

第 部 基調講演議事録

第1章：モバイルソフト・モバイルプログラム-インターネットプログラミング の機動性と安全性

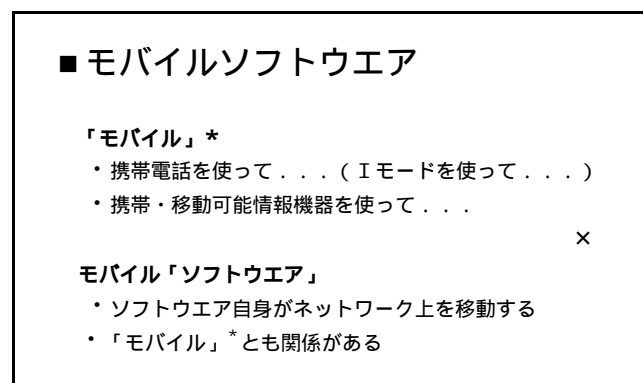
・開催日 2001年5月30日 報告者 米澤委員

【米澤委員】 ご紹介にあずかりました米澤でございます。

今日は少しテクニカルな話になるのではないかという気がいたします。ですけれども、私自身、いわゆるコンピューターソフトウェアをつくったり、その安全な作り方の研究をするなど、割と基盤的なことをやっておりまして、そういうものがどういうものかというのも何となくあまり世の中に知られていない。知る必要もないということもあるかもしれないのですけれども、そういうことでありますので、少し、技術に近い話題でお話をさせていただきたいと思います。

スライドの配付資料 16 枚、これはかなり抽象的に書いてあって、実はこの後に 20 枚ほど、もうちょっときれいなものがあるのです。これをお話ししようと思うのですが、大げさにいうと、特許ということですから、ある会社と、このソフトを世に出すという話があって、レポートで印刷されてしまうと何となく心配なところがありますので後の部分の印刷はしないということでお話しさせていただきます。

図1：モバイルソフトウェア



配付資料をみていただければいいと思いますが、ここでいっているモバイルソフト、あるいはモバイルプログラムというのは何かということ、ちょっとご説明したいと思

います。

いわゆるモバイルという言葉はネット社会では非常によく使われているわけですが、モバイルの後ろに*がついていますが、モバイル何とかと、いろいろな名詞が後につく場合があります。その場合に、一般的には携帯電話、あるいはPDAを使って、それからもうちょっと進化すると、例えば自動車の中にあるカーナビがそのままインターネットにつながってどうのこうのと、そういうことが世の中で一番いま、話題になっているのではないかと思います。ここで私が話そうと思いますのは、そういう携帯電話を使ってどうのこうのということがメインではなくて、いわゆるソフトウェア——ソフトウェアが何かというお話もしたいと思うのですけれども、コンピューターソフトです。アプリケーションといってもよいものかもしれません。それがネットワークの上をある意味で動き回っていく、そういう世界のお話をしていきたいと思います。これは、携帯を使ったモバイル何とかというものとも非常に関係してきますし、それをいろいろな形でもっと強化していくということとも関係します。

ソフトウェアの種類

これも釈迦に説法といいますが、私がこんなことを申し上げなくてもいいので

すけれども、ソフトウェア、略してソフトというと、普通、世の中で一般に使われているソフトというと、音楽ですとかビデオですとかゲーム、そういうのをソフトと呼んでいます。もっと抽象的にソフトというと、知識とか、そんなところま

図2 :「ソフトウェア」

- 「ソフトウェア」
 - 「ソフト」
 - ・ ミュージック、ビデオ、ゲーム × ()
 - 「ソフトウェア」
 - ・ プログラム
 - ・ アプリケーションソフトもその例
 - ・ その他、インターネットソフトウェア
 - エージェントソフトウェア、...
 - ウイルス
 - ・ プログラム+データ ○

3

でいってしまうかもしれませんが...。それで、ここで私がネットワーク上を動き回るソフトウェアというときにはそういう意味ではなくて、先ほどから何度もいっていますけれども、いわゆるコンピューターのプログラムのつもりでございます。我々、もっと専門的な呼び方をしますと、コードなどといったりしますが、ここで がついている

ものです。これでもちょっと違う意味なので、もう少し詳しくいいますと、ここで私がこの後、ソフトウェアといているときの意味は、いわゆるアプリケーションソフトというものに割と近いとお考えください。アプリケーションソフトというと、ワープロもそうですし、表計算もそうです。もちろん、それを移動しても構わない。あるいは数値計算のパッケージのソフトウェアなどというのも、移動させることです。

それからエージェントソフトウェア。エージェントというと代理人、人間の代理をしてくれるところの、そういうインターフェースもありますし、人間にかわっているいろいろな仕事をしてくれる、そういうエージェント、代理人のソフトウェアというのもあります。

それからウイルス。有用なものではなくて、逆に、非常に害であるわけですが、それも基本的にはプログラムです。これもネットワーク上を移動するわけです。いいか悪いかは別ですが、これもプログラムです。

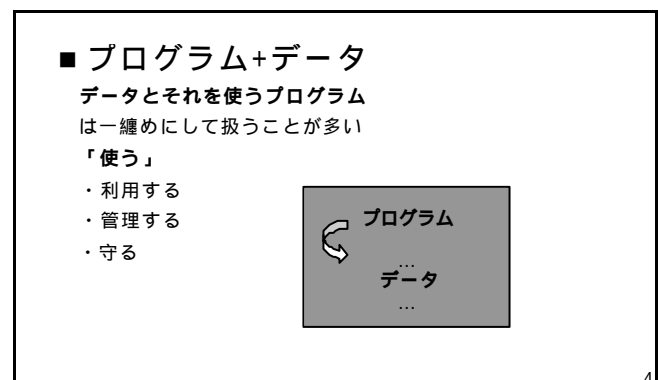
ですから、ここでソフトウェアといているのは、アプリケーションソフトみたいなものとか、インターフェースソフトウェア、エージェントソフトウェア、あるいはウイルスなども含みます。

プログラム+データ

もうちょっと技術的に正確にいいますと、ここでいうソフトウェアというのはプログラムとデータ。プログラムというのは、いわゆる手続が書いてあるプログラム。それとデータが一体化している、そういうものだと考えていただくといいかもしれません。

それで、ここでいっているプログラム+データという話は、右側の方に絵が描いてありますけれども、ここにデータがあります。このデータと、それを取り扱うプログラム、これがひとまとめでなつて、1つのパッケージならパッケージ、あるいは

図3：プログラム+データ



カプセルのようにになっている、そういうものがまとまって動く。

例えば普通のワープロソフトでも、あるいは普通の意味でのソフトであるミュージックのソフトでもゲームのソフトでも、あれもやはりデータと、それを使うプログラムのひとかたまりなわけです。ここで使うといていいますのは、そのデータを利用するとか、それを管理するとかも「使う」ことですし、あるいは、ある意味でそのデータを守る、その守るためのいろいろなことをしてくれるプログラムを守るということも、データを使う仕事の1つです。

あるいは、例えば音楽のデータの課金をする仕事も含まれます。ライセンスのことをそこに含めておくと、そういうことは全部データと一体になっている、あるいはデータを管理するところのプログラムがやるということにも、部分的に既になっていますけれども、そういうことがこれからどんどん進んでいく。

それで、ここでいうプログラムというのは
は手続のつもりですけれども、手続とデー

タというのは、これはどちらもいわゆる
知能とか知識とか機能、そんなものを表現
しています。プログラムも知識を表現
するわけです。データも知識を表現しま
す。それからプログラムもある種の機能
を表現していることになりますし、知識
も表現しているのです。一般に、プログ

ラムとデータというのは二元的に異なるもので、非常に違うものだ、全く独立した要素
があるというようなものに思われているのですが、実はここにありますように、どちら
もほとんど相互変換可能で、データで表現したものをプログラムで表現することも可能
ですし、プログラムで表現したものをすべてデータで表現するということも、ほとんど
の場合可能です。ですから、相互変換したときどちらで表現しておくかということが効
率的なのかという問題になります。それはメモリの容量の問題とか、関連して転送速度
の問題とか、あるいは管理のしやすさ、プログラムを書く上での書きやすさ、書きにく

図4：プログラムとデータ

■ プログラムとデータ

どちらも「知識」や「機能」を表現

プログラムとデータの二元性 (dichotomy)

プログラムとデータは相互変換可能なことあり

「プログラムの大きさ」/「データの大きさ」 (比率)

5

さ、そういうところで、ある機能とか知識というのを、どのぐらいの割合のプログラムとデータを使って全体を表現しておくかは、重要なファクターです。特に、ここでソフトウェアが動き回るというときに、どんな形で表現しておくのがいいかというようなことは、後々いろいろと問題になります。

ネットワーク上の移動

この辺も全く簡単な話ですけれども、インターネットに限らずネットワークの上の移動という言い方をさせていただきたいのですが、ネットワークというと、規模、その他によって3つぐらいに大きく分類していいのではないかと思います。いわゆる地球規模の非常に広域なインターネット、それにはPCがつながっていたり、サーバーというようなものがつながっている。それぞれの計算機の間、ある意味の

距離が非常に遠くて、情報の伝達の時間的な遅れが無視できないような場合です。

それからLANの話というのは、もうちょっと局地的な、地球規模ではなくて、1つの物理的な組織の範囲内ぐらいのものでしょう。

それからもっと計算機の中に入った話になりますけれども、普通のワークステーションをタイトにつなげて、それを1,000台、2,000台を合わせたもの、クラスターと呼んでいますけれども、そういうものをつなげたもので十分、昔のスーパーコンピュータの能力を超える能力がありますけれども、そういうのがまた互いに密につながっている。

図5：ネットワーク上の移動

■ ネットワーク上の移動

「ネットワーク」

- ・ インターネットのような地域・地球規模のもの
- ・ LANのような局地規模のもの
- ・ 密結合したプロセッサ集合

6

広域分散計算の基盤ソフトウェア

これも少し余談になってしまうのですが、基盤的な広域のソフトウェアをつくるときに、耐故障性ですとか、ネットワークでどの計算機とどの計算機がつながっているかというような状況で、それらの1つが壊れたりするので、つなぎ直しとか、どことどこのリンクを張り直すとか、いろいろなことを考えました。最近ですと、後でもちょっと出てきますけれども、つながるコンピューターの数で100万、1,000万、1億というような形になり得るということがあります。

それから、ここに「ごみ集め」とありますがけれども、計算機のメモリーというのは、ど

んなことがあっても有限なので、仕事に使わなくなったメモリーというのは、その時点でどんどん回収する。この回収の作業というのはソフトウェアで回収するわけですが、これは非常に重要な仕事です——要するに、メモリーをリサイクルするわけです。それをハードウェアでリサイクルするというのではなくて、実際に使っている計算機中の使用しなくなったメモリーを、またほかのことに使えるように再利用する。そういうのをごみ集めといいますけれども、そういうことがものすごく重要なファクターになっています。

なぜ移動なのか？移動の利点

ここで本題に戻ることになりますけれども、なぜ、ソフトウェアを移動するのか。ネットワークの上を移動する必要があるのか。移動するといいいことがあるのか。いろいろな要素がお互いに相互関連していますので、申し上げる順番というのはあまりはっきりしたものではないのですが、大ざっぱにいいますと、まずソフトウェアが移動することによって、全体の通信量、トラフィックが減る、あるいは減らすことができるということがあります。これはなぜかという、先ほどの、ある機能とかをデータで表現するのか、プログラムで表現するのかと。非常に巨大なデータがあって、それを扱うと

図6：広域分散計算の基盤ソフトウェア

■ 広域分散計算の基盤ソフトウェア

- ✓ 基本的な耐故障性はもちろんのこと
- ✓ ノードやネットワークの状態に応じたネットワークの動的な再構成
- ✓ 非常に多数 ($10^3 \sim 10^8$ あるいはそれ以上) の計算ノードを用いた並列計算のための基盤ソフトウェア
- ✓ 動的なネットワーク上での数々の分散アルゴリズム (ごみ集めや共有メモリ) などについて

7

ころの小さなプログラム、それでデータを全部送るよりは、プログラムを送りつ

図7：なぜ「移動」なのか？

けるか、あるいはデータのあるところにプログラムが移動して、そこで処理をした方が世の中にためになる、通信量が減るというような、これは1つ大きなファクターであります。いくらネットワークを太くするとかなんとかいっても、その時点ではある程度、高速道路ができると

■ なぜ「移動」なのか？

ネットワークを跨る通信量を減らしたい。
情報の到着を待つ間、別の仕事をしたい。仕事の負荷を分散したい（仕事をしながら）。
移動しないと見付けられない情報を使いたい。
移動先にしかない資源をきめ細かく／柔軟に使いたい。
料金が安いところで仕事がしたい。
安全な場所で仕事がしたい。

8

最初は調子がいいのですけれども、必ずトラフィックはいっぱいになるというわけで、通信量を減らす技術の研究というのはいつでもやっておかなくてははいけません。現時点でもそうです。

それから、自分が計算機を使っていて、別な仕事をほかの計算機に頼んで仕事をやらせよう。頼むぞといっておいて、なかなか結果が戻ってこない、その間、ずっと待たされて、自分の手元でやっている仕事が先に進まないために、その結果とか情報を待っている、この待ち時間を減らしたいという、そんなことです。

また実際に、これは人間の世界でもあると思いますけれども、どこかに、物理的にある場所に移って見ないと、なかなかみつけれない情報というのがあるわけです。遠くから電話して、人に聞いているだけではわからなくて、やはり現地に行って情報収集してくるというようなことが、人間の世界でもあるわけです。それと同じことが計算機のソフトウェアの世界でもある。その人間に例えるところのソフトウェアが実際にその目的の場所に行って、それでいろいろな情報をかき集めてくる。あるいは、そこで得られた情報を使って、また別のところへ探しに行く、そういうことが必要になってくる。

後で簡単な例が出ますけれども、物理的にそこに行かないとないような、ある場所に移動しないと存在しないような資源というものが実際あるわけで、例えば前にもいったかもしれませんが、天体望遠鏡が、世界で一番いいのが1つしかなくて、そこに行かないと仕事ができない。その天体望遠鏡を操作するプログラムを手元に置いておいて、そ

れで遠隔に操作すると、なかなかきめの細かい、しっかりした、やりたいことができないので、手元で操作のためのプログラムを、ソフトウェアをつくっておいて、それを天体望遠鏡のあるところに送りつけて、そこでその天体望遠鏡のサーバーみたいなものと、手元で書いたプログラム、ソフトウェアをその場所に移動させて仕事をさせる。その結果を送り返す。

それから、あとは料金が安いところで仕事をしておきたいとか、安全なところで仕事をしたいというようなことが、この辺も説明するといろいろ要素はあるのですけれども、時間もないので……。

移動の種類

移動の種類。ソフトウェアが移動する
移動の種類というのは、例えばダウンロ

図 8：移動の種類

ードする、自分の手元にソフトウェアを
どこかのサーバーから送ってもらう。こ
れは A という計算機にあるソフトウェア
を B に移動させる。それから A にあるソ
フトを、同じ A にあるソフトウェアが、
どこかに送りつける、すなわち、押しや
るプッシュ型です。ダウンロードして手
元に持ってくるのと、どこかに送りつける。

■ 移動の種類

ダウンロード型：

計算機 A に在るソフトを、計算機 B にあるソフトが
A に移動させる。

プッシュ型：

計算機 A に在るソフトを、同じ A にある別のソフトが
別の計算機 B に移動させる。

自立型

侵入型

9

もう 1 つは、ここにありますが、プログラム自身が自分で自律的に移動していくという
場合があります。それから、これはちょっとタイプは違うかもしれませんが、侵
入型といいますか、ウイルスとは少し異なり、ハッカーとかクラッカーがプログラムを
送りつけてきて、何か悪いことをするという場合です。

資源の種類

このスライドには、先ほど、ある場所に来て、そこにある資源を使おうという話をし

ましたけれども、その資源が何かということが書いてあります。資源の1つとしては非常に巨大なCPUを持っている場合がありますし、すばらしいデータベースみたいな、いわゆるデータコンテンツというのも資源です。それからさっき天体望遠鏡の例を出しましたけれども、その場所に特有に存在している装置とかも資源の1つです。それから、ちょっと種類が違うかもしれませんが、情報のブローカーみたいなもの、これも非常に重要な資源であります。

図9：「資源」の種類

■「資源」の種類

CPUによる計算力
データコンテンツ
その場所特有の機器/装置
情報ブローカー

10

いろいろ、例を用いての説明で話をしてきました。これはわかりやすいかわかりませんが、いま、どんな話が非常に盛んかといいますと、例えば100万個の、家庭用で世界中に散らばっているPCで、使っていない、あるいは部分的に使っていない、あるいは、ある時間帯は使っていない、そういうPCにプログラムを送りつけて、そこで実際、余っているCPUを使わせてもらう。こういう方式を提供する会社があります。アメリカの会社ですけれども、自分のPCをこう

いう形で使わせてもいいよというのは、ちゃんとお互いに合意してやらなくてはいいけませんから、メンバーの登録をして、自分のPCを賃貸し、あるいはボランティアで貸すというような「使わせる」という方式が進んでいる。これは後で出てきますが、非常に怖いことですけれども、そこでウイルスが入ってきたり、ウイルスのみならず変なプログラム

図10：分かり易い例

■分かり易い例

1,000,000,000個のアイドルなPCにプログラムを送って計算・情報処理をしてもらう。
検索プログラムがサイトを巡って情報を収集
特殊天体望遠鏡のきめ細かい操作プログラムをサーバに送って、そこで実行
巡回型ソフトウェアインストール・アップデート
トロイの木馬

11

が入って、自分のデータを壊されたら困りますので、安全性がちゃんと保証されているという仮定のもとに、自分のPCを賃貸します。

いまこれは、例えばエイズのための薬、免疫系なんかをやれとか、非常に高度な処理

をするためにということで、ボランティアとPCを貸す人々がたくさん募られています。

それから情報のいわゆるサーチエンジン、グー（goo）だの、ヤフー（Yahoo）だの、google だの、いろいろありますけれども、世の中のサイトを手元にかき集めてやっているというのが現状の方式です。新しい方式では、サーチエンジンみたいなもの自身がネットワーク上を動き回る。それで情報を探してくるというような、そういうことのテクノロジーが進んでおります。

いま、いくつかわかりやすい例を言ってきたわけですが、応用ソフトウェアのインストールをする、アップデートをするといった移動巡回型のソフトウェアを我々はつくっていますので、それがどんなものかというのをお話ししたいと思います。それも一つの例ですし、クラックするためのトロイの木馬的な、悪意のあるプログラムというのも、移動するソフトウェアのわかりやすい例ではないかと思います。

メタサーバ

時々、例えば計算機のソフトウェアの話で、クライアント・サーバー・モデルというのをお聞きになることがあると思います。サーバーというのは要求を処理してくれるコンピューター。そのコンピューターに、ちょっとだけ遠くからメッセージを送って何かをしてもらう。このメッセージを送る方がクライアントです。現在の方式ですと、サーバーの方に

大きなプログラムがあるソフトウェアがあって、クライアントの方は比較的小さなプログラムです。サーバーが持っているインターフェースだけではお仕着せのインターフェースしかなくて、十分なサービスが得られない。それを、ちょっと考え方を变えて、いままでのクライアントサーバー方式ではなくて——いままでのクライアントサーバー方式では、サーバーの方に割と簡単なメッセージを送るということを何度も繰り返してき

図11：サーバにプログラムを移動して実行

■ サーバにプログラムを移動して実行 メタサーバ

サーバが持つインターフェースだけでは良いサービスが得られないことがある。
ネットワークが遅いとサーバを呼び出していると効率が悪いことがある。



・プログラム自体をServerに送信する。

12

たということになりますけれども、このモバイルソフトウェアの話ですと、サーバーの方にプログラムを送り込む。そういう意味では、サーバーは非常にジェネリックなもの、あるいはメタ的なものになって、よりいろいろな機能が柔軟になるといったらいいでしょ

うか。

簡単な絵（図 12～15）を描いてきたのですが、割と細かい話になってしましますが、クライアントの方にあるプログラムが遅いと、サーバーの方に移って仕事をする。今度はサーバーの方にたくさんプログラムが移ってくると、こっちも混んでくるので、またクライアントの方に戻してやるというようなことを、かなり柔軟にやっていくということはやられるようになってくると思うのです。

それから、クライアントサーバーというと比較的近い距離を隔ててやっているようですが、こういう話がグローバルに起こり得る。大きな地球規模のスケールで起こるということになります。

ここからが入っていない部分なのですが、同じような話を二度聞くのも何でしょうが、もうちょっと具体的な例でいうと、先ほど言いましたように、今だと携帯電話でゲームソフトがぱっと移動してくる。これも一つのモバイルソフトウェアの例です。

それから、先ほど旅行の手配をしてくれるソフトウェアみたいなのも、実際に現地に行

図12：ストレージサーバを利用するプログラム（１）

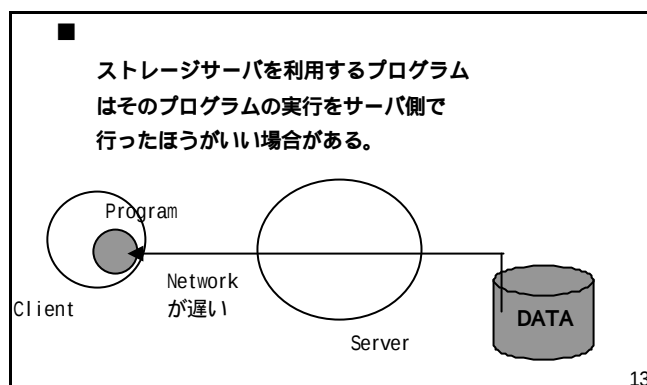


図13：ストレージサーバを利用するプログラム（２）

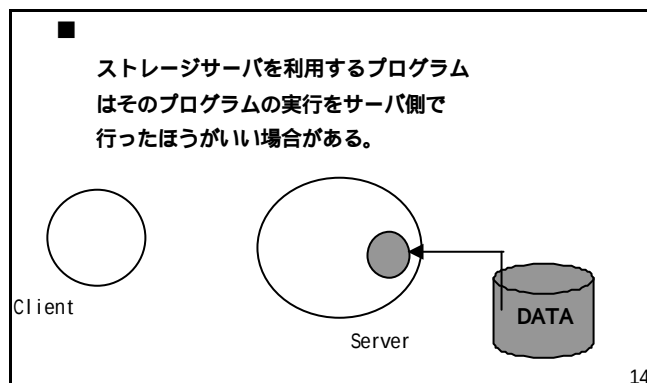
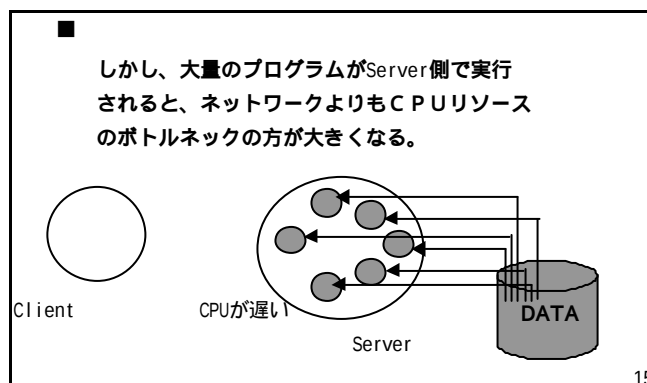
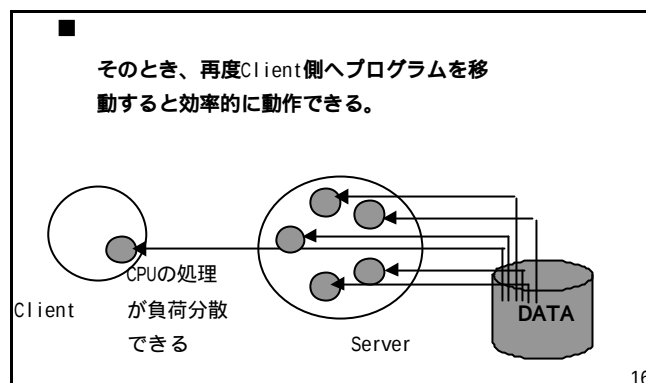


図14：ストレージサーバを利用するプログラム（３）



って情報を収集してくるというようなことを、従来のものと違ってそれをやる。

図15：ストレージサーバを利用するプログラム（４）



セキュリティ上の問題

ここで一番怖いのは、外から得体の知れないプログラムが、ソフトウェアが入ってくるわけですから、それがどんな悪さをするかわからない。ここの対策をどのようにするか。それから、これだけではなくて、逆に自分が送り込んだり、送りつけた、そういうソフトウェアが人のところに行って殺されてしまう、あるいは改編されて戻ってくるなどという、これを悪意のあるホストといいますけれども、それに壊されてしまうというような危険もあります。ですから、セキュリティの問題をどのように解決するか、いろいろと研究が進んでいるところです。ご存じのように現実にあったお話ですが、誰かがいたずらでゲームソフトと詐って、iモードでゲームだと思ってダウンロードしたら、警察に電話してくれるプログラムになっていたと、そういうのが新聞に出ていたりしました。

具体例・モバイルインストーラ

例えばネットスケープとか、あるいはワープロでもいいですけども、どんどんバージョンが上がってくる。それを一々自分でインストールなどをするのが全く苦手な人もあれば、そうする時間も全くないような人もいます。そういう人を助けるために、LAN

内のPCに必要なソフトウェアを自動的にアップデートして、インストールしてくれるとありがたいです。それを、ここでいう移動モバイルソフトウェアとして実現するお話を少しさせていただきます。

ここでは、例えばワープロのソフトを入手すると、それをLAN上のPCを訪問しながらインストールしてくれるソフトウェア・システムの話です。それは、インストールとかアップデートしたいソフトウェアのコンテナみたいなものですが、そのコンテナ自身もソフトウェアなので、それがアプリケーションソフト（例えば、ワープロ、ネットスケープ）を、LAN上のPCまで、ネットワーク上を移動してインストールをしてくれて、それが終わると、また次のPCまで行ってインストールしてくれると。それぞれ、このマシンは誰それの使っているPCだから、どんな方式でインストールしなくてはいけないとか、あるいは現在、どのバージョンが使われているから、次はどのバージョンをインストールするか自動的に決めてインストールしてくれる。そういうインストールをするプログラムがネットワーク上を動き回る。いままではシステム管理者が、一々それぞれのPCのところに行ってインストールをしていたのを、かわりにこのモバイルインストーラーというのがやってくれる、そういう話なのです。

これも、自分のマシンの中のソフトウェアが自動的にアップデートされてしまうということで、ひょっとして変なものが入れられてしまうかもしれないということがあるので、そのためのセキュリティが重要になってきますし、新しく入ってきたソフトウェアがライセンス上でほかに漏れたりしないように暗号化してやるとか、このモバイルインストーラーを扱っている人が本当に本人なのかということをチェックする、そういう認証的な話などもすべて組み込んでおります。

これも一つの例です。大体こんなところで終わりたいと思うのですが、もっと実際にどんな形で動くかというようなことは、この後に絵があるのですが、それには、基本的にはいま申し上げたようなこととか、どういうプロセスで行われるかというのが書いてあります。

【質疑応答】

【公文委員長】 うろ覚えなんですけれども、たしかいま、アメリカで1つ法案が出て

いて、この手の、売り切りではなくて、全部、ネットワークの上でアップデートするような形で商取引を行わせることを直接するのか、そんなような、それで非常に議論的になっているというニュースをみたばかりなんです、逆にこの場合の問題点はどういう.....ユーザーの側からすると、要するにお金をふんだかれる議論もある。

【米澤委員】 いま、おっしゃっているソフトウェアというのが、こういう実際のプログラムか、それとも音楽のとか.....

【公文委員長】 プログラムです。

【米澤委員】 相手がわからないからという認証性の問題をどのように解決しているのでしょうか。ちょっと調べる必要があります。

【公文委員長】 しかし、実際、売り切りをやめてしまって、買ったら全部、それが1週間おきにアップグレードされていくと。1週間たったらいまのが使えなくなるというようにしてしまえば、ほとんど海賊版はつくれませんよね。

【米澤委員】 ですから、あと何日間もつとか、あるいは何回使ったとか、そういうことをうまく記憶しておくためのいろいろなテクニックがあります。しかし、そのような方式が本当に市場で受け入れられるかどうかというのはまた別の問題なんです。でも、技術的にはそれはいろいろ考えられています。

【伊藤委員】 たしかマイクロソフトでも、次のX Pのバージョンからすべてそうです。ネット上でレジスターしないものは自動的に終わっちゃうというように、次のバージョンのオフィスはできているし、それとマイクロソフトのドットネットの戦略と、それとこの間、C . バルマン (C. Borgman) もいっていましたけれども、とにかくXMLはパソコンの発明と同じぐらいインパクトがあると。ビル・ゲイツはマイクロソフトの会社ごと、すべてXMLにかけていると。そのXMLは何かというと、これの土管なんです。それで、ドットネットの戦略というのは、まさにプロダクツからサービス、このように切りかえて、その認証機関も全部マイクロソフトが握ろうとして、全部マイクロソフトワールドにつくろうとしているのをいま、やっていると思うんです。

モバイルコードと信用

【伊藤委員】 そうですね。僕がすごく重要になってくるところは、こうやって

くると、いまのアイキャンの話に近くなってくるんですけども、どこに信用を置くかということなんです。信用しないモバイルコードは信用されていないネットワークには入っていけないわけですので、1つおもしろいなと思うのは、2004年か2005年ぐらいでインターネットの半分以上が中国人になるんです。中国のユーザーが、アメリカのユーザーも2004～2005年で超えてしまうわけなんです。そうすると、もううちのネットワークはマイクロソフトは入れないよとかいうこともできるわけなので、だからWTOとかを中心に、こういうモバイルコードと認証のコンソーシアムが出てきて、それはメンバーとかノンメンバーとか、そういうレベルになるのではないかと思いますけど。

【米澤委員】 要するに普通の、ネット社会でない社会でも信用関係というものは当然出てくるわけで、あるいはそれを技術的にどこまで押さえて——完全に押さえきれないわけですから、それを社会的なところの面でどうやってまた規制していくかという、そういう問題になってくると思いますけれども。

【公文委員長】 それが嫌だと思ったら、ネットワークから自分でつくるしか仕方がなくなるということですかね。例えばパソコンは旧型のパソコンしか持たないで、うんと古いソフトしか使わない。それでネットワークにはつながない。つなぐと壊されてしまうからね。

【伊藤委員】 それかマック（Mac）を使うとかね。だから、例えばいま、リナックス（Linux）上での世界というのも、結構そこがいろいろあるので、たぶんネットワークにつないでいても、ハードが同じでも、OSだとか、例えばJavaを使うか使わないかとか、いろいろな選択肢はあると思うんです。たぶん、コミュニティごとに自分に一番特化したシステムを使って、そのシステムのゲートウェイというのができてくると思うんです。だからカレンシーみたいなものになるんじゃないかと、私は思うんです。

【米倉委員】 それはおもしろいですよね。ユーロ圏とか달러圏と同じで、自分がこのネットワーク——ただそのときに、やっぱり心の底で1週間ごとにバージョンアップされては困るなというような、マイクロソフト型のが本当に支配するのか。結局、そんなのはフリーソフトウェアで止められないわけです。MPCだのナップスだのと、そういうカンパニーが次々出てきて、もう勝手にやってくれと出したら、もう誰も止められない世界ができてしまうんです。インフラが整備されて、ブロードバンドになってしま

って……。そんなときに全く違うビジネスモデル、要するにそんなもので儲けようと思わない世界が出てくるのか、あるいはそういう帝国の方がある意味でアナーキーだから怖くて、いいこともあるけれども悪いこともいっぱいあると。だから、お金を払ってもクローズドな世界に住みたいとか、そんな住み分けになってくるんですかね。

自由とインターネット

【伊藤委員】 この間、インターネットのガバナンスの会議に、ロスで出てきたんですが、ランドコーポレーションとか、最初のころにつくったおじさんたちがみんな出ているんだけど、彼らが言っていてすごくおもしろいポイントがあったのは、最初、インターネットをつくったときのユーザーというのは、大体自由が欲しいから、アナーキストですよ。でも、いまのユーザーとサービスを提供している業者というのは、特にその部分はインターネットに魅力を感じていないので、全部ATMとサーキットで全く自由がないネットワーク、ナップスターが存在しないネットワークに切りかえても、たぶん99%の人は損した気分にならないだろうと。だから、ある種の技術進化はインターネットによってできたけれども、例えば原子力と同じように、もう学者の世界からコマーシャルの世界、もしくは政府の世界に移ったので、このまま、ラジオももともとパケット通信とかで、みんなで自由にやろうといったら、結局終わってしまったじゃないですか。だから、インターネットのラジオ化というのが、やはり彼らはすごく恐れていて、可能性はなくはないと思うんです。

【米倉委員】 ただ、99%は困りもしないし、あまり考えないといいつつも、やっぱり自由がもつパワーというのはあって、1%にコンタミネートされると、みんな何となくうつってしまうというおもしろさがある。でもこれは戦いですよね、いかにマイクロソフトを倒すか。

【公文委員長】 そっちの流れは本当に20世紀のマスメディア型に逆戻りするという話でしょう。有料化して、1方向になってがちがちに締められて……。その限りではいいと思う人もたくさんいるのかもしれないけれども、さて……

【米倉委員】 安全安心だし、電話代もかかっているのだから同じだと思うかもしれないしね。

【伊藤委員】 あまり自由でない中国に僕がすごく期待しているのは、マイクロソフトを倒して得するのは中国なんです。アメリカは、やはり自分をたたくようになっているので、中国が例えばリナックスを本気でやって、このOSという部分はみんなにオープンにしよう。しかも中国語を中心に考えようというようにやってしまえばおもしろいんじゃないかと。アイキャンなんかどうでもいいと。中国を中心に、中国にはみんなDNSをうちが仕切るとか、そういうことをやり出すと、結構マイクロソフト戦略というのはうまくいかないと思うんです。

【公文委員長】 また東西陣営の対立に……

【米澤委員】 また冷戦の状況になりますね。

【米倉委員】 中国は確かに一番損していますものね。でも、マイクロソフトだって、最近使っているお金のほとんどはそういう途上国での海賊版対策で、もうイタチごっこでしょう。だから彼らはやはりネット支配に行くんです。

それで、ついでに話を聞くと、モバイル、結構日本は何とかだといっているんですけども、そのたくさんの覇権争いの中で日本の位置というのはどんな感じなんですか。

日本のモバイル

【米澤委員】 冒頭にちょっといったんですけども、日本のモバイルというと携帯、iモード、その技術が進んでいるといわれていますが、それは狭い意味で進んでいるんだと思うんです。もっと広いいろいろな技術があるわけで、ですからそのiモードというのはインターネットの使い方を非常に限定しているわけです。それに日本が勝っていると思ってやっていると、iモードだけでがんばっていてもあまり誇れないのではないのでしょうか。

【米倉委員】 実際にいうほどあまり強くないと。このASPで、もう本当に500ドルパソコンって、一瞬笑って捨てられたけれども、また出てくる。要するにいまのモバイルはそうなるわけでしょう。しょっちゅうつないで……

【米澤委員】 次世代の。

【米倉委員】 全部もらってくるソフトからいって、その辺はあまり強くないというように考えていいんですか、日本は。

【米澤委員】 特に進んでいるというわけではないと思います。

【伊藤委員】 おもしろい会社があるんです。小さいところ、例えばアプリックスとかアックスとか、ああいうデバイスドライバーだとか、あとリナックスのカーネルを積むとか、アメリカには当然あるんですけども、あそこはかなりノウハウがある。

あと、唯一といっちはいけないですけども、トロンが、あれはかなりトロンをうまく利用して、珍しく価値を生んだ政府のプログラムだったかなと思ったんです。

【米倉委員】 アプリックスは結構頑張っているところです。でも、これは結構ストラテジックな分野だから、iモードなんかも大事だけれども、こういうところにもっと...。でも、あまり覇権争いに入ってもしょうがないのかもしれないですけど。

【公文委員長】 今日のお話は、人によっていろいろな使い道があると思うんですが、しかし誰がどういう目的でまずそれを使い出すかによって、大変大きな社会的なインパクトが出てくるでしょう。

なぜモバイルなのか

【野村】 1つだけ、素人の質問をさせていただきますけれども、なぜモバイルかと。通信量を減らすためであると言われていたんですが、普通のソフトのイメージだと、いわゆるアプリケーション・プロバイダー・サービスというんですか、A P S、あれで、結局いままではデータだけ送っていたのが、プログラムも送られてくる。だから、むしろ通信回線上の交通量がふえるんだろうと、私は思っていたんですけども.....

【米澤委員】 そうではなくて、やはり大きなデータがどどっと行ったり来たりするということでしょう。

【野村】 それを、例えば圧縮プログラムと一緒にくっつけて、そうすれば減るということなんですか。

【米澤委員】 そうですね。データとプログラムの普通の大きさというのは、何万倍の違いがあったりしますので.....。

【伊藤委員】 単純にXMLを分析すると、少しそういうところがありまして、例えばウェブページの表示をするスタイルシートを先に送って、変わるところだけのデータをやりとりする方が全然軽いわけです。いま、例えばトランスマッシュという会社がある

んですけれども、これはやはりモバイルにリナックスを動かして、彼ら、XMLのエンジンをサーバーに置いているんです。何をするかというと、ウェブページを引っ張ってくると、HTMLを全部捨てて、XMLの圧縮したバージョンになって、そのXMLをJAVAのアプリケーションに投げて、そのJAVAのアプリケーションがそのXMLをまた表示してしまうんです。ただ、今度はローカルで全部、フォントだとか色とか全部入っているので、そうするとものすごい、ナノバンドでできてしまう。これもすごくシンプルなバージョンなんですけれども、とにかく余計なものは全部捨ててローカルにやってしまうと。

——了——