

日本大学経済学部経済科学研究所研究会

【第173回】

2010年6月19日

平成20～21年度共同研究成果報告

ネットワーク産業における標準化と互換性

日本大学経済学部教授

大 場 允 晶

日本大学経済学部准教授

中 邨 良 樹

コニカミノルタ情報システム
株式会社部長

大 宮 望

社団法人バイオ産業情報化
コンソーシアム特別研究職員

丸 山 友希夫

日本大学経済学部の大場允晶です。

私たちは平成20年度から21年度にかけて、日本大学経済科学研究所共同研究プロジェクトとして、「ネットワーク産業における標準化と互換性」に関して研究を行ないました。本日はその共同研究の内容について簡単に報告いたします。

初めに、共同研究を行なうに至った背景、研究の動機及び研究内容について、私から総括的な報告を行ないます。その後、共同研究者各自がそれぞれの研究について報告を行ないます。

ここでネットワーク産業とは、サービスを提供するために不可欠なネットワーク構造を持つ産業で、電気通信業、情報産業、交通産業などの総称でございます。特に近年では、国内及び国外を通じて、ネットワーク経済関連産業の発展は著しくなっております。

ネットワーク産業に共通する特徴を経済学的に把握しますと、主として外部性、補完性、互換性、標準化といった点が挙げられます。

これらの特徴について若干説明をしますと、ネットワークの外部性は、需要サイドの消費決定が互換可能なネットワークの規模の大小を通じて消費者相互の選好に直接的影響を与えるような技術的外部性のことであります。たとえば携帯や固定電話のサービスは、ネットワークに加入する人の数が増えれば増えるほど、利便性が高まりますし、ハードウェアの効用は、それに対応するソフトウェアの充実度に依存するわけです。

また補完性としては、ソフトウェアとハードウェアのように、他の財と一緒に消費されなければならない財を補完財と言います。消費者はコンピュータとソフトウェアを同時に購入するように、個別財ではなく、システムとして購入します。私が昔、所属した会社、コニカの製品の例で言えば、カメラにはフィルムがないと写真は写りませんので、カメラとフィルムを同時に購入します。そういったものを補完財と言います。

互換性・標準化とは、補完製品は同じ標準で制御される必要がありまして、ある機器専用のソフトウェアパッケージ全てが他の機器でも実行可能であって、その逆もまた可能であることを意味します。たとえば同一のシステムに記憶装置やプリンタなどを接続できれば、2つの機器には互換性があるということになります。

次に研究背景ですが、ネットワーク産業におけるこうした特徴を基本的な分析視点を経済学的に整理いたしますと、ネットワーク関連産業に関して、企業間及び産業間におけるネットワーク財及び関連する財・サービスの標準化及び互換性に関する競争または協調にかかわる戦略、技術開発に関する戦略などが企業利潤、消費者余剰、さらに社会的厚生に与える影響の考察などがあります。

また、国際間におけるネットワーク財の標準化や互換性に関する競争または協調に関する戦略及び技術開発に関する戦略などが各国の経済・厚生などに与える影響の考察が重要な課題であります。

工学的に見ますと、近年ではネットワークをベースとして、企業外部にもわたる複数の人間がコラボレートするような技術展開から、ネットワーク産業は複数の人間が間違いを犯しがたい技術標準を確立して、これを広く活用することで、互換市場を生成することが肝要と考えています。

特にネットワーク産業の中で内部業務に標準化が後れている情報産業に焦点を当てまして、情報システム生成に自動化の考え方を入れた新しいシステム開発方式による開発標準化の例が企業内の技術互換性にどう影響を与えるのか、これを考察することを考えています。

また、保守及びテスト工程は情報産業の中でもコストが高いと言われていています。そこで、その標準化と保守要員の最適交代問題についても大きな課題となっています。

近年、日本の情報産業は、コスト競争を背景に、中国の湾岸にあるソフトウェア産業にアウトソーシングして、オフショア開発として発展していますが、生産性向上と品質確保のために開発標準化の役割は少なくありません。そこで、中国オフショア開発を進めていくうえでの課題について分析する必要があると思います。

本研究プロジェクトにおいては、ネットワーク産業について、経済学からのアプローチと経営工学からのアプローチについて行ないます。

研究テーマとして、第1に経済学からのアプローチとして、ネットワーク経済を構成する産業及びその財・サービスを研究対象に、消費の外部性に着目して、寡占産業における財及びサービスの互換性及び標準化に関する企業の戦略が社会に

おける厚生・企業利潤などに与える効果について分析しています。

ネットワーク財及び関連する財・サービスの標準化及び互換性などの問題を考慮する場合、次の2つのアプローチがござります。それは、ネットワーク産業の特徴としての財の間の補完性に焦点を置いたアプローチと、消費の外部性に焦点を置いたアプローチです。財の間の補完性に焦点を置いたアプローチでは、各企業がシステムの構成に関して補完性を有する財の組み合わせをどのように選択するかについて分析するものです。これに対して消費の外部性に焦点を置くアプローチでは、各製品の消費者が同一または互換性を有する製品を利用する消費者の数が多ほど、当該製品を利用する際の効用が増すという状況を分析することでありませう。

本共同研究においては、ネットワーク産業を研究対象に、後者の外部性に着目して、寡占産業における財及びサービスの互換性及び標準化に関する企業の選択並びに価格または数量に関する戦略が社会的厚生、企業利潤などに与える影響について分析を行なひました。

既存研究ではいづれも正のネットワーク外部性のみが考慮されていますが、本共同研究においては、正のネットワーク外部性だけでなく、負のネットワーク外部性についても考慮したモデルに関する考察をしています。

先行研究としては、これまでなされた経済学的アプローチに関連する研究として、ゲーム理論的な分析に基づく研究の主要なものについて簡単に述べたいと思ひます。

KatzとShapiroについては、ネットワーク外部性が存在する場合における互換性と標準化の問題、新技術と導入の問題について、ゲーム理論的な考察を行なっています。

Farrellらは、ネットワーク外部性が存在する場合において、旧式の技術に関する既存ベースのネットワーク外部性によって、より効率的な新技術が採用されなくなる可能性及び効率性で劣る新技術が採用される可能性を考察しています。

これらの研究を基礎として、これまでネットワーク外部性に関する多くの研究がなされてきました。しかしながら、それらの研究においてはいづれも正のネットワーク外部性のみが考慮されて

います。これに対してわれわれの研究は、正のネットワーク外部性だけでなく、負のネットワーク外部性についても考慮したモデルに関する考察が行なわれています。

考察したモデルの概要は次の通りです。ネットワーク財を購入する消費者の効用の仮定として、購入する財またはサービスから得られる直接的な満足は、その財を消費する消費者全体及び競合する財を消費する消費者全体に依存するものと仮定いたします。

ネットワーク財については、製品差別化を考慮して、寡占企業がそれぞれの品質の異なる財またはサービスを提供すると仮定いたします。

ここで各企業が他の企業の財と互換性を有する財を生産する場合には、正のネットワーク外部性が関係すると考えています。逆に互換性を有しない財を生産する場合には、負のネットワーク外部性が関係するものと考えませう。これまでの関連研究においては、正のネットワーク外部性と負のネットワーク外部性を同時に考慮するようなものではなく、この点が本研究の重要な貢献をなすものと考えています。

本研究はこのモデルをもとに、ネットワーク産業における消費の外部性に着目して、寡占産業が提供する財及びサービスの互換性及び標準化に関する企業の選択及び価格または数量に関する戦略が社会的厚生、利潤などに与える効果について分析するものです。

このモデルにおける部分ゲーム完全均衡の特徴は、互換性に関する企業の選択が、提供される財本来から得られる効用とネットワーク外部性の効果の大きさに関するパラメータに依存して非常に異なったものとなり、そのため、利潤、消費者余剰及び社会的厚生水準も影響されることになりませう。

以上の考察を基礎に、国際間の寡占企業の競争を考え、標準化に関する政策が及ぼす社会的厚生、利潤などへの効果を分析することによって、本研究の拡張が可能と考えています。

もう1つの経営工学からのアプローチとして、大宮は、情報システム開発における設計工程に標準化を導入した新しい設計書作成作業方式による品質、生産性向上の事例のケーススタディを行なっています。また、情報システム産業における

保守要員の交代タイミングに着目した保守要員の最適な交代時間モデルの提案を行なっています。

さらに、ネットワーク産業界において、特に情報産業に着目して、情報産業における情報システム生成に自動化の導入を目的とするリポジトリを利用した開発支援システムの構築事例を丸山から発表いたします。

最後に私から、北京、済南で実施した中国現地

調査の結果を踏まえた中国のソフトウェア企業の現状と課題について、分析及び事例研究として発表いたします。

各々の研究については各共同研究者それぞれが報告いたしますが、時間の関係から全員の発表はできないことをあらかじめご了承くださいと思います。

それでは早速、大宮から報告いたします。

「情報システム開発における設計工程への標準化導入事例」

大宮 望

それでは「情報システム開発における設計工程への標準化導入事例」と題しまして、コニカミノルタ情報システムの大宮から発表させていただきます。進め方としては、1番目に背景、2番目に現状の問題点、3番目に新しい開発方式、4番目に評価、そして最後に結論という順でお話しさせていただきます。

まず背景です。情報システム開発には品質及び生産性向上の課題が存在しております。品質及び生産性向上の課題への対応策としては標準化が代表的であり、標準化の狙いとしては、互換性、少数化、安全、生産性の向上、再発防止、トラブル一掃、自動化、モラルの向上、品質の向上、共通部品化やコストダウン、会社の利益、単純化などがあります。これらによって、品質及び生産性の向上に標準化は有効であると言われております。

標準化の取り組みではハードウェアの取り組みが代表的です。製造業における標準化が特に代表的ですが、情報システム開発においても標準化の取り組みがなされてきております。しかし、標準化の段階はハードウェアとソフトウェアでは大きく異なっておりまして、ハードウェアの標準化は作業の標準化、工程の標準化、製造方法の標準化、そして生産方式の標準化という段階を経て標準化は進められていきます。ソフトウェアの標準化は、部分的な標準化、作業の標準化、工程の標準化、ソフト製造方法の標準化と、ハードウェアの標準化段階より1段階遅いイメージで標準化が進んでまいります。

ソフトウェアもハードウェアをまねて標準化を進めてまいりましたが、標準化によってハードウェアと同様な効果を得られていないのが実情であります。その理由としては、ソフトウェアは無形の生産物であり、ハードウェアと全く同じ標準化はできないと言われております。ハードウェアは目に見えますが、ソフトウェアは目に見えないといったイメージであります。

標準化の効果によって品質及び生産性の向上はソフトウェアにおいても得られるものであり、ソ

フトウェアに標準化を適用することは有用であると考えております。しかし、先ほど申し上げました通り、ソフトウェアとハードウェアでは同じ標準化が適用できません。このため、ソフトウェアに適した標準化の適用が必要であると考えております。

ここで情報システムのつくり方の工程の簡単なお説明をいたします。まず、最初に情報システムをつくるための要件を決定する工程で、これを要件定義工程と言っております。要件定義工程を受けまして、その内容を情報システムに反映するための設計工程、そしてプログラミング工程でソフトウェアをプログラミング言語によってソフトウェア化します。プログラミング工程で出来上がったソフトウェアをテスト工程でテストし、本運用で実際にソフトウェアを使うこととなります。

現在、情報システム開発における品質及び生産性向上の取り組みの中心はプログラミング工程となっております。プログラミング工程では、標準化への考え方をベースとしたオブジェクト指向言語やコンポーネントの利用など、さまざまな取り組みが行なわれております。現在の情報システムの取り組みの中ではプログラミング工程が中心と言えるのではないかと考えております。

逆に、その他の工程についての取り組みはあまり行なわれておりません。特に設計工程においては、設計書の作成方法に問題が数多くあり、多くの工数を必要としております。設計工程の品質及び生産性向上の取り組みを行なった事例もなく、今後この設計工程の取り組みが大きな課題になると考えております。

これらの課題を受けて、本研究の目的は、情報システム開発の設計工程に標準化を取り込みました新たな設計書作成作業方式による品質及び生産性向上事例の紹介を行ないたいと考えています。

補足ですが、「情報システム」についての定義を簡単にご説明させていただきます。情報システムというのは端的に言えば、「人の行なう作業を、プログラム言語によって作成されたソフトウェアが代行し、作業効率を向上させるための仕組みである」と言い換えることができると考えております。

情報システムのつくり方は建物の建築によくた

とえられます。たとえば2階建ての洋風建築、2階の窓は2つ、耐震強度は震度7、そしてエレベーターが欲しい、などといった顧客の要望を聞いて、細かな設計図をつくります。この設計図をもとに建築し、家が完成するわけですが、この工程が情報システムに類似していると言われているようです。

では実際の情報システムのつくり方を先ほどの例と同じようにご説明させていただきます。情報システムも同様に、顧客の要望をまず初めに聞いてまいります。たとえば口座への入金できるシステムが欲しい、停電しても動く情報システムで、指紋認証機能が欲しいなどといった要望を受け、情報システムも設計図を作成していきます。情報システムの場合は、これを設計図と言わず、設計書と呼んでおります。

たとえば、口座番号では5桁の口座番号を入力できるようにする。入金額入力項目では、7桁までの金額を入力できるようにする。また、入金者の電話番号を入力できるようにする。そして、数値入力では、口座や入金額、電話番号の数値をこの機器から入力します。指紋認証機器では、指紋認証ができなければ口座番号も入金額も入力できない仕組みをつくります。要望にありました停電しても動くシステムをとということで、無停電電源装置を設置することになります。

このようなことで設計書を作成し、この設計書に沿って実際に情報システムを開発します。プログラミング言語によってソフトウェアを開発し、画面上にこのような画面が表示されるような仕組みになってくると考えられます。

これらの現状の問題点を整理しますと、設計工程の課題を解決するに当たって設計工程における課題の確認を行なった結果、工程間継承情報の修正に時間がかかっていることが明らかになりました。工程間継承情報とは、複数の設計書で共通に利用される単語や文章を指しています。たとえば工程1の設計書A.ここにはA機能に数値を入力し、結果Bを得られる。工程2の設計書B、A機能は100以下は入力できないといったような場合、A機能のこの部分が工程間継承情報と呼んでいる部分になっております。

さらに詳細に工程間継承情報の説明をさせていただきますが、工程間継承情報の詳細例は以下の

通りです。担当者A、B、Cと、さまざまな担当者がおりますが、「口座番号は5桁」ということを同じように個々に入力し、設計書を作成しております。このため、入力ミス、口座番号の変換間違いなどによって、ミスが発生しやすい状況になっています。また逆に、「口座番号」という単語を変更した場合には、多くの工数が必要になってしまうこともわかっております。

この現象につきまして調査の結果、同一文章が130カ所もある場合もあることがわかっております。たとえば「口座番号は5桁」という文字が130カ所、さまざまな設計書に存在していると言い換えることができます。

設計工程における問題点は、いま申し上げました工程間継承情報以外にも多くの問題が発生していることも明らかになりました。まず1点目、口座番号は5桁と先ほど申し上げましたが、この大事な記載がない設計書もあることがわかりました。設計書がどこにしまっているのかかわからないといった基本的な問題があることもわかっております。誰もが自由に見えてしまう、これによって機密漏洩の可能性が出てまいりました。また、口座番号は他の設計書にもたくさん記載しているということで、工程間継承情報が散らばっているという状況です。設計書のフォーマットは各個人によってさまざまなレイアウトがありますので、フォーマットを毎回考えるという問題も抱えておりました。

いま申し上げました問題点をまとめると、1点目、フォーマットを毎回考え、時間がかかっている。2点目、重要な情報が欠けていることがある。3番目、今回の問題の中心でもありますが、工程間継承情報の再利用ができない。4番目、しまっている場所もわからない。そして機密漏洩の危険があるといった5点にまとめることができました。

さらにこの問題点を整理した結果、以下の2つの課題があることが明らかになりました。まず1点目、標準設計ルールの課題です。標準ルールは存在しておりますが、このルールが人に依存している関係から、強制できない状態にあることがわかりました。2点目、設計書作成ツールの課題ですが、設計書作成ツールがワープロのため、多くの設計書に共通な情報（工程間継承情報）が個々のファイルに独立し、有効に再利用できないと

いった問題があることがわかりました。

これらの課題を受けまして、新しい開発方式の検討を行ないます。いま申し上げました課題を解決するアプローチとして、標準化を強制できる仕組みが必要である。ワープロでは標準化を強制できる仕組みはできませんでしたので、これを強制できる仕組みの検討が必要であると考えました。

課題解決における具体的な方法は、設計書の標準化、設計部品による自動化・機械化を実現するために、標準化を強制できる開発支援システムを開発し、新たな作業方式によって設計書を作成する方式です。

いま申し上げました設計部品とは、前半でお話ししました工程間継承情報を部品化したものです。たとえば設計書をすべて手づくりの場合、すべてが手づくりになってしまいます。しかし、手づくりの部分に部品をはさみ込むことによって、手づくりの部分減らし、生産性・品質向上を得ることができると考えております。

この考え方は製造業で行なわれている標準化の1つであります共通部品化をソフトウェアに適した標準化にアレンジしたものであり、製造業と同様に、設計部品を利用することによって手づくりの部分減らし、品質及び生産性向上が可能になると考えています。

続きまして、開発支援システムを開発するに当たりまして、開発支援システムで対象とする設計書の簡易特定を行なった結果、70種類存在していた設計書から18種類の設計書を抽出しました。70種類の設計書は実は、さまざまな分析を行ないますと、類似している設計書が多く、これをまとめると18種類の設計書であることがわかったということです。

この18種類の設計書から、同じ単語や文章の記載を洗い出し、設計部品の取り出しを行なってまいりました。取り出し方法は、すべての設計書から同じ単語、文章を抜き出して行なっております。

さらに開発支援システムの開発を行なうに当たり、設計部品を利用される順位を調べることにしました。この順位が決まることによって、設計部品が有効に利用できるからです。たとえば設計工程において、設計工程の前半で使える設計部品、設計工程後半で使える設計部品が存在していること

がわかっております。この後半で使える設計部品は前半では使えず、前半で使える設計部品は後半で使えることはありません。逆に、設計工程全般で使える設計部品が存在していることもわかっております。

この順位を特定することによって設計部品が有効に利用できるイメージについてご説明します。たとえば茨城、塩山、横浜、牛久、甲府、埼玉、東京、日野、八王子などといった、このデータがすべて設計部品だと置き換えていただきますと、この中から東京都の日野市を選択したいといった要件があった場合、この中から「日野市」を選択することは非常に難しいと考えられます。しかし、まずこの設計部品を都道府県別に表示し、「東京都」を選択します。続きまして、東京都の中でも市の範囲である設計部品を絞ることによって、「日野市」を選択することは容易に可能になると考えています。

この例では「都道府県」が設計部品の順位としては1位、「市」が2位の順位となっております。このように順位をつけることによって、「日野市」が簡単に選択できることがわかります。これが設計部品の順位づけの有効な点であります。

いま申し上げました順位づけをさらに細かく行なってまいります。設計部品の順位によってデータを分析し、5種類にグループ化しました。これを部品種別と呼びますが、開発基準、プロジェクト情報、データベース情報、設計書情報、そしてプログラムIDという5種類に分類できることがわかりました。

開発基準情報では開発者の氏名、プロジェクト情報ではプロジェクトの名前が代表的です。データベース情報ではデータを確認するデータベースの情報、設計書情報では口座番号を入力する方法などの設計書情報、プログラムIDでは個々のソフトウェアを識別する番号となっております。この5グループの順位づけを行なった結果、1位が開発基準情報、2位がプロジェクト情報、3位がデータベース情報、4位が設計書情報、同じく4位にプログラムIDとなりました。

たとえば1位の開発基準情報で、開発者の氏名ということで私の名前「大宮」という文字を使った場合、ここで1位の設計部品を登録したときには、4位の設計書情報で「大宮」という文字を使

おうとすると、ここに1回登録することによって、何度もここで設計部品を流用することが可能になります。順位が逆転して4位に開発基準情報があった場合には、設計部品としてはほとんど役に立たないことがわかります。

分析をさらに進めていきますと、プロジェクト情報とデータベース情報が逆転していることがわかりました。企業によっては全社規模のデータベースを保持している場合、複数のプロジェクトを横断的またぐ場合があることがわかったからです。この結果、1位に開発基準情報、2位にデータベース情報、そして3位にプロジェクト情報、4位に設計書情報、同じくプログラムIDと順位を決定しました。

いま申しあげました順位づけ、設計部品の抽出をもとに、実際に開発支援システムを構築した際のイメージをごらんに入れます。まず左側に設計部品がライブラリーに格納されております。そしてこの設計部品をもとに開発支援システムを利用し、成果物である設計書を作成していきます。

たとえば口座番号、入金額、電話番号といったように設計書に設計部品を配置することによって設計書が完成していきます。開発支援システムでは、設計部品を配置することによって設計書の作成を可能としております。続きまして工程間継承情報を設計部品化し、部品を修正することによって、1度の修正で複数の成果物を自動で修正できるシステムとしました。たとえば「口座番号」という文字を何らかの違うかたちの文字に変更した場合、設計書は何枚あっても自動的に修正できるようになりました。

実際に開発しました開発支援システムをごらんに入れます。以下の図の通り、設計部品を選択し、設計書が作成可能となっております。ここにあります設計部品は順位づけがされており、必要な設計部品のみが表示される仕組みとなっております。

続きまして評価です。まず設計書作成についてのワープロと開発支援システムの違いについてごらんに入れます。ワープロ利用時には設計書のフォーマットを考える時間が1フォーマットに約1日かかっておりましたが、フォーマットを考えなくてよくなりましたので、定量的な効果として、18種類の設計書のフォーマット、18日分の工数削減

が達成されました。定性的な効果では、設計書のフォーマットが統一でき、品質が向上したと考えられます。

設計書修正の評価では、ワープロでは設計書修正には最大で1人で2日かかっておりました。冒頭で申しあげましたが、同じような文章が約130カ所にあるため、修正に約2日間かかっていたということになります。これを開発支援システムに置き換えた場合、設計部品を修正することによって完了し、約10分の1の工数で済むことがわかりました。これは定量的な効果です。

定性的な効果では、設計部品を修正することによって陳腐化が防止されています。

陳腐化といいますのは、修正している本人にはわからないような設計書に設計部品が使われている場合もあることがわかっております。その際には、設計部品を修正することによって、自分の意識していない部品の設計書まで自動で修正されるために、設計書は陳腐化されることが防止されたといったイメージになっております。

最後に結論です。いま申しあげましたように、情報システム開発における設計書作成業務に対しまして、標準化の一環として、ワープロの代わりに開発支援システムを利用する作業方式を開発し、これによって設計書を作成することによりまして、先ほど申しあげました評価の通り有効であることが導かれました。

今後の課題として、以下の3点が挙げられております。1番目、他工程への検討です。今回は設計工程を対象としましたが、情報システムにはこれ以外の工程として、要件定義工程、テスト工程、プログラミング工程など、さまざまな工程が存在しております。これら他の工程でも同様な検討を行なうことによって生産性や品質向上が得られるのではないかと考えますので、他の工程でも同様な検討が必要であると考えております。

次にネットワーク接続の検討です。この開発支援システムは社内利用を前提にしております。このため、イントラネット上での利用が前提となっております。しかし、社外開発で委託業者などにこのシステムを利用される場合、ネットワークポリシーなどの関係上、接続することができません。このため、外部業者にプログラムや設計書を担当させる場合に、どのようにネットワークをつ

ないでいくかという課題が残っていると考えております。

最後に図の検討です。本研究では単語や文章を工程間継承情報としております。しかし、情報システムの設計書にはフローチャートやさまざまな図による画面のレイアウトなど、図も同様に設計書に描かれております。この図を工程間継承情報とする検討が必要で、図を文字に変換し、同じこ

とが表現できないかと、さまざまな取り組みを行なってまいりましたが、実際にはなかなか同じような効果を得ることができませんでした。今後、このような図をどのように工程間継承情報に取り込んでいくかの検討が必要であると考えております。

以上で私の発表を終了させていただきます。ご清聴ありがとうございました。

「保守工程における業務引き継ぎ モデル設計とシミュレーション」

中邨 良樹

続きまして中邨から、情報システム産業における保守要員の交代タイミングに着目した保守要員の最適な交代時間モデルの提案に関しまして、「保守工程における業務引き継ぎモデル設計とシミュレーション」と題して進めてまいります。

研究背景、目的、それから、情報システム研究の現状と課題を共同研究者の大宮氏と一緒に抽出してきた経緯をお話します。その大宮氏から様々な実現場データをいただきましたので、そこから結果を分析して仮説設定、モデルを構築、そしてシミュレーションというような流れで研究を進めてきましたのでそれを説明します。最後に成果と課題です。

研究背景は、近年の情報システム投資は対売上高で4%ほどと言われております。研究開発投資でもいま5%以上という世の中ですので、情報システム投資は非常に大きいと思われまます。

情報システムは、企画、設計から保守まで、多くの部分を管理していかなければなりません。先ほど大宮氏から説明がありました通り、要件定義、設計、開発、テスト、保守と、いろいろな管理の部分がござりますが、これに関して多くの先行研究がなされております。たとえば見積り目の計画と実績の差異をなるべく減らすための管理手法とか、バグを出さないためのプログラミング手法とか、費用最小化問題などが挙げられます。

しかし、今回の研究で着目した点は、情報システム導入後は不具合が必ず発生し、そのための増築も必ず必要になる。そのような状況になりますと、開発SEが保守工程に取られて、新たなプロジェクトに投入できない。それが新たなビジネスチャンスを逃しているという問題があります。

そこで、保守業務を見越して、前段階の計画の精度に誤差があってもいいのではないか。ある意味、いい加減さがあってもいいのではないか。大事なのはいつ保守業務を引き継ぎせるかだということです。そこで本研究の目的としては、計画から引き継ぎまでの流れをモデル化し、最適な引き継ぎ時期の提案と投資規模比率のシミュレーシ

ンを行うというものです。

本研究の流れといたしましては、まず、最初に現場のヒアリングを行ないまして、保守時の課題や要望を明確化し、そこから仮説とモデルの設定をします。モデルを実証するために、現場の実データからモデルの推定をし、シミュレーション及び考察を行います。そして再度、結果に対する現場ヒアリングとモデルの有効性検証をしていくという流れになっております。

以上のように、実現場での課題を抽出したうえで、モデル化・シミュレーションを行なうというかたちをとっておりますので、モデルの精度としては低いということをお断りしておきます。

研究目的としては保守要員の交代タイミングを推定することを試みようと思っております。その際、3つの要素が大事になってきます。まず、不具合が起きて、それを補うための再投資（増築）の意思決定時期と増築前の期間。それから増築の規模・大きさ。そして引き継ぎのための指標の抽出。この3つの要素をモデル化して、シミュレーションを行なっています。

これがヒアリング結果と得られたデータです。日付、問い合わせ件数、問い合わせ時間。つまり、納品したユーザーが1日に開発者に問い合わせた件数と、そのときに対応した時間です。増築決定日というのは不具合が起きたので情報システムを増築しようと決定した日。そして増築分を納品した日、問い合わせ発生日等々、いろいろな項目があります。主観的かもしれませんが、このプロジェクトが成功か失敗かというデータも入れております。

それ以外に、情報システム導入後は不具合が必ず発生してしまう。そのため、担当SEがそのまま保守工程において、増築分のシステム開発・保守、さらに増築という流れを繰り返している。また、担当SEが新たなプロジェクトに移行させるタイミングを見分けることは難しい。そもそもどれぐらいの規模のライフスタイルコストがあるのかということが情報としてあったほうが、好ましいのではないか、ということがヒアリングの結果と得られたデータから見られます。

（資料1）このようなデータ及びヒアリングをもとに、次のような仮説を立ててみました。仮説1として、問い合わせ件数は、成長曲線のように、

「中間」で急激に伸び、「後半」では終息に向かうのではないかと。仮説2では、前半から中間の問い合わせの中に、増築の意思決定が行なわれるはずである。また、その増築額と意思決定されてから納品までの時間は、前半部分の問い合わせ件数によって変動するのではないだろうか。仮説3として、問い合わせの対応時間は大きく2回の山がある。1回目は納品したときの問い合わせ、2回目は増築が行なわれて新たに増築分が納品された後の問い合わせ、この2つの大きな山があって、この大きな山を基準に、業務引き継ぎが可能となる指標が抽出できるのではないだろうか。データ分析の結果から、このような仮説を立てて、この仮説をもとに実証分析をしていこうと考えています。

(資料2) モデル化に際し、いつ引き継ぎが行なわれるのか、初期の投資規模、増築の規模、この3つの点が重要ですので、それを推定できるような形にします。したがって、初期のプロジェクト規模から、この規模によってどの程度の情報システムに関する問い合わせがあるかどうかの関係をモデル化します。そして、その問い合わせ件数によって、どの程度情報システムの増築を行なうかという増築規模を意思決定します。その増築の規模によって新たな情報システムの追加分の納品日が決定され、それが納品された後に、いかにプロジェクトの主要メンバーを引き継がせるかという引き継ぎ指標を算出します。

このモデルを関数関係で説明していきます。初期のプロジェクト規模と問い合わせの件数については、初期のプロジェクトの規模が大きければ大きいほど、問い合わせ件数は減っていくのではないかと。同様に、問い合わせ件数が多ければ多いほど、それだけエラーが多いので、増築規模も大きくなっていくのではないだろうか。増築規模が大きければ大きいほど、納品日までの時間が長くなっていくのではないかと。納品までの期間が長いということは開発期間が長いわけですから、開発期間が長ければ長いほど、引き継ぎ指標までの期間も長くなってしまわないかと。こういう仮説を立てて実証推定をし、最終的にはモデルにします。

(資料3) これが推定結果です。いただいたデータをもとにデータ分析して、データから仮説をつ

くって推定するというかたちをとりました。したがって、決定係数などは精度の低いものになってしまっています。

初期の投資の意思決定支援、初期のプロジェクト規模によって、問い合わせ確率、そして最終的には引き継ぎ時間はどのように変化するか、シミュレーションを行ないました。その結果、問い合わせ件数が多いほど、納品までの時間は短い。初期の規模が大きいかほど、引き継ぎ時間は短くなる。最初に惜しみなく投資するほうが、引き継ぎ時間は短くなって新たな引き継ぎへというかたちになることが、モデル上で説明できます。

(資料4) 同様な線形モデルを利用して、問い合わせ件数の変動によって、どのように引き継ぎ時間が変化するか、簡単なモンテカルロシミュレーションをやってみました。問い合わせ確率は指数分布すると仮定し、初期投資の規模を0.01と仮定したときに、問い合わせ件数を500回の乱数発生させたときの引き継ぎの時間はどうか。結果として、平均27.83日で、標準偏差2.77日になります。

研究の成果としては、増築の意思決定期と増築額に関して情報の1つを抽出できたのではないかと。引き継ぎのための指標も抽出することができた。推定したモデルによってシミュレーションすることができ、今後の情報システム導入の支援をすることができたということを考えております。

ただし、課題もあります。最初のヒアリングの基礎データをモデルに組み込んでいないので、将来的には人数の変動によって引き継ぎ指標がどのように変化するか、プロジェクトの複雑性によってどのように変化するか、ということも取り込んだモデルにしていきたいと考えております。

データ分析の精度が非常に悪いことも、今後の課題の1つです。今回はデータを見て、そこから仮説を立てて分析して行きました。本来はデータ分析の精度をより高めてモデル化していくことが大事だと考えています。

また、線形で処理しておりますので、シミュレーションの結果は一方向になっております。これも今後の検討課題だと思っております。

以上で発表を終えさせていただきます。

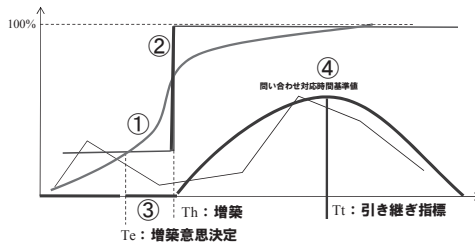
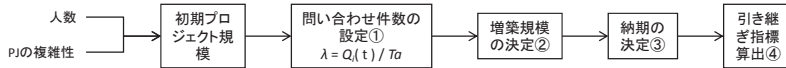
資料 1

得られたデータからモデル化へ

プロジェクトID	問い合わせ到着日	作業(時間)	問合せ件数	問合せ時間
0120	2006/04/18	0.1	1	0.1
0120	2006/04/24	0.1	1	0.1

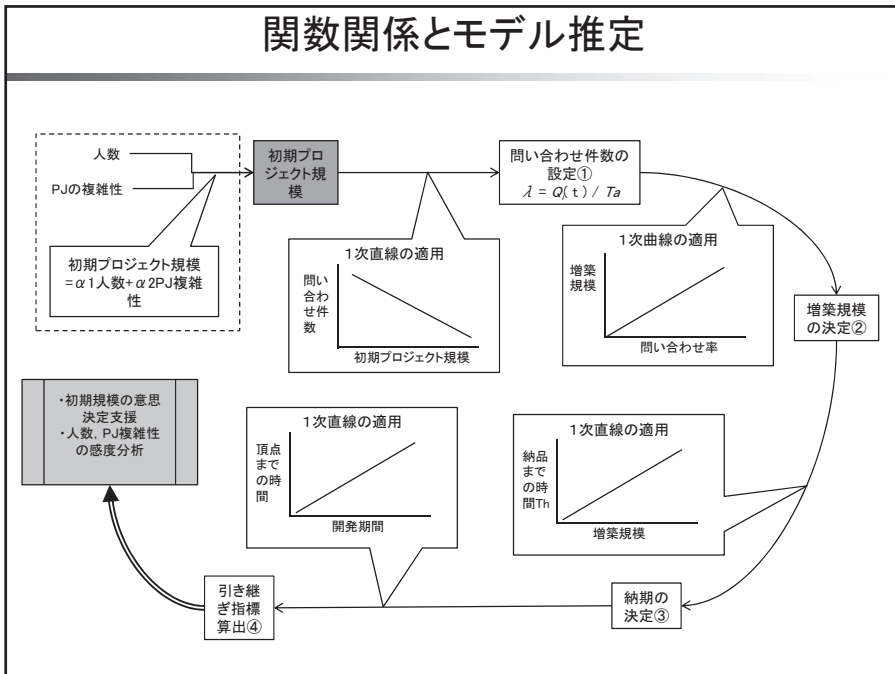
増築規模	増築日	増築規模特定日	PGM本数	DB数
	2006/4/24	2005/12/1	120	58
			1352	318

稼働状況	人数	PJの複雑性	納期遅れの有無	成功、失敗
TRUE	100	1	2	1
TRUE	2	1	1	1



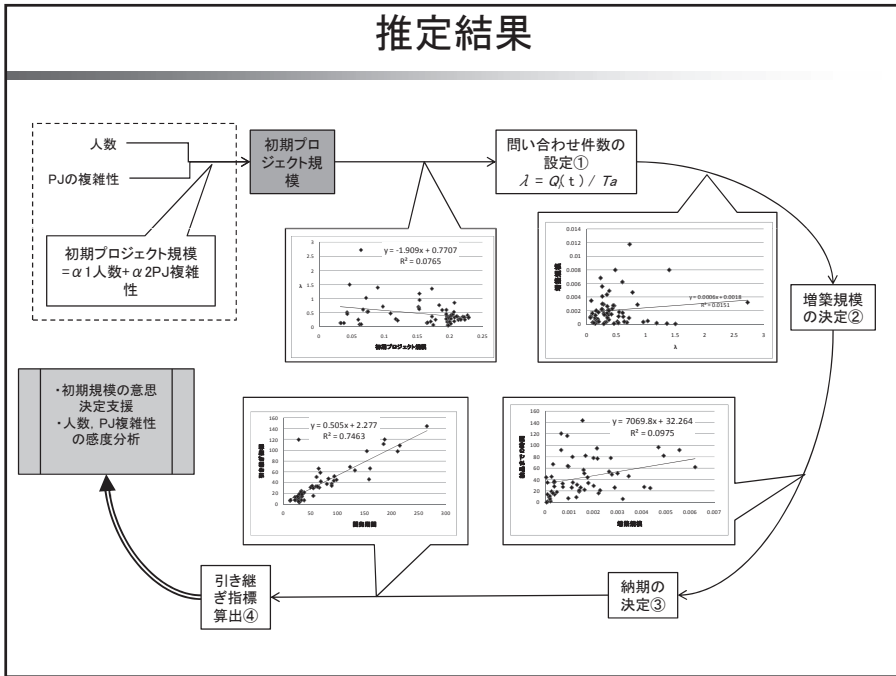
資料 2

関数関係とモデル推定



資料3

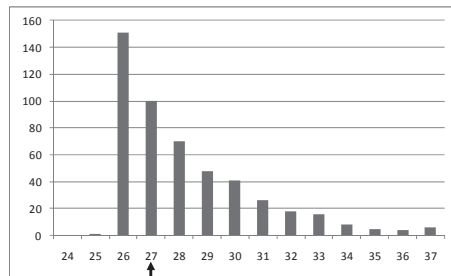
推定結果



資料4

シミュレーション2

問い合わせ確率を指数分布すると仮定し、初期規模0.01のとき、500回確率を発生させたときのモンテカルロシミュレーションの結果



考察

・平均27.83, 標準偏差 2.77となる

「設計部品リポジトリを利用した情報開発システムの提案とその応用」

丸山友希夫

続きまして丸山から、「設計部品リポジトリを利用した情報開発システムの提案とその応用」ということで報告をさせていただきます。

初めに研究背景、研究目的を述べます。その後、本プロジェクトにおける大宮氏主導による設計部品リポジトリを利用した情報システムの構築方法の提案を紹介させていただきます。ただ、先ほど大宮氏からもありましたように、ソフトウェアはハードウェアに比べて中身が見づらいので、設計部品リポジトリというものがどのように使えるか、設計部品リポジトリを利用した情報システムの応用例として、私のほうの開発事例を紹介させていただきます。そして最後にまとめという流れで報告をさせていただきます。

まず研究背景ですが、近年の情報システム開発では、市販・汎用品のワープロソフトとか表計算ソフトのように、年々機能の複雑化があります。それに従いまして、開発規模の拡大も当然あるかと思えます。使う数も多いので、開発をどんどん行なっていくと時代に乗り遅れてしまうということで、開発期間の短縮化傾向があります。しかし、いくら開発期間短縮化をしたとしても、生産数は多く上げなければいけないし、使いやすくなければいけないということで、生産性・品質向上は必要不可欠になってくる。

情報システム開発における生産性・品質向上に関する手法には、再利用と自動化という2つのキーワードがあります。再利用に関しては、部品のモジュール化というのがプログラミングのところで関連してまいります。そこで本研究目的としては、先ほど情報システム開発における工程が5つありましたけれども、その全コストの5分の2を占めるテスト工程に、再利用、自動化を導入したシステムの構築を提案しようということです。

それでは実際に設計部品リポジトリを利用した情報システムの提案ですが、「設計部品リポジトリ」という言葉はあまり耳にしません。そこで簡単な言葉の定義をさせていただきますと、「設計部品」というのは工程間を継承するものなのです。

で、「共通で使用する単語または文章」ということとなります。「リポジトリ」は、「設計部品及び設計部品に関するマッピング情報、カテゴリ情報を格納するエリア」、つまり、どこに配置するかという場所、エリアです。

(資料5) では設計部品とリポジトリというのはどういう概念かと申しますと、カテゴリ1、カテゴリ2の中に、設計部品がそれぞれあります。まずカテゴリ情報として設計部品があるわけですけれども、これを成果物フォーマット情報として、成果物1には設計部品1をマッピング情報として配置をしましょう。成果物2には設計部品2を配置をしましょう。出力すると、成果物1では設計部品1と縦に配置する。成果物2では設計部品2は横に配置する。設計部品とリポジトリの関係を示すには、このような概念があるということです。

このような概念を念頭に置きまして、設計部品リポジトリを利用した情報システムの構築をしていくわけですけれども、なぜテスト工程にシステムを構築するのか。先ほど研究目的で、情報システム開発における全コストの5分の2を占めるテスト工程に再利用、自動化を導入したシステムの構築を提案すると申しました。開発工程は要件定義、設計、プログラミング、テスト、保守の5工程がありますので、もし均等にコストを使うならば、それぞれの工程で5分の1ずつ使っていくことになるはずですが、ところが、テスト工程で全コストの5分の2を占めるということは、ここに特に力を入れようということです。

それはなぜなのかといいますと、近年の汎用ソフトを見てわかりますように、機能の複雑化、開発の拡大によって、要件定義、設計工程での仕様書づくりが複雑化しています。当然のことながら、それをテストする仕様書はもっと複雑化、拡大している可能性が高く、おそらくものすごい労力を要するだろう。したがって、このテスト工程に再利用、自動化を効率よく使えば、労力を少なくし、開発期間を短縮することもできるのではないか、ということでテスト工程に着目したわけです。

(資料6) 設計部品リポジトリを利用した情報システムの提案を図に表したのがこれです。成果物A、B、Cを、マッピング情報、カテゴリ情報、成果物フォーマットとしてデータベースに格納す

る。それを抽出するときには、共通で使う設計部品もしくはマッピング情報によって、成果物A, B, Cが出せる。このようなシステムを大宮氏主導のもとで提案を行なったということです。

システムの提案だけでは意味がないので、これで評価がよくなっていなければいけない。システムの評価方法としては、一般的にソフトウェアの品質分析で用いる手法としてDRE (Defect Removal Efficiency) があります。m/m+hという式で与えられていますが、mは開発期間内に発見したエラー数、hは開発期間内に発見しなかったエラー数です。したがって、DREが1に近づけば近づくほど、製品後のエラーが少なかった。品質のいいソフトウェアであることがわかります。さらに経営工学的なアプローチから、DREの値を用いて統計的検定を行なうことによって、システムの評価を深められることがわかっております。

以上が設計部品リポジトリを利用した情報システムの提案ですが、次に、この設計部品リポジトリを利用した情報システムの応用例を示しながら、構築したシステムをもう少しわかりやすく説明していきたいと思っております。

私の職場が実験環境にありますので、そこをどうにかして組み込めないかということでつくったのですが、幾つかの工程から成る実験環境において、各工程の実験結果及び全工程の実験結果を管理するシステムにリポジトリを利用して実験データを管理しようということです。この場合の設計部品は、実験ID、実験日、実験者などです。

実験現場の環境は、中が吹き抜けの建物で、われわれがいる居室と呼ばれるスペースがあります。吹き抜けの反対側に実験室があり、その隣に機器室があります。実験者は居室と実験室の間を移動して、実験が終わったら、戻って居室で実験結果をPCで打ち込むというかたちです。

実験作業手順として、1つの実験結果が出るまでにはこのような工程があります。まず初めに、きょうはどのサンプルを実験するか、サンプルの選択をします。それに対して実験工程A, B, Cとあって、実験工程が終わって実験結果が出ると、品質検査を行なったうえで、実験を使うチームにそのサンプルの受け渡しをする。

実験者は5名です。実験工程はA, B, C3つしかないのですが、実験者は5名いて、その5名が、ある日はAとBを担当したり、ある日はBとCを担当したり、その都度決まる。誰がAとかBとかは決まっていない。また、サンプルの選択は1名が担当して、情報があっちへ行ったりこっちへ行ったりしないように管理しています。

実験者は自分が担当した工程のチェックは当然自分で行ないます。実験結果にはエラーが出現する可能性があります。実験Cまでやってはじめてエラーが出るものもあれば、実験Bの段階でエラーが出るものもあります。もし途中でエラーが出た場合は、もう1回そこでやり直しをして次に引き継ぐ。このような実験の流れになっております。

実験結果の記録は、まずサンプルの選択をした際には必ず、きょうは誰々がこのサンプルを実験してくださいと担当を決めて、ファイルに記載します。実験が始まって、実験A、実験B、実験C、品質検査では、終わった後に必ず各工程でその結果がよかった悪かったとか記載しているわけではありません。実験結果は表計算のエクセルファイルを使って管理していますが、実験中は実験者独自のノートに記載して、その後、居室に戻ってファイルに記載する。これは実験的なエラーが多くなることに結びつくかもしれません。

情報システムを導入する前の実験結果のファイルは、1つの表計算のファイルの各シートに違う実験の結果がまざっている。工程Aのシート、工程Bのシート、工程Cのシートがあって、それぞれの実験者が実験IDを自分が担当した分を割り振られたシートからコピーしたりしてしまっていたので、既存データの上に上書きしてしまったりということも起き得るわけで、管理的には非常によくない状態でした。

情報システム導入前の状況をまとめますと、実験データを管理するファイルは1つのコンピュータの中に1つしかない。そこに実験者が実験結果を投入したり、確認をしたりする。実験を取り締まる管理者または関係者は、そのファイルを信用して閲覧し、実験の進行を確認していました。ところが、いま申したように、上書きされて投入データが消滅したり変わってしまっていることがある。各工程の結果を投入しようとしても、投入

データが消滅していて実験IDが登録されていない場合もある。また、1つのファイルをみんなで共有しているので、誰かがそのファイルを使っていると、もしくは開いたままにしていると、ほかの人は使えない。誰が開いているのかわからないために、確認にも時間がかかる。そのような不便さやムダもあったわけです。

ではどのようなシステムを導入すればいいのかということで、情報（データ）にミスがないシステムの構築、そして構築したシステムが使いやすくてはならないという、2つの課題を挙げました。そのための実際の作業として、まずRDB（Relational Data Base）の導入で、複数の人数がいっぺんにファイルにアクセスできて、確認もでき、データ導入もできるようにする。さらにそれをもう少し効率よくするために、システムをオンライン化する。

実際の構築はどのように行なったか、先ほどの建物の図で示しますと、居室に各個人のPCは1人1台あります。機器室にサーバーを置きまして、この中にRDBを構築する。同時に、実験室の各工程それぞれにPCを新設して、実験が終わったらすぐその場で書けるようにする。それをネットワークでつなぐ。サーバー、居室のPC、実験室のPC、すべてオンライン化でつながるように構築しました。

（資料7）これが情報システム構築の概要です。開発者は私ですけれども、サーバーを構築して、ネットワーク設定をする。RDBを構築することによって、実験者はRDBに対して実験結果を投入する。ただ、データベースを使い慣れていないと、データを投入するのやりづらいし、見る側も見づらいので、コンテンツを作成して見やすくしました。実験者はコンテンツから実験データを投入できる、実験関係者はコンテンツで実験結果を閲覧できる、このようなシステムをつくったわけです。

サーバーのネットワーク化には情報のセキュリティの強化が不可欠です。特にうちの研究所は人の遺伝子を扱っているのです。知的財産の保守にもかかわって、セキュリティの強化はきっちりやらなければいけないという大きな課題もありました。

RDBの構築は、サンプルの選択から、実験工

程A、B、C、品質検査、受渡しとあります。まずサンプルの選択では、実験番号の割当用のテーブルを1つつくります。そして実験工程A、B、Cではそれぞれ結果を入れるためのテーブルをつくり、そして品質検査と、各工程ごとにテーブルを作成しました。したがって、5つのテーブルを作成したことになります。

先ほどから言っていますRDBとはどういうものか、簡単に説明しますと、1件のデータを複数の項目（フィールド）の集合として表現し、データの集合をテーブルと呼ばれる表で表す方式である。その中でID番号や名前などのキーとなるデータを利用して、データの結合や抽出を容易に行なうことができる。これは中小規模のデータベースでは最も一般的な方法です。

これがどこで設計部品とリポジトリと関係するのかとといいますと、「ID番号や名前などのキーとなるデータを利用する」というのは、工程間で継承する設計部品があるということ。「データの結合や抽出を容易に行なうことができる」というのは、たとえば工程・の場合はここにその結果を配置しようといったリポジトリの概念をここで使っていることになります。

実際にどのような仕組みになるか、リポジトリを利用した情報システムの応用例を見ていただきますと、最初の工程の実験番号割当テーブルでは、本日はこの3つを実験しようと仮定したとします。いままではそれぞれの実験者はエクセルで管理されたファイルのコピーの紙を渡されて実験をやっていたけれども、RDBを構築したことによって、これで登録をすると、自動的に実験工程Aのテーブルにその実験ナンバーが入る。きょうは誰がそれを担当するかなど、付加情報も全部入る。しかも、1回登録することによって、最後の品質検査のテーブルまで、実験番号やその他の情報が全部自動的に登録される。これが工程間継承になっているわけです。

（資料8）実験割当テーブルから、実験工程が終わって、うまくいったものにはチェックのところに「1」というフラッグを立てます。最後の品質検査テーブルまで、同じようにやっていきます。実験を統括する人は、どこまで実験が進んでいるか確認するために、実験工程テーブル、品質検査テーブルのチェックの部分だけを取り出して、下

のような表にします。

たとえば1のサンプル0225aaaは、stage 1もstage 2もstage 3も1ですが、Q Chは0になっていますので、実験工程A、実験工程B、実験工程CまではチェックOKだけれども、クオリティーチェックのところで失敗しましたというのがわかります。ただし、これではどこで失敗したか、よくわからない。このような場合は、しょうがないのでもう1回最初から実験をやり直すことになります。

サンプル2の0213aaaについては、品質チェックまでOKで、Transも1ですから、このサンプルを使うところにすでに受渡が終わりましたということを示しています。統括する人はこれを見て、では次の実験を追加していこうという判断をすることになるわけです。

このような表は、慣れている人にはわかりますけれども、万人にわかりやすいわけではない。そこで、経営工学の観点から、「見える化」で、操作や表示を簡素化しようということで、コンテンツを作成しました。

最後にこのシステムの評価ですけれども、1実験当たりの実験結果のデータ作成にかかる平均時間と、1か月当たりの実験結果の平均入力ミスの回数、この2つを用いて、システムを導入したこ

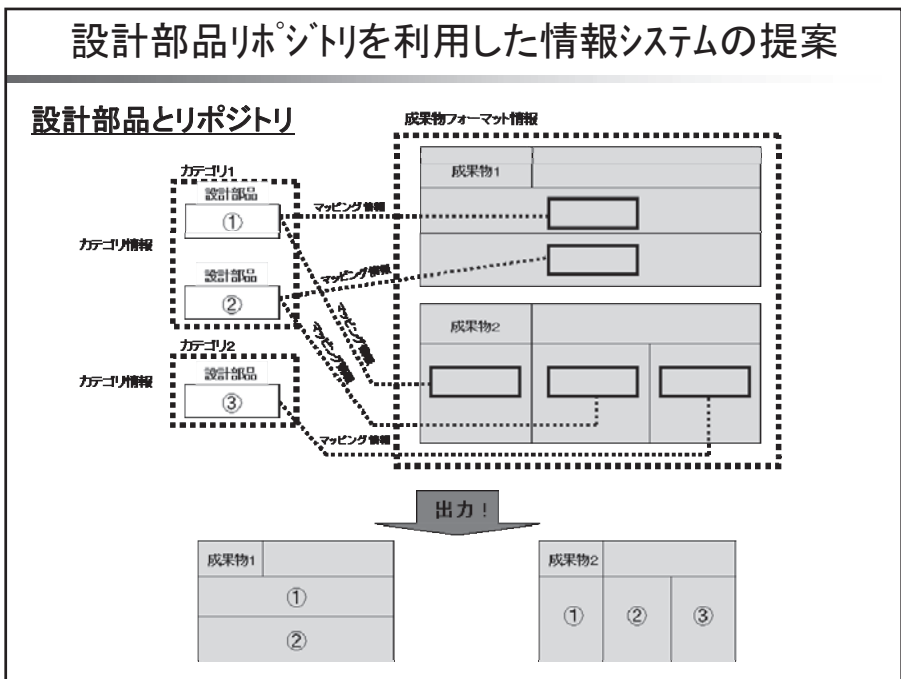
とによってどのくらい精度がよくなったか確認しました。

1実験当たり実験結果のデータ作成にかかる平均時間は、いままでエクセルで管理していたときは、5人の平均値で17.5分かかっていた。システム導入後は5分ぐらいで入力できるようになった。1か月当たりの実験結果の平均入力ミス回数も、導入前は1人9.7回だったのが、1.3回になった。本来ミスはゼロにしなければいけないのですが、どんなシステムを使ったとしても、やはりヒューマンエラーは起こってしまう。そのミスが1か月に1回程度なら、どこで間違ったか見直すことが容易になったということで、この数値はよしとすることにしました。このように、システムの使いやすさ、実験結果の精度の向上が認められたという結果になっています。

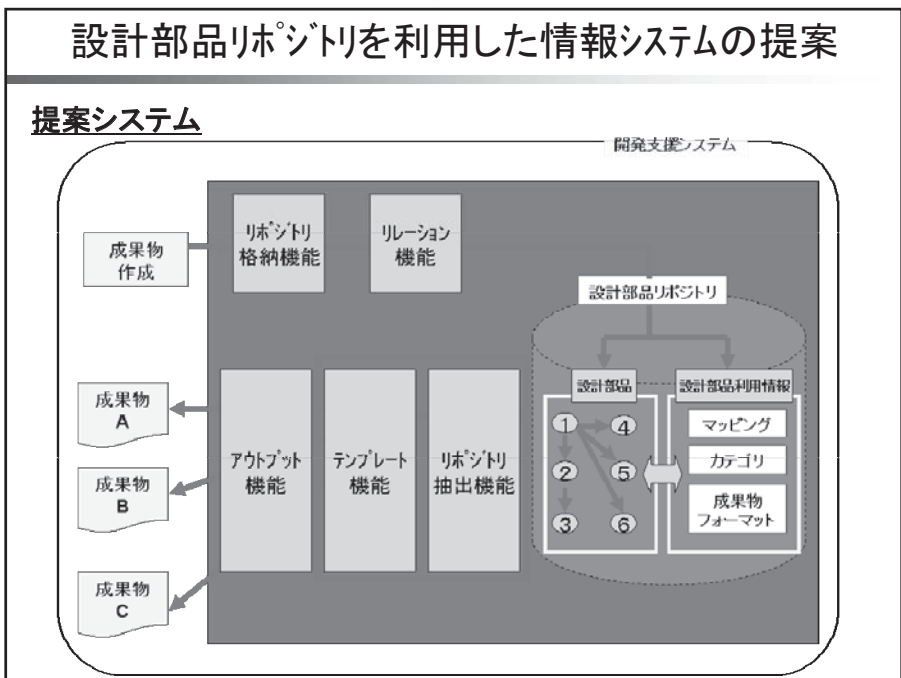
最後にまとめますと、設計部品リポジトリを利用した情報システムの提案を行ない、設計部品リポジトリを利用した情報システムの応用例を挙げ、その導入前と導入後の評価を行ない、生産性、品質、それぞれの向上が認められたという例を示しました。

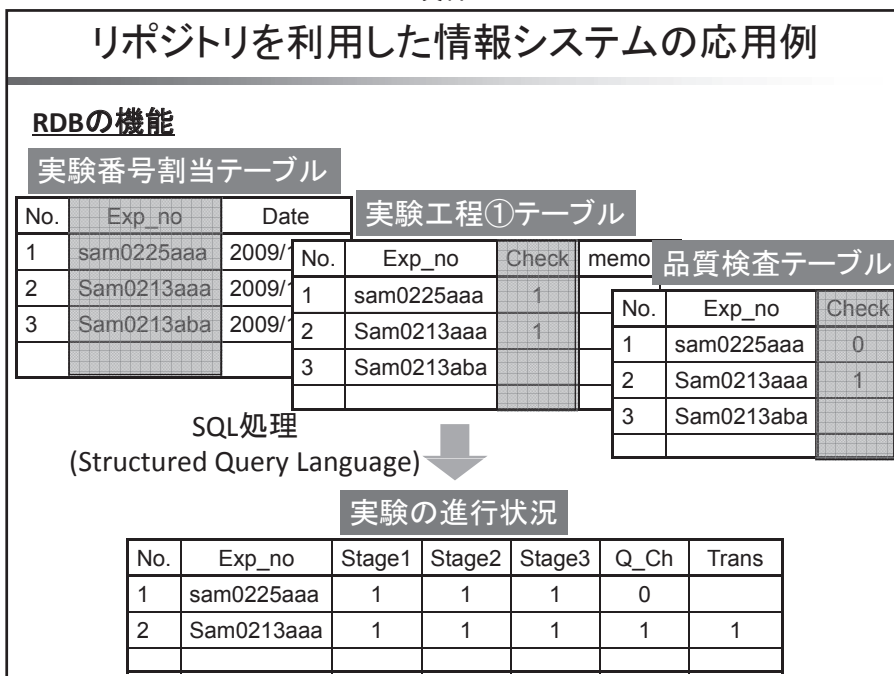
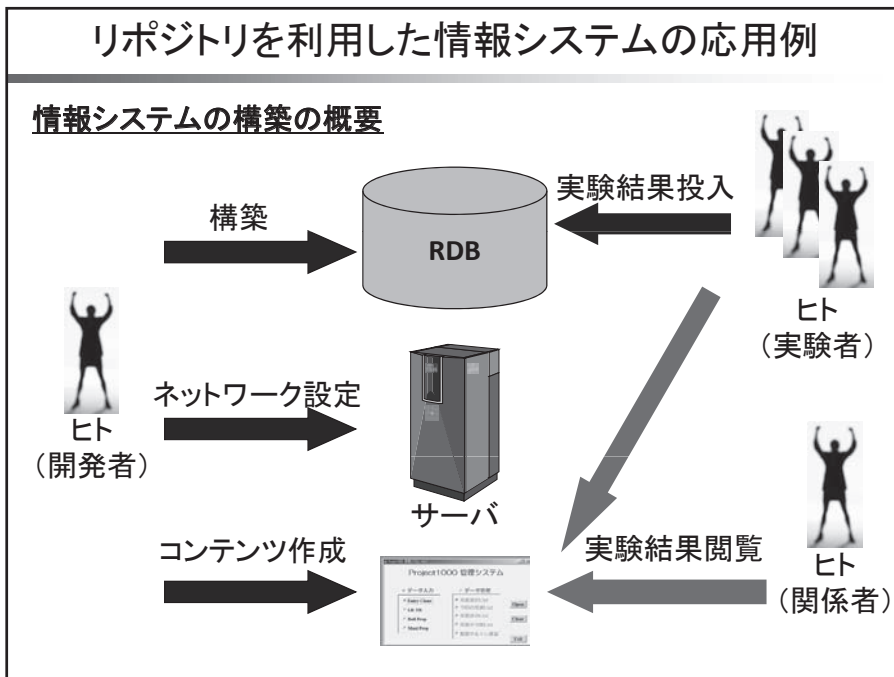
以上で報告を終わります。ありがとうございました。

設計部品リポジトリを利用した情報システムの提案



設計部品リポジトリを利用した情報システムの提案





「中国オフショア開発を進めていくうえでの 開発標準化の役割と課題」

大場 允晶

去年の3月、中国の北京と済南において、ヒアリングによって調べた内容についてご報告したいと思います。

中国におけるソフトウェア産業は1984年から萌芽期を迎え、ちょうどそれは1984年に中国ソフトウェア産業協会が設立されたのを機としております。1993年から国家級の電子情報応用工程を動かし始め、90年代半ばに外資系企業の製品の現地化あるいはセット産業の請負などが徐々に設立されて、2000年からかなり大きな発展期を迎えています。

20世紀の末時点、中国政府はサービス産業の対外解放に慎重だったのですけれども、WTOの加盟が2001年、そして2000年のちょっと前にインドでソフトウェア開発ビジネスが大きく確立してきたことを踏まえて、中国政府もソフトウェア開発とアウトソーシングの発展を重視して進んできました。2000年以降、ソフトウェア産業の設立は急激に増えてきたという状況でございます。

中国国家級のソフトウェア・パークは生産拠点の産業基地と輸出拠点の基地と2つありまして、両方重複しているのが9カ所、産業基地のみと輸出基地が2カ所で、中国の湾岸沿いに多くあることから、「オフショア開発」という言葉はここから出てきているのではないかと思います。

産業の規模は、2008年で企業数約2万軒、著作権で4万件、従業員数約3万人です。現有シェアは、アメリカが37%、EUが27%。それに対して日本、中国は同等で3位を占めています。

売上の6割は日本系で、全体の規模は小さいけれども、発展速度は早いという状況です。売上の内訳は、インテグレーション関係が近年伸びていますけれども、まだ製品とか技術開発に重点が置かれています。

ソフトウェア・サービス産業のアウトソース受託者の所有形態は半数が民営企業で、あと合資、外資の合弁というかたちで、受託企業の人員と規模は中小が多い。

ソフトウェア・情報サービスのアウトソース受

託企業の人員の学歴構成は、大半が大卒以上で、4分の3を占めています。産業としてはまだ新しいこともあって、経験年数は10年未満がほとんどです。

受託企業の顧客は、インドがヨーロッパと米国を中心にやっているのに対して、それに出後れた中国は日本を中心に発展してきているというような内訳でございます。

整理しますと、受託企業のビジネスモデルの類型としては、多国籍企業が設立したソフトウェア開発組織、ソフトウェアと製品開発が中心のところ、それから多国籍企業が設立した市場解放型組織、それからアウトソーシング大企業などがあります。

近年、オフショア開発というのがかなり話題になっているのですけれども、その背景は、日本での高止まりした人件費。それから、もともとソフトウェア産業は労働集約型の産業で、汚いとか3Kとまでは言わないけれども、大変厳しい労働環境で、それに対する嫌悪から日本での労働者が不足してきています。日本におけるプログラミング技術が2000年のちょっと前ぐらいから第4世代の言語に移ってきていますけれども、銀行などのソフトはまだCOBOLなどの3世代技術者が圧倒的に多いので、残念ながら人が不足して、それを求めているというケースも多いようです。それからITパークを中国政府が設置していることもあって、これらを背景にソフトウェア産業のアウトソーシングの進展があったのではないかと考えています。

オフショア開発とは、「システムインテグレータがシステム開発や運用管理などを海外の事業所や海外子会社に委託すること」と定義とされます。主な受注先はインドや中国の企業ですけれども、日本や欧米の企業が現地に進出して、自分の国の案件を委託するというケースも増えております。

オフショア開発の最大のメリットは、安価な労働力を大量に得られるというコストメリットです。逆にデメリットとしては、現地採用のスタッフに十分な技術が身につけていない、さらに、言葉や習慣の違いからくるコミュニケーション不足が原因で発生する納期や品質に関するトラブルが増えていることも報告されています。

ソフトウェア産業の特徴は、情報システムの生産は一般にトップダウンのウォーターフォールアプローチで行なわれていますけれども、詳細設計から製作・コーディングの部分がオフショア開発の多い範囲と見ております。

情報システムは受注生産形態をとっているのが普通で、その設計思想は自動車やプロセス産業財のように、インテグラ（すり合わせ）型のアーキテクチャーであり、パソコンや自転車のようなモジュール（組み合わせ）型と違って、数多くの部品（プログラム）を1つ1つその製品専用に最適設計することではじめて、まともな製品が出来上がる。それを最後、テスト工程ですり合わせて製品として動かせる。そういう設計思想からできている製品であることから、全体を外部に任せ切るとするのは難しく、詳細設計と製作・コーディングが中心になっていると考えられます。

中国ソフトウェア産業調査の概要ですが、2009年3月9日から13日まで、北京、済南でソフトウェア開発企業6社、日系商社1社を訪問して、日本企業の中国へのアウトソーシングの状況と中国の流通業について、情報収集と意見交換を行ないました。北京では現地のソフトウェア開発企業3社を中心に情報収集と意見交換を行ない、済南では日大と提携関係にある山東大学を訪問して、現地企業へのアンケートに関する打ち合わせをしました。また共同でソフトウェア・パークに行き、さまざまなアンケート及びヒアリングを行なった次第です。また、本発表では、インターネットによる調査企業1社を加えました。

ヒアリングの内容は、沿革と企業概要、人員構成と人材開発、強み、コミュニケーション問題、世界的不況の影響、その他としてIPO、BPOの進展方向、生産性、ネットワークの利用状況などを中心に聞いてまいりました。

時間がないので簡単にしか言えませんけれども、沿革・企業概要については、ほとんどが2000年以降の設立で、日本向けIPOが主体でした。

人員構成の特徴では、大学と連携した業務実習があって、企業が大学に講座を設けて、大学を卒業する前にITの技術講習や日本語教育をして、その中から採用する。その結果、低離職率を実現しているようです。

中国企業の強みは、一言で言えば日本語への対

応ですが、これは企業個別の強みで、たとえばトランスコスモスは日本に親会社があるため、日本の品質が確保できる体制ができています。INSIGMAはCOBOL要員が470名と豊富で、日本のレガシーシステムを新システムへ入れ替えをするコンバージョン作業に強いという特徴を持っています。TREは、ラボ契約によって、顧客直結の開発部隊として一定期間の契約で技術者を常時確保することが可能になっています。

コミュニケーション問題では、日中文化の違いや日本式業務ルールの採用に伴って、多くの問題が起きています。特に設計書の行間の解釈のズレとかスケジュール管理の認識の違いも大きい。テレビ会議をやっても、あるいは実際にフェース・ツー・フェースのコミュニケーションをやっても、十分ではないという傾向にあります。

世界的不況下の影響、その他ですが、行ったのが2009年3月ということもあって、不況の影響で午前中は閉じているとか、急に解雇したとか、成長計画がかなり狂った企業も多く見られました。その中でトランスコスモスは日本の大手企業からの受注が増加していて、開発標準化技術の向上で、売上も利益率も増加しています。

世界的不況によって新規案件が減ってきたことから厳しい状況にあつて、伸びているところもあるけれども、下がってきているところもあって、多極化しています。その大きな原因は、開発標準化技術をどのくらい持っているか、それに対する人員の確保ができていのかどうか。中国オフショア開発のメニューの中では、組み込み系とかウェブとか3Dとか、システム構築と業務サポートなどを体系化している会社もごまいます。単純なアウトソーシングよりも、もっと上流工程を進めるために、メニューを充実させて見せていく例も増えています。

これは強みとして出てきたことですが、日本に親会社があるため、日本と強いコネクションを持っているとか、日本語のできる、日本の開発プロセスがわかる技術者が多数いるとか、日本の評価確保ができる、ISO9000の認証を取得している、CMMI5を全社規模で達成しているなどで高い生産性を出していますよということを宣伝しているところもございました。

品質管理の取り組みの例としては、プロジェクト

ト管理、レビュー管理、標準化管理など、システム構築プロジェクトの実施をする中で標準化を取り入れて、ITソリューションサービスを提供していく、納期をきちっと守る体制をつくっているということで、お客さまの満足度を上げている企業もごぞいます。

まとめますと、中国オフショア開発企業は2000年以降に設立されて、単価の安さと質・量ともに豊富な人材によって急速に伸びてきています。

中国IT企業は大学と連携した事前の実習教育と日本語教育に力を注ぐ新卒採用によって、優秀な学生のみを選別育成するという体制をつくり込んでいます。

日本語のOSとか日本語で書く日報の採用などで日本企業式業務ルールでのコミュニケーションに努めていますが、実際には日中文化の違いによって、コミュニケーション問題が起きています。

サブプライムショックの世界的大不況によっ

て、新規開発の下請は非常に減って、レガシーシステムの修正による保守業務が増えています。標準化の後れなどによって、保守業務を受けられる作業者はまだまだ少ないという状況です。

中国オフショア企業のうち、開発標準化の進んでいる企業では、サブプライムショック以降でもリモートジョブメンテナンスなどの業務増が起こっていますけれども、単なる日本語日報対応レベルの企業は開発案件減の影響が大きいと思われるます。

中国IT企業は大変優秀ですけれども、インテグラ（すり合わせ）型アーキテクチャである情報作業では、個々のプログラムを動かす動作環境の統一と本番環境の中で、要求業務仕様通りに稼働可能なシステムテスト以降の業務コミュニケーションのノウハウを身につけることが不可欠で、これが確立しない限り、実際にはなかなか進んでこないだろうということがわかりました。

以上で発表を終わりたいと思います。