# 家電ネットワークにおけるサービスの動的提供機構

1Z－1

岩井　将行[1]　大越 匡[2]　中澤　仁[2]　徳田 英幸[1,2]
[1]慶應義塾大学 環境情報学部　[2]慶應義塾大学大学院 政策・メディア研究科

## 1　はじめに

ネットワークに接続される家電製品が出荷され，本格的にネットワークに対応した家電が，家庭内にまで普及しはじめた．それに伴い，分散環境構築を対象とした研究が次々に発表されている．その分散環境の中でプラットフォームに依存せず，応用範囲が広い研究の一つとして，Sun Microsystems 社が提唱する Jini[1]がある．Jini の分散環境により，プログラマは分散アプリケーションの設計や実装をより容易に行えるようになった．

しかし，Jini を家電ネットワークに応用させるためには，いくつかの問題がある．第一に，家電開発者は，ネットワークに対応した家電製品を設計する段階でどの機器と連動させるかを予測することは困難である．第二に，ユーザに対しては，複雑な機器間の設定を要求することはできない．そのため，既存のシステムでは、ユーザがセンサや家電製品などのサービス間のイベントを簡単に組み合わせ，そこから発生するイベントを動的に組替えることが実現していなかった．

本稿では，上記の問題を解決すべく Jini の提供するサービスに組み込むライブラリとして，家電間のイベント通知を動的に組み合わせる分散環境と，ユーザが家電製品の組み合わせを容易に設定，変更可能な GUI を構築するライブラリを提案する．

## 2　本システムの概要

本システムでは，プリンタ，センサ，ネットワークに対応した家電製品などのデバイスとそれを操作するソフトウエアをサービスと定義している．

また本システムは，イベントをサービス側での機器の変化及び，ソフトウエアで記述された状態の遷移と定義する．

また本システムは，サービスを提供している機器の状態変化及び，ソフトウエアの状態変化をイベントとして定義する．

本システムは，各サービスごとにイベント通知先を記述したテーブルを保持し，そのテーブルを外部から書きかえる事によりイベントの通知先を動的に変更できる．これにより，家電開発者はイベント通知先を考慮せずにネットワークに対応した家電製品を開発できるメリットがある．さらにユーザは動的にサービスから発生したイベントを動的に切り替えることができる．

## 3　本システムの特徴

本システムの性能的な特徴として安全性，柔軟性，可視性があげられる

**安全性：**

本システムは，policy file を proxy 側に保持することによって，悪意のあるサービスによる不正なシステムへの攻撃を防止及び，信頼あるサービス提供者を特定可能である．

**柔軟性：**

本システムの proxy は，サービスの既存の機能を失うことなくサービスを拡張するため，システムに参加していないサービス間との連携が可能である．

**可視性：**

システムで構築できる GUI によりユーザはサービスから発生したイベントの通知先を視覚的に決定し，変更できる．

## 4　設計

本システムは，家庭内における各サービスから発生するイベント間を動的に組み合わせることを目的としたライブラリを提供する．

### 4.1 設計方針

複数の分散されたサービスから発生するイベントの通知先の動的な変更方法として，集中管理式にサーバを作り通知先を管理する方針が考えられる．しかし，本システムは，分散的にイベントの通知先テーブルを保持する．イベント通知先テーブルを保持することにより，本システムは，上記の集中管理式モデルと比較してトラフィックの発生量とイベント通知先管理サーバに対する負荷が軽減できる．

家庭内ネットワークのように，イベントの発生量が多く，通知先の変更頻度が少ない状況のネットワークにおいては，イベント発生毎にサーバに問い合わせをせず通知を行うことが可能なため，本システムが有効であることがわかる．

### 4.2 想定する環境

本システムが動作する前提条件として以下の２つがある．

- JavaVM を搭載した家電製品
  本システムは JavaVM 上で実行される Jini のサービスとして提供される．そのため JavaVM を搭載した家電製品またはセンサの存在を前提としている．
- Jini
  Jini は，分散環境において，サービス間を簡単に接続する目的で設計され，Jini に対応した家電製品が多く発表されている．これらの理由から本システムでは Jini を採用し，サービス起動時に Jini の

---

Dynamic Events Binding System for Home Network
Masayuki Iwai[1]　tailor@ht.sfc.keio.ac.jp
Tadashi Okoshi[2], Hitoshi Nakazawa[2],
And Hideyuki Tokuda[1,2]
[1]Faculty of Environmental Information, Keio University
[2]Graduate School of Media and Governance, Keio University

API が実行可能であることを前提としている.

### 4.3 本システムの構成要素

本システムの構成要素を以下に示す.

**・イベント生産者:**
イベントを発生させるサービス.

**・イベント消費者:**
イベントの通知を受けアクションを起こすサービス.

**・Consumer Proxy:**
イベント消費者のサービスに実装されるプログラム. セキュリティポリシファイルを保持し, ファイルに記述された信頼できるサービスからのイベント通知によってアクションを起こす.

**・Producer Proxy:**
イベント生産者のサービスに実装されるプログラム. Service Binder からの命令を認識し, イベントの通知先を変更する. イベント通知先テーブル(Table)を保持する.

**・Table(イベント通知先テーブル)**
Producer Proxy に存在し, イベント通知先を保持しているリスト.

**・Service Binder:**
サービス間の組み合わせを行うプログラム. Lookup Server から現在入手可能なサービスを検索し Service Binding GUI に通知する.また Producer Proxy のテーブルを書き換え通知先を変更する.

**・Service Binding GUI:**
サービスを組み合わせる GUI. JavaBeans[2]のようにコンポーネント間のイベントのやり取りが視覚的に設定可能である.

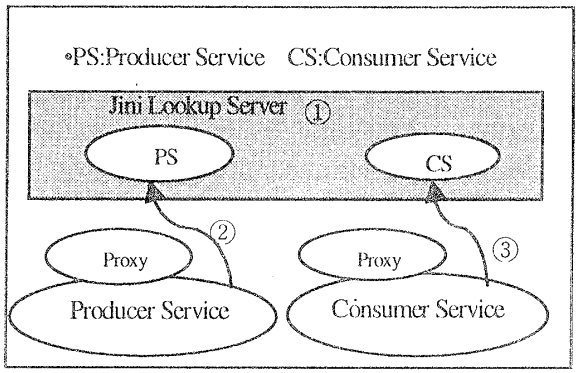


図 1 Lookup server への登録手順

### 4.4 本システムの実行手順

本システムの実行手順は, サービスの登録手順とサービス間のイベントバインディング手順に分けられる.

**登録手順**

本システムへのサービス登録手順を図 1 に示す.

1. Jini Lookup Server を起動する.
2. イベント生産者サービスは, Producer Proxy を生成し, Jini Lookup Server に自分自身を登録する.
3. イベント消費者サービスは, Consumer Proxy を生成し, Lookup Server へ自分自身を登録する.

**サービス間のイベントバインディング手順**

本システムにおけるサービス間イベントバインディング手順を図 2 に示す.

1. ユーザは Service Binding GUI を起動し, Service Binder のインスタンスを生成する.
2. Service Binder は Lookup Server 上にある Service を取得する.
3. Service Binding GUI はそれをユーザに表示し入力を受け付ける.
4. Service Binder は, ユーザから得た情報を Producer Proxy に通知しイベント通知先テーブルを書きかえる.
5. Producer Proxy は, イベントが発生するとテーブル通り新たな Consumer Proxy に通知を行う.

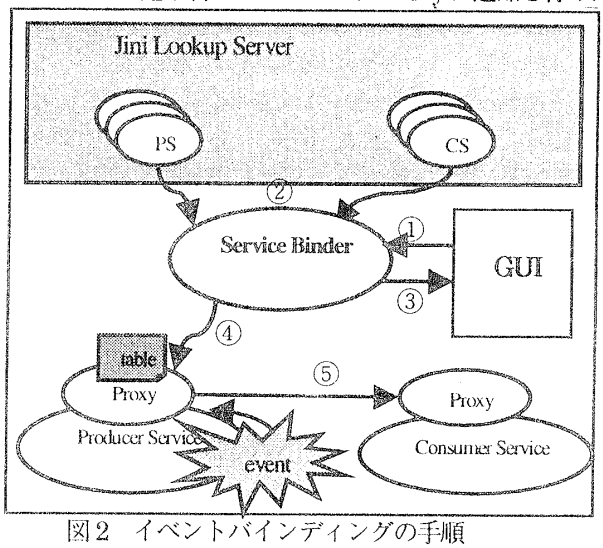


図 2 イベントバインディングの手順

## 5 まとめと今後の課題

本稿では, Jini を用いサービス間のイベント通知に関するバンディングを提供する機構を実現するシステムについて提案した.

今後は, 本システムを利用したアプリケーションのパフォーマンス評価, Lookup Server 障害時の復旧機能の実装, 複数間の Lookup Server への対応 (大規模性の実現), より詳細なセキュリティポリシの追加をなどを予定している.

さらにイベントの通知条件, 通知期間, 通知ポリシーなどを規定し, 多様な設定が可能になるように拡張予定である.

## 参考文献

[1] Sun Microsystems Inc., "Jini Technology Overview", 1999
[2] Sun Microsystems Inc., "InfoBus1.2 specification", 1999
*http://java.sun.com/beans/infobus/infobus1.2.pdf*