

INTERNET

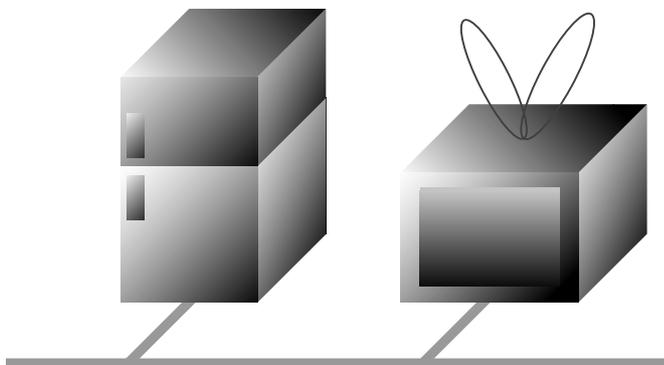
● インターネット最新テクノロジー : 第22回

すべての機器をネットワークにつなげる

Jini

Javaがもたらした衝撃は大きかった。Javaによって開発されたプログラムはCPUのアーキテクチャーに依存しない。そして、この利点はいわゆるコンピュータだけでなく組み込み機器まで広がろうとしている。Javaが真に目指していたのはネットワークを取り込むことである。今ここに、機器をネットワークにつないで簡単に動作させるためのものがJavaの利用によって実現された。これがJiniである。今回はネットワークの利用を促進させるJiniを取り上げて解説する。

宇野 浩司 日本サン・マイクロシステムズ株式会社



さまざまな機器がつながり出す

今やコンピュータチップの搭載された機器は、特定の専門分野にだけでなくオフィスや家庭に広く浸透した。そしてコンピュータをはじめとする機器を、ネットワークで結ぶことによって新しい使い方を生み出そうとしている。

インターネットによってコンピュータ自体の存在意義が変わったように、機器がネットワークにつながることによって、今まで持っていた役割以上の価値が生まれてくるはずである。たとえば、家庭にあるテレビやビデオ、エアコン、電子レンジ、湯沸し器、洗濯機などが

すべてネットワークで制御できれば、今までにない新しい使い方が出てくる。それは、さまざまな機器が相互に結びつくことによって1つのコミュニティとして働くことになるからである。これを簡単に実現してくれるのが「Jini」である。

Javaの利点を最大限に利用する

Jiniを理解する前に、まずJavaについて知らなければならぬ。なぜなら、JiniはJavaの技術に基づいたコミュニティを形成するためのアーキテクチャーだからである。

Javaのコンセプトとして「Write Once,

Run anywhere」(一度プログラムを書けば、どのプラットフォームでも動く)がある。これは、Java言語で書かれたプログラムならば、JavaVM(パーチャルマシン)さえ搭載されていけばどのようなアーキテクチャーのマイクロプロセッサを使っているかわからない機器でもプログラムの開発ができるようになる。たとえば、カメラやCDプレイヤー、テレビ、ビデオといったありとあらゆる機器もJavaVMが搭載されていけば同じようにプログラムが動くわけである。

また、Javaはソフトウェアを簡単に部品化でき、この部品化されたソフトウェア(クラス)はネットワークを介してJavaVM間を移動できる。したがって、クラスはどのマシン上にあってもよく、プログラムは必要なときに必要なクラスを必要な場所にダウンロードして動作できる(図1)。つまり、ネットワークにソフトウェアが分散された環境でもプログラムが動く仕組みを持っているのである。

複雑な垣根を取り払う

冒頭でも述べたが、Jiniはさまざまな機器やソフトウェアをネットワーク上で簡単に共有できる技術である。そして、ネットワークに散らばったこれらのものを結び付け、それぞれがうまく機能するようにコーディネートする技術である。

言葉を換えて言えば、あるサービスを受けようとするものと、そのサービスを提供するものとの間にある複雑な垣根(たとえば機器の特性やCPUの種類に対応したデバイスドライバのインストール)を取り払い、ネットワークにつなぐだけで両者のサービスのやり取りを実現させるのがJiniである。機器がネットワーク上のどこにあるのかわからなくても、何ができる機器なのかわかっていたら、機器同士を簡単に接続できる。

では、このような機能がどのように実現されているのかを解説する。

Jiniを演じる役者は3人

Jiniでは通常三者の間でやりとりが行われる。三者とはサービス提供者（アクセスされる機器）、サービス利用者（機器に対してアクセスする側）、そしてそれをコーディネートするサービス管理者である。

この三者の動きは次のようになる（図2）。

サービス提供者はネットワークにつながると、ネットワーク上のサービス管理者を探し出す（Join）。

サービス管理者は、サービス提供者のソフトウェアを登録できる環境を持っている。そこで、「私はあなたのソフトウェアを登録できます」という返事をサービス提供者に返す。

サービス管理者は複数いてもいいので、サービス提供者は返事もらったサービス管理者の中からソフトウェアを登録するものを1つ選ぶ。そしてサービス管理者にサービス利用者が自分とコミュニケーションするためのソフトウェア（プロキシ）を登録する（LookUp）。プロキシはJava言語で書かれたクラスの集まりである。

次にサービス利用者がネットワークにつながるとネットワーク上にあるサービス管理者を探す（Discovery）。このとき、無造作にサービス管理者をいくつも探してもいいし、自分が使いたいサービスを登録しているサービス管理者を探し出してもいい。

サービス管理者は、サービス利用者が望んだサービスが登録されていれば、「私はあなたが望んだサービスを登録しています」という返事をサービス利用者に返す。

と同様にサービス管理者が複数いれば、サービス利用者は複数の中から自分が使

たいサービス管理者を決め撃ちで選ぶ。もしくは、サービス管理者のリストから機器を使う人が選択してもいい。

サービス利用者側でサービス管理者を選ぶと、自分が利用したいサービスに合致したサービス提供者のプロキシをもらう（LookUp）。

サービス利用者がプロキシを受け取るとサービス管理者の役目は終わり、あとはサービス利用者がプロキシを使用してサービス提供者とコミュニケーションを直接行う。

以上が一連の動作であるが、これらは自動で行われるためユーザーは何も意識せずにネッ

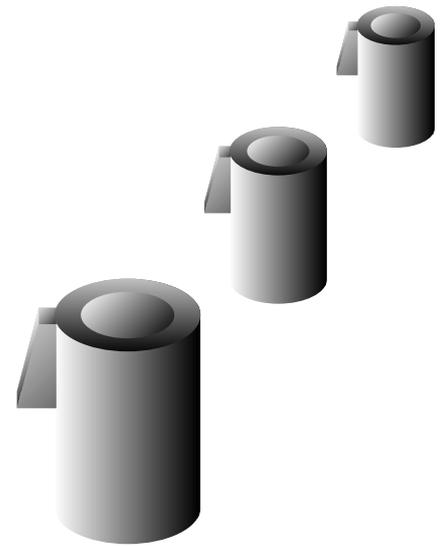


図1 Javaの動的なクラスのロード

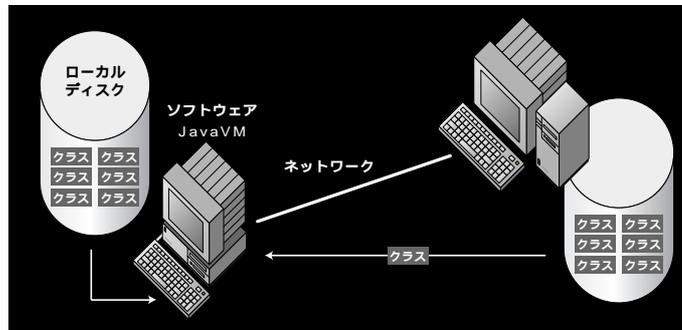
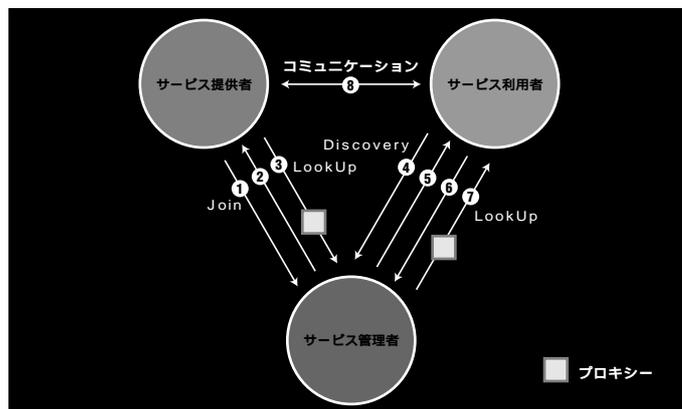
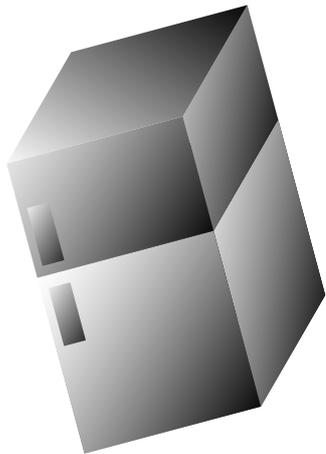


図2 Jiniの動作原理





ネットワークにある資源を利用できる。またJini環境ではサービスの管理者は必須ではなく、ネットワーク上に1つも存在しなくても、直接サービスの提供者と利用者がやりとりすることによってサービスを提供し利用できる。

プリンターから簡単に印刷できる

この例をネットワークプリンターの場合で考えてみる(図3)。コンピュータがサービス利用者でプリンターがサービス提供者の場合である。

Jini対応のプリンターには、コンピュータの画面イメージを印刷用言語(ポストスクリプトなど)に変えるソフトウェアや自分と通信するための情報(IPアドレスなど)を含む

Javaのクラスが組み込まれている。これがプロキシーとなる。

プリンターがネットワークにつながると、サービス管理者(LookUpサーバー)を見つけて出してプロキシーを管理者に登録する()。

コンピュータから文書を印刷する場合、プリンターがあるかどうかをコンピュータがサービス管理者に問い合わせる。そして、サービス管理者が自分に登録されているプリンターの一覧をコンピュータに返し、その中からユーザーは使いたいプリンターを選ぶ。

コンピュータはユーザーが選んだプリンターのプロキシーをLookUpサーバーを介して受け取り()、プロキシーを実行して画面イメージを印刷イメージに変換して直接プリンターにその内容を送る()。

このようにコンピュータには印刷のためのデバイスドライバがインストールされていなくても、プリンターにあらかじめ組み込まれているプロキシーを持ってくことで、印刷というサービスを実行できる。つまり、Jiniの環境さえあれば、ユーザーがプリンターのドライバをコンピュータにインストールしたり、ネットワーク上のどこにあるかを考えるような複雑な作業がいらなくなる。

役者全員が主役を演じる

ここで、コーディネーターとなるサービス管理者(LookUpサーバー)は巨大な保存領域を持つ必要はない。大規模にソフトウェアをサービスとして登録するのであれば、巨大なデータベースが必要となるが、単なるサービスの管理者であれば、PCのようなものでかまわない。もし、家庭のような環境であれば、セットトップボックスや電話のようなネットワークの入り口になるものがその役目を果たしてもいい。

一連の動作を現在のネットワークと比較してみると、Jiniではコーディネートする管理者の役目が終わると、先ほどのコンピュータとプリンターのようにサービス提供者とサービス

図3 Jiniを使ったプリンターの例

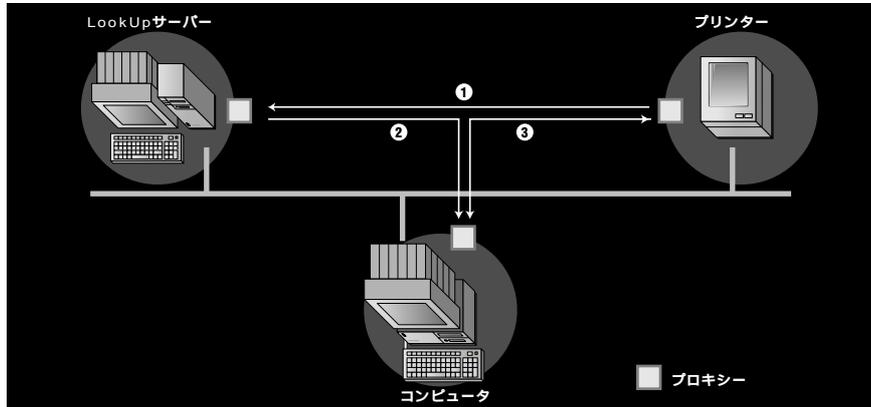


図4 3つのコンポーネント

	インフラ	サービス	プログラミングモデル
Java	JavaVM	JavaAPI	JNDI
	RMI	Beans	エンタープライズ Java Beans
	Javaセキュリティモデル	JTS
Java + Jini	Discovery/Join	Leasing	2 Phase Commit マネージャー
	Distributed Security	2 Phase Commit	Java Spaces
	Lookup	Events

利用者間で直接やり取りが行われるピアツーピアの関係である。つまり、Jiniでは全員が対等な関係になっている。これに対し、現在の特にPCのネットワークでは、機器（プリンターなど）は必ず管理者であるサーバーを介さなければならない主従関係である。

さまざまな適用が考えられる

例としてプリンターを取り上げたが、今までネットワークにつなぐことは考えられていなかったものにもJiniは適用できる。たとえばディスクやCD-ROMドライブもそうだが、カメラやスキャナー、PDA、または冷蔵庫やテレビもまったく同じように接続できる。カメラで撮影した、あるいはスキャナーで取り込んだ画像をネットワーク上のどこかのディスクに保存したり、PDAをネットワークにつなぎアプリケーションをダウンロードして使ったりすることが簡単にできる。

Jiniはインターネットやイントラネットのようなネットワークから1台のPCのような小さなモジュール単位のものまで使える技術である。Jini自体はトランスポートの通信プロトコルに依存せずに、ネットワークを介してリソースを共有するアーキテクチャーである。したがって、特定の通信プロトコルでネットワークの特性を活かした環境を作れば今までにないものが構築できるだろう。

Jiniを構成する

3つのコンポーネント

さらに踏み込んで、Jiniに含まれるコンポーネントについて解説しよう。

Jiniは3つのコンポーネントから成り立っている（図4）。3つのコンポーネントとはすなわちJiniを使うためのインフラとプログラムを作るうえでの取り決めであるプログラミングモデル、そしてこれらの2つを使ったサービスである。

Java自体が持っているコンポーネントを考

えると、インフラとはJavaVMやリモートプロシージャコールのためのRMI、セキュリティを取り決めるJavaセキュリティである。そして、プログラミングモデルであるJavaAPIやBeansを使ってソフトウェアを作ればすべてのプラットフォームで動くものとなる。また、拡張機能としてJNDIやEnterpriseJavaBeansといったサービスを提供している。

Javaをネットワーク環境向けに拡張

Jiniでは上記のJavaの環境に次のものが加わる。

インフラとして先ほど説明したDiscoveryやJoinといったサービス管理者やサービス自身を探すときに使われる共通のソフトウェアインターフェイスが用意されている。さらに、サービス管理者を見つけたあとにソフトウェアをアップロードしたりダウンロードしたりするためのLookUpというソフトウェアインターフェイスがある。

プログラミングモデルとしてLeasingというサービスの期間を限定できるインターフェイスがある。Leasingによって、サービス提供者がサービスを提供する時間を設定できる。ほかには、トランザクションの処理を規定する2 Phase Commitやネットワーク上の事象のやり取りを決めるためのEventsがある。

また、サービスとしてすでにあるものが、トランザクションを管理するための2 Phase Commit マネージャーとJava Spaces である。Java Spaces は実際のサービス管理者として、サービス提供者からのサービス（プロキシ）を保存したり取り出ししたりする役目を果たす。

以上のように、JavaはJiniを使うためのインフラであり、JiniはJavaの技術をネットワーク環境でより利用しやすくしたアーキテクチャーの総称であることがわかるだろう。

魔法の効力はこれから

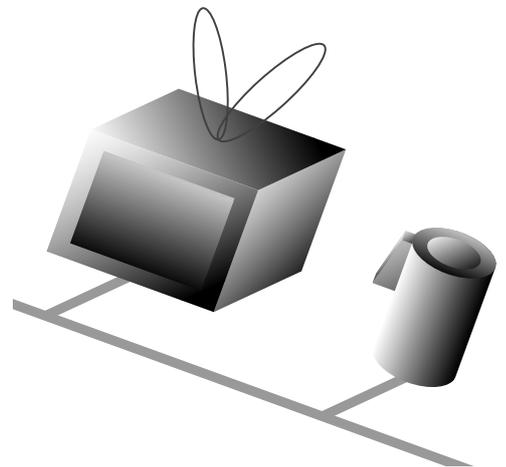
Jiniはこの8月に発表されたばかりである。現在サン・マイクロシステムズではさまざまな企業と提携してJini対応製品を出していく予定である。

Jiniはサン・マイクロシステムズのビル・ジョイによって開発された。彼はJiniによって変わるネットワークのビジョンを持っている（本誌278ページ参照）。

Jiniの語源はアラジンの魔法のランプから出てくる妖精「ジニー」である。妖精によってネットワークにかけられた魔法、すなわち多くのユーザーが待ち望んでいた「つなぐだけ」のネットワーク環境が何をもたらしてくれるかは、これからである。

参考 URL

URL <http://java.sun.com/products/jini/>





[インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

株式会社インプレスR&D

All-in-One INTERNET magazine 編集部

im-info@impress.co.jp