

## 6-2 分散協調型ホームネットワークサービス構築基盤

### 6-2 *Distributed and Cooperative Service Platforms for Home Network Services*

山崎達也 沢田篤史 西村俊和 高岡真則 多鹿陽介 美濃導彦  
YAMAZAKI Tatsuya, SAWADA Atsushi, NISHIMURA Toshikazu, TAKAOKA Masanori,  
TAJIKI Yosuke, and MINOH Michihiko

#### 要旨

ゆかりプロジェクトでは、家庭内の情報家電のみならず、将来の普及が見込まれるセンサ・ロボットなども対象に、機器内の各機能を協調させてサービスを提供する、分散協調型ホームネットワークサービス構築基盤の研究開発を行ってきた。そして、主たる研究成果として基盤を構成する2種類のミドルウェア「ゆかりコア」「ゆかりカーネル」の開発を行った。ゆかりコアは機能連携のメカニズムの必要最小限の部分をミドルウェア化したものであり、ゆかりカーネルはより柔軟なユーザ適応のための機能追加を行ったものである。これらの基盤を様々な実機のアプライアンスに実装し、機能協調によるホームネットワークサービスを実環境で構築することにより、提案基盤技術の有効性を実証することができた。

In the UKARI (Universal Knowledgeable Architecture for Real-lffe appliances) project, we studied and developed distributed and cooperative service platforms for home network services, which provide services by cooperating functions from networked appliances (NAs). The NAs include consumer appliances as well as sensors, robots, and so on. Main research results were "Ukari-Core" and "Ukari-Kernel", which constitute the platforms. UKARI-Core is a middleware with must and core modules of distributed function cooperation. UKARI-Kernel is, moreover, another middleware as an extension of UKARI-Core with additional functions for flexible user adaptation. We implemented the middleware into various kinds of real appliances and constructed home network services in a real environment to evaluate the efficiency of the proposed service platforms.

#### [キーワード]

ネットワーク家電, ホームネットワーク, 機能協調, ミドルウェア  
Networked appliance, Home network, Function cooperation, Middleware

### 1 まえがき

ハードウェア・ソフトウェア技術の発展に伴い、家の中にある様々な機器が多機能化・高機能化され、単体としての付加価値は確実に高まってきている。一方、家電機器のデジタル化及びネットワーク化も年々進み、情報家電をネットワーク化することによりホームネットワーク環境を構築していこうとする活動も行われている。ホームネットワークが提供するサービスとして、機器間

でのコンテンツのやり取りや遠隔からの機器操作が実現されてきているが、サービスが一つのメーカーの機器だけに閉じていたり、各分野の決まった系統(例えば、白物家電系やオーディオ・ビジュアル(AV)機器)内でしか利用できなかったりといった現実の課題があり、今後の標準規格化が望まれている。

そのような状況の下、現状では利用者が進化する技術に適応・同化することにより技術の進歩の恩恵を受けているが、それでも付加された機能を

すべて利用するようなことはほとんどなく、逆にそこに搭載されていない他の機器の機能を借りて使いたいというような場合がある。また、一つの家庭の中に同類の機能が重複してある場合には、不具合の発生時等に一時的に代替手段としてその機能の貸し借りができることと便利である。ゆかりプロジェクト<sup>[1]</sup>ではこれらの要件を満たすために、機器の持つ一つ一つの機能を分解し、ネットワーク経由で各機能を単位として利用することを可能にする、分散協調型ホームネットワークサービス構築基盤に関する研究開発を行ってきた。機器間の機能連携を行うプラットフォームとしてまず開発したのが「ゆかりコア」であり、ゆかりコアを更に拡張し、より柔軟にユーザ適応できるサービスアプリケーションを可能とするための機能を追加したものが「ゆかりカーネル」である。本稿ではこの二つのホームネットワーク基盤について述べる。なお、ゆかりコア、ゆかりカーネルは幅広くネットワークに接続される様々な機器(アプライアンス)を対象としており、それらを総称してNA (Networked Appliance)と呼ぶこととする。また、一つの機能が他の機能とは独立した形で決められたインタフェースで外部と接続される時、その機能をFE (Function Element)と呼ぶものとする。

## 2 関連研究と提案方式の位置づけ

ネットワーク上に分散した機能を連携させるシステムに関して、これまでも様々なものが提案されており<sup>[2]–[7]</sup>、ホームネットワークを指向した研究開発<sup>[8]</sup>や標準化<sup>[9]</sup>も進行中である。分散オブジェクト技術やWebサービス技術の進歩により、ネットワーク経由で公開されたコンポーネントを相互利用する基盤は整いつつあるといえるが、これらのコンポーネントを利用してサービスを構築するためには、あらかじめコンポーネントのインタフェース仕様の詳細を知っておく必要がある。例えば、UPnP (Universal Plug and Play)<sup>[6]</sup>やECHONET (Energy Conservation and Homecare Network)<sup>[9]</sup>などのサービス解決技術では、要求に応じて必要なサービスオブジェクトを発見することはできるが、あくまでも実機器を単位とした接続を規定しているため、機器を連携させたアプ

リケーションを構築するには、発見された個々のインスタンスのインタフェースを詳細に把握してプログラムを構築する必要がある。また逆に、コンポーネントを構築する側でも、特定のアプリケーションを想定したインタフェース設計を行わざるをえない場面もあり、再利用性向上の阻害要因となっている。ゆかりコア、ゆかりカーネルでは、このような要求や問題点を解決するために、機器単位でなく機器の持つ個々の機能単位で連携させることができるネットワーク接続基盤を提供している。

また、将来は家電のみならずセンサやロボットなどもホームネットワーク上で互いに連携して、家庭環境で状況に適応したサービス提供を行っていくものと思われる。このような将来普及すると考えられる機器にも容易に対応するために、ゆかりコア、ゆかりカーネルでは機能の類型化・抽象化と、個別に特性を記述できる詳細情報の記述を巧みに組み合わせて機能発見等を行うフレームワークを備えている。以下でこのフレームワークについて述べる。

まず、FEで扱われるメディアの種別と、メディアに対する操作を二つの軸としてFEのタイプを表1のように類別する。ここでメディア種別はTrue/Falseを表す論理(Boolean)、数値(Numeric)、文字(Text)、音声(Audio)、画像(Image)、映像(Movie)に分類され、操作タイプは、生成(generate)、消費(consume)、合成(mix)、変換(transform)、蓄積(store)に分類される。このようにFEのタイプを簡潔な表記で大別することにより、FEをカテゴリごとに速やかに発見することを可能としている。

表1では、FEタイプは2文字か3文字のアルファベットあるいは数字の組合せで表されることを示している。表1中のすべての組合せに対して、最初の大文字アルファベットがメディア類別を示していることは共通であり、生成、消費、蓄積に関しては2文字目の小文字アルファベットが施される操作タイプを表している。合成に関しては2文字目が合成を表すmとなっており、その後の3文字目が合成されるメディアの数を示している。例えばAm2は二つの音声メディアを合成するFEであることを表している。変換に関しては、2文字目が変換後のメディア種別を表しており、

表1 FEタイプの分類

メディア種別	操作タイプ				
	生成 (g)	消費 (c)	合成 (m)	変換 (t)	蓄積 (s)
論理 (B)	Bg	Bc	Bmk	BXt	Bs
数値 (N)	Ng	Nc	Nmk	NXt	Ns
文字 (T)	Tg	Tc	Tmk	TXt	Ts
音声 (A)	Ag	Ac	Amk	AXt	As
画像 (I)	Ig	Ic	Imk	IXt	Is
映像 (M)	Mg	Mc	Mmk	MXt	Ms

3文字目に変換の操作タイプを示すtがつけられている。例えばATtは音声メディアを文字メディアに変換するFEであることを表している。

FE間の協調連携を行うためには、表1に示されるFEタイプの分類では粗すぎるため、次にFEを連携させるために詳細情報をやり取りする必要がある。詳細情報が記述されているのがFECAPであり、例えば画像のメディア種別に対しては、コーディング方式やサイズなどの情報が記述されている。すなわち、まず表1で分類され

るFEタイプでサービスの構成要素となり得る機能群を絞り込み、より詳細な情報はFECAPを相互にやり取りすることでサービス構築に実際に使える機能が決定される。例えばホームネットワーク上に数十個のNAがあり、各NAが平均5個のFEを提供できるとした場合でも、機能連携サービス構築の際に検索の対象となるFEの数は数百のオーダーになってしまう。この場合すべてのFEから膨大なFECAPを取り寄せ判定することは現実的ではなく、提案方式に基づきFEタイプで大きくFEをふるい分けて選別した上で、FECAPの詳細情報を用いてサービス構築に利用可能なFEを決定する手法の方が効率的で有効であると思われる。

### 3 ゆかりコア

#### 3.1 全体構成と基本モジュール

図1にNAが家庭内ネットワークを通じて相互に接続され、単機能同士で協調動作をするためのシステムの全体構成を示す。各NAは幾つかの実際のハードウェアに対応した機能を有し、それぞれの機能に対して外部ネットワークとのインタフェースとなるソフトウェアオブジェクトとしてのFEが存在する。そして、我々の提案するホームネットワーク基盤では、NAはサービスのセットアップと制御を行うユニシエータと、サービスに必要な機能を提供するレスポндаに分けられ

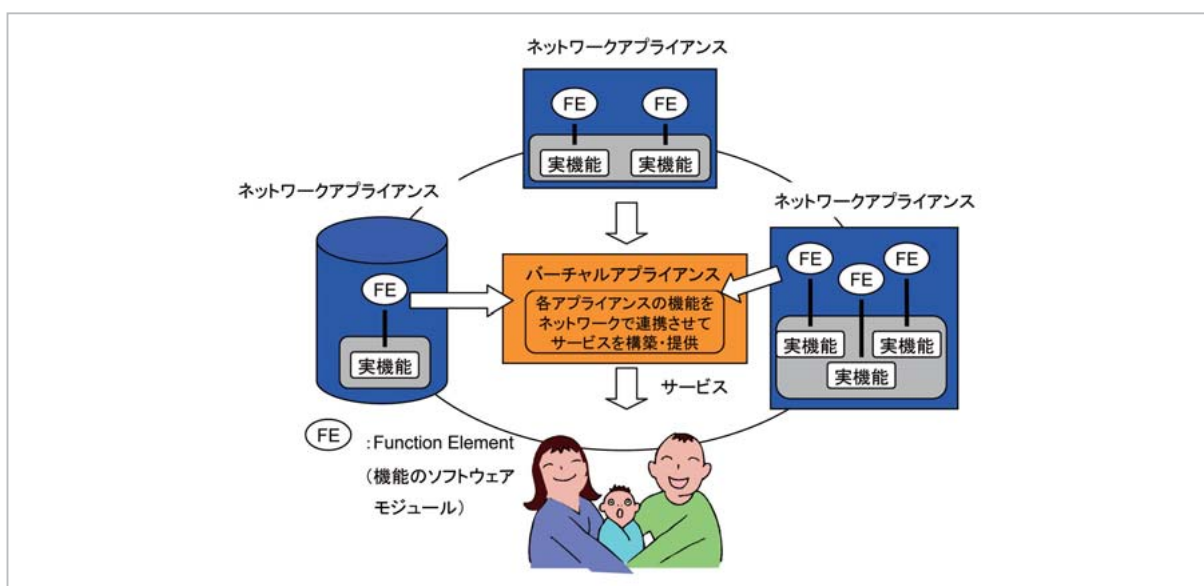


図1 ゆかりコアシステムの全体構成

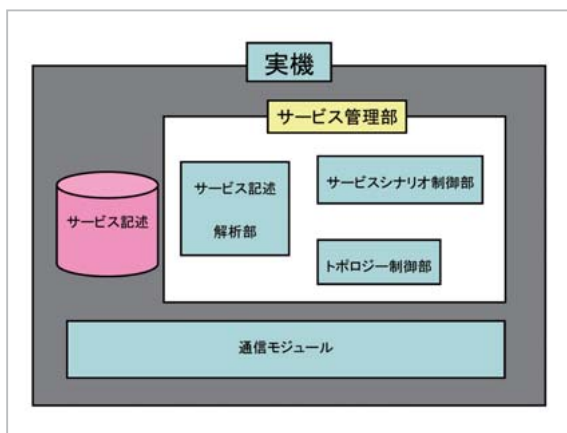


図2 イニシエータの構成

る。ただし、イニシエータは物理的に機器ごとに独立しておく必要はなく、一つの機器内にレスポンドとともに共存させることも可能である。

図2にイニシエータの基本モジュール構成を示す。イニシエータ内の主要なモジュールはサービス管理部であり、これは主にサービス記述解析部、トポロジー制御部、サービスシナリオ制御部から成る。

サービス記述解析部は、XML (eXtensible Markup Language) で記述されたサービス記述を解釈するモジュールであり、サービス記述は(サービス)シナリオ記述とトポロジー記述から構成されている。トポロジー制御部はトポロジー記述に基づいて通信モジュールを介して必要な機能を発見し、その間のネットワークパスを設定するモジュールである。サービスシナリオ制御部はシナリオ記述に基づいて各FEに対して制御メッセージを送信し、返ってくるイベントを受信するモジュールとなっている。

図3にレスポンドの基本モジュール構成を示す。レスポンドは主にNA自体を管理するNA Manager と、そのNAが提供するFEを管理するFE Manager から構成される。FE ManagerはNAが提供するFEと一対一に存在する。

NA Manager 内のNA記述解析部では、XMLで記述されたNA記述を解釈し、イニシエータからのリクエストに対して、そのNA記述により対応するFEがあるかないかの判定を判定部で行う。

FEM 管理部は複数のFE Manager を管理するモジュールであり、FE Manager 内のモジュール

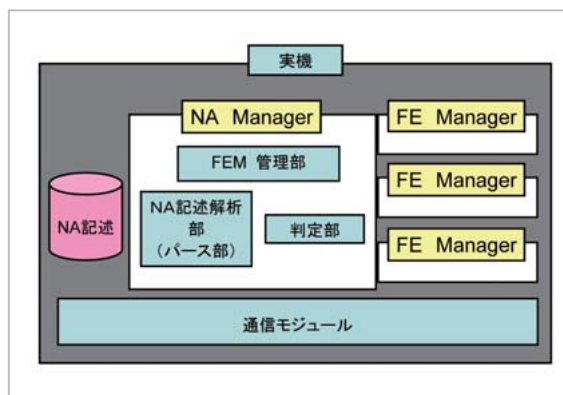


図3 レスポンドの構成

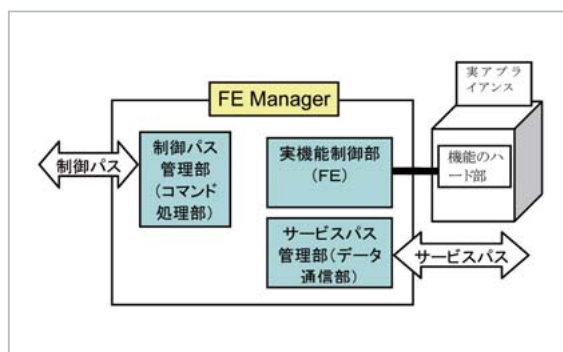


図4 FE Manager の構成

構成は図4に示すとおりである。FE Manager 内の実機能制御部は、実アプリケーションの機能ハード部のプロキシーを行うモジュールであり、制御バス管理部は制御メッセージ(コマンド)を送受信するネットワークパスを管理するモジュールとなっている。サービスバス管理部はFE間を接続し、データの送受信を行うネットワークパスを管理するモジュールである。

### 3.2 動作プロトコル

以下で、動作プロトコルに関して説明する。まずイニシエータはサービスを構成する機能をネットワーク上で発見しなくてはならない。機能発見のためにネットワーク上に送信するメッセージをBidと呼ぶ。Bidはあるネットワーク範囲のレスポンドに対して出されるが、Bidを受けたレスポンドがもし当該の機能を有していた場合は機能提供を申し出る返信メッセージを、Bidを送信したイニシエータに返す。このメッセージをOfferと呼ぶ。その後、イニシエータは、詳細情報を記述したFECAPという記述要素の固まりを、Offer

を返してきたレスポндаに要求し、レスポндаはそれに応じて当該 FECAP を送信する。ゆかりコアではこの FECAP 送受のやり取りを省けるように、一定期間イニシエータが一度受信した FECAP を保持(キャッシュ)できるようにしている。また、イニシエータは一つ以上の Offer が返ってきた場合、一つのレスポндаを選択しなければならないが、現在はこの選択はアプリケーションの実装にゆだねている。最終的に、機能提供をするレスポндаが決定されると、イニシエータはそのレスポндаに同意するメッセージを返す。このメッセージを Agree と呼ぶ。また、選択しなかったレスポндаには Disagree のメッセージを返す。

サービスを構成する機能が複数ある場合は、所定のすべての機能を見つけるように上記の手順を繰り返し、サービスを構成する所定のすべての機能が見つかった場合、イニシエータは機能を提供する FE 間を接続しデータを送受信する、サービスパスを構築するためのメッセージを各レスポндаへ送信する。以上がサービスを構築するための、機能発見及び制御/サービスパスの構築の一連の流れとなっている。

## 4 ゆかりカーネル

### 4.1 全体概要

前節で説明したゆかりコアは、サービスの構成要素となる FE をサービス記述に基づき発見し、FE 間の相互接続を可能とすることで、機能協調のための基本的なメカニズムを与えている。しかしながら、様々な機器が混在するホームネットワーク環境においてユーザに適応したサービスアプリケーションを効率良く開発・提供する基盤として、ゆかりコアの単純な FE 発見と協調の機能だけでは不十分であるため、以下の(1)から(3)の3点においてゆかりコアの拡張を行い構築した機能協調プラットフォームがゆかりカーネル[10]である。

- (1) ユーザ別の権限による NA・FEへのアクセス制御
- (2) アプリケーション構造だけでなく振舞いを記述する枠組
- (3) 各種センサのデータを精細に表現するため

### のデータ型

それぞれの拡張に関して簡単に列挙する。

(1)のアクセス制御機能については、レスポндаの持つ NA Manager がアクセス制御リスト(ACL: Access Control List)により各 FE への接続を制御するよう拡張を行った。

(2)の振舞い記述については、ゆかりコアではサービス記述と呼ばれていた FE 間の接続関係記述をトポロジー(topology)記述と言い換え、新たにサービスアプリケーションの振舞いを形式的に表現するためのシナリオ(scenario)記述を導入した。その上で、このシナリオ記述を解釈しサービスの実行を制御する機能を新たにイニシエータに実装した。ゆかりカーネルでは、トポロジーとシナリオ両者の記述を併せてサービス記述と呼ぶ。

(3)については、多種多様なセンサから出力されるデータを表現し、FE 発見時にサービスアプリケーションの目的に適応したセンサ FE の選択を可能とするため、SI 単位系などを用いた標準的な数値表現に加え、複合データ型などの追加定義を可能とするよう NA 記述の仕様改訂を行った。

以下の節で、ACL とシナリオ記述に関する概要を述べる。

### 4.2 アクセスコントロールリスト

ゆかりカーネルでは、NA・FE へのアクセス権限をユーザ別に設定し、プライバシー保護あるいはリソース競合時の解決手段として用いるため、アクセス制御を実現するアプローチを ACL として導入した。ゆかりカーネルで実現するホームネットワークサービス上のアクセス制御を実現するため、ACL は以下の要件を満たす必要がある。

- 各機器・サービス等の優先順位と利用可否といった利用ポリシーを表現する記述でなければならない。
  - プライバシ保護の観点から、利用者ごとに利用可否が記述できなければならない。
  - 同じ機器内の複数の機能がそれぞれ異なったサービスで利用されるため、記述単位は機器ごとではなく機能ごとであるべきである。
  - ゆかりカーネルで実現されるサービスは、機能の組合せによって構成されるので、サービスごとの優先度を準備する必要がある。
- これらの要件を満たすため、ゆかりカーネルに

おける ACL では利用者、機能、サービスそれぞれに固有の識別子 (ID) を与え、それぞれの組合せでのサービスの利用可否、あるいはサービスの優先度合を数値化し、記述している。すなわち、ACL は利用者 ID、サービス ID、機能 ID、優先度の四つ組で示され、利用者 ID、サービス ID、機能 ID の組合せを問い合わせキーとすると、それに応じて利用可否や優先度を返答として得ることができる。

ACL を用いた優先度制御の方法は幾つか考えられる。一つは優先度の値が大きい組合せを、そうでない組合せよりも優先してサービスを行うという方法であり、また機器利用禁止であることを著しく優先度が低い状態であるとみなすことにより、非常に強いアクセス制限をかけることもできる。より詳しい優先度制御、ACL を利用するためのユーザインタフェース、具体的な利用形態等に関しては、文献[11]を参照されたい。

### 4.3 シナリオ記述

ゆかりコア及びゆかりカーネルにおける FE の協調は、特定のオブジェクト間の密結合ではなく、抽象的なインタフェースを公開しながら、まばらな関係で作用し合うという特徴を持つ。このため、サービス指向の考え方でとらえるのがより適切と判断し、振舞い記述の枠組として、BPEL4WS[12]に基づく XML ベースのシナリオ記述方式[13]を設計した。図 5 にゆかりカーネルにおけるシナリオ記述の概要を示す。図 5 中央部の網掛け部分が一つのサービスアプリケーションに対するシナリオ (BPEL4WS では process タグにより表現される) に相当し、ゆかりカーネルで新たに追加された枠組である。シナリオ記述はイニシエータにおいて解釈実行される。そこでは、アクティビティ (activity) とアクティビティ間の制御依存関係を表すリンク (link) により、制御の遷移手順を記述することができる。また、シナリオと外部とのデータのやりとりの結果得られるデータは変数 (variable) に格納され、プロセス内の他のアクティビティによって参照可能となる。

図 5 右部に示す FE インタフェース (FEI) は、シナリオと FE との間でやりとりを行うためのポートである。FE インタフェースはトポロジー記述の要素である FE と一対一に対応して定義さ

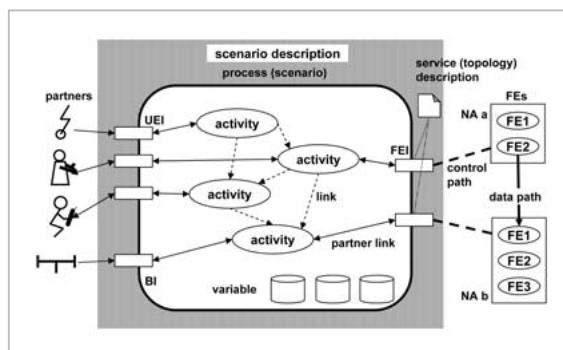


図5 ゆかりカーネルにおけるシナリオ記述

れ、実際にサービスへ参加する FE の制御を FE 発見・接続、FE 機能の実行、FE 間のデータ通信の各面から行うためのインタフェースを提供する。例えば、インタフェースに定義された FE 発見・接続のためのオペレーションを呼び出すことで、トポロジー記述内の該当する型の FE を発見して制御パスの確立を行うことができ、機能実行のためのオペレーションを呼び出すことで FE 機能の実行開始・終了・中断などを行うことができる。また、データ送受信に関するオペレーションを呼び出すことで、FE 間のデータパスを経由したデータ通信を行うことができる。

一方、図 5 左部のユーザ・環境インタフェース (UEI) は、FE に分類することのできないサービス要素とのやり取りを行うためのポートの役割を果たし、主にユーザや外部サービスとの間で制御やデータを受け渡すために用いられる。UEI にはポートを通じた制御やデータ交換のためのインタフェースが自由に定義できる。

また、同じく図 5 右部に示すブローカインタフェース (BI) はエンドポイントを指定することなくメッセージを交換するためのポートであり、イニシエータがあらかじめ関心のあるメッセージタイプをブローカに登録することで、該当するタイプのメッセージが流れた時に通知を受けることができる。BI によって、一つの機器やセンサが発生するイベントから複数の異なるシナリオが発火するといった形態のアプリケーション提供が可能となる。

ゆかりカーネルのシナリオ記述言語では、プロセスの構成要素となるアクティビティの種類を表 2 に示す八種類とした。BPEL4WS との相違点

表2 アクティビティの種類

receive	FEI, UEI, BI を通じてデータを入力
reply	FEI, EI, BI を通じてデータを入力
invoke	FEI, EI, BI を通じてオペレーションを起動
assign	変数にメッセージの内容を代入
terminate	シナリオの実行を停止
pick	FEI, UEI, BI を通じて入力されたデータ種に応じたアクティビティに遷移
findNA	FEI を通じ、キーワードなどを指定してNAを発見
findFE	FEI を通じ、トポロジー記述上のFEを発見

として、FE 発見・接続に必要なインタフェースを FE インタフェースの要素として暗黙に定義している点、エンドポイントを指定しない通信形態をブローカインタフェースとしてサポートしている点、pick 以外の構造要素をシナリオから排除し解釈実行系の論理を単純・軽量化することで非力な組み込み系機器での実装をねらっている点が挙げられる。

## 5 実装例

我々はゆかりコア・ゆかりカーネルを、提案するホームネットワーク基盤上のミドルウェアとして開発し、様々な家電・センサ・ロボット等の実機へ実装することにより、実ホームネットワークにおけるサービス構築の実証を行ってきた。その結果、ネットワーク上での機能発見、機能連携を行い、機器単体では提供できないサービスも、複数の既存の機器の機能協調により実現することができることを確認してきた。本稿では、代表的なものとしてゆかりコアにより実現した例を示す。なお、ゆかりコア、ゆかりカーネルはオープンソフトウェアとして研究開発目的に広く公開している[14]。

実装システムの構成を図6に示す。ゆかりコアは、実機のデジタルカメラ、プラズマディスプレイテレビ、冷蔵庫、ドアホン、焦電(人感)センサに実装されている。ドアホンはボタンのみのOn/Offスイッチで、On/OffイベントはZigBee(ZigBeeはKoninklijke Philips Electronics N.V.の登録商標)を用いたセンサーネットワークを通じて、ゆかりコアのアダプタボードに直接収容され

ている。また、焦電(人感)センサは、焦電型赤外線方式のもので、5m以内の特定角度内にいる人など赤外線を放出するものの存在を検知するものである。検知した場合、イベントとしてゆかりコアのアダプタボードに通知される。実装システムの概観を図7に示す。

デジタルカメラ、プラズマディスプレイテレビ、冷蔵庫が本実装システムではレスポنداに相当し、そこに実装した実行モジュールの構成を図8に示す。また、実機の機能ハード部は機能別に単位化されており、それらに対して制御及び通知を行うモジュールが実装してある。具体例としては、プラズマディスプレイテレビでは文字(テキストデータ)を表示する機能を単位化し、レスポنداのFEとの結合を行ったことがこの実装に相当する。サービス記述にも実アプライアンス機能を表

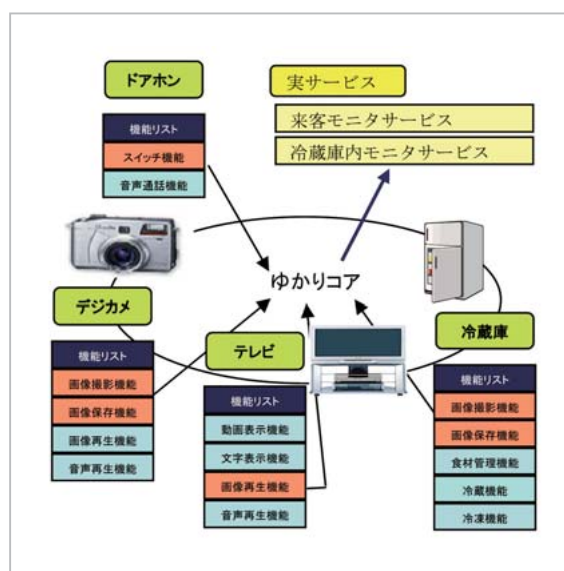


図6 実装システム構成

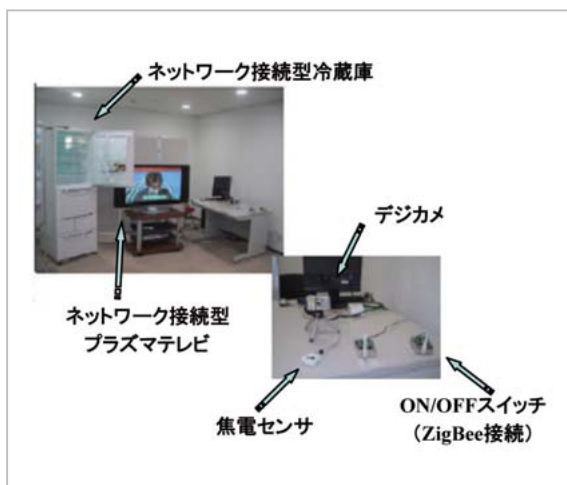


図7 実装システム概観

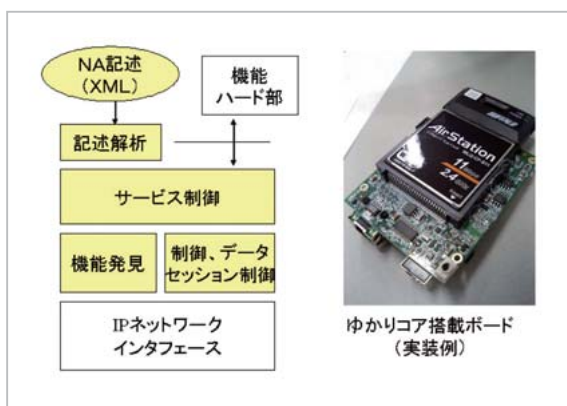


図8 実装モジュール構成と実装例

現した上で、実装システム上での動作を検証した。プログラム自体は、C 及び C++ にて記述し、XML 解釈用のパーサ組込みを含め、IP ネットワーク上のプログラムとして、Linux 環境上で動作させた。

我々は、開発したミドルウェアの実環境での検証を含め、リアルな家庭の生活環境で情報通信技術の有効性を実証するため、実生活型実証実験テストベッド「ユビキタスホーム」を2004年4月より構築している[15]。2005年度の1年間で、このユビキタスホームにおいて、構成の異なる家族4組と研究関係者1組の計5組を被験者として、生活実証実験を行った。これらの実験では、それぞれの被験者グループが約2週間ずつ、食事や睡眠・通勤などを含め、それまでの日常と同様の生活パターンでユビキタスホームで実際に生活しつつ、その中で用意された生活支援サービスを継続

的に利用し、サービスの評価を行っている。我々はサービス利用に関して実ユーザからアンケート・評価用紙への回答・インタビュー等のフィードバックを得るとともに、各種センサで実生活の行動データを取得することができた。

この生活実証実験の中で、無線タグでユーザの位置を把握した上で、最も近いところにある音声メディア生成アプライアンス(通常はスピーカ)を発見し、データパスをその場で構築した上で洗濯機の終了メッセージをユーザに通知するというサービスを、ゆかりコアで実施した。約2週間の実験期間中、このサービスは安定して動作し、ユーザからも「家が(ユーザの)いる場所が分かって教えてくれるので、びっくりした。便利に思う」というようなコメントを得ることができた。

## 6 むすび

ゆかりプロジェクトで研究開発してきた、NAの各機能を単位としてネットワークで協調連携させることによりホームネットワークサービスを構築する、分散協調型ホームネットワークサービス構築基盤の概要を紹介した。機能連携のメカニズムの必要最小限の部分をミドルウェア化したものが、ゆかりコアであり、より柔軟にユーザ適應できるサービスアプリケーションを可能とするため、アクセス制御・シナリオ記述・センサ拡張の三点でゆかりコアに機能追加したものがゆかりカーネルである。これらの基盤を用いることで、

- アプリケーションに依存しない形式での機能記述の枠組み
- 機能連携サービスの構造や振舞いを形式的に記述する枠組み
- 広範囲のアプライアンスに共通な機能の抽出と類型化
- 実環境、実機器に適應した柔軟な機能発見方式

を実現することが可能となった。

ゆかりコア、ゆかりカーネルは研究開発に利用してもらうため、オープンソフトとして公開しており、リアルタイムストリーミング対応などの各種改良が行われている[16]。



## 参考文献

- 1 美濃導彦, “ゆかりプロジェクトの概要”, 本特集.
- 2 The Oxygen Project Website: <http://oxygen.lcs.mit.edu/>
- 3 The Ninja Project Website: <http://ninja.cs.berkeley.edu/>
- 4 南 正輝, 杉田 馨, 森川博之, 青山友紀, “ユビキタス環境に向けたインターネットアプリケーションプラットフォーム”, 電子情報通信学会論文誌, Vol.J85-B, No.12, pp.2313-2330, 2002.
- 5 R. Gupta, S. Talwar, and D. P. Agrawal, "Jini home net-working: a step toward pervasive computing", Computer, Vol.35, No.8, pp.32-40, 2002.
- 6 Universal Plug and Play Website: <http://www.upnp.org/>
- 7 HAVi Website: <http://www.havi.org/>
- 8 J. Nakazawa, Y. Tobe, and H. Tokuda, "VNA: A Serverless Distributed Architecture for Integrating Information Appliances", IEICE Trans. Fundamentals of Electron., Commun. & Comput. Sci, Vol.E84-A, No.7, pp.1610-1623, 2001.
- 9 ECHONET Website: <http://www.echonet.gr.jp/>
- 10 沢田篤史, 西村俊和, 山崎達也, 美濃導彦, “機能協調基盤ゆかりカーネルにおけるアクセス制御と振舞記述”, 電子情報通信学会技術研究報告, モバイルマルチメディア通信研究会 (MoMuC2006-60), Vol.106, No.409, pp.13-18, 2005.
- 11 西村俊和, “ゆかりカーネルにおけるプライバシー保護とアクセス制御, 第3回ユビキタスホームワークショップ”, pp.117-122, 2006.
- 12 T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickobic, and S. Weerawarana, "Business Process Execution Language for Web Services", Version 1.1, 2003.
- 13 沢田篤史, 多鹿陽介, 山崎達也, 美濃導彦, “機能協調型家電ネットワークのためのサービスシナリオ記述方式”, 情報処理学会研究報告, ソフトウェア工学研究会 (2004-SE-145), Vol.2004, No.87, pp.97-104, 2004.
- 14 UKARI Project Open Website: <http://open-ukari.nict.go.jp/>
- 15 Tatsuya Yamazaki, "Ubiquitous Home: Real-life Testbed for Home Context-Aware Service", Tridentcom2005 (First International Conference on Testbeds and Reserch Infrastructures for the DEvelopment of NeTworks and COMmunities), pp.54-59, 2005.
- 16 森村吉貴, 山崎達也, 美濃導彦, “分散協調基盤における QoS を考慮した動的ストリーミングサービス制御”, マルチメディア, 分散, 協調とモバイル (DICOMO2007) シンポジウム, pp.1602-1606, 2007.



やまざき たつや  
**山崎達也**

知識創成コミュニケーション研究センターユニバーサルシティグループ研究マネージャー(旧情報通信部門けいはんな情報通信融合研究センター分散協調メディアグループ主任研究員)  
博士(工学)  
マルチメディア情報通信処理技術



さいた たひろし  
**沢田篤史**

南山大学数理情報学部教授(元情報通信部門けいはんな情報通信融合研究センター分散協調メディアグループ専攻研究員兼務) 博士(工学)  
ソフトウェア工学

にしむら としかず  
**西村俊和**

立命館大学情報理工学部准教授(元情報通信部門けいはんな情報通信融合研究センター分散協調メディアグループ専攻研究員兼務) 博士(工学)  
計算機仲介コミュニケーション



たかおか まこと  
**高岡真則**

日本電気通信システム株式会社  
NCOS ラボラトリ主任  
ネットワーク

たしか しょうすけ  
**多鹿陽介**

元株式会社東芝 博士(情報学)  
ホームネットワーク



みのうみ ひこ  
**美濃導彦**

京都大学学術情報メディアセンター教授(元情報通信部門けいはんな情報通信融合研究センター分散協調メディアグループリーダー兼務) 工学博士  
3次元モデル処理、環境メディア、創作活動支援