4Q-4

# Jini環境における サービス自動生成に関する研究

井上 芳丈　中村 次男　小川 均
立命館大学大学院理工学研究科

## 1 はじめに

今日のネットワーク環境は，端末だけで構成されるコンピュータネットワーク環境から，様々な機器が相互に協調できるような環境へと変わってきている．

このような環境を構築する技術として Sun Microsystems社のJini技術がある．Jini技術によって様々な機器はネットワークに接続するだけでそのネットワーク環境の中に組み込むことができる[1]．また Jini ではネットワークに接続されたハードウェアやその上で動作するソフトウェアなどのリソースを『サービス』として抽象化する[2]．サービスとして扱うことで，サーバの移動や追加があった場合もクライアントがそれを気にする必要は無い．

しかし，サーバ・クライアント間の通信を行うために Jini では JavaRMI(Remote Method Invocation)が利用されており，既存プログラムなどを用いてのサービス生成は困難である．

そこで，既存プログラムからのサービス生成が自動で行われると，サービス生成の効率が上がると思われるのでこれを提案する．

## 2 サービス生成の自動化

### 2．1 サービス登録情報

Jini ではサービスのさまざまな情報を Lookup サービスが管理し，サービス同士はその情報をもとに通信する．Lookup サービスは，Jini 上のサービスを「サービスアイテム」の形で管理している．ここでこのサービスアイテムの中身には，そのサービス自身のインスタンスとサービスの名前や場所，機能など，それに Lookup サービスが一意に割り振るサービス ID が含まれている．

### 2．2 サービスの自動生成

前述したように，Lookup サービスへの登録を行うには，新サービスの名前や場所，機能が必要である．つまりこれらの情報を管理者が認識さえしていれば，つぎの二つのテンプレートを用意することで Jini 技術を利用したサービスの登録・生成を自動で行うことができる．

- JavaVM(オブジェクト)間通信にJavaRMIを用いたソースコードの作成
- Lookup サービスへの登録に必要な名前，場所，機能，使うクラスの情報の挿入

したがって管理者の負担を軽減でき，サービスの生成は効率よく行える．また，新サービスの管理者は JavaRMI や Jini の知識を特に必要とせず，誰でも気軽にサービスの提案，生成が行えると考えられる．

リモコンサービスでの具体例を図 1 に表す．図 1 の 22 行目では LookupDiscovery クラスを用いて、近くに存在するLookupサービスの検索が行われる．ここであらかじめ登録する Lookup サービスのホスト名，ポート番号がわかる場合は，LookupLocator クラスを用いることもできる．33～35 行目では、Lookup サービスに登録するサービスアイテムが用意されている．ここでこのサービスの名前を JiniRemoconServr としている．38 行目で実際に登録が行われる．41 行目では登録したサービスの ID を参照している．

## 3 リモコンサービス

### 3．1 目的

以上の Jini の性質を理解した上で，本研究では特にリモコンを用いた家電のサービスを提供する．既

```
        /* JiniRemoconSrvImp.java */
01| import java.rmi.*;
02| import java.rmi.server.*;
03| import net.jini.core.discovery.*;
04| import net.jini.core.entry.*;
05| import net.jini.core.lookup.*;
06| import net.jini.lookup.entry.*;
07| public class JiniRemoconSrvImp extends
08|         UnicastRemoteObject implements
09|             JiniRemoconSrv, DiscoveryListener{
10|
11|
12|   public static void main(String[] argv){
13|     JiniRemoconSrvImp jrSrv = null;
14|     LookupDiscovery discovery = null;
15|     System.setSecurityManager
16|           (new RMISecurityManager());
17|     try{
18|       jrSrv = new JiniRemoconSrvImp();
19|     }
20|           (中略)
21|     try{
22|       discovery = new LookupDiscovery
23|           (LookupDiscovery.ALL_GROUPS);
24|     }
25|           (中略)
26|
27|     public void discovered(DiscoveryEvent dEvt){
28|     Entry[] attrs;
29|     ServiceItem item;
30|     ServiceID id;
31|     ServiceRegistrar[] registrars = null;
32|     registrars = dEvt.getRegistrars();
33|     attrs = new Entry[1];
34|     attrs[0] = new Name("JiniRemoconServer.");
35|     item = new ServiceItem(null, this, attrs);
36|     try{
37|       ServiceRegistration reg =
38|           registrars[0].register(item,10000000L);
39|       System.out.println
40|           ("SucsessServiceRegisteration.");
41|       id = reg.getServiceID();
42|       System.out.println("Server: id = " + id);
43|     }
44|           (中略)
45| }
```

RMIクラスとJiniクラスのパッケージの追加

リモコンサービスの内容をここに記入

Lookupサービスの発見方法の指定

サービスアイテムの登録

図 1 JiniRemoconSrvImp.java

存の家電はネットワークに対応しておらず，Jini 環境に直接接続し利用することができない．そこで家電に備わっている機能の最も代表的なものとしてリモコン機能に注目し，これを Jini 環境の一つのサービスとして生成，利用することを提案する．リモコン制御信号は，パソコンとシリアル接続された装置から発信する．

### 3．2 リモコンサービス利用までの流れ

サービスの管理者はサービスアイテムにリモコンの制御信号を付加し，これを Lookup サービスに登録する(図 2-A)．Lookup サービスは受け取ったサービスアイテムを登録し，同時に付加されているリモコン制御信号をデータベース化して保存する(図 2-A-a)．よってユーザやアプリケーションは端末から家電を操作することができる(図 2-B)．また，携帯電話や PDA など携帯端末からの家電操作の利用も実現可能であると考えられる．

すでに Jini 環境にサービスとして取り込まれている家電と同一の家電を新規サービスとして Jini 環境に取り込む際，Lookup サービスが保存されているリモコン制御信号データベースを参照することで，同時にリモコンサービスまでも提供することが可能となる(図 2-A-b)．

## 4 今後の課題

Jini の動作環境の整備と Lookup サービスにリモコン制御信号データベースを実装中である．

### 参考文献


[1]「Jini アーキテクチャの仕様 Revision1.0」，Sun Microsystems (1999).

[2] 夏川勝行，森田晃平：「ネットワーク管理への Jini 技術適用法の検討」信学技報 IN99-112, (or TM99-78, OFS99-65) (2000-01).


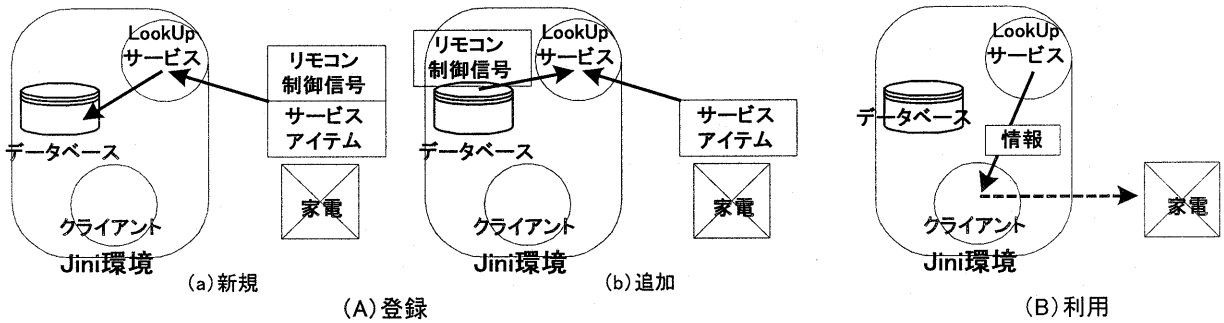


図 2 リモコンサービス利用までの流れ