

情報家電機器間の動的な イベントバインディング機構の構築

岩井 将行¹ 楠本 晶彦¹ 中澤 仁¹ 徳田 英幸^{1, 2}
¹慶應義塾大学大学院 政策・メディア研究科 ²慶應義塾大学 環境情報学部

本研究では、情報家電機器間のイベントを動的に組み合わせることを可能にするイベントバインディング機構を構築した。近年家庭内へのネットワークの導入により、独立で動作する家電機器に代わり、ネットワーク接続機能を持った情報家電機器が登場し始めた。本イベントバインディング機構では、それらの情報家電機器を相互に利用する枠組みを提供する。機器ネットワーク障害への対処、未知の情報家電機器の検索と発見、情報家電機器間の組み合わせを動的に変更できる動的変更機能が Jini を用いて実現されている。これにより、情報家電機器間同士での状態の通知や相互利用が可能になる。

Dynamic Redirectionable Distributed Event Model for Intelligent Home Appliances Network

Masayuki Iwai¹ Kusumoto Akihiko¹ Jin Nakazawa¹ Hideyuki Tokuda^{1, 2}

¹Graduate School of Media and Governance, Keio University
²Faculty of Environmental Information, Keio University

In this research, mutual communication mechanism of networked appliances with a built-in communication interface is proposed, designed and implemented. As the network spreads over the home, the networked appliances that can be controlled through network have appeared. However many of networked appliances are not used automatically each other. To provide the structure in which more networked appliances are connected each other, network fault obstacle, searching of unknown networked appliances, and function of dynamically changing the combination of each networked appliances are needed.

1 はじめに

今日ネットワーク接続機能を備えた情報家電機器が多く生産されるようになった。また通信ネットワーク形態の多様化に伴い、一般的に家庭内で用いられるネットワークの形態も Ethernet, IEEE1394[3], IEEE802.11[4] など有線、無線通信分野で多様化している。ネットワークインタフェースを持つ情報家電機器が上記の様に、家庭内ネットワークに接続されるシステムを本稿では、情報家電ネットワークと呼ぶ。情報家電ネットワークの利用形態は2つある。第1の利用形態は、図1の上を示す様に、情報家電機器をネットワーク経由でユーザが操作することである。この方法は、操作する情報家電機器の制御プログラムを対象の機器側に保存し、ユーザがソフトウェアコンポーネント及びXML[10]に基づいた制御ファイルをダウンロードし、そのソフトウェアを通してネットワーク経由の情報家電機器操作を可能としている。Jini[6], UPnP[9], HAVi[5], VNA[11]などは上記の利用形態を実現できる。情報家電ネットワークの第2の利用形態は、ネットワークに接続された情報家電機器同士が協調し、インテグレートされた作業をユーザに対して行うことである。図1の下に示すように、情報家電機器の状態変化(以下 イベント)を、ユーザを介さずにメー

カや用途の異なる情報家電機器に通知する。それによりユーザは、機器の設定を行うだけで良くなり、ユーザの情報家電機器に対する操作等を削減できる。

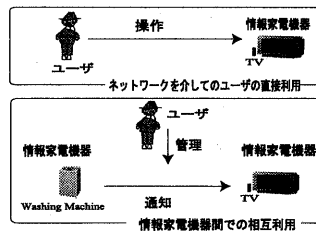


図1: 情報家電ネットワークの利用形態

情報家電機器同士が協調する例としては、「洗濯機の洗濯終了情報をユーザの近くにあるTVの画面に表示し、ユーザに洗濯が終わったことを通知する」、「インターフォンが押されると携帯電話に通知が来る」などの今までの家電の枠組みでは、あまり関わりのない機器間の相互利用が可能となる。しかしこのような情報家電ネットワークの第2の利用形態の場合、情報家電機器を設計開発する段階で、イベントの通知先の機器の名前及び通信先アドレスを認識し、イベントをネットワーク上の通知先に伝達するプログラムを機器に予め組み込んでいなければなら

ない。しかし情報家電ネットワークでは、予め組み込むことは、第2の利用形態の柔軟性を著しく損ねてしまう。

本稿では、この問題点を解決するためにイベント通知先を動的に変更できる動的分散イベント通知機構を提案する。

以下に本稿の構成を示す。第2章では、情報家電ネットワークの必要な機能を明らかにし、既存の分散イベント機構が情報家電機器間の相互利用に必要な機能を果たしていないことを述べる。第3章では、情報家電ネットワークに適したイベント通知機構「動的分散イベント通知機構」(DRAGON)を提案する。第4章では、動的分散イベント通知機構を基にしたシステムを容易に構築できるツールキット: DRAGON/TKの実装について述べる。第5章では、DRAGON/TKを基に構築したシステムの定量的評価及び、DRAGON/TKとJiniの比較評価を行う。第6章では、今後の課題について述べ、第7章で、本稿のまとめを行う。

2 既存分散イベントモデルの問題点

前章で、情報家電機器同士が協調し、インテグレートされた作業をユーザに対して行う事の利便性を述べた。本章では、情報家電機器間の協調作業のための必要機能を述べ、既存分散イベントモデルで機器間の協調作業を行う事が困難であることを述べる。

2.1 情報家電ネットワークの必要機能

機器間の協調作業を行うためには情報家電機器間で情報のやり取り(イベント通知)が行われなければならない。本稿では、情報家電機器をソフトウェアとハードウェアが協調したサービスと定義する。サービス間でのイベント通知機構の必要機能は、検索発見機能、動的変更機能、耐障害機能である事を述べる。

検索発見機能

情報家電ネットワークでイベントの介在してサービス間の相互利用を行う場合、イベントを発生する側および、イベントを消費する側双方のサービスを検索可能でなければならない。双方のサービスを検知できなければ、両者を組み合わせることは不可能になる。また、サービスには様々な役割と属性情報が保持されているため、ユーザが様々な検索条件を用いて検索できる必要がある。例えば、テレビやパソコンなどのディスプレイを持ったサービスを、その名前やネットワークアドレスでなくディスプレイの解像度や、大きさ、カラー情報などの属性条件からも検索発見可能である事が望まれる。

動的変更機能

前節で述べたように、情報家電ネットワークには用途やメーカーの異なるサービスが散在し、サービスが連続してイベントを配送し続ける場合が多い。そのため、あるサービスが他のサービスにイベントを配送する場合は、ユーザがサービスが動作中にイ

ベントの通知先を自由に設定し変更可能でなければならない。つまりイベントを生産するサービスとイベントを消費するサービス間を動的に常に組替え可能である必要がある。

耐障害機能

情報家電機器は、頻繁に電源を遮断されたり、ネットワークから切断されたりする事故が多く、ユーザによって接続地点(ネットワークアドレス)を変更される場合が多い。そのため、ネットワーク的な障害等に対してシステム全体が影響を受けない耐障害性を保持する必要がある。

2.2 HAVi制御モデル

HAViは、Sonyや松下電器などの家電機器メーカーが中心となって策定された規格で、AV機器がIEEE1394規格のインタフェースで統合されたホームネットワーク内で利用できる。HAViアーキテクチャでは、デバイスを制御機器と被制御機器に分類している。制御機器はDevice Control Modelを介して被制御機器を制御する。被制御機器は、当該機器を制御するために必要なコードを保持しており、ユーザインタフェースも含まれる。しかしHAViは、テレビとビデオのように相互の機器の役割があらかじめ定まっている機器制御が想定されている。そのため、HAViの固有のシステムでは、様々な用途やメーカーの機器の混在するネットワークへの応用が困難である。さらに、複数の情報家電機器を相互利用する場合、様々な機器に対してイベント通知先を変更できる枠組みをHAViに付け加えなければならない。

2.3 Jini分散イベントモデル

Jiniは、応用範囲が広い分散環境構築技術の一つである。Jiniではサービスは、ディスカバリプロトコルを用いて、Lookup Service(LUS)の発見を行う。LUSは、分散環境下にある各サービスの情報を管理する。このとき、LUSにはサービスの遠隔参照オブジェクト(Stub)がコピーされる。サービスを利用するプログラムは、LUSに対して利用したいサービスの情報を与え、合致するサービスの遠隔参照オブジェクトを得て、遠隔メソッド呼び出し(RMI[7])によって当該サービスを利用する。これにより、情報家電ネットワークにおいて情報家電機器の遠隔制御を実現できる。また、登録されたサービスは、Leaseによって時間を利用した管理がなされる。LUSは、Lease期間が超過すると、保持していたサービスを廃棄する。サービス側は、そのリース期間終了前にサービスをLUSに再登録する。

Jiniを情報家電ネットワークに応用するにはいくつかの問題がある。イベント通知相手を予めプログラムに組み込むでおかなければならず、イベント通知先の動的な変更が不可能である点とイベント消費サービスがLUS登録されないため、他のサービスからの検索が不可能であり、ユーザがサービスの組み合わせを選ぶ段階で表示できない問題点がある。

このため、動的変更機能と検索発見機能を満たしていないと言える。

2.4 ECJ分散イベントモデル

ECJ[8]は、非同期イベント駆動型の分散システム構築用フレームワークである。発生頻度が高く、サイズの小さいイベントを高速に伝達する場合に適している。イベント駆動型システムにより、ネットワーク経由の高速なイベント配送が可能になる。ECJは、イベントIDとイベントハンドラをマップをするテーブルを、Local Managerが保持しているため、イベント通知先の動的な変更に対応できる。しかしサービス発生時の自動発見や様々な属性を用いた検索が行えず、検索発見機能を満たしていない。このため、情報家電ネットワークに適した分散イベントモデルとは言えない。

3 動的分散イベント通知機構

本稿では、JiniやHAViをはじめとする既存の機構に存在しないイベントの通知先を動的に変更可能にする機能を有する『動的分散イベント通知機構』(DRAGON)を提案する。DRAGONは、JiniやHAViによって分散管理されている情報家電機器上に提供され、それらの既存分散システムに検索発見機能、耐障害機能を追加する。本章では、情報家電機器の機能別のモジュール分割を述べ、それを基にDRAGONが果たしている機能と役割を述べる。

3.1 情報家電機器の機能別分割

個々の連携しない独立動作する家電機器をソフトウェアで制御する場合、図2のようにセンサモジュール及びアクションモジュールにソフトウェアの役割を分割できる。

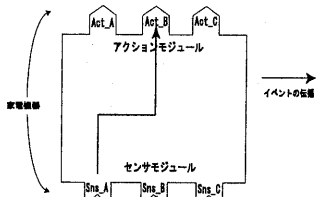


図 2: 独立した家電機器のモジュール分割

センサモジュールとは、プッシュ式ボタンなどのユーザの操作を受け取る部分と、温度センサなど実世界からセンシングを行うモジュールである。アクションモジュールとは、音、映像などのメディアデータの実世界への出力、ヒータやモータの電源の投入部分などの実世界に関連した操作を起こすモジュールである。

動的分散イベント通知機構で扱っている情報家電機器は、上記で述べた2つのモジュール以外に、イベント出力モジュール、イベント入力モジュールを保持する。イベント出力モジュールは、図3左で示すようにセンサモジュールで感知した情報をイベントオブジェクトに格納し、イベントとしてネットワークを経由して、他の情報家電機器に送り出すモジュールである。

イベント入力モジュールとは、図3右で示すようにイベントを受け取り、機器を操作するアクションモジュールに渡すモジュールである。

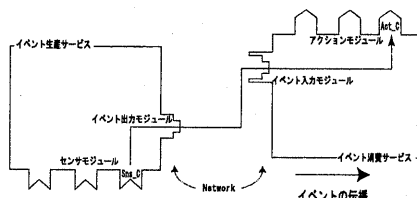


図 3: イベントモジュール部の追加

各情報家電機器のイベント出力モジュール及びイベント入力モジュールが本システムで提供され、ネットワーク上で組み合わせることが可能になっている。こういった機能別モジュール分割を行うことによって情報家電開発者はイベント送信、受信部分を考慮する事なく情報家電制御ソフトウェアを開発できる。

本稿では、イベント出力モジュールを保持している情報家電機器をイベント生産サービスと呼び、イベント入力モジュールを保持している情報家電機器をイベント消費サービスと呼ぶ。その両モジュールを保持するサービス中でもアクションモジュール及びセンサモジュールを保持せず、イベントに対する処理のみを行うサービスをイベント管理サービスと呼ぶ。

動的分散イベント通知機構は、図4のように左のイベント出力モジュールと右のイベント入力モジュールを動的に組み合わせることができる。そのためイベント生産サービスは、イベントの通知先を考慮することなく通常の家電機器と同様に設計開発可能になる。

以上のモジュール機能別分割により、連続的にイベントを発生している状態のサービスでもサービス全体を操作することなくイベントの通知先を切り替えることが可能になっている。

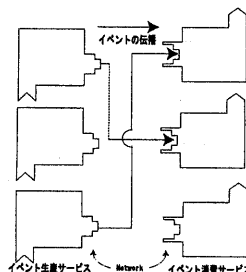


図 4: イベント出力モジュールとイベント入力モジュールの動的な組み合わせ

3.2 イベント生産サービスとイベント消費サービスの検索と発見機能

動的分散イベント通知機構は、イベント生産サービスの管理をJiniのルックアップサービス(LUS)によって行っている。LUSにイベント生産サービス

の情報を保持させ、他のサービスからの検索を可能にしている。本システムが提供するイベント組み替え管理サービス (Binder サービス) は、イベント消費サービスの情報を保持し、イベント消費サービスの検索を可能にする。Binder は Jini のサービスとして実装される。

各サービスのイベント出力モジュールとイベント入力モジュールは、互いの存在を前もって認識していない。そのため各サービスは、情報家電ネットワークに接続された環境で、互いを動的に検索発見可能でなければならない。図5に検索と発見機能の動作概要を示す。

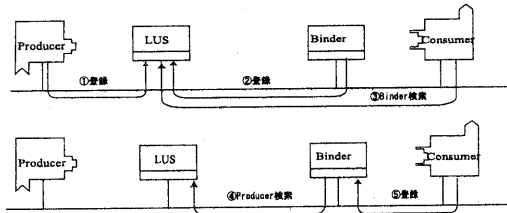


図5: サービスの検索と発見機能

イベント生産サービスはLUSに自分自身を登録する。Binderも同時にLUSに自分自身を登録する。イベント消費サービスはLUSに対しBinderの検索を行い、発見したBinderに自分自身を登録し、情報を公開しユーザからの検索を可能にする。Binderは、LUSに対してすべてのイベント生産サービスの検索を行い、ユーザに情報を公開する。

3.3 イベント生産サービスとイベント消費サービスの動的な組み合わせ機能

図4のようにイベント生産サービスのイベント出力モジュール部分とイベント消費サービスのイベント入力モジュール部分とを動的に結び付けることによってサービス間のイベント通知を可能とする。動的なイベント通知先変更の動作概念を図6に示す。

イベント切り替え管理サービス (以下Binder) はLUSを発見し、設定用のGUIアプレットを登録する。ユーザは、クライアントプログラムを起動し、LUSからBinderを検索する。クライアントプログラムは、LUSからBinder及びGUIアプレットをダウンロードする。GUIアプレットは、ユーザからの組み合わせ命令を受け、Binderに情報を送信する。Binderは、イベント消費サービスに通知先の変更命令を発生する。イベント消費サービスは、命令に従いイベント生産サービスにイベント登録を行う。イベント生産サービスは、イベントが生じると登録に従い、イベント消費サービスに通知を行う。以上の仕組みでサービス間の動的な組み合わせを実現できる。

3.4 イベント生産サービスとイベント消費サービスの障害動的適応機能

イベント消費サービス及びイベント生産サービスは、通常の家電製品同様、緊急の電源の切断や移動

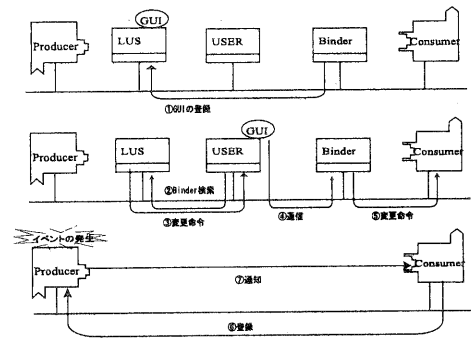


図6: 動的なイベント通知先変更の実現

によりネットワークから切断されてしまう場合が多い。そのため障害時にも適応できる機能を持たなければならない。

イベント消費サービスの障害時

イベント消費サービスに、ネットワーク切断の障害が発生する。イベント生産サービスは、イベント通知を行うい、イベントが到達不可能である状態を検出する。イベント生産サービスは、イベント消費サービスをイベント登録リストから抹消する。現在のイベント登録リストの更新をBinderに報告する。これによって、他のサービスへは影響なくイベント通知が継続が行える。

イベント生産サービスの障害時

図7に示すように、イベント生産サービスはLUSに対し定期的な更新を行う。BinderはLUSに対し特定のイベント生産サービスの障害及び発生に対して報告を受けるための登録を行う。イベント生産サービスに障害が発生すると、イベント生産サービスからの定期的更新が不能になる。LUSは、Binderに対して障害通知を行い、Binderはその通知をもとにイベント消費サービスに通知をする。イベント消費サービスは命令に従い再度検索を行い、イベント生産サービスの障害に適応できる。

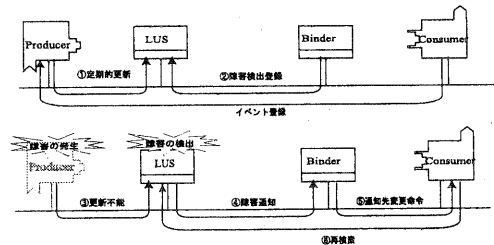


図7: イベント生産モジュールの障害動的適応機能

3.5 イベント管理サービスの役割によるイベント配送の柔軟性

イベント管理サービスは、イベントの管理以外には行わないプログラムだけで実装されるサービスである。以下のイベント管理サービスにより柔軟なイベント配送が行えるようになる。現在5種類のサービスを開発している。

- イベント組替え管理サービス：上記で述べた Binder サービスである。イベント通知先切り替えを行うサービスである。
- イベント統合管理サービス：複数のイベント生産サービスを登録できる。このサービスはすべてのイベントが到着した時点でイベント消費サービスにイベント通知を行う。
- イベント分配管理サービス：ある一つのイベント生産サービスからのイベントを複数のイベント消費サービスに通知するサービスである。
- イベント時間管理サービス：イベントが通知されると同時にイベントの状態を保持し、設定時間になるとイベント消費サービスに通知するサービスである。
- イベントキュー管理サービス：複数のイベント生産サービスから連続して通知されるイベントを一つのキューで管理し、イベント消費サービスに通知を行うサービスである。

これらのモデルは、ペトリネットのトランジションモデル [2] を参考に構成されている。

4 動的分散イベント通知機構の実装

本章では、動的分散イベント通知機構構築用のツールキットである DRAGON/TK の実装について述べる。

4.1 システム全体の概念

システム全体の流れを図8に示す。左の洗濯機及び電話機の情報家電機器をイベント生産サービスとし、それぞれ洗濯終了イベントや電話呼び出しイベントを発生できると仮定する。右のテレビはイベント消費サービスとし、テレビに文字を表示したり、アラームを鳴らすなどのアクションを起こすことが可能である。

仮にユーザは、聴覚に障害を抱えており、電話やインターフォンの鳴っている状態をテレビで視覚的に確認したいと考えているとする。ユーザが行なうことは、手元の PC や小型端末機器 (PDA) 等を利用して、サービスが表示されている GUI 上のテレビのアイコンと電話機のアイコンを結び付けるだけである。その通知先変更命令は、サービス間の結び合わせを管理しているサービスに送信される。その通知先管理サービスは、通知先変更命令をテレビサービス上で起動しているイベント消費サービスに通知する。テレビサービス上で起動しているイベント消費サービスは、通知先変更命令に従い電話サービスに対して自分自身の登録を行なう。この段階で電話が鳴るイベントが発生した場合、上記の登録にしたがってイベントがテレビサービスのイベント消費サービスに通知され、テレビサービスが「文字で知らせる」というアクションを起こす。

4.2 主な構成要素の実装

本節では、DRAGON/TK において重要な構成要素の実装について述べる。

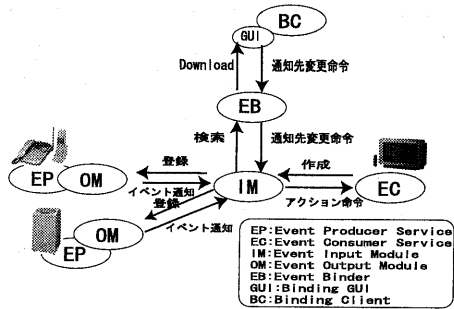


図 8: システム全体の概要

Event Producer Service

3.1節で定義したイベント消費サービスである。センサモジュール、イベント出力モジュールを必ず持ちイベント消費サービスに通知を行うサービスである。

Event Output Module

Event Producer Service の一つのモジュールである。Event Producer Service を LUS に登録し、LUS に対し一定周期で登録を繰り返す作業をおこなう。また、Event Input Module からのイベント登録を受け、イベントが発生次第イベント通知を行う。

Event Consumer Service

3.1節で定義したイベント消費サービスである。イベント入力モジュールとイベント出力モジュールを保持している。テレビに文字を表示するサービス、電灯を ON にするサービスなどに相当する。

Event Input Module

Event Producer Service から通知されたアクション命令を Event Consumer Service に引き渡すモジュールである。Event Binder Service の検出を行い、Event Producer Service に対してイベント通知登録を行なう。常に LookUp Service 上のイベント生産サービスの変化 (消滅, 新規登録, 属性の変化など) を見張っている。また、Event Binder からの通知先変更命令を受け付け、登録先を変更する機能を有する。

本システムでの Event Input Module の実行手順を図9に示す。

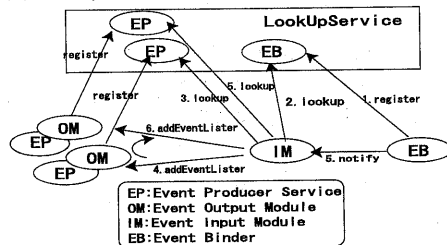


図 9: Event Input Module の実行手順

1. Event Binder は、LUS に登録される。
2. Event Input Module は、Event Producer Service を検索する。

- Event Input Moduleは、さらに Event Producer Serviceを検索する。
- Event Input Moduleは、発見した Event Producer Serviceの Event Output Moduleにイベント登録する。
- Event Binderは、変更通知命令を Event Input Moduleに送る。
- Event Input Moduleは、それに従い、新たな Event Producer Serviceを検索する。
- Event Input Moduleは、発見した Event Producer Serviceの Event Output Moduleにイベント登録する。

Event Binder

Event Binderとは、サービス間のイベントの組み合わせを行なうサービスである。Jini Lookup Serviceから現在入手可能なサービスを検索し、次に説明する Event Binding GUIに通知する。また Event Input Moduleにイベントの通知先変更する命令を通知する。Event Binderは、Lookup Service上のイベント生産サービスの変化（消滅、新規登録、属性の変化など）を検知している。また、イベント消費サービスからのイベント登録を受けると同時にイベント消費サービスの属性情報を獲得しているイベント消費サービスの検索が可能になる。本システムにおける Event Binder 実行手順を図10に示す。

- Event Binderは、Lookup Service上にある Event Producer Serviceを検索する。
- Event Binderは、LUSに自分自身を登録する。
- Event Binderは、Event Input Moduleからの登録を受け付ける。
- ユーザは Binding Clientを起動し、Event Binderを検索する。
- Event Binderから、Binding GUIをダウンロードする。
- Binding GUIは、ユーザにサービスの情報を表示し、イベント通知先書き換え命令を受け付ける。
- Binding GUIは、通信により Event Binderに情報を送る。
- Event Binderは、ユーザから得た命令を Event Input Moduleに通知する。

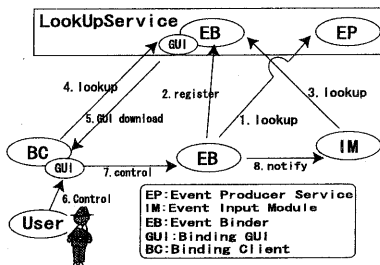


図 10: Event Binder の実行手順

Binding GUI

Binding GUIとは、ユーザがイベント通知先を組み合わせる段階で利用する GUIである。図11の様にユーザにはアイコンを結びつけることにより、ネットワークや機器間の相互利用を容易に設定できる環

境が提供される。Binding GUIは、ダウンロードして遠隔利用を提供しているためマルチキャストが通過しないルータを越えて用操作が可能である。そのため家電ネットワークの外部からの機器の組み合わせを設定できる。

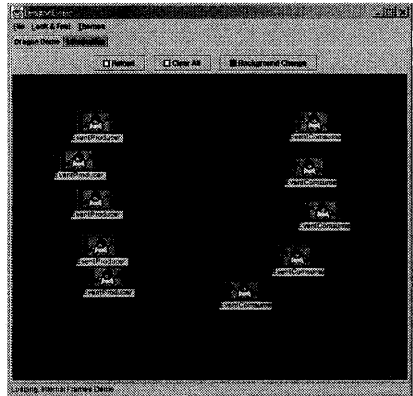


図 11: GUIの画面ダンプ

5 DRAGON/TKの評価

イベント伝搬にかかる時間の測定及び、イベント通知先切り替え能力を測定し、DRAGON/TKが想定する情報家電ネットワークにおいて必要な速度を満たしていることを示す。また DRAGON/TKを Jiniと比較し、追加した機能を評価する。

5.1 イベント伝搬にかかる時間の測定

一台のマシン上で起動しているイベント生産サービスが複数のイベント消費サービスにイベント通知を行うまでのイベント伝搬に関わる時間を測定する。

測定方法

測定には100Base-Tで相互接続された、Sun Microsystems社のワークステーションであるUltra 30をイベント生産サービスの実行に1台使用した。表1にUltra 30の性能を示す。測定用アプリケーションには、一秒ごとに1024byteイベント付加情報を保持させたイベントの伝搬を行うイベント生産サービスを作成し、イベント消費サービスを1台から13台のマシンで実行し、イベント通知を図12の様にGUIにより設定した。すべてのイベント消費サービスは、イベント生産サービスにイベント登録を行う。イベント生産サービスは、すべてのイベント消費サービスに送信する。その送信始めから送信終了までの時間を、100回測定した。

表 1: 測定環境

CPU	UltraSPARC-II 248MHz
メモリ	128MB
OS	Solaris2.6
JDK version	Solaris_JDK_1.2.1.03
Jini version	1.0.1

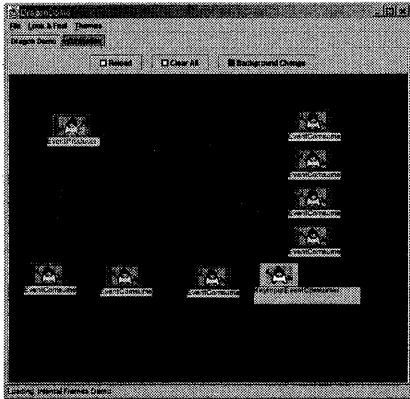


図 12: 複数のイベント消費サービスにイベント通知中の画面

測定結果

グラフ 13は、横軸にイベント消費サービスの数、縦軸にすべてのイベント通知が完了する総時間を示す。

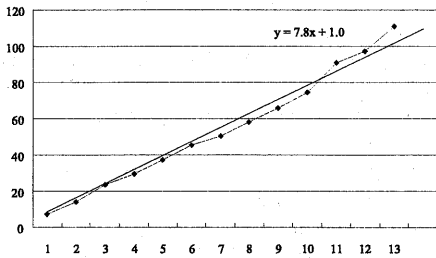


図 13: 通知時間測定結果

情報家電ネットワークは、ホームエリアの使用を対象としているそのため、想定している情報家電ネットワークでは、目覚まし時計のアラームを鳴らすなど1秒単位でイベントの発生を行う情報家電機器が多く存在する。1秒を情報家電ネットワークの基準単位とし、測定結果から得られた線形回帰曲線の近似式より、上記のグラフの傾きを7.8に近似でき毎秒128個の消費者サービスに同時に通知できることが分かる。この値は、家庭内での情報家電ネットワーク中では、これほど情報家電機器が設置されている事は想定しがたく、一つのイベント生産サービスが通知を行う数としては充分と言える。

5.2 イベント通知先切り替えにかかる時間の測定

ユーザがGUIにより操作を繰り返してから切り替えがイベント生産サービスに反映されるまでの時間を測定する。それにより、イベント通知先切り替えにかかる時間は、比較的に短い事を示す。通知先変更命令は、AB区間からFG区間まで逐次伝搬される。各区間の説明を以下に示す。

- 区間AB: BindingGUIクラスに対するユーザの入力から描画処理時間
- 区間BC: アプレットが通信に要した時間

表 2: 切り替え時間測定結果 (ms)

AB	BC	CD	DE	EF	FG	Total
1.6	31.6	0.1	35.6	99.3	48.4	216.6

- 区間CD: EventBinderクラスで消費された時間
- 区間DE: Event Input Moduleクラスに通知先変更命令が通知される時間
- 区間EF: Event Input Moduleクラスで消費された時間
- 区間FG: Event Input Moduleクラスで発生したイベント登録がEvent Output Moduleに完全に登録されるまでの時間

測定方法

測定用アプリケーションは、一秒毎にイベント伝搬を行うイベント生産サービスを作成し、イベント消費サービスを1台イベント通知命令をGUIにより発行してからその命令が反映されるまでの時間を測定する。同一ホスト上で、Binding GUI, Event Binder, Event Consumer Service, Event Producer Serviceを起動した。イベント登録と登録のキャンセルを5秒ごとに繰り返し、GUIに接続を示すラインを描画してから、通知先命令が反映されるまでの時間を100回測定した。

測定結果

測定結果を表2に示す。ユーザがイベント切り替え命令を描画させてから切り替え反映されるまでの時間は、約216ミリ秒必要とした。このうちDRAGON/TKのオーバーヘッドはAB+CD+EF=101msである。このため、設定と同時にGUIのライン描画を100ms間点滅描画させることにより、ユーザに遅延を感じさせることを避けた。

5.3 Jiniとの定性的比較評価

動的分散イベント通知機構によってJiniの不足機能を補い情報家電ネットワークの必要機能を満たす事が可能になったかを評価する。評価項目を以下に示す。

検索発見機能: Jiniでは、イベント生産サービスのみ検索可能であるのに対して、DRAGON/TKはイベント消費サービス双方から検索、様々な検索条件を用いた自由度の高い検索に対応している。

耐障害機能: Jiniではイベント生産サービスの障害時の検知十分にされないのに対して、DRAGON/TKでは両サービスのネットワーク断絶や障害に対応できる機能ある。

動的変更機能: Jiniでは一度イベント通知を変更するには、意図的に通知先を切り替えるコードを始めに記述しておかなければならない。DRAGON/TKではイベントの通知先を動的にユーザが変更できる機能がある。

簡易設定機能：DRAGON/TKにはイベントの通知先の切り替えをユーザによって容易に変更可能なGUIを提供している。

表3の様にDRAGON/TKが情報家電ネットワークに必要な機能を満たしたイベント通知機構であると云える。

表 3: Jini イベントモデルとの機能比較

評価項目	J ¹	D ²
LUS障害機能	×	×
検索発見機能		
イベント生産サービス発見機能	○	○
イベント消費サービス発見機能	×	○
耐障害機能		
イベント生産サービス障害検知機能	△	○
イベント消費サービス障害検知機能	○	○
柔軟なサービス検索機能	○	○
動的可変機能	×	○
簡易設定機能	×	○

¹Jini, ²Jini+ DRAGON/TK, ×...実現していない, △...一部実現している, ○...実現している

6 今後の課題とまとめ

現在のDRAGON/TKは、ごく基本的な機能しか提供していない。将来は、本研究を情報家電機器に組み込まれるプログラムとして広く使用されることを目標にしている。今後以下のような機能について拡張していく予定である。

6.1 音声インタフェースへの対応

音声によりユーザからの要求を認識し、その認識に基づき情報家電機器を結びつけるプログラムによりGUIだけでなく声による組み合わせを可能にする。ViaVoice SDK, Java Technology Edition[1]を用いて実装する予定である。

6.2 Lookup Service (LUS) 障害時の復旧機能の実装

LUSの障害は、情報家電ネットワークでは、他のサービスと同じ頻度で発生すると考えられる。本システムでは、検索部分でLUSを利用しているだけであり、一旦イベント登録されたイベント消費サービスとイベント生産サービスは、LUSが障害にあっても不具合が生じる事はない。しかしLUS障害時には、新しいサービスの登場に対処できなくなるため、障害を関知した場合自動的にLUSを復旧する枠組みが必要になる。

6.3 地理的情報を基にしたサービス検索の実現

現在、サービスの検索は、サービスの名前またはサービスの属性で検索を行うことができる。しかしユーザが利用するサービスの検索発見を容易にするためには情報家電ネットワークは、「この家の2階にあるイベント生産サービスの発見」などという地理的状况からサービス発見検出を行う必要がある。

7 おわりに

本稿では、情報家電機器間のイベントバインディング機構の構築について述べた。現在、情報家電機器同士による自律的な制御は実現されていない。将来、急速な発展が見込まれる情報家電ネットワークは、情報家電機器と情報家電機器が状態の変化(イベント)を人間を介さずやり取りする場面が想定される。そのために必要となる機能は、検索発見機能、耐障害機能、動的可変機能である。本研究ではこれらの問題点に対して有効な動的分散イベント通知機構(DRAGON)を提案した。またDRAGONを基にした、システムの構築を支援するツールキット(DRAGON/TK)を構築した。本研究によって今までユーザの手を介して行ってきた情報家電機器制御が情報家電機器間で相互に行なわれるようになり、その設定も容易に行うことができる。情報家電機器をユーザが自由に組み合わせる環境が構築可能になり、ユーザが思い描く新しい役割を情報家電機器間に動的に割り振る事も可能になる。

参考文献

- [1] IBM . The ViaVoice SDK, Java Technology Edition V1.
- [2] A.Al-Dallal and K.Saleh. Protocol synthesis using the petri net model. PDCS 1997.
- [3] IEEE. Standard for a high performance serial bus, 1999.
- [4] International standard for wireless lan medium access control (mac) and physical layer(phy) specifications, 1999.
- [5] Sony, Matsushita, Philips, Thomson, Hitachi, Toshiba, Sharp, and Grundig. Specification of the Home Audio/Video Interoperability (HAVi) Architecture, May 1998. <http://www.havi.org/home.html>.
- [6] Sun Microsystems, Inc. Jini architecture specification. <http://www.javasoft.com/products/jini/specs/jini-spec.pdf>.
- [7] Sun Microsystems Inc. Remote Method Invocation Specification, 1996.
- [8] 宮澤隆幸, 海邊裕. Javaによる非同期メッセージ配送フレームワーク ECJ. 情報処理学会 システムソフトウェアオペレーティングシステム. 情報処理学会, May 1999.
- [9] Universal Plug and Play Forum, 1999. <http://www.upnp.org>.
- [10] World Wide Web Consortium. Extensible Markup Language(XML) 1.0, February 1999.
- [11] 大越匡, 中澤仁, 田村陽介, 望月 祐洋, 戸辺義人, 西尾信彦, 徳田英幸. VNA : 仮想情報家電の実現へ向けて. 第59回情報処理学会全国大会, September 1999.