

## VNA: 仮想情報家電の実現に向けて

4B-1

大越 匡<sup>1</sup> 中澤 仁<sup>1</sup> 田村 陽介<sup>1</sup> 望月 祐洋<sup>1</sup>  
戸辺 義人<sup>2</sup> 西尾 信彦<sup>3</sup> 徳田 英幸<sup>1, 2, 3</sup><sup>1</sup>慶應義塾大学大学院 政策・メディア研究科 <sup>2</sup>慶應義塾大学 SFC 研究所<sup>3</sup>慶應義塾大学 環境情報学部

## 1 はじめに

近年、家電機器はマイクロプロセッサを搭載して情報処理能力を増し、「情報家電」機器に変貌しつつある。その中でも、他の機器と何らかの物理媒体でネットワーク化されて積極的に他の機器と情報交換を行うことで、より高度な機器制御を行う情報家電機器は、Networked Appliances (例えば、[1] 参照) と呼ばれる。Networked Appliances は、IEEE 1394 [2]、電力線通信、IrDA、Ethernet などの伝送物理媒体による家庭内部でのネットワークと、インターネットの普及により家庭外部とのネットワークという両方の側面から期待が高まっている。また Networked Appliances を相互接続するためのソフトウェアの標準化活動も活発化している。しかし、現状の Networked Appliances に対する取り組みは、機器同士を連携させるという見方にとどまっており、ネットワークで接続することにより新たな視点からアプリケーションを構築するという見方が不十分である。こうした観点から、我々は、情報家電を柔軟に構成するための VNA (Virtual Network Appliance) アーキテクチャを提案する。

本稿では、第二章で VNA アーキテクチャの概要を解説し、第三章にて VNA アプリケーション構築事例を紹介する。第四章では関連研究との比較を行い、第五章で本稿を結ぶ。

## 2 Virtual Network Appliance

本章では、VNA (Virtual Network Appliance) アーキテクチャについてその目的、想定環境、機能、システム構成、およびソフトウェア環境について述べる。

## 2.1 背景および目的

情報家電機器がネットワークへ接続されると、他の情報家電機器のみならず、パーソナル・コンピュータ (以下 PC とする) 上で動作するソフトウェアや、新たに温度セ

ンサ、位置検出センサといった各種センサ、各種コントローラとも連携動作させることが可能となる。これまでのアプローチでは、そうした連携動作を行うのに、個別の物理的な機器 (例えば、PC、電子レンジ、プリンタ、デジタルカメラ) 単位で相互に接続するという見方を超えることはなかった。

これに対して、VNA アーキテクチャは、散在する物理的な機器のソフトウェアが提供する機能もしくは機器が有する部分機能を組み合わせて複合機能を有する仮想的な情報家電機器 (VNA) を構築するためのソフトウェアアーキテクチャである。VNA を構築する目的として、既存機能展開と新規機能生成の 2 種類がある。

## ● 既存機能展開:

既存機能を、複数の機器が有する機能を組み合わせて実現する。通常、家電機器では、動画を記録する [機能] ものとして、ビデオレコーダ [機器] があるように、機能が特定の機器にバインドされている。VNA では、「間に合わせ」的に部分機能を組み合わせることにより、特定機器とのバインドの制約をはずす。例えば、「動画を記録する」ために、PC、キャプチャデバイス、ハードディスクを「間に合わせ」で組み合わせる。既存機能展開により構築されたアプリケーションを“仮想化機器アプリケーション”と呼ぶ。

## ● 新規機能生成:

従来知られている機能ではなく、新しい機能を、複数の機器が有する機能を組み合わせて実現する。これは、情報家電開発段階において、従来存在する物理的な機器がソフトウェア機能単位で分解されて、さらに各種センサと組み合わせることにより行われる。生成されたものは VNA となる。例えば、目覚し時計、パントスタ、コーヒメーカーを組み合わせ、「目覚し朝食セットクッカー」を生成する。新規機能生成により構築されたアプリケーションを“創造的機器アプリケーション”と呼ぶ。

VNA アーキテクチャは、以上の 2 つの目的に対して、利用者に柔軟に VNA を構築する手段を与えることを目指す。

VNA: Toward the Realization of Virtual Network Appliances  
Tadashi OKOSHI<sup>1</sup>, Jin NAKAZAWA<sup>1</sup>, Yosuke TAMURA<sup>1</sup>,  
Masahiro MOCHIZUKI<sup>1</sup>, Yoshito TOBE<sup>2</sup>, Nobuhiko NISHIO<sup>3</sup>,  
and Hideyuki TOKUDA<sup>1, 3</sup>

<sup>1</sup>Graduate School of Media and Governance, Keio University  
5322, Endo, Fujisawa, Kanagawa 252-8520, Japan  
E-Mail: <slash@ht.sfc.keio.ac.jp>

<sup>2</sup>Keio Research Institute at SFC, Keio University

<sup>3</sup>Department of Environmental Information, Keio University  
本研究は、情報処理振興協会 (IPA) 次世代アプリケーション事業「次世代アプリケーション共通基盤 RT-HDI (Real-Time Human Device Interaction) プラットホームの研究開発」にて行われている。

2.2 特徴

VNA アーキテクチャを用いて情報家電アプリケーションを構築することにより、利用者毎のカスタマイズと、動作の動的適応が可能となる。

1. 利用者毎のカスタマイズ: 同一の情報家電機器を利用する場合においても、異なる利用者に論理的に異なる情報家電機器を与えることができる。たとえば、パワーユーザに対しては色彩、速度等詳細に渡る制御が行なえる操作ボタンを与え、簡単に使いたいユーザに対しては基本操作ボタンのみがある仮想的な VCR を生成すること等が考えられる。
2. 動作の動的適応: VNA を構成する個々の要素の組み合わせを変更することにより、情報家電機器の動作を動的に変えることができる。

2.3 想定環境

本システムは、多様なネットワークで接続された情報家電機器あるいはセンサ機器を対象として、利用者側計算機におけるそれらの機能あるいは部分機能の組み合わせが可能である。以下に、物理デバイスおよびネットワークの各観点から、本システムの動作環境を定義する。図 1 に VNA の想定環境を示す。

2.3.1 物理デバイス

本システムでは、ネットワークあるいは計算機に接続可能な、情報家電機器、センサ、およびコントローラを対象としている。これらのデバイスは、Java 仮想マシン(以降 JavaVM と呼ぶ)の有無によって、以下の 2 つに分類できる。

**JavaVM 搭載型** JavaVM を搭載し、本システムの提供するソフトウェアが動作可能なデバイス

**JavaVM 非搭載型** センサをはじめとする計算機に接続して用いるデバイスで、接続された計算機が JavaVM を持つ

JavaVM を搭載可能なデバイスとしては、計算機(PC)、情報家電機器、携帯情報端末が挙げられる。これらのデバイス上では、本システムの提供するソフトウェア環境が動作する。これに対して JavaVM を搭載しないデバイスとしては、小型センサやその他の小型デバイスが挙げられる。これらのデバイスは、JavaVM を搭載する計算機などに接続され、本システムの提供するソフトウェア環境は、接続された先の計算機上で動作する。

2.3.2 ネットワーク

本システムでは、上記デバイスが何らかの接続性によって相互接続される環境を対象としている。物理デバイスは、Ethernet, IEEE1394, 赤外線 (IrDA), 無線, シリアル (COM) ポート, 電力線, 電話線等様々な物理層およびデータリンク層で接続される。また、ネットワーク・トランスポート両層においても、TCP/IP, IEEE1394

トランスポート, COM ポートによる point-to-point 接続等の多様なプロトコルで接続される。これらの接続性に関する物理デバイス間の差異は、後述する VNA コンポーネント間通信支援機構によって隠蔽される。このため各デバイスや利用者側計算機では、物理層からトランスポート層までの様々なネットワークプロトコルを透過的に利用可能である。

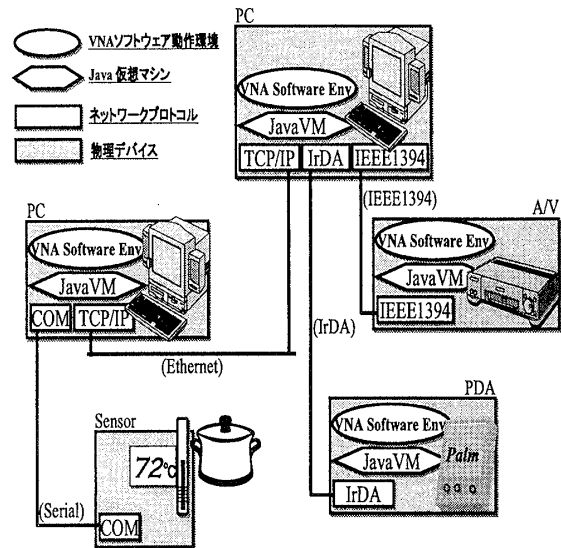


図 1: VNA の想定環境

2.4 システム構成

図 2 に示すように、VNA アーキテクチャのシステムは、VNA コンポーネント、VNA コンポーネント間通信支援機構および VNA サービスインタフェースの各サブシステムによって構成される。VNA コンポーネントは、ネットワークに対してその機能あるいは部分機能を提供する物理デバイスやソフトウェアモジュールであり、VNA コンポーネント間通信支援機構を用いて互いに通信する。VNA コンポーネントは、利用者側計算機で組み合わせることによって、VNA となる。このとき、VNA サービスインタフェースとは、利用者側計算機上で動作し、利用者の組み合わせ動作を支援する GUI を含むユーザインタフェースを指す。

各 VNA コンポーネント内あるいは VNA コンポーネントが接続されている計算機内では、当該 VNA コンポーネントを管理するソフトウェアモジュールである、“Serdget”が動作している。Serdget は、クライアント側スタブおよびサーバ側スタブで構成され、クライアント側スタブはユーザインタフェースおよびサーバ側スタブの遠隔参照を伴っている。図 3 に、Serdget の概念図を示す。図 3 において利用者は、クライアントスタブを利用者側計算機に移送することによって、対応する VNA コンポーネントを制御できる。このとき、各 VNA コンポーネント内

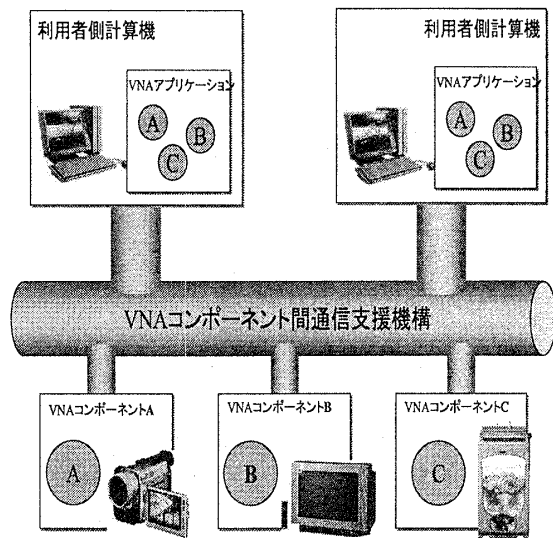


図 2: VNA アーキテクチャ・システム構成

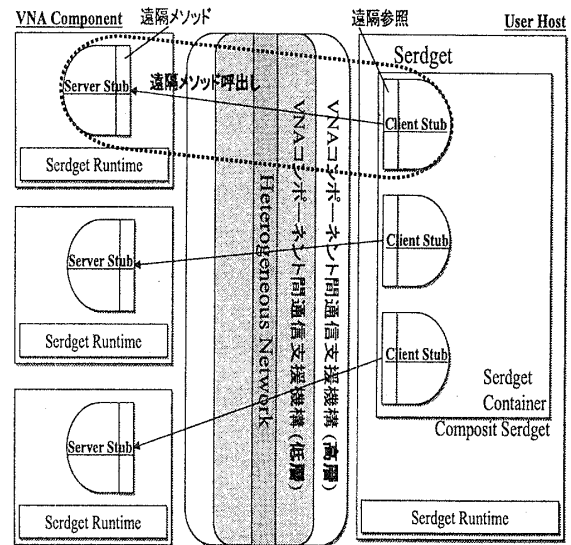


図 3: Serdget の構成

および利用者側計算機内で Serdget 動作環境が起動している。

Serdget のクライアント側スタブを、利用者側計算機で動作する Serdget コンテナを用いて複数組み合わせ、コンポジット Serdget (Composite Serdget) を作成できる。コンポジット Serdget は、組み合わせられた Serdget の各機能を提供する VNA コンポーネントであり、それ自体も他の Serdget と同様に組み合わせ可能であると同時に、VNA のソフトウェアの実体でもある。したがって VNA は、既存の情報家電機器やソフトウェアモジュールと同様に、VNA コンポーネントとしても扱える。

「コンポジット Serdget を作成する」ということは、Serdget が先にあり合成していくというボトムアップ的な考えに立っている。一方、抽象表現としての Serdget が先にあって、そこで利用する要素となる Serdget を動的に取り込んでいくトップダウン的な考え方もある。トップダウン的な構成例としては、動的に参加/離脱の可能な電子会議用の Serdget がその要素である参加者デバイスを動的に増減させることなどが考えられる。このようなトップダウン的なアプリケーション構成を支援するために動的生成型 Serdget (Generative Serdget) を導入する。動的生成型 Serdget では、その構成要素である Serdget を仮想 (generic) クラスとして表現し、その個数 (occurrence) を不定とすることにより、要素となる Serdget の動的な生成/削除を可能とする。

Serdget のクライアント側/サーバ側スタブ間通信、およびコンポジット Serdget における Serdget 間通信は、遠隔メソッド呼出しによって行われる。本通信は高層と低層に分けられ、両者とも VNA コンポーネント間通信支援機構が提供する。同機構は、高層通信として前述の遠

隔メソッド呼出しを提供しており、低層通信として各物理デバイスの接続性の差異を吸収するヘテロジニアスネットワークを提供する。従って、異なる種類のデータリンクおよび下位層プロトコルを用いる VNA コンポーネント間においても、遠隔メソッド呼出しを用いた通信が可能である。

### 2.5 機能

VNA は、制御データあるいはメディアデータの集約機能および転送機能を持つ。ここで制御データとは、対象機器の制御を目的としてネットワークに送信されるデータを指し、メディアデータとは、当該機器が扱う画像や音声などのデータを指す。このとき、上記の各データに関して VNA は以下の機能を持つ。

#### 制御データ

**集約** VNA は、1つ以上の Serdget を介して、同時に複数の VNA コンポーネントを制御できる。

**転送** ある Serdget に対する制御データを、宛て先情報をもとにして他の Serdget へ再送信できる。

#### メディアデータ

**集約** VNA は、1つ以上の Serdget を介して得られたメディアデータを同時に表示、再生できる。

**転送** ある Serdget を介して得られたメディアデータを、他の Serdget に転送し追加処理等が可能である。

データの集約および転送に関する設定は、Serdget コンテナを用いて利用者が容易に変更できる。このため、特定の環境に基づいて作成された VNA を、他の環境でも設定を変更するだけで容易に利用できる。また VNA は、既存の情報家電機器やソフトウェアモジュールと同様に VNA コンポーネントとして扱えるため、他の VNA コンポーネントと組み合わせたり、再帰的に新たな VNA の構

成要素となることができる。

### 3 VNA 事例

本章では、VNAとして仮想化されたVCRである“VVCR (Virtual VCR)”アプリケーション、在宅での遠隔学習用テレティーチングアプリケーションの例としての“Virtual 教室用マルチテレビデオ”アプリケーションを例に、実際のVNAアプリケーション構築手順および利用手順を述べる。

#### 3.1 VNA アプリケーション構築手法

VNAアプリケーションは、“仮想化機器アプリケーション”，および“創造的機器アプリケーション”の2種に分類される。前者の例としては“VVCR”が、後者の例としては“Virtual 教室用マルチテレビデオ”が挙げられる。

前者の構築アプローチは、既存の家電機器を機能の集合として仮想化し、それぞれの機能に対応する実世界の機器として、実際の家電機器もしくは同じ機能を実現する異種の家電機器の部分機能を組み合わせて、仮想的な家電機器を実現するアプローチである。例えばVVCRは、既存の家電機器であるVCRを仮想化したものである。VCRでの再生ということでは、映像表示機能に対応する実世界の機器としてTV受像機を利用することが一般的である。しかし、TV受像機が利用者の周辺に存在しない場合には、PC上の映像出力用ウィンドウをTVと同じ機能を提供するVVCRの部分機能として組み合わせることが考えられる。

後者の構築のアプローチは、実世界に存在する既存の家電機器にとらわれず、新しい家電機器としてVNAアプリケーションを作り出すアプローチである。ここでは、在宅環境において、“Virtual 教室用マルチテレビデオ”を「教室において先生と複数の学生が各自ネットワーク接続されたPCを携帯し、先生のPCの画面を全員で共有閲覧できる機器」と定義する。実際には単体でこのような機能を実現する機器は存在しないが、VNAアプリケーション構築者は、想定される利用者の利用モデルを元に、VNAアプリケーションを構築する。動作時にはVVCR同様、VNAアプリケーションの各部分機能に、それと同じ機能を提供する実世界の機器を組み合わせ、動作する。

#### 3.2 VVCR

VVCR (Virtual VCR)-アプリケーション (以下VVCR) は、VNAアプリケーションとしてその機能が仮想化されたVCRである。ここでは、画像再生時におけるVVCRの機能構成、および利用者の利用手順を述べる。VVCRを利用することで、利用者はVCRプレーヤからの再生データをネットワークを通して遠隔の機器で鑑賞したり、遠隔からの再生制御といった利用が可能となる。

#### 3.2.1 機能構成

図4にVVCRの機能構成図を示す。VVCRでは、VCR再生を構成する機能部分であるVCRプレーヤのデータ読み込み機能、ディスプレイの映像表示機能、スピーカの音声再生機能が、それぞれVVCRを構成する機能部品として仮想化される。

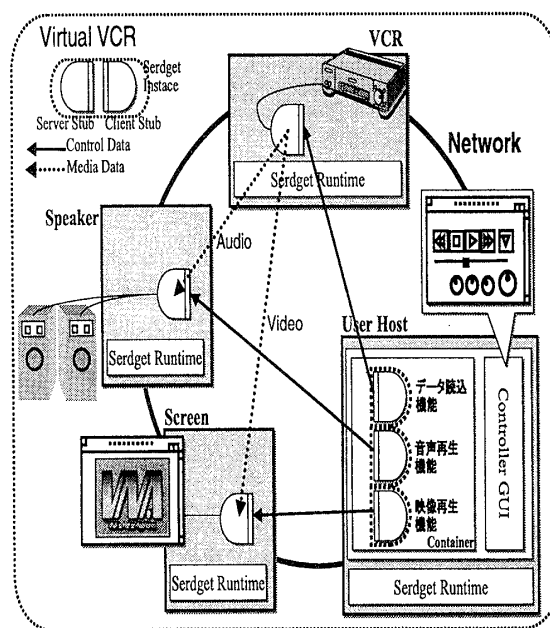


図 4: VVCR の構成

VVCR 利用者は手動または自動で、各機能部品として動作する Serdget インスタンスを設定することで、ネットワーク上に散在する Serdget から柔軟に VVCR を構成できる。

#### 3.2.2 利用手順

VVCR 利用者の利用手順、およびその際の本アーキテクチャにおける動作手順の概要は以下の通りである。

1. VVCR 利用者は、VVCR 構築者によって作成された VVCR のソフトウェア実体である VVCR - コンポジット Serdget を、インターネット上のサイト等より取得する。  
この時 VVCR - コンポジット Serdget 内部には、それを構成する Serdget となる、特定の機器の Serdget に関する設定は含まれない。
2. 利用者は取得したコンポジット Serdget を起動する。
3. 利用者は VNA サービスインタフェースの提供する GUI 等を用いて、VVCR のデータ読み込み機能、映像表示機能、音声再生機能といった構成機能を、ネットワーク上の特定の Serdget インスタンスと設定する。この場合 VNA アプリケーション次回起動時からは、構成機能はユーザによる再設定のない限り指定された Serdget インスタンスと自動的に設定される。

もしくは利用者は、構成機能を例えば「おばあちゃんの部屋のビデオ」といったディレクトリサービスへの問い合わせと設定する。構成機能は、その問い合わせを用いたディレクトリサービスからの検索結果の Serdget インスタンスと設定される。この場合 VNA アプリケーション次回起動時からは、ユーザによる再設定のない限り構成機能は毎回のディレクトリサービスからの検索結果である Serdget インスタンスと設定される。利用者は、上記のいずれかもしくは両方で各構成機能が設定された VVCR - コンポジット Serdget を保存することで、利用者の利用用途にカスタマイズされた VVCR を作成できる。

4. 利用者は再生、停止などの利用を実際に行う。

### 3.3 Virtual 教室用マルチテレビデオ

Virtual 教室用マルチテレビデオアプリケーションは、PCを使った画面共有閲覧アプリケーションである。授業を行う教室と遠隔から受講する在宅学習者をネットワークで結び、画面を共有したテレティーチング等を用途とする。

授業において、教室の教師、および自宅等遠隔地から受講する学生の全員がネットワークへ接続された各自のPCを使用する。授業中は教師のPC画面が学生全員のPCへと表示される。質問やディスカッション時には、制御権を持つ教師が画面の切り替え操作を行って、任意の学生の画面を共有を行ったり、任意の画面へ書き込みを行う切り替えを可能である。

#### 3.3.1 機能構成

図5に本アプリケーションの機能構成図を示す。本アプリケーションを構成する機能部品は、画面の描画情報をネットワークへ公開する描画情報出力機能、描画情報を元に画面を描画する画面描画機能、キーボード・マウスの入力をネットワークへ送信する入力送信機能、入力送信機能からの入力情報を受信する入力受信機能の4機能である。

本アプリケーションは、動的生成型 Serdget によって実現される。動作時には利用者（教師等）によって設定された受講人数等の数に応じて Serdget 内に4機能に相当する部品がそれぞれ作成される。作成された各機能部品は、各参加者側の Serdget として設定され、動作する。

#### 3.3.2 利用手順

本アプリケーション利用者の利用手順、およびその際の本アーキテクチャにおける動作手順の概要は以下の通りである。

1. 利用者は、本アプリケーション構築者によって作成された Virtual 教室用マルチテレビデオのソフトウェア実体である動的生成型 Serdget を、インターネット上のサイト等より取得する。

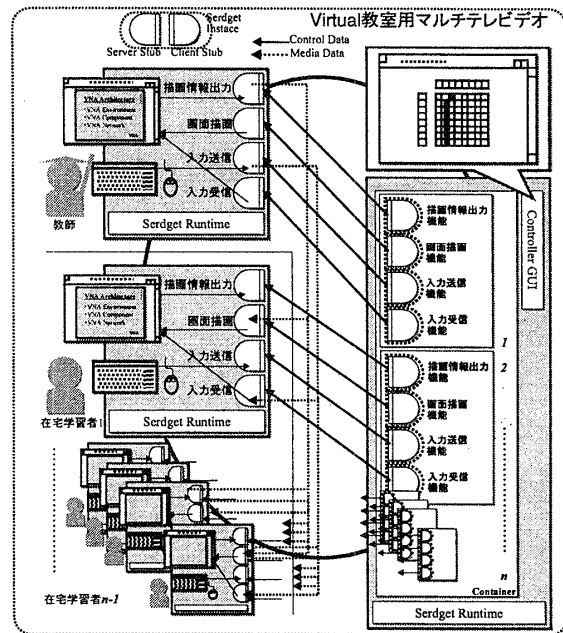


図 5: Virtual 教室用マルチテレビデオ構成

この時、動的生成型 Serdget 内部には、それを構成する Serdget となる、特定の機器の Serdget に関する設定は含まれない。

2. 利用者は取得した動的生成型 Serdget を起動する。この時起動時の引数もしくは起動後の設定として、使用する人数を設定する。
3. 利用者は、動的生成型 Serdget 内の各構成機能を、教師および履修学生名簿等の情報からなるディレクトリサービスへの問い合わせを設定する。問い合わせを元にディレクトリサービスから検索された Serdget が、参加人数分の各構成機能を設定する。利用者は、各構成機能への設定が行われた動的生成型 Serdget を保存することで、利用用途（授業等）毎にカスタマイズされた動的生成型 Serdget を作成できる。
4. 先生として登録された利用者は、画面および入出力の切り替え制御を行い、本アプリケーションを利用する。

## 4 関連研究

Networked Appliances の相互接続性を定める仕様として、HAVi (Home Audio/Video Interoperability)[3], Jini (Java Intelligent Network Infrastructure)[5], UPnP (Universal Plug and Play)[6] がある。Jini, UPnP に比較して HAVi が AV(Audio Video) 機器接続に特化しているという違いはあるものの、いずれも物理的「機器」を視点において、プラグ・アンド・プレイによる機器同士のスムーズな接続、共通プロトコル・コマンドの実行に重点を置いている。これに対して、我々が本稿で示す VNA アーキテクチャは、「機器」を仮想的に機能の集合体として扱い、「機器の機能や部分機能をネットワークを介して

組み合わせ、利用者毎の情報家電機器」を提供するものである。また、VNAアーキテクチャは情報家電開発にも適用可能である点が他とは異なる。反面、VNAアーキテクチャは、機器のプラグ・アンド・プレイについては規定しないので、以上の仕様と補完関係にある。

分散する情報家電機器を制御する方法として、ユーザが携帯型コンピュータを用いて遍在する機器を統合的に制御するシステム、CUES (Control for Ubiquitous Embedded Systems)[7]が提案されている。CUESでは、移動コードや移動エージェントを含む複数の通信形態をサポートし、このために制御機器上にSMAF (Simple Mobile Agent Facility)と呼ぶ実行環境が動作することを前提としている。遍在機器の統合的制御の方式として、自動サービス統合 (Automatic Service Integration) およびユーザ補助によるサービス統合 (User Assisted Service Integration) の2つを想定しており、実現例としてプリントサービスの統合取り上げている。これに対して本研究では、自動/ユーザ補助によるサービス統合に相当する分類に加え、3.1節で示したボトムアップ/トップダウンアプローチによる統合という分類の次元に基づくサービス統合(すなわちVNAアプリケーションの構築)を支援している点に特徴がある。

Ad-Hoc Mobile Service[4]では、ユーザの移動先で提供されるさまざまなサービスと、携帯端末を介してインタラククションを行うためのシステムアーキテクチャを実現している。サービスの一部には移動先の照明やVCRなどネットワーク接続された機器の制御を含んでおり、携帯端末をユニバーサルインタラクタ (Universal Interactor) として捉え、ユーザと実世界に存在するさまざまな制御可能オブジェクトとのインタラククションの体系化を目指している。移動先で利用可能なサービスを発見するための資源発見プロトコルや、移動にともなう制御対象機器(とその制御用インタフェース)の変更に応じて携帯端末のユーザインタフェースを動的に対応づけるプロトコル(transduction protocol)、複数の機器を連携させる複合動作をスクリプト言語を用いて定義する機構を提供している。我々の研究同様、サービスを組み合わせることを念頭においているが、一つの「機器」として見せるという視点はない。

我々は、先にポゼッションシステム [8, 9] に基づく分散システム上でのアプリケーションの構築方法を提案した。ポゼッションシステムでは、遍在する機器やソフトウェアモジュールを利用するアプリケーションの構成部品を魂 (Soul) が肉体 (Body) を憑依する (possess) アナログに基づいて分類し、構成部品同士を柔軟に組み合わせる機構を実現している。元々適応的な協調作業支援アプリケーション構築への利用が想定されているため、VNAを目指したものではないが、本VNAアーキテクチャに

おける Serdge 間の結びつきに活用することができる。

## 5 おわりに

本稿では、情報家電を柔軟に構成するためのVNAアーキテクチャとその構成要素について概要を述べた。VNAアーキテクチャでは、VNAコンポーネントを組み合わせ、VNAとしてアプリケーションソフトウェアを構築するという方法により、仮想的な情報家電機器を定義することができる。こうした方法により、個別の物理的な機器をつなぎあわせるという視点を超えて、1つの「機器」がネットワーク上に散らばる機能の集合体として存在するという見方が可能となる。

今後の課題として、(1) 現在定めたVNAアーキテクチャの範囲内のVNAの具体例の構築、(2) 利用可能な機能もしくはサービスが動的に変化する環境でのVNAを生成するための仕組み(動的変化に対応したディレクトリサービス、Serdgetの動的切り替え機構等)の具体化がある。

## 謝辞

本研究を行うにあたり、RT-HDIプロジェクトに参加されているCarnegie Mellon University、北陸先端科学技術大学院大学、電気通信大学、NTT情報流通プラットフォーム基盤研究所、東芝研究開発センター通信プラットフォームラボラトリー、横河電機情報技術開発センター、日本アイ・ビー・エムES事業他の皆様から多くの助言をいただいていることを、感謝申し上げます。次第です。

## 参考文献

- [1] *Proceedings of IEEE Workshop on Network Appliances*, (1998.10).
- [2] <http://www.1394ta.org>.
- [3] Sony, Matsushita, Philips, Thomson, Hitachi, Toshiba, Sharp and Grundig: Specification of the Home Audio/Video Interoperability (HAVi) Architecture (1998). <http://www.havi.org/home.html>.
- [4] Hodes, T. D., Katz, R. H., Servan-Schreiber, E., and Rowe, L.: Composable Ad-hoc Mobile Services for Universal Interaction, *Proceedings of MOBICOM'97*, pp. 1 - 12 (1997).
- [5] Sun Microsystems, Inc.: Jini Architecture Specification (1998). <http://www.javasoft.com/products/jini/specs/jini-spec.pdf>.
- [6] Universal Plug and Play Forum: Universal Plug and Play (UPnP) (1999). <http://www.upnp.org>.
- [7] Kangas, K. and Róning, J.: Using Mobile Code for Service Integration in Ubiquitous Computing, *Proceedings of the 5th Mobile Object Systems Workshop* (1999).
- [8] Mochizuki, M. and Tokuda, H: Possession system, *Proceedings of ICDCS'99*, (1999.5).
- [9] Mochizuki, M. and Tokuda, H: Possession system: adaptation support Middleware for Collaborative Multimedia Applications in Java, *Proceedings of ICMCS'99*, (1999.5).