

第18回R&Dシンポジウム 基調講演

「情報通信技術の動向と展望 “Open”, “Smart”, “Standard”」

東京大学大学院 情報学環 教授
越塚 登 氏

1994年東大院理学系研究科博士課程修了（情報科学専攻）、博士（理学）、1994年東工大院情報理工学研究科・助手、1996年東大院人文社会系研究科助教授等を経て、2009年同大学院情報学環教授。なお、2002年3月より、YRP ユビキタス・ネットワーク研究所・副所長を兼務。専門は計算機科学、特に、ユビキタス・コンピューティングやオペレーティング・システム、コンピュータ・ネットワーク、ヒューマン・インタフェース等の研究に取り組んできた。総務省・「ICT国際競争力会議・基本戦略分科会」委員等、多くの政府委員も歴任。



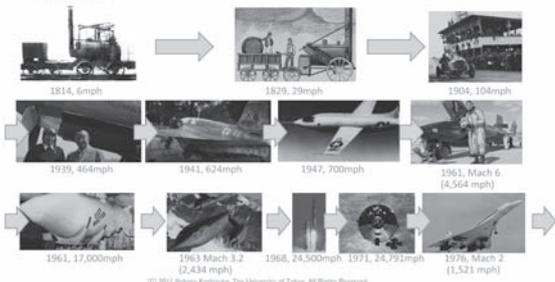
1. はじめに

近年の情報通信技術 ICT の動向と展望ということでは、いろいろなことを話したいと思いますが、今日は“Open”、“Smart”、“Standard”、この3つのキーワードから将来的なことを含めお話しします。

まず、最初に「進歩が止まった時代」ということで話をしようと思います。ICTと言われると、たぶん非常に先進的な分野で、ハイテクで、イノベティブで、実は技術がどんどん進んでいると多くの方は思われていると思います。確かに社会に対する影響力であるとか、我々の生活を向上させる力は、どんどん加速していることは確かだと思います。ですが、実はコンピュータの本当の技術的な進歩に関しては、最近、特にここ5～6年、進みが遅くなってきているのではないかとされています。最初にこのことから話をいたします。

Wall Street Journal, April 7, 2011の記事より

▶ “Putting on the Brakes: Mankind Nears the End of the Age of Speed”
■ As Space Shuttle Nears Retirement, Flyboys Pause to Reflect on a Slowdown”



今年面白い記事がありました。『ウォール・ストリート・ジャーナル』という、株取引をやられている方だとよく見ておられる誌と思いますが、この誌において、鉄道、輸送など、交通の分野の話がありました。人間が移動することに関する高速化の歴史についての話で、産業革命あたりの1814年に最初の鉄道の機関車が出て、1時間あたり6マイル、つまり時速約10キロでした。時速約10キロ

2005年のロードマップのようにではなく、2004年の時点で1つ開発計画を中止してからというもの、この辺をうろうろしています。2007年になってロードマップを書き直したのですが、これも計画どおりにはいかないようです。LSIの進化の法則をゴードン・ムーアという方が論じ、ムーアの法則と言いますが、それに沿わなくなっている。物理的な限界だと言われています。だから、コンピュータは、相当のイノベーションがない限り、これからそんなには速くはなりません。コンピュータが速くならないとはどういうことかという、今後通信も速くなりません。なぜかと言うと、通信は光で行っていますが、それを切りかえるところはコンピュータが行っているからです。これが速くならなければ通信も速くなりませんから、情報通信の速度は頭打ちの時代を迎えていると言えます。これは先ほどの、交通の高速化と少し共通点があるのではないかと思います。

これが意味することは？

- ▶ 技術の進歩が止まり、技術開発が不要ということではない。
- ▶ 単に、より速く、より大きく、といった単純明快な重厚長大型技術目標が終焉
 - ICT分野で、唯一のこっている重厚長大型単純明確分野、パソコン
- ▶ より複雑な技術目標を設定し、それをクリアする取組みへの転換期に
 - コストパフォーマンス、信頼性、合目的性、標準化、新しい価値の創造、
→ 最終的には総合力
- ▶ 日本の技術者は単純明快な目標をこなすのは得意だが、目標自体が複雑化した時に、とまどうところがある。
 - ICT分野でみられる現象

©2011 Nakano Kazuhiko, The University of Tokyo, All Rights Reserved

このことから私が申し上げたいことは、これで技術の進歩が止まったということではなくて、また、止まってから技術開発が不要というわけではなくて、単に、より速く、より大きく、といった単純明快な重厚長大型の技術目標というのは、交通の分野もコンピュータ分野も終焉してきているのではないかと考えています。ICTの情報通信の分野で唯一残っている単純明快な重厚長大型の分野は、スーパーコンピュータぐらいだと思います。そうすると、ICTもより複雑な技術目標を設定して、それをクリアする取組みをするという転換期にあります。それはコストパフォーマンスであったり、信頼性であったり、合目的性であったり、標準化であったり、または新しい価値の創造であったりと、よ

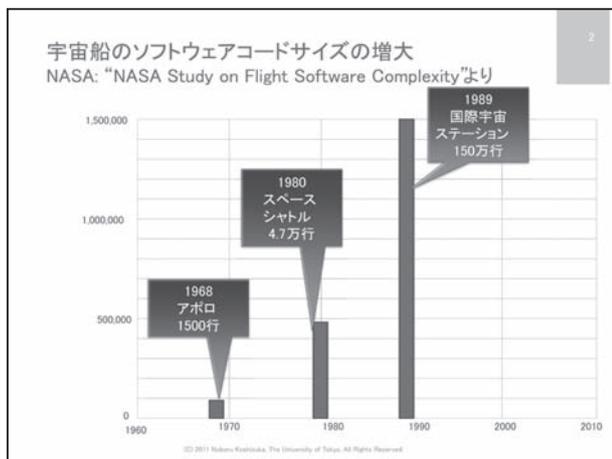
り具体的に言えばテクノロジーだけではなくて、社会システムとか企業とかビジネスなど、最終的にはこれらの総合力といったものが求められているわけです。

ここで私が非常に不安なのが、日本のICTの分野だけではなくて、全ての理系の分野がそうですが、日本の技術者は、単純明快な目標を、例えばより速くとか、より高くとか、より大きくこなすということは非常に得意です。いろいろ言い方はありますが、人から目標を与えられてそれをこなすのは得意といえます。

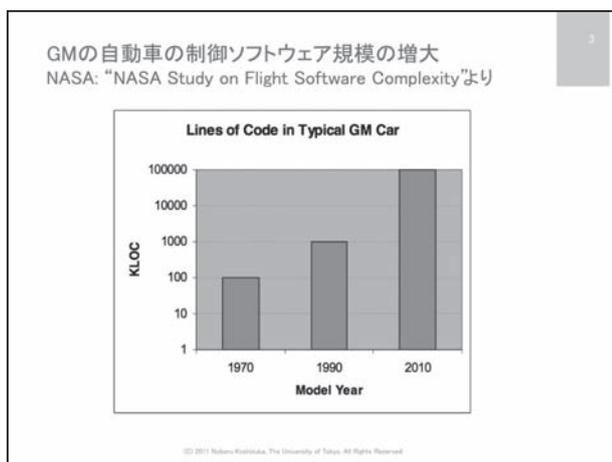
しかし、ここに少し挙げてありますように、目標もコストパフォーマンスとか、いろいろ複雑なことを言うと、やはり日本の技術者は戸惑ってしまうところがあるのかなというように思います。そこで、今日は少し技術目標がより複雑な話になってきている、特に信頼性などに特に着目して話をしたいと思います。

ICTの現状と展望ということですが、これもいろいろとすることができます。面白おかしい話もたくさんあります。きょうは改めてものすごく堅い話をしたいと思います。現状と展望というのはいろいろな面があります。けれども、ここでまず1つ注目しようと思っているのは、ソフトウェア規模の増大ということなんです。これは、鉄道分野と非常に関係があるということで調べてきたのが、NASA（アメリカ航空宇宙局）の資料です。鉄道も制御システムが過去から何十年もの歴史があって、非常に大きな資産になっていると思います。それは、制御システムという分野では宇宙でも同じです。NASAの中のいろいろな宇宙開発におけるソフトウェアのコードサイズというのは非常に大きくなってきています。

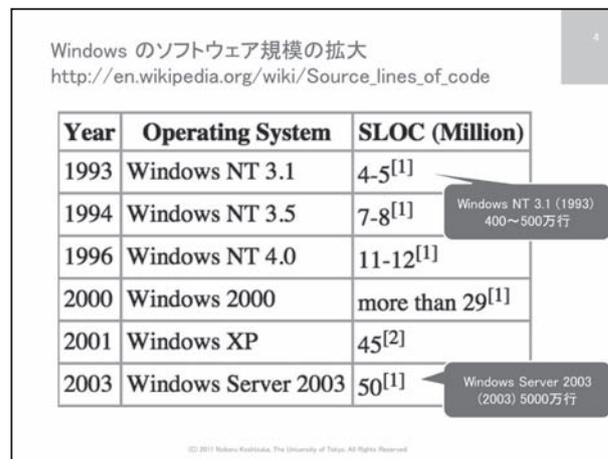
1968年がアポロ計画のソフトウェアの規模、次がスペースシャトルの規模です。そしてその次が宇宙ステーションの規模です。これを対数軸ではなくて、普通のスケールにすると、20年間の間で膨大に増えていることがわかります。



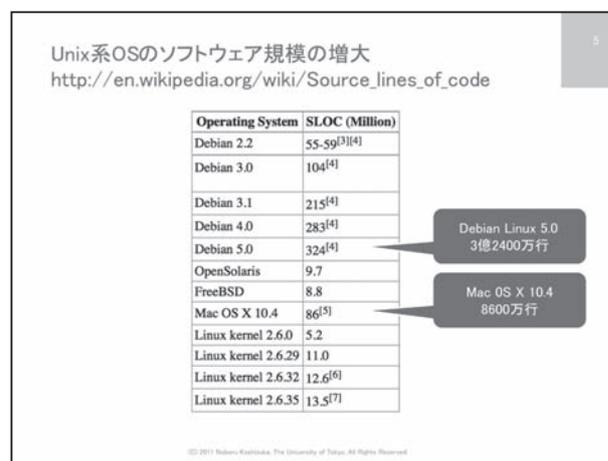
宇宙の分野だけではありません。今日のパネリストに日産自動車株の方がいらっしゃいますけれども、自動車の分野も同じです。次はアメリカの自動車会社 GM 社の資料についても説明します。



GM 社の自動車の制御ソフトウェアの規模ですが、これもものすごく増大しています。縦軸対数軸で、KLOC の K は 1,000 という意味です。そして line of code ですから、何千行のソフトかという単位ですが、それが 1970 年、1990 年、2010 年と、指数関数的に増えています。



また、コンシューマー的な方を着目しますと、Windows があります。Windows も、ものすごく増えています。1993 年の Windows NT 3.1 という、非常に懐かしくもありますが、その時のソフトウェアの規模が、400 万から 500 万行ぐらいでした。それが 2003 年の、今から 7~8 年前では、5,000 万行となっています。10 年間で 10 倍コード量が増えていることとなります。



Windows というのも最も大きいものではなくて、最も大きいものでは、オープンソース系と言われるもので、少しお聞きになったことがあると思いますが、Linux はものすごく、今一番多いものの行数は、3 億 2,400 万行です。もうここまで天文学的な大きさになるとどうしていいかわかりません。Mac OS についても、私も使っていますが、かなり大きく 8,600 万行あります。博士論文書いた時に覚えているのですが、私のプロトタイプシステムは 3 万行でした。2 年間かけて 2 万行書きましたから、その 1 万倍です

からとてつもない分量ということになります。ほかにも皆さまが普段使っている、基本ソフトの OS ですとか、Firefox、ブラウザなども軒並み何百万行、何十万行という非常に膨大な数となってきています。

他の主要なソフトウェアの大きさ
http://msquaredtechnologies.com/m2rsm/rsm_software_project_metrics.htm

Project	Report	No. of Files	aLOC	
 Linux Kernel 2.6.17	Total Languages & LOC Report	15,995	4,142,481	412万行
 Firefox 1.5.0.2	Total Languages & LOC Report	10,970	2,172,520	217万行
 MySQL 5.0.25	Total Languages & LOC Report	1973	894,768	89万行
 PHP 5.1.6	Total Languages & LOC Report	1316	479,892	47万行
 Apache Http 2.0.x	Total Languages & LOC Report	275	89,967	89万行

© 2011 Nakano Kazuhiko, The University of Tokyo. All Rights Reserved

ここまで大きくなると、プログラムが動くほうが奇跡だと思えます。何億行のプログラム、間違いが1個あったら動かないわけですから、動くほうが奇跡です。私がよく言う事で、やはりコンピュータ科学や情報科学、また ICT の本質というのは、どこまで複雑なものを動かせるかということにあると思えます。おそらく、今かなりその限界までできていて、神様に近づこうとしているのではないかと思うぐらいの複雑さをマネージしようとしているのが今の ICT です。

こうしたソフトウェアの肥大化がもたらす最大の問題は何かと言ったら、やはり信頼性です。これだけのものをまともに動かそうと思ったら、それはたまりません。我々の業界でも Windows が落ちるのをとりあげて、バカにするような事を言う方があります「Windows しよっちゅう落ちる」と。「Windows に車の制御なんかさせたら大変だ」というようなことを言われた方もあります。ですが、「じゃあ、あなた Windows を作ってみなさい」と言ったら誰も作れません。ここまで複雑なものを作るのは、人類にできるかできないか、本当に人類の限界に挑戦しているぐらい複雑です。その中でこうしたシステムの信頼性をいかに担保していくかというのは、非常に重要な課題です。これがまだ Windows パソコンだったら、その場で自分のパソコンがクラッシュしても誰かが死ぬということはありません。けれども、鉄道とか自動車などの分野は非常にライフクリティカルです

ので、こういったところで信頼性が損なわれている状況でしたら、これはもう大変なことです。ですから、こういった超巨大なソフトウェアの中でどうやって信頼性を担保するのかということが、非常に重要なわけです。

システムの信頼性向上へのこれまでの技術的な取り組み

- 人海戦術によるテスト
 - スマートではないが、日本で多くとられる手法
 - ある程度の規模までは通用するが、よく考えた方法には、長期的には淘汰される。
- システムのモジュール化、コンポーネント化、部品化、ブラックボックス化
 - モジュールプログラミング
 - オブジェクト指向
- Formal Methods (形式的手法)
 - 数学的手法によって、システムの振舞いの正しさを証明する
 - システムの仕様をプログラム言語で記述(いわば、システムの条件)
 - プログラムの手続きが、仕様を満たすかを自動的に証明する
 - 他にもいくつか方法がある

© 2011 Nakano Kazuhiko, The University of Tokyo. All Rights Reserved

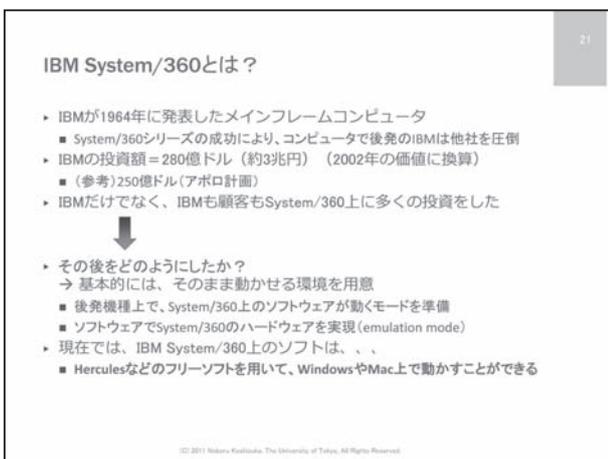
その時に、これまで伝統的に取組まれてきた方法がいろいろとあります。一番上は、方法と言っているのかよく分かりませんが、日本でもよくやるのが人海戦術です。スマートではないけれども日本でもよく取られています。ですが、これはある程度の規模までは通用しますが、ヨーロッパで行われているような、良く考えられた方法には、やはり長期的には淘汰されるようなところがあります。このような事から、コンピュータ分野でよく言われるのですが、「ICT の分野は理系ではない」、「じゃあ何系なんだ」というと、「体育会系」という話があったりします。徹夜に徹夜を重ねて、ソフトウェアをやられている方は、そうだなと思っていただける方もいらっしゃるかもしれませんが、そういう分野なわけです。

そういったところをきちっと技術的、体系的にも取り組もうという動きは、これまでもいろいろとあります。1つはシステム自体のモジュール化や、またコンポーネント化、部品化をしてとにかく分けることなどです。大きいシステムを小さく分けて、小さい中でマネジメントしてそれを組み合わせていく。そうしたそれぞれの固まりというのは、これまでのいろいろな資産をモジュール化しているということがあります。

ここで1つ例を挙げたいのが、例えば我々 ICT の分野でもどのようなことをしてきたか。過去の膨大なシステム資産をモジュール化してきた例を紹介します。



このスライドは IBM システム 360 です。これは私が生まれる前の 1964 年、もう 50 年近く前のシステムです。スライドの IBM 社というのは、実はアメリカのコンピュータの中では後発メーカーです。先発メーカーにはもっといろいろなメーカーがありました。その後発であった IBM 社が世界を制覇するきっかけになったモデルが、このシステム 360 とされるものです。1964 年に発表したメインフレームを開発するために IBM が投資した額は、最近のお金の価値に換算すると 3 兆円投資したと言われています。アポロ計画が 250 億ドルで、IBM のシステム 360 への投資が 280 億ドルですから、アポロ計画よりも多くの資金を投入して作ったシステムといえます。そういう意味では非常に膨大な資産です。



ところがコンピュータはどんどん進歩しますから、これだけ投資して、しかも IBM 社だけではなく、IBM 社の顧客の方もシステム 360 の上にいろんなアプリケーションを作

りますから、そういった資産とか投資もどんどん行われてきているわけです。これが、最終的にどうしたのかというと、この後にいろんな機種が出てきます。けれども、1つの方法はそのまま使います。そのまま使えるように、後発機種の上ではシステム 360 上のソフトウェアが動くものを準備することや、またシステム 360 はハードですけれども、これをソフトで作ったりもします。ソフトで作って、ソフトの上でシステムが動く。今はコンピュータが、どんどんもう 40 年、50 年たって速くなりましたから、実はシステム 360 と同じ動作をするシステムが、このパソコン上で動きます。それを動かすソフトウェアが、実はみんなオープンソースです。この後のキーワードで「オープン」というのが出てくるのですが、既にオープンソースになって外に出ていて、誰でも動かせるようになっていきます。こういったようなモジュール化の技術というのはやはり次々に行われてきています。

次に、これは世界最初のコンピュータの ENIAC です。



ENIAC も今はソフトウェアで実装されていて、パソコンの上で動きます。だから ENIAC と同じプログラムがパソコンの上で動きます。これを仮想技術とか仮想マシン、仮想機械というのですが、そういう技術を使って過去の資産をそのまま動かすというようなことは、最近の ICT の中で非常にポピュラーな手法になってきています。

これまで、学術分野、小規模ソフトウェアへの適用に限られていたが、

Formal Method

こちらも過去からの取組みの中で私も非常に注目しているのですが、フォーマルメソッドとって、実はコンピュータプログラムというのは、言い方が難しいのですが、数式の固まりです。数式の固まりなので、ソフトウェアの仕様、どのようにしなければならないという技術規格というものを、通常日本語とか自然言語で書くと思いますが、それをシステム仕様自体をプログラム言語で書いて、実装したソフトウェアがあると、それが最初に作った技術規格と合っているか否かを、実は機械で数学的に証明することができます。そうすると、テストなんかする必要がありません。技術の規格書があって、プログラムのソースコードがあれば、これをつき合わせてきちんと規格書通りにソフトができていくかということ、機械的に証明できるという方法です。今このようなことを申し上げると「何言ってるんだよ」と「そんなものは学術分野で小さいものはできるかもしれないけれど、実用的なものは無理」と、おそらく皆さま直感的にそう思われたと思います。

Airbus A380の制御システムがFormal Methodを利用



僕もつい3年ぐらい前までは、そう思っていました。「このようなものは実用的には駄目で、最後はやはり泥臭いテストが必要だ」と、ずっと思っていたのですが、あるヨーロッパの技術者と話をして知った事実が、最近就航したエアバスのA380にはすごく大きな制御システムが搭載されていますが、実はこの制御システムが、フォーマルメソッドを使って、数学的な証明をすることで信頼性を確保しています。

「えっ」と思って、では鉄道分野はどうなのだろうということ、少し調べますと、鉄道分野でもかなりの取組みがありました。

Formal Methods: 3つの鉄道プロジェクト

- SECAM System (1988年より作業開始)
 - GEC Alsthom, MATRA Transport, and RATP
 - 21,000行のソフトウェア (Modula-2)
 - 63%がSafety-Criticalな部分
 - 安全性の検証に100年分の作業を要したが、Formal Methodによって、安全性の向上に成功
- Line 14 of the Paris Metro (1998年より稼働)
 - 86,000行のソフトウェア (Ada)
 - 115,000行の仕様記述 (s)
 - 27,800カ所の数学的証明
- Paris Roissy Airport shuttle (2007年より稼働)
 - 158,000行のソフトウェア (Ada)
 - 183,000行の仕様記述 (s)
 - 43,610カ所の数学的証明



これはヨーロッパの例ですけれども、後ろがSECAM (セカム) という、1988年のプロジェクトです。これはModula-2 (モジュラーツー) というプログラム言語で書かれた2万1,000行のシステムの検証を、特にセーフティクリティカルな、安全性にかかわる部分に関して数学的証明を使うことで安全性を確保しています。その次はフランスの地下鉄の14号線です。これが1998年より稼働が始まっているものですが、これも8万6,000行のソフトウェアを2万7,800カ所の数学的な証明をすることによって安全性を確保しています。新しいところでは、2007年のパリの空港のシャトルです。この制御では更にソフトウェア数が増え、15万8,000行です。こちらも4万3,610カ所の数学的証明をすることで成り立っています。こういった事例からも、もう体育会系ではなく、徐々に理系になってきています。

19

ADVANCE Project

- プロジェクト名称
 - Advanced Design and Verification Environment for Cyber-physical System Engineering
 - FP7 ICT Programの一部
- 概要
 - Cyber-Physical Systemの開発手法として以下の二つを開発
 - a unified tool-based framework for automated formal verification
 - simulation-based validation
- Members
 - Software Technologies Ltd (CSWT)
 - Alstom Transport Information Solutions (AT)
 - Systemel (SYSTEMEL)
 - University of Düsseldorf (UDUS)
 - University of Southampton (SOUTHAMPTON)

© 2011 Noboru Kunitake, The University of Tokyo. All Rights Reserved.

今もこういった FP7 (エフピーセブン) というヨーロッパの研究開発プロジェクトの中で、そういったフォーマルメソッド、つまり数学的証明を使うようなこともだいぶ行われてきています。

ここまでは、取り組まれている典型的な 2 つの方法を紹介しました。1 つはモジュール化すること、もう 1 つは数学的証明をすることですけれども、これにもやはり限界があります。やはり次々と新しい機能が増えてきて、次々と増えてきたら、それはたまらない。日本の技術者は非常に優秀な方が多いので、そのように増えてきたことに対して実はずまく扱ったりするのですが、それにもやはり限界があります。

ここで新しい方向性ということを少しお話します。

まず 1 つ目は非常に非技術的な方法ですが、問題意識としては、過去の膨大な資産をいつまで使い続けているのか。蓄積が増えてくると、例えば新入りの技術者が入って、過去の膨大な資産を理解するだけで一生が終わってしまうほどの蓄積ができたかどうか、というようなことがあります。そうすると、やはりある程度考えないといけないことになり、それは大胆な経営判断によってシステムを簡素化していくことと言えます。

また、過去とのしがらみを切っていくというのも、これも 1 つの重要な方法ではないかというように思います。

ここで、いくつか例を挙げたいのですが、例えばこれは、高速道路の自動料金収受システムです。

25

【例1】Singapore ERP (Electronic Road Pricing)

日本のETC



SingaporeのERP



© 2011 Noboru Kunitake, The University of Tokyo. All Rights Reserved.

このスライドは左側が日本で、右側がシンガポールです。日本の ETC は IC カードにしましたが、現金も可能にしました。そのため、バーが必要になり、また一般と ETC の道が分かれていて、管理の人もあります。人がいるとスツと出てきて事故も起きたりもして結構危ないらしいです。シンガポールでは、車に ETC を付けることは義務になっていますから、そうすると非常にシステムの簡素化になって、スライド右にあるような単純なシステムで済むようになっていきます。このように、現金のやり取りという過去のしがらみを断ち切ることによって、決済システムが非常に簡素化しているという例です。シンガポールに関してはさらにこの先があり、全ての車に ETC が付いているので、例えば駐車場の決済とかそういうものにも ETC を使っています。日本だとできないシステムです。それがシンガポールだと、全ての車に付いているので、それができます。このように、非常に効率化も図ることができます。

26

【例2】バスの料金ボックス(日本)とICカードリーダー(中国)

日本のバスの料金ボックス



中国のバスのICカードリーダー



© 2011 Noboru Kunitake, The University of Tokyo. All Rights Reserved.

次の例はバスです。バスでいつも思うのは、日本のバスの料金ボックスは素晴らしいです。これは現金がコインだけではなくお札、磁気カード、そしてICカードも使えて、さらにICカードのチャージまでできてしまうという箱です。その代わりこちらは確か非常に高額です。次にスライド右の中国のバスを見てください。こちらはICカードがなかったら乗ってはいけません。これはある意味、技術ではなくて経営判断といえます。どのようにするかを判断することによって非常に簡素なシステムがここにできています。



次の例は、普通の券売機です。スライド左はJR 東日本の券売機ですが、これも高機能に加え、目の見えない方が買えるようにテンキーまで付いています。私は、国際標準に最近よく携わっていて、ジュネーブによく行くのですが、ジュネーブのトラムの券売機、これはまず基本はICカードがないと券が買えません。コインが使えますが、コインのおつりが出ません。ですから、きっちりお金を用意しないと余分にとられてしまいます。この券売機のシステムですが、中を開けたのを見たら、非常に簡単なシステムが入っているだけです。しかし、これもある意味で単なる時代遅れなのかもしれませんが、そういう料金を取るというプロセスの過去とのしがらみを切ることで、非常に簡素なシステムになっているといえます。これらから、やはりどういふふうに着目し、どこを切るかというところは、その後のシステムを簡素化することに関してはものすごくインパクトを持つというふうに思います。

ここまでの基本的によく言われることです。

2. KEYWORD1 “Open” アプローチ

ただ、そういった大きなシステムを動かす時のもう1つの、新しいアプローチとして、キーワードの1つ目に入るのが、“Open” というアプローチです。

このように見ていくと、ICT をに携わる者の問題意識の1つは、もはや ICT のシステムは巨大になり過ぎてしまって、1つの組織体では管理したり運用したりということが出来ないぐらいの規模と複雑度を持ってきてしまったのではないかということです。

これと同じようなことを違う言葉で言っているのが、次にあるこちらです。



この写真の方はライナス・トーバル氏で、Linux という OS を作った方ですが、自分の作った Linux という OS を全部公開して、世界中みんなと一緒に作ろうよということをした人です。そんなに大事なソフトウェアを売り物にすることもなく、そのままオープンにしておいてみんなで作ろうよという方法にしてしまったかということ、ここには1つの哲学があり、それは信頼性の問題と言えます。

コンピュータで言うと、バグというのが障害とか欠陥という意味ですが、英語で「Given enough eyeballs, every bug is shallow」って書いてあります。「十分多くの人の目を通せば全てのバグは非常に簡単なんだ」ということをおっしゃっているわけです。1つの会社の中、1つの開発チームの中だけで信頼性を向上させようと思っても、人間には限界があり、また、人数にも限界がありますのでやはり信頼性に限界が出てくるといえます。それよりは、もう地

地球上 70 億人の人に公開してもらった方が信頼性を上げるには非常に有効だとする、そういうのがこのオープンという方法です。

そうやって出来てきているものが、このオープンソースソフトウェアというもので、ソフトウェアを公開して、世界中のプログラマーの貢献によって開発が進んでいます。その世界中のプログラマーというのは、ボランティアという場合もあれば、実は裏で会社が雇用し支えている場合もあります。必ずしも全てボランティアばかりではありません。

32

オープンソースソフトウェアシステムの利点

- ▶ 開発チームの人材が優秀
 - オープンソースソフトウェアは、世界中のソフトウェア技術者のトップ5% によって開発されている。
- ▶ 世界中でテスト
 - 世界中でソフトウェアがテストされる
 - ソフトウェアの中身を世界中でレビューされる
 - 問題があれば報告される
- ▶ 開発スピードは素早い
 - 開発→リリース→不具合修正のサイクルが非常に早い

© 2011 Nakano Kaitaku, The University of Tokyo. All Rights Reserved

こういうシステムのやり方の利点は、そもそもこういうオープンソースソフトウェアにかかわっている人たちには優秀な方々が多く、技術者のトップ 5% 程度の人がこういったものを開発していると言われています。世界中でテストをされ、開発スピードはどんどん速くなり、新しいものが出てくるといふ、そういう利点を持っています。

ただ、コンピュータの歴史では、今オープンというと結構珍しいとか「あっ」と思うかもしれませんが、最初コンピュータは全部オープンでした。

実は 1960 年代とかそれぐらいのコンピュータは全てオープンでした。IBM のマシンを買うと、回路図も付いてくれば、ソースコードも付いてきたものです。しかし、それをしなくなって秘密になったのはいつからかという、実は IBM スパイ事件、日本の企業が向こうの秘密をなるべく取って行って、IBM の互換機を日本で作ろうというようなことを始めたあたりから、その対応をするために知的所有権戦略というのが出てきて、クローズにしてブラックボックス化をし始めます。これは非常に日本に対してうまくいって、日本の ICT

産業を弱めることには成功したわけです。

33

歴史

- ▶ 本来は、
 - そもそも、コンピュータ分野はソフトもハードもオープンだった。
 - IBM スパイ事件後の、米国の知的所有権戦略転換以後、ブラックボックス化。
- ▶ 1980年代：Free Software運動の始まり
 - GNU (GNU is Not Unix)とFSF (Free Software Foundation)
 - Free = 無料
- ▶ 1990年代：Free Softwareの台頭とビジネス化
 - killer softwareの出現：Linux (1991), Apache (1995)
 - Distribution Business: RedHat (1994)
 - 大手ベンダー支援：IBM, Intel, Oracle



MIT, FSF, R. Stallman氏

© 2011 Nakano Kaitaku, The University of Tokyo. All Rights Reserved

そういったブラックボックス化に反して、今のオープン化というのはその後に出てきた流れで、1980 年代にソフトウェアをオープン化しようという動きが、このリチャード・ストールマン氏という方を中心起こります。それが最初は単なる大学の中の話でしたが、1990 年代になると、そういったオープン化で本当にビジネスになるのだろうかと考えられて、実際はそれを使って膨大なビジネスがなされるようになってきています。皆さまご存じの通り、携帯電話の Android のソースコードは全部公開されています。それできっちりとしたビジネスになって高い信頼性を確保しています。皆さまお使いのもので、Linux とかウェブサーバーの APACHE、Fire fox などのブラウザ、そしてデータベースも今オープンのものがかなり使われています。

34

サーバー、PC分野のオープンシステム



OS



Webサーバー



ブラウザ



データベース

© 2011 Nakano Kaitaku, The University of Tokyo. All Rights Reserved

こういったサーバーとか PC 分野だけではなくて、制御とか組み込みの分野に関しても、同様の動きがあります。

私どもの紹介になりますが、私どもは TRON プロジェクトと、T-Kernelと呼んでいる組み込み OS に携わっています。これも 1984 年から、組み込みのシステムをオープンにして、そこでいろいろな人の手を入れてとんとんいろいろな取り組みを行っています。

すでに、今の日本の組み込み OS のシェアの 50% ほどをいただいている、私どもの大学で最初始めたものをオープンにしてそれを皆さまに使っていただいております。

35

組み込み、制御分野のオープンシステム



T-Kernel ITRON

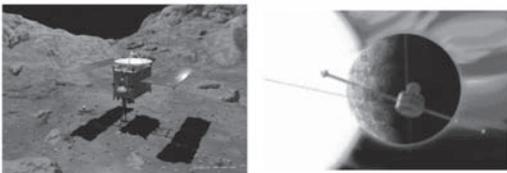
TRON, T-Kernel

日本の組み込みOSのシェアの約50%程度

©2011 Noboru Kozuka, The University of Tokyo. All Rights Reserved

36

はやぶさ、あかつき、イカロスの制御OSはTRON



©2011 Noboru Kozuka, The University of Tokyo. All Rights Reserved

37

自動車関連制御にもTRONが多く採用

カーナビ(デンソー) Lexus(トヨタ自動車)



©2011 Noboru Kozuka, The University of Tokyo. All Rights Reserved

38

BOSCHはカーナビにT-Kernelを採用

OPELや他の車種で使われている(国外)



©2011 Noboru Kozuka, The University of Tokyo. All Rights Reserved

39

TRONが利用されている製品

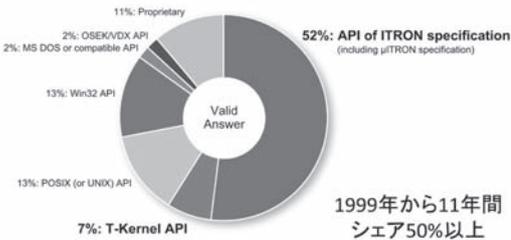


AQUOS シャープ
DVDレコーダー 日立
EXLIM カシオ
Colorioプリンタ EPSON
電子ピアノ ヤマハ

©2011 Noboru Kozuka, The University of Tokyo. All Rights Reserved

40

TRONは59%のシェア



API Specification	Share (%)
Proprietary	11%
OSEK/VDX API	2%
MS DOS or compatible API	2%
Win32 API	13%
POSIX (or LINUX) API	13%
T-Kernel API	7%
API of ITRON specification (including μITRON specification)	52%

Valid Answer

1999年から11年間
シェア50%以上

©2011 Noboru Kozuka, The University of Tokyo. All Rights Reserved

今これはどういふところに使われているかと申し上げると、例えばこういった宇宙、はやぶさ、あかつき、イカロスといったようなものや、自動車や、カーナビなど、もちろんこういった家電製品にも使われていて、今 59% 程度のシェアを日本の国内で持っています。こういった制御の部分においても、オープンというのはとんとん使われてきているわけです。

また、もっとショッキングなことというか、私自身も「えーっ」と思ったのは、実は9年前にオープンシステムの国際会議がフランスのカヌヌであり、私も発表をしましたが、その会議で横にいた方が米国国防総省の方でした。これが会議の写真ですが、国防総省の役人の方がいらっしゃいます。



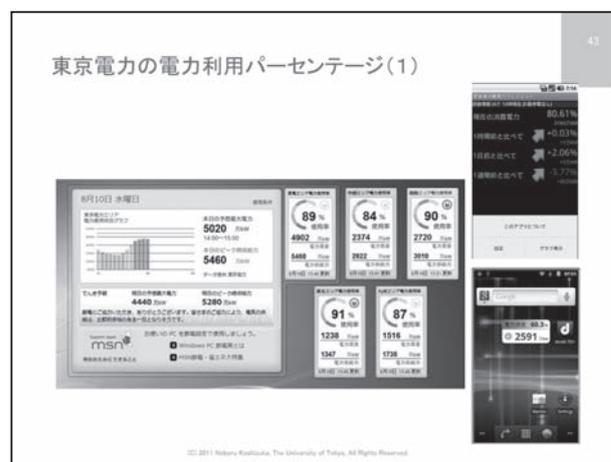
普通、軍のシステムといったら最も秘密で、オープンなんてあり得ないと思っていたのですが、2002年の時点でアメリカの国防総省が兵器を作るなど兵器のコントロールシステムをオープンシステムにする話をしていました。特にアメリカですと、制御システムの非常に大きなマーケットの1つはミリタリーです。その中でもこうやってオープンなものをどんどん使っていくことが行われています。オープンにすることで信頼性が下がるとかではなく、適用する分野に勿論向き不向きはありますが、非常に膨大なソフトウェアを、信頼性高くメンテナンスしていく時には、このオープンというのは非常に有効な手法の1つとなってきています。

加えて、オープンにするというのはこういった基幹システムだけではなく、最近はもう少し違う分野にも来ています。

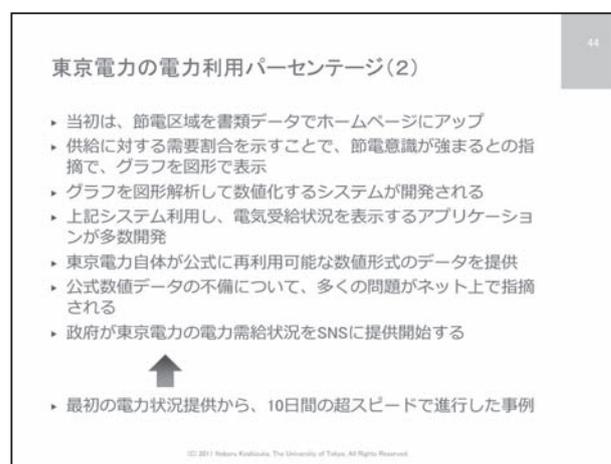
次は制御システムなどの硬い方の話ではなくて、もう少し柔らかい価値創造などについてお話します。

それはオープンデータです。ソースコードではなく、データをどんどんオープンにしようというものです。情報をオープン化することで、市民やユーザーの方、また、お客さまと一緒にシステムを開発していこうというような考え方がここ数年ものすごく流行りになってきています。

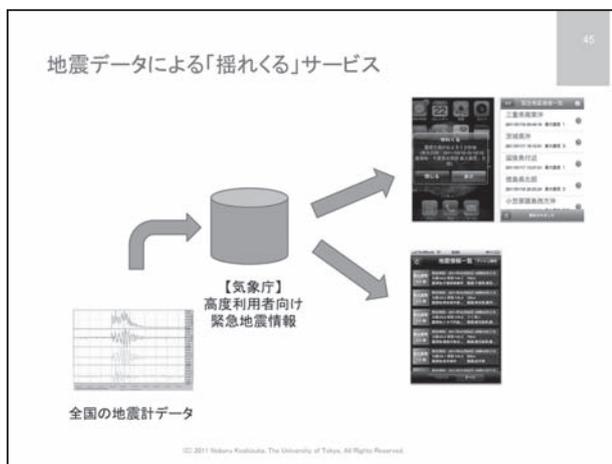
その代表的な例がこちらです。



これは緊急時の例ですけれども、東京電力株の電力のピークがどうなっていて、節電がどうだということを見るようなソフトウェアが今はたくさんあります。このソフトウェアをスマホに入れられている方もいらっしゃったり、自身のパソコンに入れている方もいらっしゃると思います。実はこのソフトウェアは東京電力株が作ったものではありません。東京電力株が行っていることはデータを公開していることだけです。データを公開することから、こういったソフトウェアを作る人が何十、何百と出てきてこのようなソフトウェアが出来たわけです。しかもこれができるまでが、データを公開してからできるまで2週間です。たった2週間でこれだけのものができあがって、こういう電力をモニターすることができるようになってきています。

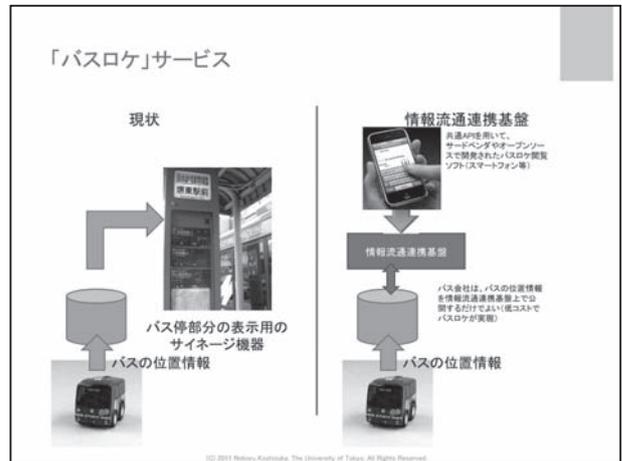


その経緯はこちらに少し書きましたが、最初は東京電力(株)も、電力のデータの数値を公開したわけではなくて、どれくらいだったかというグラフの画像だけを公開していました。一般のプログラマーの間から、数字があったらもっと便利だという話で「数字を公開してほしい」という話がありましたが、グラフの画像しか公開していませんでした。そうしたところ、世の中にはいろいろな方がいて、そのグラフの画像から数値データに逆変換する方が出てきたんです。それがネットに公開されました。ネットに公開したら、その数値データを使ってソフトを作る人が出始め、そうしたらさすがに東京電力(株)も、「それだったらもうデータを公開しましょう」という話で、データを公開しました。そのデータを使って、またよりソフトがいろいろ出てきてという。これが2週間、いえ、プロセスが全部で10日間です。10日間でどんどん進んだというようなことです。これがデータをオープン化するという、そのオープンデータの威力をわが国で示す最も代表的な例ではないかと思えます。

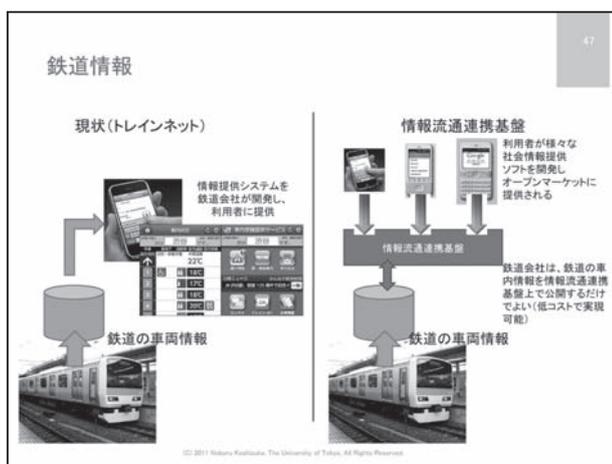


これは本当に一般のユーザの方の例ですけれども、つまり中小のベンダーさんのように小さいソフトウェア会社が関係してくるという意味になりますが、もう1つ地震絡みですが、「揺れる」サービスみたいなものがあります。これもデータは、地震計を持つ気象庁が公開しているわけですけれども、このソフトを気象庁が作っているわけではありません。これも気象庁はデータを公開するだけで、そのデータを使っていろいろなアプリケーションが出てきており、有料のものもあれば無料のものもあります。企業が作ったものもあれば一般ユーザが作ったものもあるということで、こ

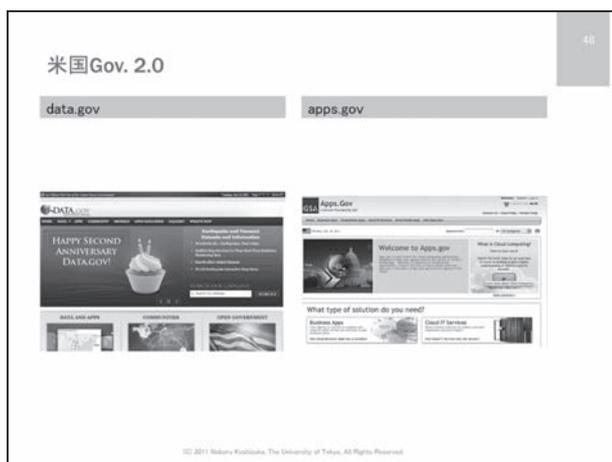
いった高度利用者向けの緊急地震情報をソフトにするといいことがどんどん起こっています。



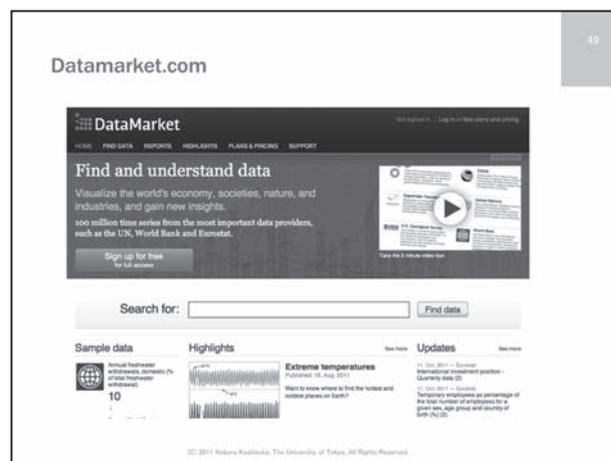
これと似たような話はいろいろあります。交通の分野で言いますと、バスのロケーションシステムというものがあります。左側が今でもよくあるもので、バスの位置情報をデータベースに溜めて、それをきちっとスライドのようなサイネージのすごい設備を作って、そこに今どこでバスがどういふふうにと走っていて、いつ到着するということを表示します。しかし、もう今の時代でしたら、皆さん携帯電話も持っていますしスマホを持っている方もあります。こんな高価なサイネージがなくても、バスの位置情報をデータベースで公開して、誰でもプログラムできる環境さえ提供すれば、これは簡単にはずです。おそらく企業などにとって、例えば、JR 東日本でこれやろうと思ったらものすごく簡単にはずです。公開さえすれば、それを使ってバスのロケーションサービスのソフトなんていうのはたくさん出てくると思います。今はそういう時代です。それだけの開発力をお客さま自身であったり、ユーザの方自身が持つようになっている。これを利用しない手はないというふうに思うわけです。



最近私も非常に楽しませていただいて面白かったのが、JR 東日本で行っているトレインネットです。これは山手線中の空調温度が分かったり、どこが混んでいる、または空いてるなどが分かって非常に面白かったんです。これも今はおそらく実験とか先進的な取り組みでやられて、全部JR 東日本自身で作られていると思いますけれども、将来、こうだったら面白いと思うのは、そのデータを公開してしまう。データを公開してしまえば、もう世の中鉄道好きな方がたくさんいますから、コンピュータも詳しく、そういう方々がおそらくどんどんこういうものを、競争して作ると思います。やはりオープン化というのは非常にパワーがあると思います。



こういったこと、例えば行政でも非常に注目されていて、アメリカでは今ガバメント 2.0 というのが非常に流行りです。これは行政サービスを提供するところに市民の開発力を利用しようというものです。今までは、情報を出していくのは、最後にユーザの市民の目に届く紙のレベルとか文字のレベルまで全部役所がやっていました。しかしこれからは、役所はデータだけを出します。データだけ出したら、そのデータを利用していろいろなところ、いろいろなサービスをするのは、もう皆さんがやってくださいということです。そういうことでデータを公開していくというのが data.gov ですとか、こういうサイトがあって、政府のデータがどんどんここに公開されています。それだけではなくて、その公開したデータを使ったアプリケーションもあります。



例えばスマホの中に入れるソフトとか iPad の中に入れるソフトですとか、それを見ると行政情報がバッと出せるわけです。そういうものを集めたサイトもあります。このような市民の開発力を使って行政サービスをやっていこうというのは、もうアメリカだけではなくて、世界中の非常に大きな動向です。

これは政府だけではなくて民間でも随分あります。そういうデータを出すことを、お金を取って商売にしているようなところもあります。いろんな観測データとかセンサーのデータ、例えば JR 東日本でいろいろお持ちのデータで出してもいいものがあつたとします。それを無料で出すのではなくて有料で出しています。有料で出して、きちっとお金も取っています。そういうビジネスモデルも成り立っているといえます。尚且つ、こういったガバメントが出したり民間が出した

りして、政府がよくやっていることは、アプリケーションのコンテストです。



オープンデータを使って、このようなアプリケーションコンテストを行って、賞金を出したりします。このようなことを、JR 東日本などが行ったらすごいと思います。鉄道データを出してコンテストを行ったら日本中の鉄道ファンが集まりますよね。色々なものや、相当いいソフトができるんじゃないかなと思います。だから、こういうやり方というのはこれからの今風なやり方と考えます。先ほどの膨大なソフトウェアの信頼性をどのように上げるかという硬い話もありますが、こういった柔らかいところの新しい価値創造にもオープン是非常に有効だと思います。

ただ、最後にオープンということに関して、一言申し上げたいのは、やはり全てをオープンにすれば良いというものではないのです。これには戦略が必要です。これは経済産業省のレポートからコピーさせていただきましたが、アメリカでいろいろなオープンを進めている IT の企業の中で、どういう戦略を取っているかということの表から、いくつかあげてみます。

	特許戦略	オープンソース戦略
インテル	・自社の半導体技術を中心に、特許を取得。	・半導体に係るインターフェース開発等を中心としたプロジェクトを展開。
IBM	・新リスクの軽減の観点も含め、特許を取得。ライセンスを推進。 ・アマゾン社に対してビジネス・プロセス特許の不法侵害で提訴。	・Linux 推進を目的としたライセンス管理会社 OIN を設立。 ・SOAP、SAML、XML 関連技術 150 件をオープン化。
サンマイクロシステムズ	・新リスク軽減の観点から、特許を取得。ライセンス志向せず。 ・無料ソフトウェアの開発者に対して特許侵害を主張しない。	・自社特許技術を積極的にオープン化 (29 団体に参画)
マイクロソフト	・自社のソフトに係る特許を取得。 ・侵害していると思われるものは提訴。	・ノベル社、デル社と Linux の相互互換性を強化する共同開発契約を締結。 ・コア以外の分野でオープンシェア化の動き。
グーグル	・ページランク技術の独占権を保有。 ・一部検索アルゴリズム等は秘匿化。	・検索システムはオープン。 ・OIN にエンド・ユーザとして参画。

© 2011 Noboru Kashiwagi, The University of Tokyo. All Rights Reserved.

例えば最も分かりやすいところでグーグルをあげようと思いますが、グーグルの今やってる検索サービスは自由でほとんど使えますし、Android という携帯電話の OS のソースも全て出してオープンでやっています。ですけれども、彼らは全てをオープンにしているわけではありません。彼らの一番コアになる技術、例えば検索する技術について、これはものすごい秘密主義を取っています。論文の一本も出ていません。特許も出ていません。全部秘密です。特許にすると権利は維持できますが、公開されてしまうのでやはり秘密は守れない。そのため特許にもしないという、それぐらい明確な方針を取っています。ですから、通常、グーグルのような規模の企業になると、我々の学術分野であれば、大体研究所ができていろんな論文が出てくるものですが、グーグルの論文というものはあまりありません。マイクロソフト社は最近大量の論文を出していますが、グーグルはそういう技術発表とかあまりありません。そういうような戦略も 1 つのやり方です。やはりどこをオープンにしてどこをクローズにして、どこを競争力の源泉として守っていくかというのは、よく考える必要があるだろうというように思います。

3. KEYWORD2 “Smart” スマート

2 番目は “Smart” ということです。Smart、この後パネルディスカッションの時に(株) NTT データ経営研究所の三谷様からお話があると思いますが、一般的にはよく IT とか ICT を使うと生産性が向上すると言われています。

54

ITによる生産性向上(経済産業省資料より)

【IT投資と生産性上昇の相関係数(※)】

	製造業	非製造業
日本	0.19	0.03
米国	0.35	0.41

(出所)JCERデータベース
米商務省“Fixed Assets Tables”, “GDP by Industry Data”
(※)1995/1990 と 2000/1995 のIT比率(=IT資本ストック/全資本ストック)の変化とTFP 変化率の相関係数

- IT投資がある一定程度、日米ともに、生産性向上に貢献している。
- 米国と日本を比較して、製造業の生産性向上への寄与は顕著だが、非製造業への寄与は限られている。
- ITによる生産性向上は、まだ可能性が残されているのではないか？

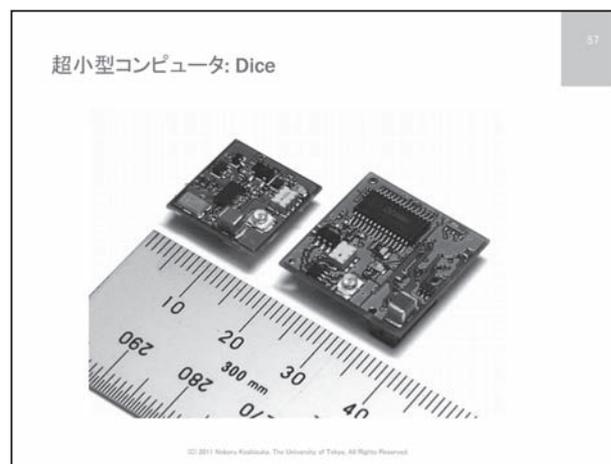
© 2011 Nakano Publishing, The University of Tokyo. All Rights Reserved.

経済産業省の例ですけれども、日本とアメリカの、ITの技術を使ってどれぐらいの生産性向上に寄与したかということを示す表です。アメリカは製造業、非製造業含めてかなり寄与していますが、日本も製造業はかなり寄与していますが、非製造業はあまり寄与していません。おそらく詳細なことはパネルディスカッションの時にお話が出てくると思います。ただ、ここから私が申し上げたいのは、普通の伝統的な今までのITによる生産性向上ができたということなんです。私はIT、ICTによる生産性向上はまだまだ可能性が残されていると思っています。

そのためのキーワードが、2 番目の “Smart” です。着眼点としては、従来の IT、ICT というのは情報のデータしか扱いません。だから、デジタルなデータになったものに関しては効率的に扱えますが、データになってないものをデータにする、そしてそれを効率化するというような試みは実はあまり行われてきていません。あくまでもデータになって、コンピュータの中に入ったものを使って、いかに効率化するかというわけです。私はユビキタスという分野をずっと研究しているのですが、これは実は情報化されてないものをいかに情報化、効率化するということが、究極的には最も重要なことといえます。そういうことをやると、もっと

いろいろな効率化ができるのではないかとようになっていて、いくつかの例をご紹介したいと思います。

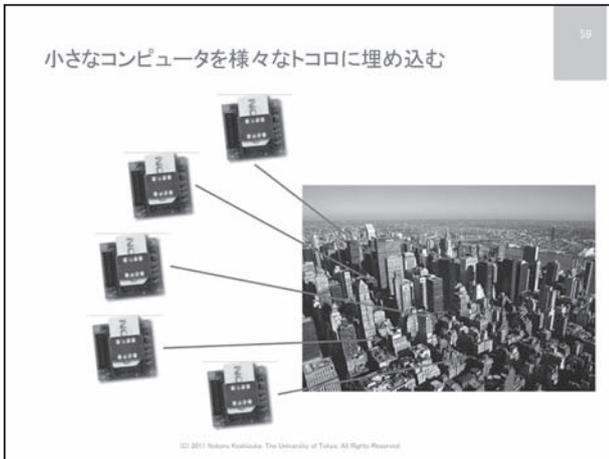
さらなる効率化のためには、データ化の過程を効率化する。そのためには、コンピュータをスマートにするだけではなく、コンピュータではないもの、物をスマートにする、場所をスマートにする。そうすることによってさらなる効率化ができるのではないかと、というふうに考えています。



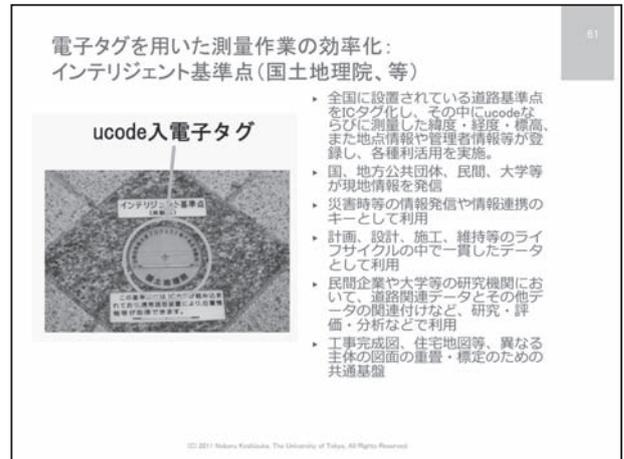
今、これは私の研究所で開発したものですけれども、約 1.5 センチとか 1 センチ角のコンピュータです。もう今ではさほど珍しいことではありません。この中にコンピュータがあり、メモリがあり、通信があり、これでワイヤレスの通信をすることができるセンサーものっています。そういう小さいコンピュータです。



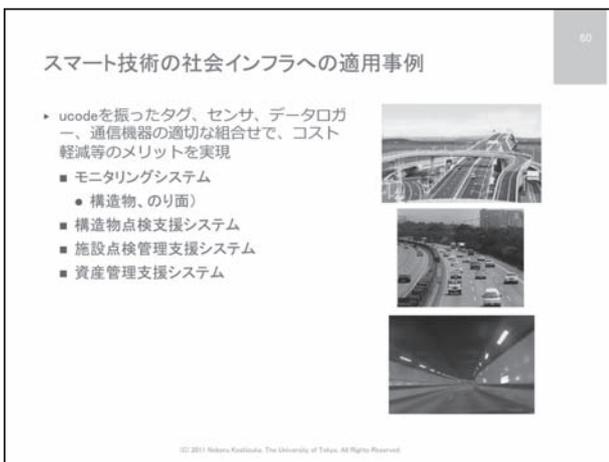
ちょうどサイコロと同じぐらいの大きさなのでこれをよく我々は「ダイス」と呼びます。



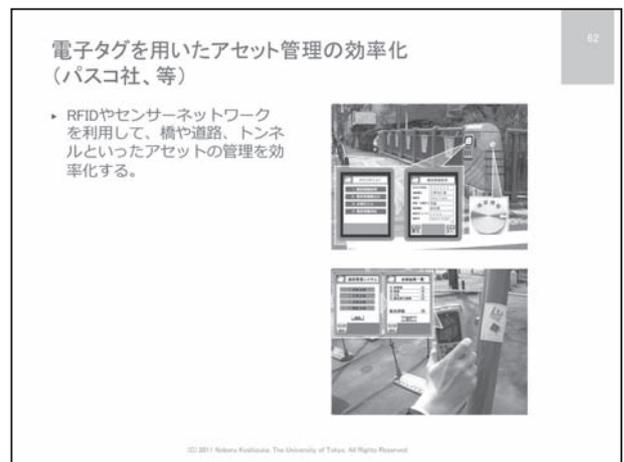
こういう小さいコンピュータをいろいろなところに埋め込むと、いろいろなことができます。つまり、スマート技術の社会インフラへの適用が可能となります。



上記のように測量することで、電子タグにデータが入ってしまえばそれこそいろいろな効率化を可能にする技術といえます。



ここに題目として、モニタリングシステムとか、構造物点検、施設点検管理、資産管理など、少し難しいことが書いてありますが、例えば、現場の作業に密着してお話しますと、1つ目の例は、国土地理院がやられていて、おそらく鉄道分野で測量というのも非常に大事だと思いますが、その測量を効率化するために、測量のための基準点にコンピュータを埋め込んで、測量をする機械とこの基準点が自動的に通信し、その測量の機械のセットアップを効率化しようという、そういう試みです。



次は、パスコ社と我々がやらせていただいている効率化ですが、社会インフラ、ここではアセットとなっていますが、そういうものを管理します。点検管理するために橋や遊具とか、そういうものにコンピュータを埋め込んで、点検を効率化するものです。

63

電子タグを用いたコンクリート管理の効率化： 電腦コンクリート(住友大阪セメント)

- ・ 供試体トレーサビリティシステム
 - 大量の供試体の固体管理と試験の効率化・正確性の向上
 - 供試体とは
 - 練り混ぜ直後のコンクリートから取ったサンプル
 - 多数作り、28日後に圧縮力を加えて目標強度が得られているかどうかをテストする
 - 従来、固化した表面に手書きで情報を書き込んできた
- ・ コンクリート製品トレーサビリティシステム
 - 「品質」「製造方法」「生産年月日」など、各製品固有のデータを購入者も確認
 - 建築物のユーザの方々の品質に関する不安を解消し、加りたい情報をいつでも提供できる「建築物トレーサビリティシステム」へと発展予定



©2011 Nihara Kaitiaka, The University of Tokyo. All Rights Reserved

次に、これはセメント会社と一緒にやっているものです。セメントの中にチップを埋め込んで、そのチップの管理とかメンテナンスをやはり効率化しようというものを、住友大阪セメント(株)とやっています。

64

電子タグを用いた住宅部品の点検作業の効率化 (財団法人ベターリビング)

- ・ メーカーが認定部品を販売する際、住宅部品ひとつひとつに ucode タグ (電子タグ) 付きの証紙が貼られて出荷。
- ・ 各認定部品の個品ごとの家屋に設置されたかの管理を少ない手間で実施可能
 - 既に、トラブル時の回収を効率化する実績
- ・ 設置業者は、RFID上の ucode と設置・保全・廃棄等の作業情報を自動的に紐付けし、ベターリビングが運営するサーバに送信・登録し、トレーサビリティ情報を管理。



©2011 Nihara Kaitiaka, The University of Tokyo. All Rights Reserved

また、これは住宅に付ける火災報知機です。火災報知機の部分に小さい電子タグを付けて、これを使ってバッテリーがいつ切れるなど、そういうことの点検のコストを削減するシステムを構築しています。

それと次は、道路や橋にセンサーを取り付けて、車で走るとそのセンサーの振動のデータとか、例えばクラックが入るといったデータが計測できるとか、そういうデータを自動的に収集している例です。

65

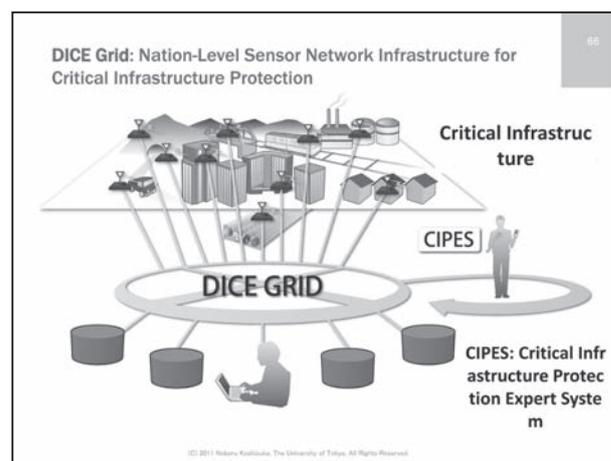
センサーネットワークを用いた、高速道路点検情報の自動収集 (NEXCO東日本)



©2011 Nihara Kaitiaka, The University of Tokyo. All Rights Reserved

今までの IT というのは、例えばデータが集まった後にはこういうスケジュールで修繕を効率化しましょう、というようなことを扱っていたと思います。ですが、どこを修繕したらいいか、というデータを集めるために、橋自体をスマートにすることは行っていなかったと思います。このように、まだ IT の効率化で残されているのは、物など今までコンピュータではなかったものにコンピュータを埋め込んでいくことで、どこまで効率化できるかというところが、まだ残されています。

今我々大学では、こういったことを非常にスケールを大きく考えていまして、社会全体のクリティカルなインフラストラクチャーは全部コンピュータの中に入れて、その状況が自動的にモニタリングできるような、よく IBM 社さんとスマートプラネットとかスマータープラネットと言われていますが、我々はこれを「スマートランド」というように呼んで、地球全体ないしは日本の国土全体をスマート化していくことによっていろいろなことを効率化できるのではないかと、ないしは、信頼性を向上させることができるのではないかとということに取組んでおります。



4. KEYWORD3 “Standard” 標準化

最後に3番目のキーワード“Standard”です。標準化というと、技術的な分野と非技術的な分野の境界のところ、非常に政治的な場でもあり、私もここ10年ぐらい情報通信の国際標準をやっています。ジュネーブの中に国際標準をやる機関がたくさんあり、そこに日本の技術を持って行って、それを国際標準にするために毎月のように各国とネゴシエーションをする、またそれだけではなくて、色々な調整を図りながら国際標準を勝ち取るということ、ここ10年ぐらいやってきております。

そして、これを行う時にいつも危機感を覚えるのは、日本は標準化が下手だということです。いつも標準化の場において思うのは、日本より悪い技術が次々に標準になっていく、という事です。ですから、これはなんとかならないものかと、常日頃思っています。逆にヨーロッパは、人間関係とか標準化とかそういう話し合いが、すごく上手いのです。その力を使って技術的に不利な部分をどうにか挽回しようということをやっています。そういった場面を私も非常に多く見えています。

す。我々のICTの通信も、情報の送り手と受け手が共通の規格を採用しなければ通信は成立しません。通信において、国際標準というのは非常に重要です。

世界最初の国際標準というのは、実は電信、テレコミュニケーションでした。それが今国連の組織としてITU (International Telecommunication Union 国際電気通信連合) という組織になっています。今の携帯電話の3G、4Gというような、熾烈な競争が繰り広げられているのはこのITUという場ですし、私がユビキタスの技術を持って行って今国際標準化を進めているのもこちらの中です。

標準化の重要性: 鉄道と情報通信

- ▶ 鉄道
 - 根源的仕組みから標準化が不可欠
 - 車両と線路
 - 車両とホーム
 - 電気
 - 信号
 - ...
- ▶ 通信
 - 情報を送り手と受けてが共通の規格を採用していなければ、通信が成立しない
 - 世界最初の国際標準は「電信」
 - → それが現在のITU (International Telecommunication Union) に

©2011 Noboru Kodaira, The University of Tokyo. All Rights Reserved.

鉄道における標準化の経緯

- ▶ 明治5年の新橋～横浜間の鉄道が開通以来、日本の鉄道技術は国鉄を中心に発展
- ▶ 日本の鉄道全体をカバーしてきた技術標準
 - JIS (Japan Industrial Standard): 日本工業規格
 - JRS (Japanese National Railways Standard): 日本国有鉄道規格
 - その他団体基準
- ▶ 国鉄の分割民営化以後、日本全体の鉄道規格の骨格が曖昧な状態
- ▶ 世界や欧州では、国際化・グローバル化を念頭に置いた技術標準化が進められている。

- ▶ 我が国における通信と様々な面で類似
 - ISO → ITU
 - JIS → TTC
 - 国鉄/JR → 電電公社/NTT

©2011 Noboru Kodaira, The University of Tokyo. All Rights Reserved.

ただ、少し謙虚に見て、この様に捉えています。鉄道も通信の分野も、標準化というのは非常に重要で、おそらくそれができた時から標準化というものがなされている分野だと思います。鉄道であれば、それこそ線路が標準化されていなければ車両は走れませんし、車両とホームも標準化が必要です。供給する電気も必要ならば、信号の標準化も必要です。これは通信も似たような状況といえま

次の話は私が申し上げると、逆に皆さまには釈迦に説法かと思えますけれども、日本の鉄道でも長い標準化の経緯があります。明治5年に開通してから、日本の鉄道全体をカバーしてきた技術標準はJISであったりとか、また、国鉄の時の日本国有鉄道規格です。こういったものが日本のある意味デジュールスタンダード、デファクトスタンダードになってきましたが、分割民営化されたところから、少しずつこういった骨格があいまいになってきているのではないかと思います。

こういった状況が本当によく似ているのは、通信分野もそうです。通信分野も、国内規格があって、電電公社がいらっやって、電電公社の規格というのが日本を席卷していたわけです。けれども、多少こういった標準化を、しかも国際化の中で国際的に展開する時には、やはりビッグな会社でいていただいたほうが非常によかったですけれども、分割されてしまうことによってなかなかそこが苦しくなっています。

70

標準化戦略

- ・ 一般的に、日本の工業技術や生産技術の質や水準は高い
 - 工場や生産現場から出てくるものの品質の高さは恐らく世界一
- ・ ただし、ものの価値は工場を出荷した時に決まるのではない
 - 他のモノや機器との相互運用性による更なる価値の増大
 - 保守、運用のコスト
 - 品質だけでなく、価格も重要
- ・ 日本の教訓：零戦
 - 運動性能のよい、素晴らしい戦闘機だったが、
 - 工業的標準化がなされていなかったため、簡単に修理ができなかった。
 - 米国の戦闘機は、2台から部品を集めて1台を作ることが可能なつくり

↓

- ・ 標準化の重要性
 - 特に欧州は標準化が得意



© 2011 Noboru Kunitake, The University of Tokyo. All Rights Reserved

標準化がなぜ重要かということを少し先ほどまでざっくりばらにお話しましたが、私が思うに、一般的にやはり日本の工業技術とか生産技術の質とか水準というのはものはすごく高いと思います。ある意味で工場や生産現場から出てくる物の品質の高さは、私はやはり世界一だと思っていますし、いまだに信じています。例えば大学にいて学生を見ているけれども、物を作らせた時に留学生と日本人の学生とを比べると、同じ学力だったら日本人のほうがものづくりは得意です。外国の留学生は話したり、論文を書かせるとロジカルな思考はすごいです。日本人はどうもロジカルな思考が少し弱いところがあるのではないかと思います。ロジカルな思考と物を作る能力はあまり関係ないみたいです。物を作る能力はピカイチですよ。我々の分野でしたら、日本のプログラム、僕は凄いです。けれども、今の時代は冒頭でお話しましたように、単に速ければいい、単に大きければいいという時代が終わった中で、話は複雑になっています。物の価値というのは工場を出荷した時にすごく速いとか、すごく大きいとかそういうことで決まるのではなくて、例えば、ほかのものや機械と相互運用ができる、それによってさらに物の価値が増大したり、保守や運用のコスト、そして品質だけでなく価格ももちろん重要です。いろいろな要素があるわけです。そういう中で、標準化というのは非常に重要です。

標準化されていない良い工業製品の日本の教訓を紹介いたします。私もコンピュータ屋ですので実は根っからのオタクですが、オタクというのは2つの系統が大体あって、鉄道オタクか、もう1つは軍事オタクというのがあるの

ですが、僕は軍事オタクの方でした。その軍事オタクで、知っていることはゼロ戦です。これは素晴らしい技術を持つ、運動性能が素晴らしい戦闘機だったのですが、いかんせん標準化が全然なされてないので、いったん壊れるとどうしようもなかったのです。壊れてしまうと、天下の大職人がやっこないと修理できない。だから、よくアメリカの戦闘機は、壊れた1機と壊れた1機2台で部品を集めて1台を作るなんていうことが可能だったわけですが、日本のゼロ戦はそんなこと全くなく、いったん壊れるとなかなか難しい。これは標準化がやはりなかなかうまくいってなかった時の1つの例です。

このように、工場から出てくる物はいいのだけど、その標準化は日本が弱いところと言えます。ヨーロッパというのはたくさんいろいろな小さい国もあって、戦争の歴史も長いことから、そういうところの切ったはったというのはものすごく能力が長けています。標準化は非常に重要です。

4.1 標準化 1 攻撃的標準化

特に鉄道分野ということもありますが、我々の ICT という分野も含めて言えば、1つはやはり攻撃的な標準化。特に IT、ICT 情報通信関係は、大量生産することによって低価格効果が非常に大きい分野です。例えばあるソフトウェアの開発費が10億円とします。10億円で1個しか製品を作らなければ価格は10億円です。これ10億台を出荷すれば1円です。特にハードウェアもそういう傾向があって、たくさん作れば作るほど安くなります。だから、例えば我々が iPad を見て、技術的にはすごいとは全く思いません。どこでも作れます。日本でなければ作れないようなことは全くありません。けれども、あの価格はすごいです。あの価格で部品を調達してきたというのはすごいことで、なかなかできません。それは、やはり多くのマーケットを抱えているから安く部品を調達できるわけです。多くのマーケットを抱えているというのは非常に重要です。iPad とか iPhone というのは、どちらも標準化したものではなくてアップル社の単独の機種ですけれども、ただし、非常に膨大な市場、マーケットを抱えていますから、アップル社は非常に安く部品を調達することができます。だから作れるわけです。

アメリカの会社なのでこのような方式を取りますが、普通ヨーロッパだとどういふやり方を取るかとすると、標準化するわけです。標準化して皆でマーケットを1個にします。そうすることによって大量生産効果を出して、そこで安いものを出していくというアプローチを取っていくわけです。マーケットの力、市場の広さというのは、それが技術力になってしまう面があります。これはたぶんICTだけではなくて、もしかしたら鉄道分野もそうかもしれません。日本だとしても鉄道というと日本の国内だけになってしまいがちですが、ヨーロッパで覇権を取ればヨーロッパ全土がマーケットですから、なにか部品を出すにしても全体に売れるわけです、そうすると非常に安いものがでてくる。それが仮に日本よりも質が悪かったとしても、非常に安いということになれば、これはまたマーケット圧力がいろいろ出てくるということになります。どうも今の日本のICTだけでなくどの工業分野もそういった悪い標準化のサイクルに少し入っているような気がして、僕はなんとかこれをやはり打破しないとイケないと思います。日本は不得意ですけども、この標準化を頑張らなければならぬと思っています。

4.2 標準化 2 防衛的標準化

これをもっと即物的に言う、防衛的な標準化という話があって、標準化をしないとなんで困るかというもっと即物的な話です。

74

WTO

- ▶ WTO=World Trade Organization (世界貿易機構)
- ▶ 多角的貿易交渉の結果を実施する国際機関として1995年1月に発足
- ▶ 日本は、1994年12の国会での承認を得てWTO発足と同時に加盟
 - WTO協定へ批准



© 2011 Nakano Koshiro, The University of Tokyo. All Rights Reserved.

ここで出てくるのがWTO (World Trade Organization 世界貿易機構)です。日本も加盟しておりますが、ここで重要な協定がTBT協定と言われるものです。

75

TBT協定

- ▶ 「貿易の技術的障害に関する協定」
 - TBT=Technical Barriers to Trade
- ▶ 経緯
 - 1979年4月: GATTスタンダードコード合意
 - 1994年5月: TBT協定として改訂合意
 - 1995年2月: WTO協定に包含
- ▶ TBT協定はWTO一括協定となっており、WTO加盟国全部に適用
- ▶ 政府調達に際しては、TBT協定遵守が不可欠

© 2011 Nakano Koshiro, The University of Tokyo. All Rights Reserved.

こちらはおそらく皆さんの中でも、ご存知の方もたくさんいらっしゃると思いますけれども、貿易の技術的障害に関する協定というもので、日本もこれに加盟しております。

76

TBT協定(概要)

- ▶ 政府調達で、技術基準を定める場合、その目的に合致する国際標準規格があるときは、その国際標準規格に準拠しなければならない。
 - 民間企業は関係ない(ただし、JRは政府扱い)
 - 国際標準がない分野も関係ない
 - 軍事など、国家安全に係る調達は例外

↓

- ▶ 国際標準規格よりもよい技術を保有していても、政府調達不可

© 2011 Nakano Koshiro, The University of Tokyo. All Rights Reserved.

いろいろ言い出すと長くなりますが、少しまとめると、政府調達で技術基準を定める場合は、その目的に合致する国際標準規格がある場合はそれに準拠しなければならないことになっています。民間企業は関係ありませんが、おそらくJR東日本は国際的には政府扱いになると思います。あと、国際標準がないような分野についても関係ありません。3つ目には、アメリカはこれを良く使いますが、軍事や国家安全保障にかかわる調達はこれまた例外です。ですからアメリカは、ハイテクを国家安全保障のものだと言って

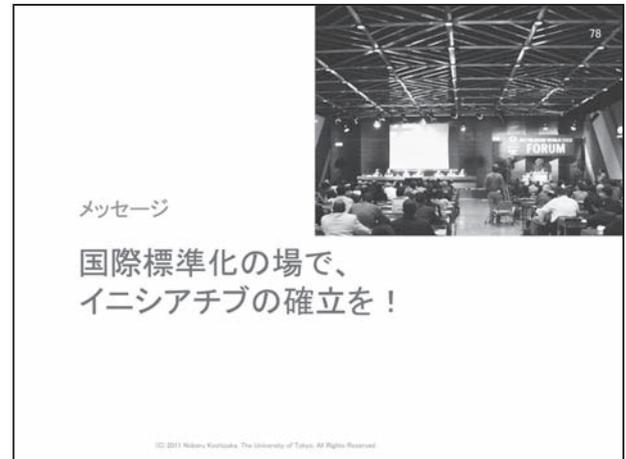
クローズな調達を行うということがありますが、日本はおそらくこれをやる外交的パワーはありません。ですから、国際標準をとりながら進めていかないと難しいかと思います。

このような協定があると、国際標準規格よりもよい技術を国内で持っていて、かつ、それを自分たちが開発したものであっても、政府調達は不可能なわけです。ですから、こういったことがあると、ますます、特に社会インフラになるような分野、鉄道や交通もそうだし、我々の通信分野もそうです。よく僕らはユビキタスといって、道路などにコンピュータを埋め込むこともやっていますが、そういう時も政府調達が絡んできますので、国際標準化を獲得するということが絶対不可避です。ですから、こういうことを一生懸命やらないといけないと思います。



少し、この例を出すことをためらったところもあるのですが、おそらく皆さま方一番ご存じなのは Suica の例だと思います。ソニー(株)さんの標準化活動、Suica の IC カードの技術というのは、それまでにあったヨーロッパの IC カードの技術と比べて段違いにいいです。どこがいいかというと、一言で言うと、IC カードの読み取り装置とカードの間の通信が行ったり来たり行ったり来たりするのですが、そのカードに行く通信とカードから戻ってくる通信が、ヨーロッパとは違う方式です。ヨーロッパの方式は行きと帰りが違います。FeliCa とか Suica は、行きと帰りが同じの対称型の通信なんです。対称型の通信だと何ができるかというと、読み取り機同士を背中合わせにして通信ができるのです。当時、ものすごい技術的なアドバンテージがあったわけです。おそらく JR 東日本も大変ご苦労されて、国土

交通省と一丸となつてはねのけて今日に至っていると思います。こういうことがおそらくこれからもいろんな分野でどんどん起こってくるのではないかと思います。



今日は3つのキーワードを申し上げてきましたけれども、最後に私のほうで申し上げたいことは、ISO とか、僕らの分野だったら、あと ITU とかそういう分野がいろいろあります。そういう国際標準の場に行くと、どうも日本は国力と比べるとプレゼンスが低くて、日本の方がこういうところに行くと、本当に日本人らしいのですが、真面目なんですよ。全ての会議に出てメモを取ってる人が多いのです。

5. おわりに

しかし他国では、ものすごい政治力などを感じ、やはりすごいなというか、日本ではちょっとあり得ないようなこともあります。

しかし、我々もどんどん国際的なプレゼンスを高めていくということ、ICT もそうですけれども、全てのこういった技術系の分野でもなされていくといいのではないかと思います。

私の話はここまでにさせていただきます。どうもありがとうございました。