまたはブラウザで動作させる環境およびデータベース・サーバへアクセスするインターフェース、資源管理エージェントへアクセスするインターフェースが必要である。

6.2 設計

本節では、要求定義で述べた環境とインターフェースについて検討する。

プログラム動作環境

ユーザの要求を処理するためのプログラムを動作させる環境が必要である。WWW サーバ側でプログラムを動作させる環境としては CGI が、ブラウザ側でプログラムを動作させる環境は JavaApplet が一般的である。

CGI は多くの言語を実行できる柔軟な環境を持っているがリクエストごとにプロセスを fork し処理が重たいことが問題となる。FastCGI と呼ばれる高速な CGI も存在するが一般的ではない。また、データ・ベースサーバへアクセスするため、容易に排他制御機を行える機構を持っていることが望ましいが、CGI は排他制御が困難である。

こういった問題に対応するため、サーバ側のプログラム動作環境として本研究では JavaServlet 環境を用いる。JavaServlet はサーバ側で動作する Java であり、オブジェクト指向のプログラミング環境を享受することで開発の効率化が得られる。また、その他の Java 技術とも親和性が高いため Java テクノロジーが活かせることも魅力である。JavaServlet は JavaVM をあらかじめ起動させ、マルチスレッドで動作するように設計されているため、処理速度の向上が図られており、CGI に比べてかなり高速である。また、Java は排他制御を行うための機構を持っているため、データベースと連携したシステム構築が容易である。

JavaApplet はブラウザ側で動作するプログラムである。JavaApplet は処理速度の問題が懸念されるが、Java の豊富なクラスライブラリを用いて高機能な GUI を実現することができる。

TCP/IP インターフェース

Java のクラスライブラリは TCP/IP インターフェースを提供している。これを用いることで資源管理エージェントプロセスへメッセージを送る機能の実現ができる。サーバ側でこの機能を実現するには JavaServlet を用いる。また、クライアント側に JavaApplet を提供することで、クライアント側からメッセージを送ることも可能である。両者とも同機能を実現できるが、サーバ側で実現する場合にはクライアントと資源管理エージェントプロセスの間に WWW サーバが介入するためオーバーヘッドが生じるだけでなく、無駄に CPU 資源を消費する。この理由により資源管理エージェントプロセスへアクセスする機能はクライアント側で実現するのが理想的である。

データベース・インターフェース

Java からデータベース・サーバを利用するための標準アプリケーション・プログラム・インターフェースとして JDBC(Java Database Connectivity) がある。JavaServlet 環境で JDBC を用いることにより、データベースと連携したコンテンツを提供することが可能となる。データベース・サーバをアクセスする GUI は JavaApplet を用いても実現できるが、排他制御の実現とパフォーマンスに関して次の問題が考えられる。Java は排他制御機構を持っているが JavaApplet 間での排他制御は困難であるため実現しにくい。また、データベース・サーバへアクセスする機能を JavaApplet で実現することで、処理がさらに重くなることが懸念される。このため、データベースサーバへアクセスする機能はサーバ側で行う方法が、効率的で構築も容易である。

動作フローの設計

これらの検討を元に、本研究では JavaServlet、JavaApplet、JDBC を使って GUI を構築する。ユーザ認証サービス、ユーザ管理サービス、ネットワーク情報サービスはデータベースアクセスが中心であるため JavaServlet と JDBC を用いて構築する。これらのサービスの動作フローを図 6.1 に示す。JavaApplet を用いた DV 接続サービスの動作フローを図 6.2 に示す。

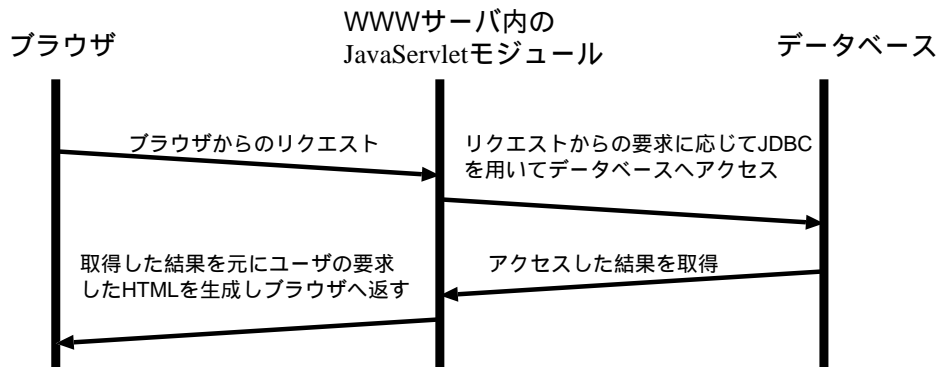


図 6.1: JavaServlet と JDBC を用いた GUI 動作フロー

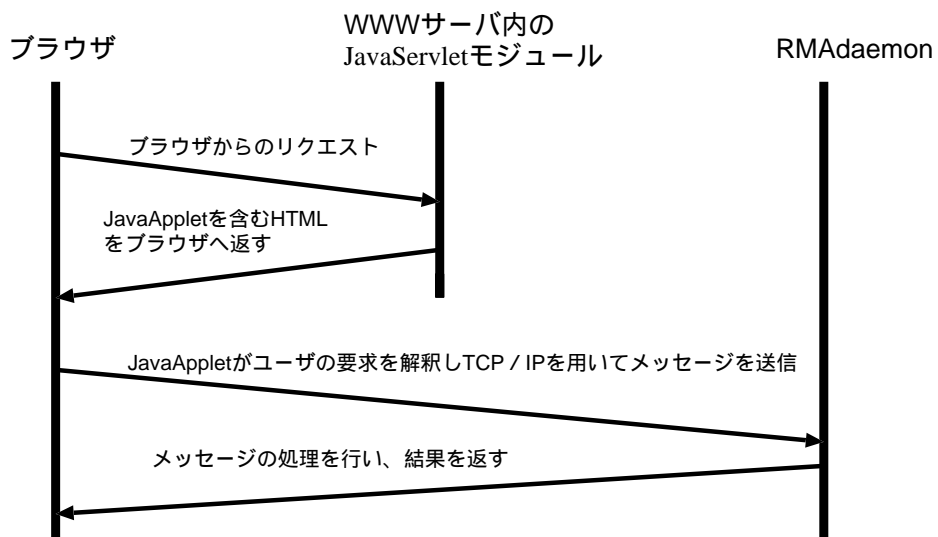


図 6.2: JavaApplet を用いた GUI 動作フロー

第 7 章

実装と動作実験

本研究で設計した資源管理エージェントの実装を行った。本章では実装の詳細とその評価について述べる。

7.1 実装システム

資源管理エージェントの実装は FreeBSD ver 3.0 上で行った。FreeBSD 上では以下の、三つのプロセスが動作する。

1. SQL daemon
2. Resource Management Agent daemon
3. HTTP daemon

以下で、それぞれにおける実装について詳しく述べる。

7.1.1 SQL daemon における実装

リレーショナルデータモデルをサポートする DBMS として PostgreSQL を用いた。PostgreSQL はデータベース言語の国際標準である SQL をサポートしており、多くのプラットフォームで動作するフリーのリレーショナルデータベース管理システムである。ソースコードも公開されているため、バグフィックス、改良が常に行われており、安定動作が得られる。また、ODBC ドライバや JDBC ドライバの提供や幅広い言語へのインターフェースの提供によってアプリケーション開発も容易である。

この PostgreSQL を使って、5 章で設計した論理モデルの実装を行った。

7.1.2 Resource Management Agent daemon の実装

このプロセスは、JAIST VideoLAN システムの DCS(Device Control Server) をベースにして実装を行った。このプロセスは資源管理エージェントの中心となるプロセスで、ユーザやターミナルシステムからのメッセージを受信し、処理を行う。その処理過程において、このプロセスは必要に応じて資源管理データベースに登録、削除、更新、検索などの処理を要求する。本研究では、このプロセスとデータベース間のインターフェースを実装した。

このインターフェースを実装するために libpq インターフェースを用いた。libpq インターフェースは C 言語を使って PostgreSQL サーバに接続し、問い合わせを行うためのアプリケーション・プログラム・インターフェースである。以下に、libpq インターフェースを用いて PostgreSQL サーバへ問い合わせの手順を示す。

1. 環境変数の設定
2. データベース・サーバとの接続
3. 問い合わせの実行
4. 問い合わせ結果の処理
5. 問い合わせの後処理

環境変数の設定では、サーバのホスト名、オプション、TCP ポート番号、デバッグ出力先、データベース名、ユーザ名、パスワードを設定する。環境変数設定後 libpq インターフェースを用いてサーバに接続する。サーバへ問い合わせを要求し、問い合わせ結果の処理を行い、メモリー領域とサーバ間の接続を解放する。

7.1.3 HTTP daemon における実装

ユーザが JAIST VideoLAN を操作するために必要な環境は WWW ブラウザだけであり、WWW ブラウザに JAIST VideoLAN 用の Web コンテンツを提供するため、資源管理エージェントは WWW サーバを動作させている。WWW サーバには Apache を用いている。Apache は NCSA の上位互換として作成された信頼性の高い堅牢な WWW サーバで、フリーソフトである。また、Apache もソースが公開されており、問題が生じた場合は即座に修正やパッチリリースが行われるため、安定した動作が得られる。

Apache はモジュールと呼ばれる各種のプラグイン機構によって、機能の拡張が可能である。図 7.1 に示すように通常のパブリッシングドキュメントを扱うにはモジュールは不要であるが、Perl などの CGI 環境、JavaServlet 環境などを構築する際には、モジュールを Apache に組み込む必要がある。

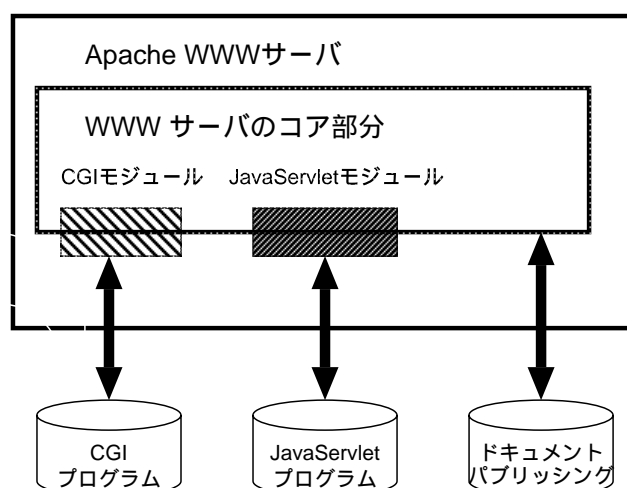


図 7.1: Apache サーバとモジュール構成

本研究では The Java-Apache Project が提供する Apache 用の JavaServlet モジュールである mod_jserv を組み込んで JavaServlet 環境を構築し、その環境上でユーザ認証サービス、ユーザ管理サービス、ネットワーク情報提供サービスのプロトタイプを実装した。DV 接続サービスを提供する JavaApplet は、メーカーから提供されているものを用いた。

7.1.4 動作実験

JAIST VideoLAN 上で実装した資源管理エージェントの動作実験を行った。動作実験としてユーザ登録サービス、DV 接続サービス、ネットワーク情報提供サービスそれぞれを行った。

サービス開始までの動作手順を以下に示す。実システムでの動作フローを図 7.2 に示す。

1. ユーザ認証フェーズ

ユーザからの認証情報とデータベースのユーザ情報をマッチングさせ、認証を行うフェーズ

2. サービス検索フェーズ

認証が確認されたユーザに提供できるサービスの情報を検索を行い、それらの情報を元にサービス選択 GUI を作成し、ユーザへ提供するフェーズ

3. サービス選択フェーズ

ユーザの希望サービスの GUI の情報を取り出し、ユーザへサービス GUI を提供するフェーズ

4. サービス開始

サービス GUI がユーザに提供されて、ユーザはサービス開始となる。

実装したプロトタイプ GUI を使って各サービスは設計した動作フローにそって動作し、サービスを提供することが確認できた。

本研究で設計した資源管理エージェントを実装し、実際のシステムにおいてプロトタイプの GUI を使ってユーザにサービスを提供した。実装したそれぞれのプロトタイプ GUI のイメージを図 7.3、図 7.4、図 7.5、図 7.6、図 7.7に示す。動作実験においてデータベースには図 7.8に示すようなインスタンスが生成され、これらの情報を元にサービスを提供している。

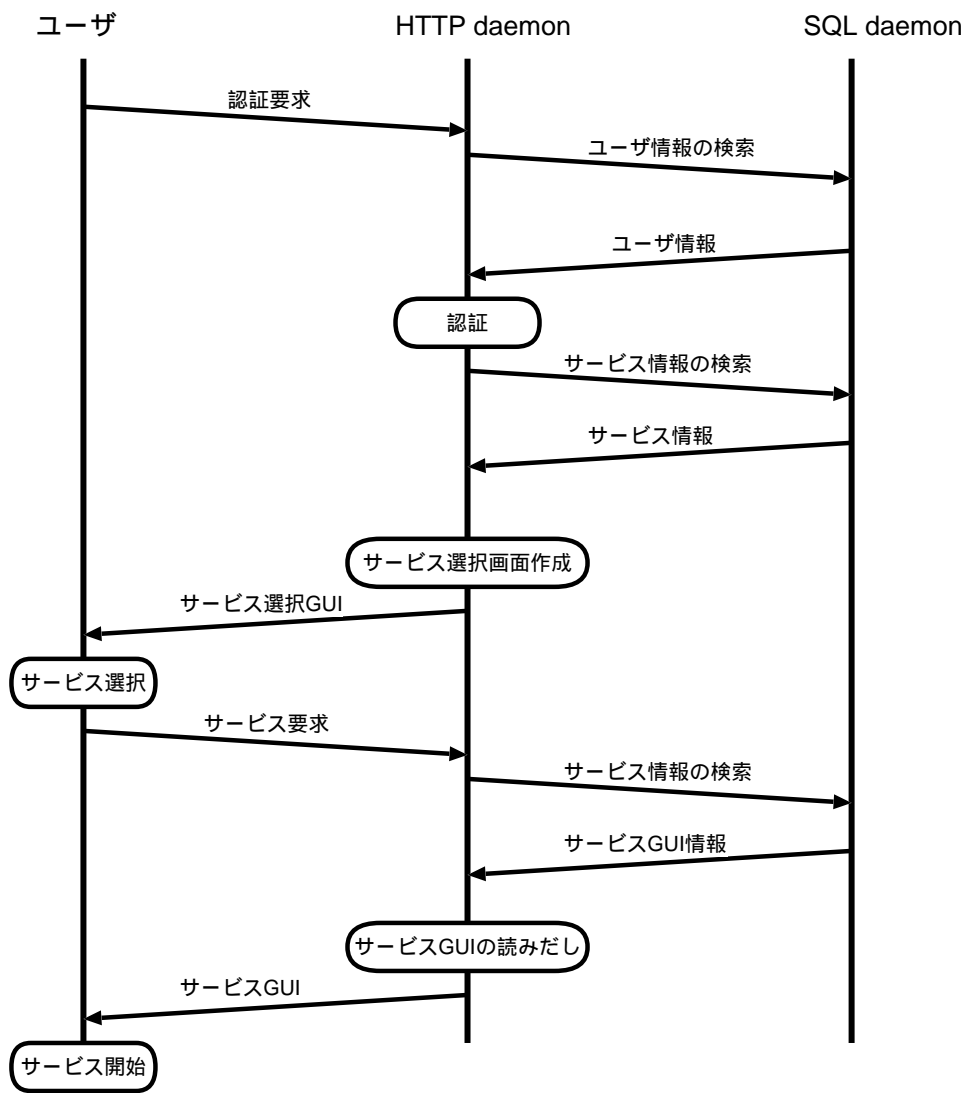


図 7.2: サービス開始までの動作フロー

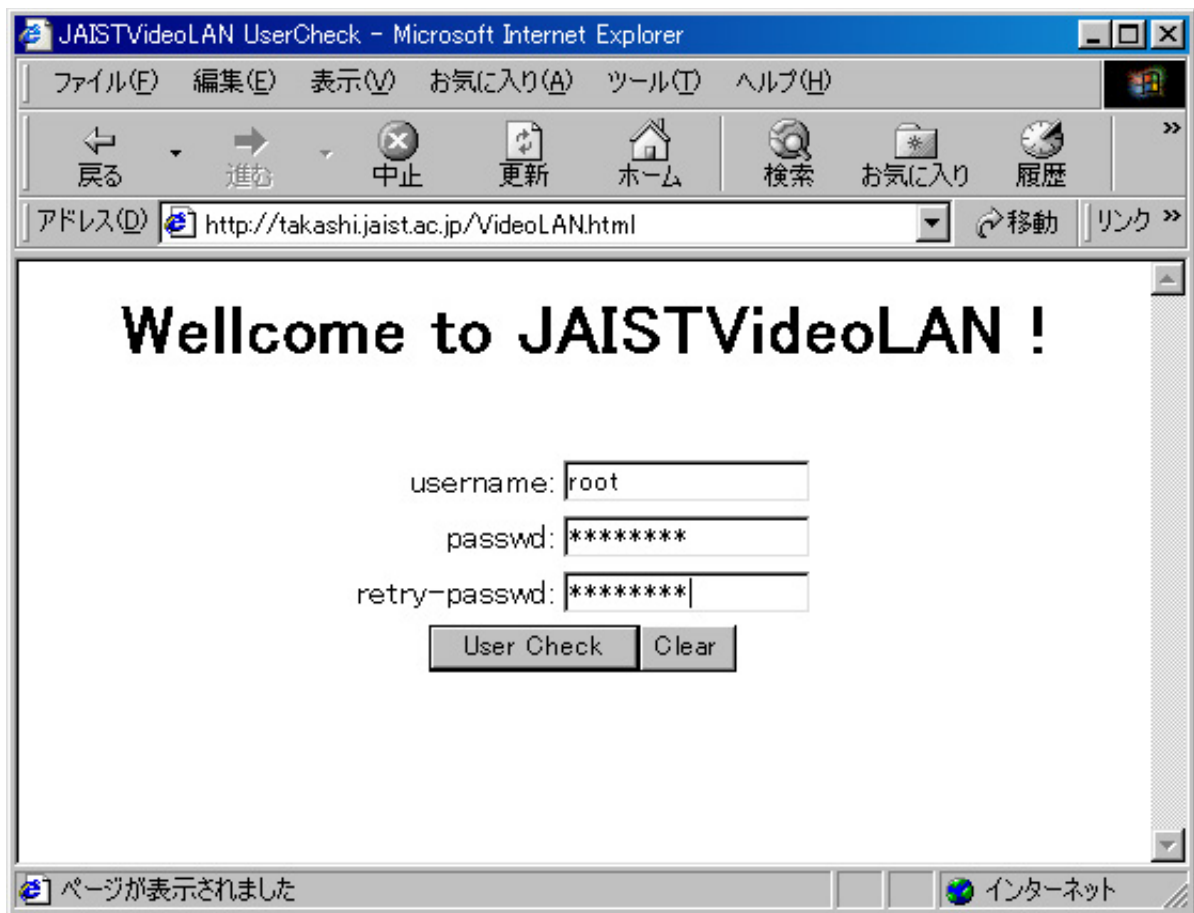


図 7.3: ユーザ認証 GUI

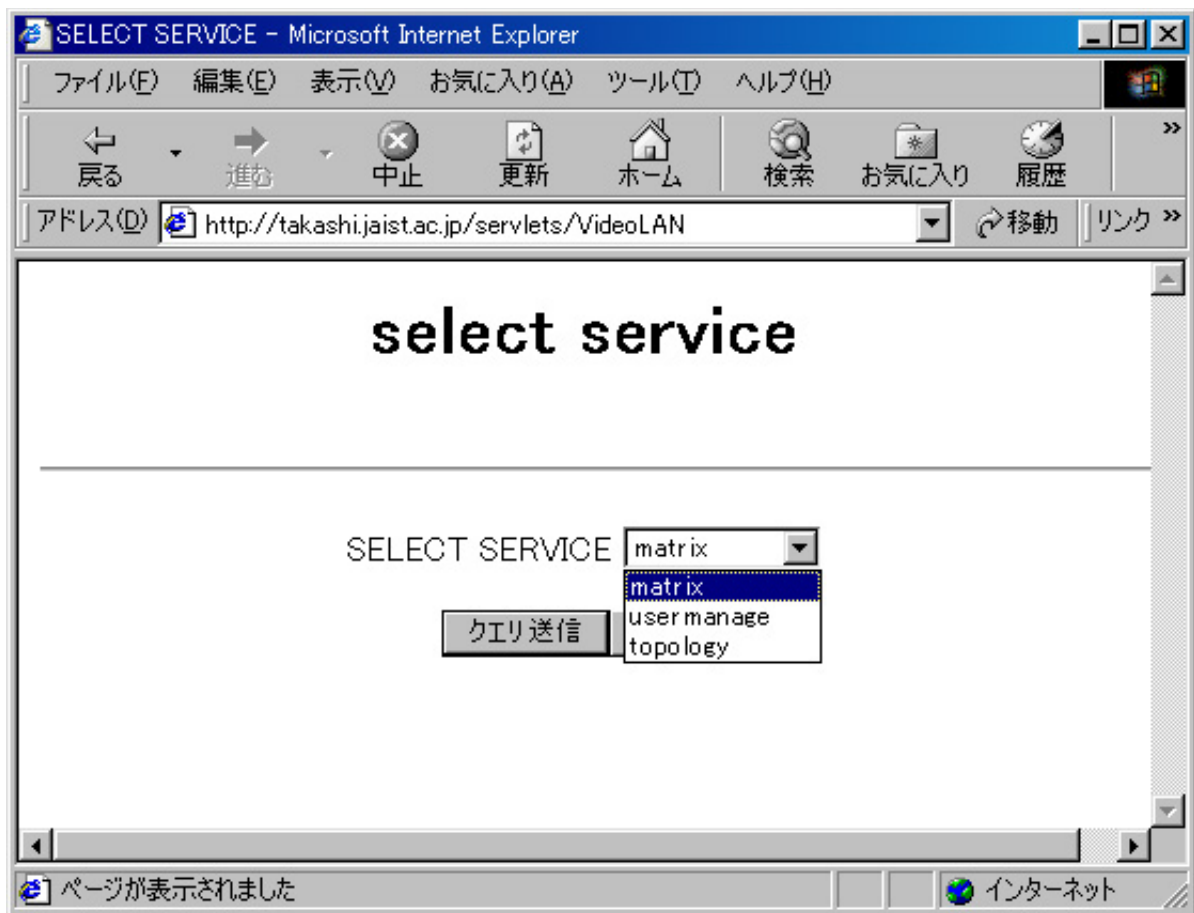


図 7.4: サービス選択 GUI



図 7.5: ユーザ管理 GUI

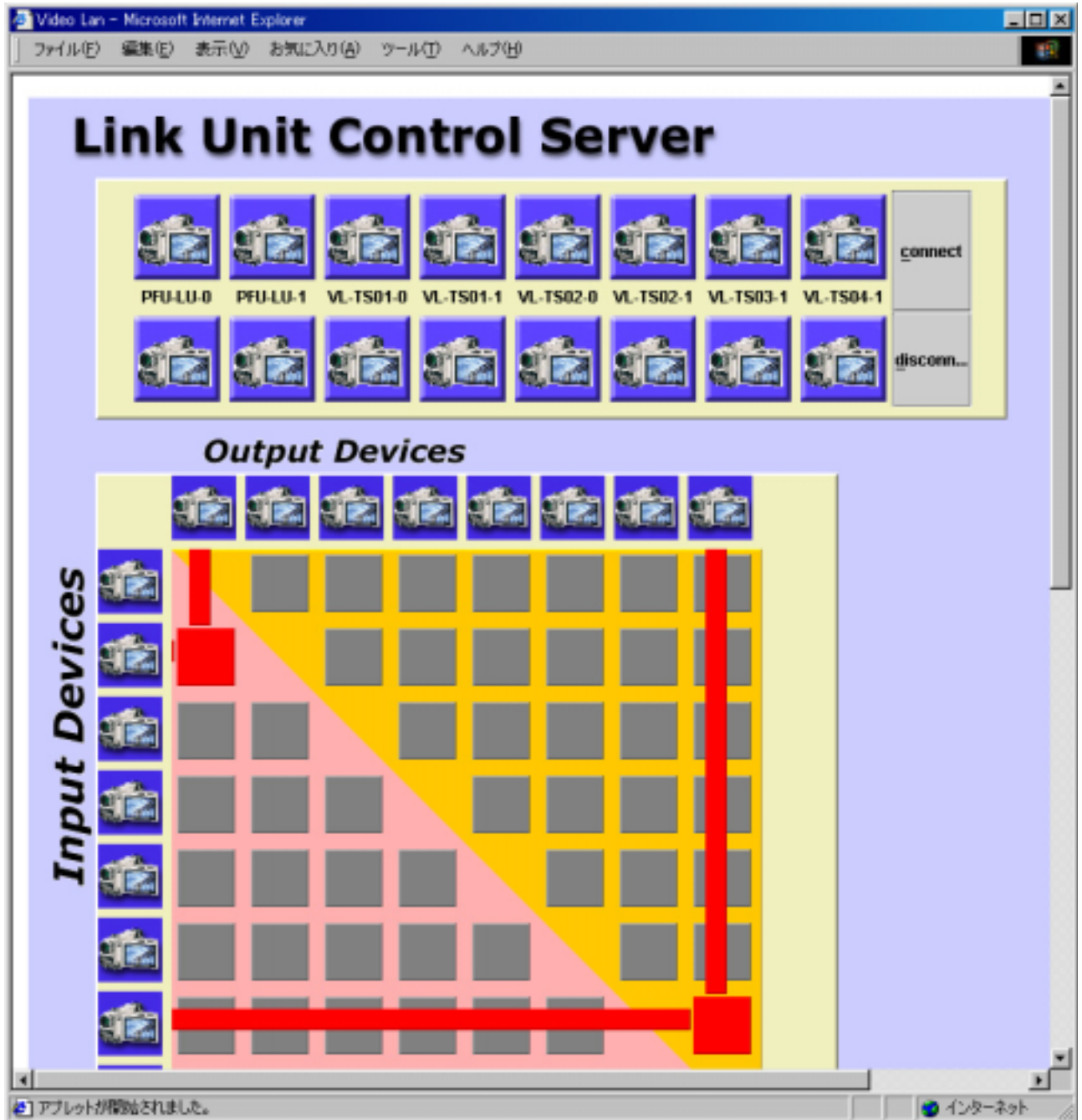


図 7.6: DV 接続サービス GUI

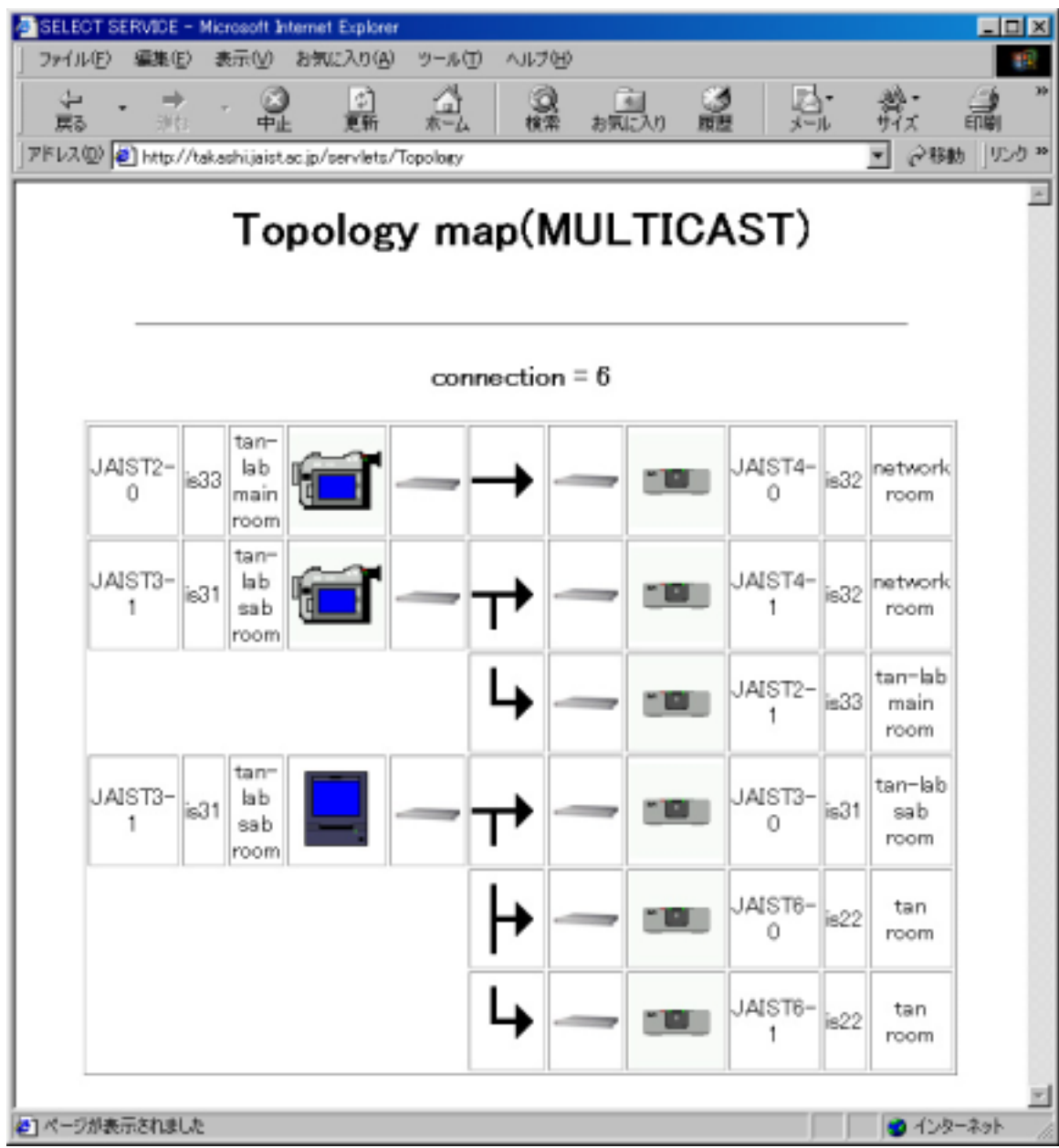


図 7.7: トポロジーマップ

user_information

username	userfullname	userid	mailaddress	user_service	comment
root	administrator	1	root@jaist.ac.jp	7	root
mtani	Masayuki Taniguchi	2	mtani@jaist.ac.jp	6	ma-chan

service

service_id	service_name	service_device	service_gui	service_comment
1	matrix	RMA	**/matrix.html	version1
2	usermanage	RMA	**/UserRegister.html	version1
3	topology	TSC	**/Information.html	version1

device(1)

service_device	device_name	building_location	room_location
TSC	JAIST4	is32	network room
TSC	JAIST2	is33	tan-lab main room

device(2)

network_location1	network_location2	device_service
47.00.05.80.FF.E1.00.00.00.F2.1A.3F.77.08.00.46.01.10.C5.	150.65.44.204	null
47.00.05.80.FF.E1.00.00.00.F2.1A.3F.77.08.00.46.01.10.D7.	150.65.44.202	null

dv_device

dv_name	dv_nuid_hi	dv_nuid_lo	parent_device	direction	serial_number
test-dv	134235649	23048179	JAIST2	IN	null
test-dv	134235651	1325184	JAIST4	OUT	null

connections(1)

connection_number	src_con_info1	src_coninfo2	src_con_info3	src_con_info4	src_con_info5
1	134234649	23048179		1	null

connections(2)

dst_con_info1	dst_con_info2	dst_con_info3	dst_con_info4	dst_con_info5	bandwidth	content
---------------	---------------	---------------	---------------	---------------	-----------	---------

図 7.8: データベースのインスタンス

第 8 章

考察

マルチメディアネットワークシステムとして JAIST VideoLAN を対象にあげ、資源情報を集中管理する資源管理エージェントを構築した。本章では構築した資源管理エージェントについて考察する。

8.1 SQLdaemon

本研究では、資源情報を管理するために、専用のデータベースではなく、DBMS を用いて実現した。DBMS を用いることで以下の利点を得られる。

- 独自の専用データベースを構築するのに比べ、安定した動作が得られる。
- 独立したデータベースをアプリケーションで共有できるので、システムの拡張性が高くなる。
- 物理的な構造を意識せず、論理的なレベルで構造を考えることができ、インターフェースだけの実装ですむため、今後の変更や拡張が容易である。

これらの利点は、システムに高い柔軟性を与えるが、システムが拡張されるにつれ、データベースへのアクセスが集中することは確実であり、アクセスの集中によってパフォーマンスの低下、システムダウンも考えられる。そのための解決策を考える必要がある。

分散データベースの技術を用いることでアクセスを分散させ、パフォーマンスの低下を防ぐことができる。また、システムダウンから早急に復旧するために、セカンダリ・データベースに資源情報をミラーリングしておき、ホットスタンバイさせておくことで耐故障化を図ることができる。

本研究では、DBMS を用いた資源管理用データベースを構築したが LDAP(Lightweight Directory Access Protocol) で実現することも可能である。LDAP はディレクトリサービスを提供する。ディレクトリサービスとは名前がつけられた情報の検索、更新などを行ってくれるサービスである。これを用いてデータベースを構築することもできる。しかし、ディレクトリサービスは以下に示すような利用形態で用いるべきでないとしている。

- ディレクトリを汎用データベースのように扱うこと
 - そのような利用形態を念頭において作られたものではない
 - トランザクションベースのサービスを提供するには不向き
- ファイルシステムとして扱うこと
 - 自分のシステムの全ファイルをディレクトリに格納しようとするのが適切ではない
- ローカルで自分自身しかアクセスしないような形態
 - ローカルで自分自身しかアクセスしないような形態では意味がない
 - いろいろな場所、マシンからアクセスされてこそ意味がある

そのため、LDAP を用いた資源管理用データベースは望ましくない。しかし、資源管理用データベース内のいくつかの情報は、LDAP を用いて実現すると効果が得られると考えられる。ユーザやサービスの情報は頻繁に更新されるものではなく、トランザクションする必要もないため、LDAP で実現すると負荷分散やパフォーマンスの向上につながると考えられる。

8.2 Resource management agent daemon

資源管理エージェントプロセスは、ユーザやデバイスからのメッセージを処理するための重要な機構である。このプロセスには本研究で設計したメッセージ、今後追加される新しいデバイス間のメッセージが実装されるため、より高機能になる。また、システムが大きくなるほど、デバイスからの要求が増え、アクセスが集中する。そのため、高機能化を図るだけでなく、耐故障化や信頼性の向上を熟慮する必要がある。現在の課題としては資源管理エージェントの多重化が挙げられている。

8.3 HTTP daemon

本研究では GUI を WWW ブラウザに提供するシステムを構築した。クライアント側、サーバ側のどちらも Java 技術を用いているため Java の能力を十分に発揮できる環境である。そのため高い拡張性を持っている。特に家庭内ネットワークのミドルウェアである Jini や Havi に対応した家電製品を JAIST VideoLAN に組み込む場合にその拡張性が生かせると考えている。Jini は Java 技術によるものであるため無論のこと、Havi も Java 技術の採用を決めている。このため、Jini や Havi 対応の家電製品コントローラを容易に利用できると考えている。また、組み込み Java チップを持つ製品が実用化されつつあり、こういった製品を JAIST VideoLAN から制御することも十分可能であると考えている。

今後の課題としては、本研究で構築したプロトタイプの GUI の改良、Jini や Havi に対応するアーキテクチャの設計が挙げられる。

8.4 資源管理エージェント

家電機器と計算機が融合されるネットワーク環境において、資源情報を集中管理する資源管理エージェントを構築した。JAIST VideoLAN において資源管理エージェントは資源情報を管理するだけでなく、サービスや GUI の提供などさまざまな役割を担っており、JAIST VideoLAN の中心的な存在である。そのため、高い信頼性を持つことの重要性を挙げ、その解決策については述べている。

今後、資源管理エージェントの課題として、サービスの充実が挙げられる。現行の JAIST VideoLAN のサービスに更なる付加価値を加えることが重要である。これを実現するための項目を以下に挙げる。

- 資源情報管理の拡張
- 異なるプロトコル体系の統合
- 異なるデータ形式の対応
- Jini や Havi が構成するネットワークとの連携

資源情報管理の拡張 現行のシステムで管理している資源情報は、JAIST VideoLAN のサービスを実現する上での設計になっている。この設計に加え、JAIST VideoLAN のサービスを柔軟に扱えるために必要な資源情報を管理していくことが重要である。例えば、こ

のシステムを利用した動画配信サービスを行うにあたって、課金機構が必要になったときに、資源管理エージェントが管理する資源情報で実現できるような、利便性の高い情報管理をしなければならない。これを実現するためには、さまざまな方向からの情報の洗い出しが必要である。

異なるプロトコル体系の統合 現在、Video on Demand・サーバやリアルビデオ・サーバなどのマルチメディアサービスを実現するシステムが存在する。これらは異なったプロトコル体系であるため、システムの目的は同じであるが相互接続できない。これらのプロトコル体系を共通する概念で統合すれば、多くのマルチメディアサービスと豊富なコンテンツをユーザへ提供することができる。これを実現するには、異なるプロトコルを変換する機構を構築することが必要である。

異なるデータ形式の対応 現行のシステムは DV データ形式だけを扱うシステムである。DV データは高品質で編集が容易であるが、広い伝送帯域が要求される。そのため JAIST VideoLAN では DV データを流すためには約 30Mbps のコネクションを確立している。しかし、視聴が目的の動画であれば DV データ形式よりも MPEG 形式の方が狭い伝送帯域で済むため効率的である。また、ユーザによっては画像よりも音声を重視した動画配信を望む事も十分考えられる。

こうしたニーズに対応していくために、異なるデータ形式を扱うためにリアルタイムなデータ形式変換機構を構築する必要がある。

Jini や Havi が構成するネットワークとの連携 家庭内ネットワークを構築するためのミドルウェアとして Jini や Havi の開発が進められている。今後、Jini や Havi に対応した家電製品が家庭内ネットワークを構築すると思われる。こうした家庭内ネットワークと JAIST VideoLAN を接続することで JAIST VideoLAN のサービスを家庭内に提供できる。また、家庭内ネットワーク間をつないだ新しいサービスなども考えられる。そのため、家庭内ネットワークとの接続も今後の課題である。

本研究では、家庭内ネットワークと JAIST VideoLAN は親和性が高いと考えている。その理由として、IEEE1394 や Java 技術の適応などが挙げられる。

IEEE1394 は家庭内ネットワークのインフラを提供する最も有力な候補であり、ターミナルシステムは IEEE1394 を搭載している。家庭内ネットワークと JAIST VideoLAN を接続する技術はすでに整っていると言える。また、Jini だけでなく Havi も Java 技術を使って、コントローラなどのソフトウェアを提供すると発表している。本研究では GUI

に Java 技術を用いているため、Jini や Havi 対応の家電製品を操作するコントローラの Java のコードを再利用できる。Jini や Havi 対応の家電製品のコントローラを構築する環境も整っていると言える。

ネットワークインフラと GUI の環境が整いつつあるため、今後は、家庭内ネットワーク資源の情報を扱うためのフレームワークおよび、新しい資源情報を元にした新サービスの検討をすることが必要である。

第 9 章

おわりに

本研究では、家電製品と計算機が融合する新しいネットワーク環境において問題となる資源管理について検討し、資源を集中管理する資源管理エージェントを提案した。

はじめに、資源管理エージェントを構成する 3 つのコンポーネントをそれぞれ設計した。

資源管理エージェントプロセスにおいては新しいデバイスである中間ノード間のメッセージとその動作フローの設計を行った。

JAIST VideoLAN を対象として概念設計、論理設計を行い資源管理用データベースの論理モデルを設計した。

WWW ブラウザに GUI コンテンツを提供するための検討を行い、Java 技術を用いたサービス GUI の設計を行った。

設計した資源管理エージェントを実装し、その動作評価を行った。最後に、動作実験および今後の課題点について考察を行った。

謝辞

本研究を進めるに当たり、終始御指導を賜りました、丹康雄助教授に心から深く感謝致します。また、適切なお助言をして頂きました篠田陽一助教授に深く感謝致します。

さらに、苦勞を共にした、木村範彦君、谷口雅幸君、中田潤也君に感謝致します。

最後に、貴重な御意見、御討論を頂きましたソニー株式会社関係者方々、丹研究室の皆様および植松氏の御援助に対し、厚く御礼申し上げます。

参考文献

- [1] 丹 康雄, "JAIST Video LAN - 実世界指向マルチメディアネットワーク", 人工知能学会 FAI 研究会, SIG-FAI-9802, 1999
- [2] 丹 康雄、野村 隆、田守 寛文, "家電的ユーザーインタフェースを有する大規模マルチメディア LAN システム", 情報処理学会 Interaction'99, 1999.
- [3] Yasuo Tan, "Scaling up IEEE 1394 DV Network to an Enterprise Video LAN with ATM Technology", IEEE International Conference on Consumer Electronics '98, 1998.
- [4] Takashi Nomura, Tetsuaki Kiriya, Hiroshi Yamamoto, Atsushi Maruyama, Hiroshi Takizuka, "New Protocol Architecture ASEL", IEEE International Conference on Consumer Electronics '98, 1998.
- [5] H. Tamori, "The AV Controller on PC Using AV Plug", IEEE International Conference on Consumer Electronics '98, 1998.
- [6] "JiniTM Architecture Specification. Revision 1.0", Sun Microsystems, Inc, 1999.
- [7] 木村範彦, 丹 康雄, "マルチキャスト通信プロトコルにおける中間ノードによる資源の有効利用法", 情報処理学会, 第 59 回 全国大会, 1999.
- [8] 倉岡貴志, 丹 康雄, "AV 系ネットワークシステムにおける資源管理法に関する一手法", 情報処理学会, 第 59 回 全国大会, 1999.
- [9] 北川 博之, "データベースシステム", 商晃堂, 1996.