

Jiniテクノロジー:Jiniによる自発的・動的ネットワークの実現

Jini Technology : Achievement of voluntary and dynamic network by Jini

堤佳之, 谷村勇輔 (知的システムデザイン研究室)

Yoshiyuki TSUTSUMI, Yusuke TANIMURA (Intelligent Systems Design Laboratory)

Abstract This paper describes about Jini technology that achieves voluntary and dynamic network system. It doesn't distinguish software and hardware, and offers dynamic, rebuildable object network environment. Its potential range of application is extraordinary vast. On the other hand, permeation to industry field is only beginning.

1 Jiniとは

パソコンとテレビやラジオ, プリンタやデジタルカメラ, さらに照明や暖房器具などの様々な機器によってネットワークを作り, 相互利用することは物理的な接続やドライバなどの問題があり決して容易ではない. 実際にそのようなネットワークを作るには人的資源と時間が必要となる. Jiniの目的はこの問題を解消すること、つまり様々な機器を接続するだけで相互利用を可能とするネットワークを作り, 人的資源と時間を節約することである.

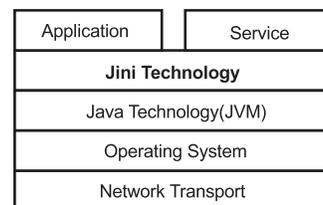


図 1: 基本構造

2 基本技術と特徴

JiniテクノロジーはJavaの動的にダウンロード可能なプログラムやダウンロードされたプログラムを安全に実行するためのセキュリティ, さらにJavaアプリケーションが全てのJava仮想マシン(JVM:Java Virtual Machine)環境のある機器上で動作することを基礎として, それらをネットワーク環境へと拡張したものである.

Jiniの特徴としては次のようなものがある.

1. 適用範囲の広さ

図1に示すようにJiniはJVM環境があればどのようなOS, ハードウェア上でも動作する.

2. 自発的ネットワーク

現在のネットワーク環境ではネットワーク全体を手動で管理する必要があるが, Jiniではそれぞれの機器がCPUを持ち, 自発して動的に自身を管理するので手動で管理する必要が無い.

3. サービスによる統一的环境

Jiniではハードウェアとソフトウェア, ファイルやネットワークなどの区別を無くし, 全てを

サービスと呼ばれるオブジェクトとして扱い, 従来のようにそれぞれの特性を意識して使用する必要の無い単純で統一的环境を作っている.

4. Lookupサービスによる相互通信

Jiniではサービスがお互いに通信するためにLookupサービスと呼ばれるサービスを用いる. Jiniネットワークに接続するサービスは, 接続すると同時に自分自身を属性と共にLookupサービスに登録しJiniネットワーク上での参照を可能とする.

5. リースによるアクセスと自己管理

サービスはリースという期限付きのアクセス権を得ることで利用でき, 継続して利用するためには期限が切れる前にリースを更新しなければならない. 期限が切れるのはユーザーがサービスを利用する必要がなくなったときやネットワークに障害が発生したときなどである.

3 必要環境

Jiniシステムを実現するのに必要なものとして, まず機器同士を接続するためのネットワークがある.

ネットワークに必要とされる帯域は接続する機器によって異なり、ディスプレイやプリンタのように広い帯域を必要とする機器もあれば、マウスやキーボードのように狭い帯域で十分な機器もある。

次にそれぞれの機器が行おうとする処理に見合ったメモリの量と処理能力がそれぞれの機器自身に必要である。これは JVM を動作させるためにも必要である。近年では小型で安価、かつ高性能なプロセッサが開発されているので、それをあらゆる機器に搭載し JVM を走らせることは容易である。

また、能力が不十分な機器であっても処理を他の機器（コンピューターなど）に任せることにより Jini システムの実現が可能である。

4 Jini の実際の動作

ここでは例としてデジタルカメラで撮った写真をプリンタで印刷するという例を考える。

1. ネットワークへのプリンタの接続
 - (a) プリンタがネットワークに接続されると、プリンタ自身がネットワーク上の適切な Lookup サービスを探しだす。
 - (b) 印刷サービスを属性と共に全て登録する。印刷サービスの属性としては印刷できる色数や使用できる紙のサイズなどがある。
 - (c) 登録したサービスに対するリースを得る。リースは期限内に更新されなければならない。
2. 印刷サービスの選択
 - (d) カメラがネットワークに接続すると前項と同じ手順で登録し、リースを獲得する。
 - (e) ネットワーク上に登録されている印刷サービスの一覧を Lookup サービスを通じて獲得し、一覧の中から適切な印刷サービスをユーザーとカメラが選択する。一つの印刷サービスが選択されると、カメラは印刷サービスのリースを獲得し、印刷のための print メソッドを直接起動できるようになる。
3. プリンタの設定
 - (f) ユーザーは印刷をする前にカメラを通して印刷サービスに含まれる configure メソッドを起動し色数や紙のサイズなどのプリンタの設定を行うことができる。
 - (g) 印刷が終了したときにそのことを知るためにはカメラ自身を印刷サービスに登録すればよい。
4. 印刷する
 - (h) ユーザーはカメラを使って印刷サービスの print メソッドを起動し印刷を行う。

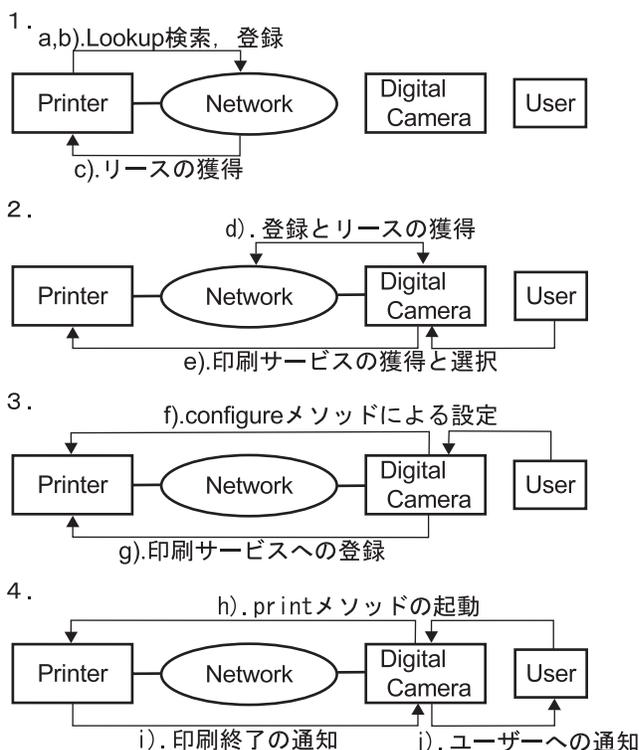


図 2: 動作の例

- (i) 印刷が終了するとカメラは印刷サービスから印刷終了の通知を受け取る。
- (j) 必要があればそれをユーザーに通知する。

5 Jini の課題

以上で述べたことより、Jini を使用すれば自発的で動的なネットワークを容易に構築できることがわかる。

しかし、Jini が動作するためには JVM が動作する必要があるのでハードウェアメーカーが JVM を実装しないと利用出来ない。既に同様の技術として情報家電分野での「HAVi」、Microsoft の提唱している「Universal Plug & Pray」が存在し、Jini が成功するためには技術的な問題だけではなく、近年の Java の普及の波に乗って如何にして他の規格よりもシェアを伸ばすか、もしくは他の規格との共存の道を歩むのが Jini の課題となっている。

参考文献

- [1] Sun microsystems 『JiniTM コネクション・テクノロジー』(Sun microsystems, 1999)
<http://www.sun.co.jp/jini/>