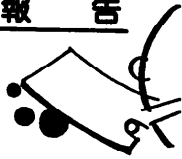


報 告



パネリスト

パネル討論会

利用者・開発者それぞれの世界と役割†

水野 睦郎¹⁾、 金井 一成²⁾、 橋本 茂司³⁾
 山本 毅雄⁴⁾、 司会 松谷 泰行⁵⁾

1. ここで論じたいことは

司会(松谷) ただいまからパ
 ネルディスカッションを行
 います。

このパネルディスカッション
 は、利用者と開発者というもの
 の役割とか、利用者はこうしてもらいたい、あるいは
 開発者はこうしてもらいたいということについて、
 お互い本音を話し合いたいというのが一番のねらい
 です。

たとえば、利用者と開発者は一体にならなきゃなら
 ないということは、だれもが言いますし、いろんな本
 とか論文にも書かれております。しかし私の聞きかぎ
 りでも、エンド・ユーザからみると、一体とか協力と
 か言っても、何していいのかわからないなんていう話
 も出てきますし、開発者はちっともエンド・ユーザの
 ことを分かってくれないという話もございます。

片一方で、開発者の方からは、これまたいろいろ言
 い分があるようでして、利用者というのはい言いたい放
 題を言って、そのとおりにやっておったらどうなるか
 分からんということも言われております。私はどれも
 本音の議論だろうと思います。そのあたりについて忌
 憚のない意見の交換をしていただきたいと思っております。

一口にシステムといいますが、構築していく形態
 にはいろんなタイプがございます。

ざっとみましても、第1に、一つの組織体の中にシ
 ステム部とかコンピュータ部門とか呼ばれているところ
 があり、そこが中心になってシステムを開発してゆ
 く形態です。

2番目の形態としては、コンピュータ部門なんかな
 いところで、いきなりどこか、メーカとかソフトハウ
 スといった外部の方に委託しながらやっていくという
 形態があります。

3番目は、たとえばデータベースサービスのように
 利用者が不特定多数になり、開発者はそれに対して
 一つのサービスを提供する、こういう形態のやり方
 です。

それぞれの形態において、ニュアンスの違いはある
 と思いますし、この三つの形態の全部について、ディ
 スカッションできるかどうか分かりません。しかし、
 できるだけ議論を尽くしたいと思います。

それでは、パネラの方をご紹介します。

まず、水野先生は、水野調剤薬局の経営者でござい
 ます。水野先生は、第5回情報システム研究会で、エン
 ド・ユーザの立場からの論文を発表していただいで
 おります。そういう意味で、水野先生には、エンド・
 ユーザの立場を代表していただきたいと思っております。

橋本先生と金井先生には、開発者の立場を代弁して
 いただきます。

橋本先生は、現在、東レシステムセンターというこ
 とで独立しておりますけれども、ついでの前までは東
 レのコンピュータ部門でした。ですから、第1の開発
 者の立場かと思えます。

金井先生はメーカということで、比較的の第2の開発
 者の立場かと思われまます。

最後に、図書館情報大学の山本先生には、ご自由
 にいろんな角度から意見を言っていたきたいと思っ
 ております。では、最初に水野先生からお願いいたし
 ます。

2. エンド・ユーザから見ると

水野 水野でございます。私は薬剤師でございま

†日時 昭和61年12月5日(金)、18:30~20:30

場所 筑波研修センター

1) 水野調剤薬局、2) 日電、3) 東レシステムセンター

4) 図書館情報大学、5) 新日鉄情報通信システム(株)

て、10年前まではほとんどコンピュータに触ったこともありませんでした。それが薬のシステムをつくってみようということで、浦先生が学校の先輩ということもあり、お話を伺ったりして、一つのシステムをつくってみました。調剤というのは、ご承知のとおり、処方箋によって薬をつくって患者さんに渡すわけですが、それに関するシステムをつくり使っております。



昨年、開発に関して利用者の立場で感じたことを発表してみろということで、その開発での経験を発表させていただきました。その話ときょうの話は多少重なると思いますので、まとめてみますと、まず、利用者というのはコンピュータをあまり知らない。開発者の方は利用者の専門職業を知らない。初めて会ったときに、まだ両方の話が通じない間に、基礎的なことを話してしまわなきゃならない。これは大変致命的なことなので、ユーザの方もコンピュータ側も、ぜひ素人言葉レイマン・ランゲージで話をしましょう。素人言葉じゃないと、とてもコミュニケーションがうまくいかないんじゃないか、ということが一つです。

次に、専門職業の仕事というのは何によらず、歴史とか伝統とか、学問、技術、技、経験など隠された部分がある。たとえば、毒薬の取り扱いなどというのが500年近い歴史をもって、ああでもないこうでもないやって、やっと薬剤師が毒薬の取り扱いがうまくなった。また19世紀ぐらいから麻薬が出てきた。今世紀には抗生物質や、精神活性の薬が出てきた。これは30年も40年も前の話なんですけど、社会の中でこういうものを扱うとなると、数百年ぐらいの歴史がないと、うまく扱えないというようなことがあります。物事の処理に適切な社会通念ができてくるまでにかなり時間がかかるし、なかなか外側から見えない。そういった仕事の裏側にある背景を理解してからシステムをおつくりいただきたい。と同時に、そういう専門業務が入ってきますと、技術とかノウハウの進歩が早うございますから、メンテナンスの簡単なものじゃないと、とてもシステムとして使っていけないということです。

3. 相談相手は？

水野 システムづくりの一番初めに相談をするのは、ハードウェア会社のセールスの方が多いわけです

けれども、ハードさえ売ればいいという姿勢が強過ぎる。コンピュータの1社1元方式は薬の世界からみれば全く不思議なことであります。武田薬品の薬しか使わないということはないのでありまして、武田だろうが、第一だろうが、それらの薬を混ぜ合わせるのが薬の世界です。そういう仕事をやっている人間からみますと、富士通さんは富士通さん、日立さんは日立さんの機械しか使わないというのは全く不思議な話であります。

一方ソフトウェア会社の方に、ソフトウェアについて相談してシステムを考えればいいのかというと、利用者の側から適当なソフトウェア会社が探せないという問題があるわけです。ソフトウェア会社の自己主張が不足しているというところもありますし、注文ならつくってあげましょうという姿勢は、むしろソフトウェア資産の集積が足りないのではないかと、ある程度は出来合いで、ある程度は注文というところが、われわれとしてはお金の面からもいいところなんじゃないかと思えます。

4. コンピュータは道具でなく仲間

水野 このパネルに全くの素人は私しかないんだということを開きまして、震えておるわけなんですけど、それなりに心臓強く、われわれのような程度の高くない利用者にとって、コンピュータというのは何なんだということからお話をしたいと思えます。

一般の社会通念からすると、コンピュータというのはミラクルマシンであります。なんでもやってくれる、大変に高価である、非常にハイテクな製品である、絶対に理解できない、そんな意味の社会通念があるわけであります。そのマシンを企業に導入することとはどういうことか。導入という言葉の中に、さあ、そこでいよいよわれわれの企業も近代化したとか、ステータスが高くなったとか、企業もそんな時期になったとか、それだけの規模になったとか、省力化とか、この辺が現実なんでありましょう。しかし、経営者の中には、タイプライタや、ワープロ、電卓を買う、作業機を買うつもりで買う、コンピュータにそんな見方もあります。

そういうことからすると、経営的な見地からすると、コンピュータを入れるということは、人が減って仕事がうまく動くようになれば良いわけです。経営者の方はコンピュータを導入したという感覚でありましようが、実際にコンピュータが導入される現場で、

働いている人たちは、コンピュータが仲間に入ってきたという感じになっているんじゃないかと思うんです。つまり、道具としてより、むしろ仲間、友達と考えた方がいいんじゃないかと思うぐらい、仲良くなる人もいれば、反発する人もいるわけです。

つまり、コンピュータをその企業とか組織が雇った生物とみたほうがいいんじゃないか。日本の場合には、外国人にあまりなじみがないのでありますけれども、むしろ異星人が入ってきた。火星人をマーチャン(Martian)と称すれば、マーチャン人種が入ってきたというか、マーチャン生物が企業の中に入ってきたと考えたほうが自然です。

そこで人間とコンピュータの比較をしてみたわけがあります。つまり中枢として、コンピュータは非常に計算とか記憶領域が大であります。人間のほうは連合域など、現在では意味不明の中枢があるわけです。コンピュータは末梢神経に関しましては未発達。感覚器は、情報を受け取る力が非常に弱い。人間の場合には、目であるとか耳であるとか、受け取る情報量は非常に多く、その中から取捨選択して記憶をしたりしている。情報の捨て方が非常にうまい。ところが、コンピュータの場合にはそうもいかない。人間だったら2〜3日絶食してもなんということはないんですが、コンピュータは一瞬の停電で全く使い物にならなくなることがあります。循環器、生殖器など、部品については、いろいろ交換できます。つまり同種の場合にはいいのでありますけれども、移植性がないこともある。また、体形がだんだん小型化しているなどコンピュータがもし生物とすれば、こんな生物的特徴があります。

生物的でない特徴としては、欲がない。生存とか権力欲とか性欲とか金銭欲とか、こういうものが全くないわけでありまして。教育も全くの個別教育で、学校教育じゃない。また、進化のスピードが非常に速い。人間は遅いわけでありまして。

コンピュータの導入とは、こういう生物をその企業が雇うことになるんだと。人間でこんな変わった性格をもっている人がいたら、どこの会社でも採らない、そういう特別な生物が隣に来てしまった、そういう生物と一緒に仕事をしているというのが導入後の利用者の認識だというふうにお考えいただいたらいいんじゃないかということです。

さて、普通働いている人とコンピュータとのコミュニケーションでありますけれども、SEとかプログラ

マとかオペレータとか、昔は必ず通訳である特別の職業人がいたわけです。

私が30年前に車を買おうと思いましたが、「運転手はつかなくてもいいですか」と聞かれたことがあるんです。今は、それをソベツとかロールスロイスを買う場合しかそういうことは聞かれないうと思うんですが……。コンピュータとの間には運転手のような通訳が必要であると今でも考えられていますが、実際には同僚として直接にコミュニケーションをとらなきゃいけないのであります。

経営者レベルでは、うちでもコンピュータを使い始めましたということですけど、一緒に働く立場になりますと、どうしても擬人化、生物化せざるをえなくなる。それが一面では生物と非常に違った性格をもち、反面、生物らしいことをやる。ことに最近では専門家はAIなどで人間らしく使おうというような努力をされているわけで、よけいに難しくなっている。単民族社会から多民族社会への変化であります。さらに最近、今までの手続き的な言語から、だんだん宣言的な言語に変わって、分かりやすくなってきた。通訳なしに、自分たちでのコミュニケーションが少しずつとれてきているんじゃないか。つまりそういった生物との協調作業というか、特徴を認め合って、人間の欠点と機械の欠点とをカバーしながら、両方で仕事をする状態を創り出す。これが今後の課題でありましょう。

たとえば日本人の社会の中に米国人とかドイツ人とか入ってくれば、日本の人たちもドイツ語や英語を勉強するでしょうし、相手も日本語を勉強するだろうと思うんです。いくつかの言葉を使うマルチ・リンガル・ピープルというのでしょうか、利用者の立場でも多少ともコンピュータの言葉を理解しよう、コミュニケーションをつくろうということが必要なんじゃないかと考えているわけです。

5. 利用者の役割

水野 開発側はシステムをつくってしまえば、それで仕事は終わりになりますけれども、つくられたほうは、いつまでもそのシステムとつき合わざるをえない。ですから、利用者の立場で教育の問題が重要であると思っております。

その話はまた後にいたしまして、開発者側に関しまして、ざっくりばらんに言わせていただきますと一般の無理解を前提として、コンピュータというものはなんでもできます。なんでも任せなさいというような、む

しろコンピュータ信仰というようなところがあるように思います。とにかく「システムをつくって差しあげましょう」「じゃお願いします」ということになるわけですが、そういう形では、いざできたシステムがなかなかうまく動かないだろうと思います。

結局は、システムをつくっている間にしても、ユーザとコンピュータと開発をなさる方々の、コミュニケーションをどうとっていくかが一番大切なところなんだろうと私は思うんですが、むしろその辺は利用者側の努力、先ほど申しあげた教育の問題などがあるんじゃないかと思います。

一つの例としまして、ある会社のAとBの課が、コンピュータを入れたときに、Aの課のほうは、コンピュータを全くの道具と考えまして、今までやっていたことをみんなコンピュータ化しました。B課のほうは、むしろ反対に、自分たちの仕事をコンピュータ寄りに変えてからコンピュータ化しました。課の規模としては、A課のほうがB課の3倍ぐらい大きくて、仕事も複雑だったのでありますけれども、実際にできたプログラム、COBOL でつくりましたので、ステップ数で比較しますと、A課は初めにオンライン処理が1万5,000ぐらいでできるはずだったのが、5万5,000ステップとかなり大規模なプログラムになってしまった。それに反しましてB課のほうは、コンピュータに都合のいいような仕事にすることとしたら、ほとんど初めに考えた1万5,000ステップと同じ大きさで、オンラインのプログラムが当初の計画から多少ふえたくらいであります。また、記憶の容量に関しまして、A課は最後には大幅にふえているわけですが、B課のほうはそんなにふえていない。

現在、A課のオンラインシステムは、あんまり大き過ぎて、あるプログラムを動かすと、負荷がかかって、ほかのプログラムがとまってしまい、大変評判が悪いというようなことでもあるわけがあります。ちょっとした教育の差で、うまくシステムができる場合とそうじゃない場合があるんだなということで、びっくりした次第です。その一つの例であります。

6. 医療の例に学ぶ

水野 システムの開発を医療に当てはめて、患者をユーザとして、医師を開発者としてみると、非常に類似性があります。

ここに示したのは聖路加看護大学学長の日野原重明先生の医師の薬物治療についての考え方です。たとえ

ば「患者は医師と薬剤師の指示を誤って聞く」つまり、利用者は開発者の言うことを誤って聞くという意味にとってください。「指示を守らない」。これはコンピュータがちゃんと動いているのか、間違っていて動いているのか区別がつかない、そういうふうと考えてみますと、大変システム開発の開発側と利用者側に似ています。

また、医師のほうに反省がありまして、患者に薬のみを与えて情報を与えないのは無責任な医療だと。これは、コミュニケーションをするやり方を与えないで、ただオペレーションだけをやらせようとしても無理じゃないかと。それから、患者さんから情報を得ようとしなさい、そのように患者を教育しない。これもやはり開発者と利用者のこととしておもしろいほどにあてはまります。

それから、開発者（医者）はシステム（薬）をつくったことだけを記録して、患者が飲んだ量、つまり、実際に、本当にどういうふうに使っているかということ記録しないというのはサイエンスではない。医療でも、実は一部でこういう反省が行われているわけがあります。

医療の中には昔からの問題がいろいろあります。たとえば、現在、いろいろな薬の副作用が問題になっておりますが、これはクロニジンという薬の副作用について、日本の会社が出している説明書であります。説明書で見ますと、「副作用、精神神経系とか循環器」つまり「ときに不安があり、めまいがあり、またまれに幻覚があり」、こういうことが書いてあります。ここには副作用が一つ残らず書いてありますが、実は書いてあるだけで、お医者さん側からはあまり使えないのであります。

どうして使えないかといいますと、一つの例をあげますと、USP・DI（米国薬局方・医薬品情報）というアメリカの国家的規格があります。これの患者アドバイスという分冊に「たくさん飲み過ぎたときはどうなりますよ」というのが一番上に書いてあります。それから、「こういう症状が起きたら、すぐお医者さんに行きなさい」と書いてあります。「こういう症状が起きたら、なるべく早くお医者さんに行きなさい」「こういう症状が起きたら、その次に診察を受けるときにお医者さんに話をしなさい」これだと非常に分かりやすいことはいまでもありません。

この医療従事者用の分冊には、すぐに治療を行わなきゃならない副作用はどういうものである、それは

何によって起こるんだという道筋が多少書いてあります。同じ副作用の情報でも、お医者さんにとって大変使いやすいデータベースだろうと思います。このように、何が起こりますよということだけでは始まらないわけでありまして、現在、われわれ薬剤師がつくる情報なども、数が多いものを一番初めに置いておく、数が少なけりゃ下へいく。しかし、数が少なくても大切な、生き死にに関係ある副作用というのが大切でありますから、データベースの中にプライオリティをつけないきゃいけないという反省があるわけです。

7. 利用者から見た課題

水野 利用者から見ると、今までの人間と機械という関係が、むしろ人間と仲間になってるんじゃないか。そういうことからすると、利用者として、むしろそういうことを自覚すると同時に、自分がそういう生物たちと話をする、コミュニケーションをする方法、言葉などについて、自分で教育を受けよう、そういうコミュニケーションの重要性を認識しましょうということなのだと思います。先ほど申しあげましたように、でき上がったシステムに、利用者側の技術的な内容が入りますと、非常に変わりやすい。それを開発者に全部おんぶして直してもらおうということは、とうてい不可能でありますから、自分でも貢献せざるをえないだろう。それでなきゃ、使えるようなシステムにならないんじゃないか。

そのソフトウェア、ことに医療などの場合には、企業組織体はそんなに大きくありません。どうしても標準化というか、みんなで共通して使えるような、つまり社会的なシステムというのができてこない、難しい問題（システムをつくり直す。データを修正していく）が残されるわけです。そういったデータの修正を、個人レベル、一つの企業、組織レベルで考えたら、これからの進歩にとても追いつかなくなるんじゃないか。むしろ社会がよいソフトウェア資産をもっており、多少のお金を払う必要はありましようけれども、みんなで使い合うということが大切なんじゃないかと思います。われわれ、医療に従事しておりますので、専門技術を含んだシステムについての問題点、そんなことできょうの責めを果たさせていただきたいと思います。どうもありがとうございました。

8. 日常の態度がシステムに現れる

橋本 東レシステムセンターの橋本です。

初めに、このシンポジウムを通じまして、情報システムにつきまして、いくつか考えることがありましたので、それをご紹介したいと思います。

私、昨晚ここに到着し、図書館情報大学までタクシーに乗ったわけです。途中の飲食店街が、昔来たときと比べてずいぶんきらびやかになったので運転手さんに「この辺は安全なんですか」と尋ねたところ、「ほとんど大丈夫だ」という答が返ってきました。

「ほとんど」というのはなんとなく大丈夫だろうと考える人もいます。ちょっと不安だから、もう少し情報を収集して、「ほとんど」のぐあいをもう少しと数量的に調べようという人もおります。中には人から聞いただけじゃ安心できないから、自らいろいろ行ってみようという行動的な方もおられます。

一方、たまたま連れがいた場合、連れというのはお客様だとします。システムで言う場合は利用者に該当すると思うんですが、そのときに、先ほどの第1のタイプですと、大体大丈夫だ、適当なところに入ってしまおうと。後でしまったということになるケースがあるということです。

そういう方がいるかと思いますが、お客様の意向は全く考えずに、自分が好きそうな店に行く。安心なところに行くタイプもありましようし、冒険好きな人は、お客様はそういうのは好きじゃないのに、冒険してしまう。

また、連れの気持ちを察して、フィットしたところに連れていく。もしそういうところを自分が知らなかったら、フィットするようなところを探して連れていく、こういうタイプもあります。

こういうふうには人間にはいろんなタイプがあるわけですが、システム開発に関しても全く同じでございまして、人間のタイプそのものがシステムに表現されてくる、こういうようなことを昨晚は考えました。

きょうは、3時ごろ参加者名簿をお配りしたわけです。今まで何回かシンポジウムに参加して、参加申し込みをした人の名簿をいただくことはあるんですけども、本当に来た人の名簿をいただくことはほとんどないですね。後から最終的なものがくることもありますけれども、そうしますと、あまり意味がない。

今回はとにかく参加者名簿をなるべく早くつくって、少なくとも夜の懇親会を楽しくやりましようとい

うことから、お配りしたわけですが、情報を提供する側はタイムリに提供するサービス精神が大事だと思います。

ただ、こういう学会の会合に限らず、一般の情報システムをつくる場合でも、お客さまに喜んでもらうというのが最も重要なんですけども、やはり経済性の原則がありまして、ある予算の範囲内でなるべく最適なものをつくらなくちゃいかん。これが開発側にとって大変なわけです。

お客さまに喜んでもらうという考え方をしたシステム開発でないと、やがてお客さん自身が必然的に離れていってしまうという経験を、ずいぶん私どもはもっております。

ついでに名簿の話なんですけれども、実は、皆さんにお配りした名簿は工夫しました。午前中に打ちまして、皆さんに修正を頼んだのは、1ページに十数名しか打っていません。かなり大きな字で、しかも間隔をあけておきました。これは修正しやすいようにとのことです。最終的には1枚におさめました。ワープロの技術が進んでいるからできた芸当です。システム開発においても、進んだ技術は率先して使うというのが非常に大切だと感じます。

もう一つ、この名簿はずいぶん汚いですね。きょう使わせていただいた複写機は、濃淡が1から9までセットできるんですけども、一番薄いほうでやった。それでこの濃さです。一番濃いレベルとほとんど差がありませんでした。1から9まで、せっかくいいレベルで機能を備えていながら、やっていることは同じにしか見えません。

システム開発の立場から見ると、開発側はずいぶんいろんなことをやっているつもりだけど、利用者はちっとも感謝してくれない。利用者側が「この程度しかやってくれていないのか」と思っているからです。

こういうことはよくあり得ることでして、この複写機につきましても、利用者がこれで満足していたら、いつまでもこのレベルなんです。メーカーに注文をつけないと良くならない。利用者は常にいいものを欲しがるといふ感覚、開発側は利用者にいいものを提供するという感覚がないと改善されない。当たり前のことですが、こういうことは日常生活の中でもかなりあるんじゃないかという気がしています。そして日常生活態度がシステム開発に最も端的に表れるんじゃないかということでございます。

9. 利用者にとって最大のメリットは

橋本 今まで私どもは東レのシステムをすでに30年近く開発してきたわけですけれども、その中でずいぶんいろいろな経験をしました。今どんな状況になっているかといいますと、システムのメンテナンスを最小にするのが、結局、利用者にとって最大のメリットであるということ。もう一つは、利用者が手軽に気楽に自分でやれる環境を整える。この2点がやはり基本じゃないかと考えております。

このために、過去の経験を踏まえまして、システムをかなり柔軟にしております。

たとえば会社では組織変更なんて日常茶飯事でございます。会社が分離するとか、二つの会社と一緒にすることもあります。つくるべきシステムにそういうことをある程度予測し、その機能を盛り込んで長いライフサイクルのシステムにすることなどは今では常識になっています。

ただあまりやり過ぎると問題でありまして、いろいろなことを全部予測して、これからこういうことも起こる、こういうことも起こるといふことまでやりますと、いつになってもシステムができ上がらない。これが恐らく一番利用者の不評を買うこととなります。最も起こりやすいパターンだけ考える、これは過去の経験とか、少しアイデアのある人だと、大体考えつくことだと思います。

別の例をあげますと、どこの会社でも原価計算をしますね。特に製造業ですと、いろんな工程を経ているいろんな製品ができ上がってきます。原価計算をやる人というのは、どちらかという、技術系出身よりも事務系出身、要するに経理とか会計に強い人がやるわけです。コンピュータにマッチした原価計算システムのつくり方なんていう本はあまりないものですから、とんでもないものをつくってしまうことがあります。

たとえば、現状の工程、工程間の物量移動のみを前提としてシステムをつくったことがありました。30年代から40年の初めは、東レもかなり景気が良かったものですから、現状の工程を単純に能力アップするとか、もう一つ工場をつくるとか、そういうようなことが多かったんですけども、だんだん不況になってきますと、物流の流れを変えるとか、あるいはある工程を止めてしまうと、工程が変化するようになってきたわけです。そうしますと、そういう工程の変化に自由自在に耐えられるシステムになっていませんでし

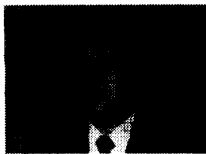
た。また、会計の本を読みますと、回収工程がある場合、計算方法が、何回か繰り返して、どこかで収束したらやめると書いてあるんです。化学会社ですと、かなり回収の工程が多いものですから初期値が悪いとループしちゃうことがある。実際は連立方程式で解けるわけですから、きちんとネットワークモデルとしてとらえて逆行列を求めてやれば、済んでしまう。こういうセンスは利用者側にはあまりありませんから、こういうふうにすればすっきりといくよという教育を開発側はしなくてははいけないと考えます。

勉強した人の中には、得意になって、これもできるあれもできると、いろんな機能をつけ加えてしまう。これはソフトウェアパッケージでしたらいいんですけども、一般の企業では、少なくとも経理計算をやる場合、あまり必要ありません。

このような悪い例が起こらないようにするためには、基本的にはシステム開発者がもう少し勉強する必要があるんじゃないか。勉強するというのは、いわゆるコンピュータ技術のことでなく、マトリックスとか、連立方程式とか、逆行列とか、こういうものを使ってモデルを整理するといった勉強のことで。利用者側もこうしたモデルを理解する必要があります。開発者と利用者が協力し合って、そういう仕組みを理解していく。そうしますと、非常にライフサイクルの長い柔軟なシステムが構築できるんじゃないかと考えています。

10. 開発の実態

金井 私は、主として日本電気の中で、病院とか医療の関係のシステム開発を担当しております。



きょうの前提は、開発者の立場から、「開発者の責任と利用者への注文」という題をいただいておりますので、その点で発言させていただきたいと思います。

お客さまの中に実際に現場で端末をたたいたり、あるいは帳簿を見たりする利用者がいて、さらにコンピュータ室、あるいはコンピュータ係という開発者がいらっしゃる。私の場合その開発者の方々をハード的、ソフト的に支援するという立場で、きょうお話をさせていただきます。

私の担当している病院フィールドでいきますと、専門的な教育を受けた開発者というのが一般的には非常

に少ない。むしろいろいろな事務系統の仕事をされていて、現場の作業に精通しているのが、一応コンピュータの担当になるというのが一般的でございます。そういうことを前提にしてお話させていただきます。

病院のシステムは運用のシステムでございますから、技術的にどうのこうのということよりも、コンピュータに携わる利用者の方々から、トップを含めて幅広くコンセンサスをとって進めていく必要があります。コンピュータ用語を素人の方々に翻訳していく人材の確保が必要ですができにくい。したがってたとえばシステムの根幹にかかかわるような仕様の変更というのが、実際に端末が全部設置されて、いよいよ最後の仕上げのテストをするというようなときに出てきたり…。これが建物なんかですと、建物が建っていく過程が目に見えるので建物がほぼできてから、5階建を6階建にしるとか、6階建を5階建にしるとかいう話はないわけですが、そういうような問題が出てくる。

メーカーのSEは、10年とか20年とかいうご経験がされているユーザのコンピュータ部門のSEに比べますと、やはり外から見てそのシステムに携わることですから、業務知識ということについては、どうしても乏しくなる。院内、あるいは企業内の真のシステム化ニーズの把握が乏しくなる。

したがって、メーカーサイドが外側から見て出した仕様書というのは、あくまでもたたき台としてとらえていただきたい。本当のお客さまのニーズと、メーカーが出した仕様にはギャップがどうしてもできてくる。このギャップが、最初のうちは書類上のチェックですから、それほどはっきり見えないものが、だんだんと工程が進みシステムが見えてくるに従って大きくなっていくというのが現状ではないかと思えます。

そういったギャップを埋めるため共同作業をする。最初の基本設計書、仕様条件、あるいは工程とか大きな機能とか、そういうものをお客さまとメーカーが共同で決めていく。そして、何回も定例的な打ち合わせをしながら、だんだんと基本設計、概要設計、詳細設計というような形で詰めていく。その過程で、最初はお客さまの方では、ハードとかOSとか、分かりにくいコンピュータ知識をだんだんと吸収してくる。逆にこちら側のほうは、お客さまの本当のニーズとか、組織とか、意思決定のルートとか、そういうものをだんだんと理解していく。通常、非常に長い時間かかってギャップをだんだんと埋めていく。そこには、基本的に

は信頼関係が作用すると思います。

まとめといたしましては、私がコンピュータのシステム開発に携わっている実感でございますけれども、たとえば建築というものを捉えますと、これでも大きなトラブルはしょっちゅうあるそうですが、少なくとも見やりとか設計図面というものが、発注主の方より分かるような形で図式化されている。建築には数千年の歴史がある。その中で表現方法がだんだんと培われてきたものだと思うんです。

翻ってコンピュータのシステムをみますと、たかが十年の歴史である。現在まだ仕様書という文字で書いたものでやっている。先ほど言いましたように、組織上の背景とか真のニーズというのを掴むには、最初の段階ではなかなか掴み切れない。だんだんといろいろな現場の方々の意見を聞きながらやっていくうちに固まってくる。そのため実際に端末の前に座ったときとドキュメントで見たときと、全然イメージが違うというようなことになってくる。

したがって、コンピュータのフィールドというのは、すべてについて未成熟ということをお認めざるをえないと思います。これを今、一步一步、環境条件を整えながら埋めていっているんだと理解しています。

一般論になりますが、メーカのSEとしての開発者の役割というのは、ハードウェア、OS、そして、それに関する基本的な知識をお客さまに提供していく。それにとまって、標準のソフトウェアとか、開発者のツールを提供していく。それから、何にも増して、そのお客さまの業務、あるいはお客さまのニーズというものを修得していく。この知識の修得というのは、ただ外から見たのではなくて、中に飛び込んで、泥まみれになって、お客さんと一緒に開発していくということの中から培われてくるものじゃないかと考えております。

11. 利用者へのお願い

金井 それから、利用者に対する注文というものは、はなはだ僭越ですけども、非常に確保しにくいわけですが、最低人数のSEの確保が必要ではないか。専門的な技術をもった方の確保は非常に難しいかもしれませんが、たとえば現場に非常に習熟していて、大きな仕様のディスカッションとか、院内のコンセンサスづくりとか、そういうことのできるセンスというか、そういうものをもった方の確保が必要ではないかと思えます。

それから、設計とかコンサルテーションといったプログラムという形になる前の作業に対するものには大きなコストがかかっている。この認識の不足が、コンピュータ開発作業の成熟の一つの大きなネックになっているのではないかという気がします。

先ほど言いましたように、仕様の決定とかコンセンサスづくりというのは、お客さまに分担していただかなければなりません。そのためには、そういったコーディネータに当たる方々がお客さまの中に必要で、その確保が成功への一歩ではないかと理解しています。

12. 発注者の問題点

山本 図書館情報大学の山本でございます。



私の考えますシステム開発の問題点の中からいくつかピックアップしてお話しました後での対策を今までのお話の中で出てきたことはなるべく除いて、いくつか申し上げたいと思います。

私、特に学校や官公庁関係のシステムの設計とか構築のご相談にあずかる経験が今まで割合多かったものですから、主としてそういうシステム開発について気のついたことを申し上げます。

こういう場合、開発が学校や官公庁の中だけで行われることが最近では少なくなりました。開発者が発注側と受注側に分かれるのが一般です。発注側が要求仕様を作り、概略設計は共同作業、それから先は受注側のメーカやソフトウェアハウスが開発するのが多いわけです。ここでまず、発注者側の問題点から申し上げたいと思います。

まず最初に、発注者側が比較的小さい組織である場合とか、大きい組織でも情報が専門でない、あるいは専門でないと思っているところに問題が多いわけです。

たとえば、発注担当者に技術的知識がないとか、いくらあるけれども古い知識を金科玉条にしているとか、逆に、勉強はよくするけれども、勉強し過ぎて新しいことに浮ついてしまっているとか、いろいろあります。小さい組織で、開発担当者が1人か2人である場合、ワンマンになっちゃう恐れがあるんですね。たとえば官公庁の場合、使われる金は自分の金じゃなくて、国民の税金なんですけれども、自分で払うような気になって、おれの足跡をシステムに残してやろうという気持ちになってしまう。これはいい場合もある

わけです。過去の技術に安住しないで新しくしていくというドライブにもなるのですけれども、恣意的システム作りになる危険性もあるわけです。

しかし、こういうのはまだいいほうで発注側の責任の所在がはっきりしない場合、多分一番やりにくいんじゃないかかと思えます。こういうとき、外部からわれわれみたいなアドバイザを臨時に呼んでみても、大学の先生なんていうのは大体無責任なものですから、業務の実際を知らずに不適切な助言をしてしまうこともあります。それから、調査検討の時間とか権限がどこにもない、あるいはないと思うことが問題ですね。たとえば予算が決まる前に金を使う準備をやっちゃいかんとかいうようなことで、いろいろ問題が起こっている。だから、前のシステムを使っているうちに次のシステムの検討を十分やっておかなきゃいけないんですが、そこが難しいわけですね。前のシステム自体やその運用にいろいろ問題がありますから、目先の問題の処理に追われているうちに、次のシステムの発注の時期がきて、予算が決まってからあわててシステム検討委員会などをつくるけれども、どうやって検討すべきかも分からないということが多いようでありま

す。こういうことでは計算機会社にとても太刀打ちできないので、発注先に発注仕様書を書いてもらうというような悲惨なことになる例もあるわけで、これでは困るんです。

発注者と実際のエンド・ユーザとの関係というのは、このシンポジウムの焦点なんですけど、やはり問題があることが多いようです。ワンマン型の人は、おれがみんな分かっているという感じになるわけですね。あと、特に運用責任者の方では、うるさいユーザがいて困る、ユーザを甘やかしちゃいけないという調子で、ユーザの声に耳をふさいでしまう人もいます。

比較的大きな組織になると、関係者だれもがエンド・ユーザの声を聞く手段や権限がない、あるいはないと思っているという場合がしばしばあります。これは民間企業にもよくあるように思われます。しかし、発注担当となったら自分の時間を使っても何しても、とにかくユーザのご意見を伺うということが必要なんじゃないか。このあたりはまた対策のほうで申しますけれども。

もう一つは、ユーザとは仲よくしているんだけれども、そのユーザが古い型のユーザであるという問題です。大体、現在のシステムの大ユーザというのは、将

来のシステムでは古いタイプのユーザになる場合が多いわけです。昔、昭和48年のことですが、これに関連して統計数理研究所の方が大変いい調査をなさいました。大学の大型計算機センタの次期システムについて、ノンユーザとユーザに分けて、どういうシステムが欲しいか調査されたわけです。すると、当時の大ユーザはすべてバッチ処理の時代ですから、ユーザからは、バッチのターナラウンドが2週間になっているので、これを2、3日にしてくれとか、メモリを少しふやしてくれとかいう地道な要望が多く出ました。一方、ノンユーザから TSS というものがあるそうだがこれは使えないか、文献検索システムが欲しいとか、画像処理とか、未来志向の意見が活発に出たのです。こういうことは今でもあることだと思います。

13. 受注者の問題点

山本 さて、受注側ですが、一般的にはよくやっておられると思うんですけども、いくつか問題があります。まず、日本の会社の縦割体質で、たとえば官公庁担当の人と会社担当の人が全然関係ないというのが問題です。多くの場合、官公庁の業務というのは常時忙しいことはないんです。ですから業務システムもパフォーマンス要求が厳しくないことも多いんですが、ときどき非常に忙しい業務をこなさなければならぬシステムがあります。そういうとき会社関係で厳しい仕事をしておられる方に教えてもらったらどうですかと言うんですが、それがなかなかできないようです。こういう場合、外国系の会社を見ていると、社内のコミュニケーションがいいように思われます。一生懸命頑張るのはいいけれど、自分の会社のプロダクトを十分知らないように見受けられる例もありました。

日本の会社のいいところは、受注側が発注側と一体化して、没入して、一生懸命頑張ることだと思います。日本のソフトウェアがなんとか言われますけれども、新日鉄とか国鉄とか、いろんなところのシステムが、できたときには世界に冠たるシステムなんです。ただ、それはそこ1社、あるいは1校、1センタ向けのシステムにとどまっています、これを一般化した道具というのがなかなかない。自分はこの担当の部だからというわけで一生懸命やる。また別なところを担当すると、それを一生懸命やる。その繰り返してある。すると、時がたつにつれてその経験が残らないで、逆に品質が下がっていくこともある、ということ

を感じます。この場合、品質というのは故障率とか、処理能力とか、そういう計測しやすい面の品質ではありません。こういう面は大体向上します。

去年、銀行のキャッシュ・ディスペンサ端末のユーザインタフェース調査をやったんですけど、新しい端末は機能が増えているけれど、使い勝手は昔最初に作ったときほどきびしく考えられていないシステムもあるように思いました。ここでも、経験の一般化と継承が必要だと感じます。

14. エンド・ユーザの問題点

山本 最後にエンド・ユーザ側の問題ですが、エンド・ユーザが全く意見を言わないのが一番困るわけです。発注者側にしても困りますし、受注者側も困ります。だから活発に意見を言ってもらわなきゃいけないんですが、日本人のユーザ、特に官庁関係のユーザの特性というのは、使ってみてだめだったら文句を言わずに黙って離れていくということなんです。もっとも全然言わないわけではなく、外では言っているわけですから、中できちんと言ってもらおうテクニックというか、その体制が必要だと思うんです。

もう一つは、文句は言ったけれども、全然直らんじゃないかということですね。これはタイムラグの問題で、特に大きいシステムではタイムラグも大きいので、注文はちゃんときており、対策はいつでもできるんだということをフィードバックすることが重要じゃないでしょうか。ただ、ときどきタイムラグが10年以上かかったこともあって、これでは次第に注文をつける気もなくなります。

もう一つ、発注者側、あるいは受注者側からエンド・ユーザの問題としてよく言われることは、注文がきつ過ぎる、あるいは細か過ぎるということです。たとえば、ユーザが専門家の立場から、どうしてもこれは譲れないと主張されると、細かいものを個別につくらされるので、ちょこちょこことごまかしてシステムを納入しますとか、あるいは、初めからシステムを知らないユーザは相談から除いておく、というようなお話がよくあるんです。

この対策としては、エンド・ユーザを外すのではなく、エンド・ユーザにこれから作るものをよく知ってもらうことが必要です。使ったことも、見たこともないものについて意見を出すと、どうしても、あれも必要これも必要、全部やれという話になりがちです。これから作るシステムのモックアップとか、あるいはま

ず、類似のものをエンド・ユーザに使ってみてもらって、意見をきくことができれば、意見も建設的になります。

発注側、受注側とも、エンド・ユーザの声を本音で大事にするということが一番重要なんじゃないかと思うんです。建前として一応大事にすると言いながら、本当は知識水準が低いんじゃないかと思っているのはよくない。

それから、エンド・ユーザ調査の方法論をつくる必要がある。実際自分でシステムを開発しても、エンド・ユーザというのはこう行動するだろうとか、こう思うんじゃないだろうとか、いろいろと考えるんですけども、本当のところ、それをどうやって調べるかが難しく、結局は主観的に決めてしまうことになりがちです。また、発注側に回った場合も、たとえば受注側に「エンド・ユーザはこうなんじゃないか、だからこうしてください」と注文すると、「そうは言っても本当は分からないんじゃないでしょうか、それは先生の主観じゃないですか」と言われる。そこで今、ユーザ調査というか、人間の心理的なものの調べ方を少々勉強し始めているところです。

それから、ユーザ教育がもっと必要じゃないだろうかと思います。ユーザ教育の最初にまず、ユーザの権利と義務を教える必要があります。ユーザの権利は選択の権利であり、義務は批判の義務です。

もう一つ、工程についての知識をもってもらいたいという話がありましたが、エンド・ユーザにとってもこれは重要なことだと思います。

こういう要求を出したら、それはどのように実現されるのか、どのくらい時間とお金がかかるのかというようなことは、ある程度知っている必要がある。

私たちが学生の教育をするときにも、ユーザあるいは発注者としての教育を重視しています。うちの大学は6割以上いろんな企業でシステム関係者になるんですが、あと3割ぐらい、図書館に行ってユーザになる学生がいます。そこで、ユーザになったときに、ユーザの最終的な権利というのはどのシステムをとるかということである。また、発注のときも、どういう特別な機能をつくってくれとか細かいことを言わないで、全体としてどういう仕事をしたんだ、どういうふうにとりかかるとかスピードでできなきゃいけないんだということをはっきり言う。相手にそれができなかったら、さっさとほかのものを買い。もし不満でもつくっちゃったら、今度はこれをしつこく使った結果をフ

ードバックして、作ったところに文句を言う。聞いてもらえなかったら、一般性のある文句から公にする、というようなことが大事だと教えています。

15. 改善の方向は

山本 最後に、官公庁でのシステム開発と、民間でのシステム開発のそれぞれについて、改善のためにかねがね感じている点を二・三のべたいと思います。

まず、官公庁のほうでは、情報処理や情報管理の専門家を重視し、もっと数をふやしてもらいたい。情報処理というのは、国家公務員の1種にあるにはありますが、ほとんど研究所向きになっています。情報管理に至っては、項目にもありません。図書館学は、以前上級甲種だったこともあるのに、今は2種になってしまって、大学図書館向きのスペシャリストという扱いです。大体、官公庁の仕事のきわめて大きい部分がきちんと待遇することと、さらに、情報管理と情報処理なので、この方面の専門家をきちんと採用し、一般行政試験に情報管理や情報処理の問題を入れるべきだと感じます。それによって、官公庁のシステムはずっとよくなっていくでしょう。

次に、システム開発におけるフリーランスのコンサルタントの活用ですね。たとえば在日外人の間では、パソコン・コンサルタントが何人もおり、趣味でやっている人もたくさんいて、電話をかけると親切に教えてくれますが、小さい事務所のシステム開発やシステム選択に活躍しているようです。特に比較的小さい組織の方にお勧めするのは、技術士の活用です。現在まだ多くの技術士の方々は会社に勤めておられて、「老」というには遠い方が多いのですが、次第に知識も経験もある方々が独立されてゆくので、こういう人にコンサルタントになってもらうと有効ではないでしょうか。今一つは学生で、学生は一般に無責任なものです。システム開発のアイデアを求めるには、そこがいいんじゃないかと思えます。

まともなくいろんなことを申しましたが、以上です。

司会 どうもありがとうございました。

コンピュータは生物ではないかという水野先生のお話から始まりまして、山本先生には、なんとなく今まで出たことについてまとめていただいたというようなことで、一応パネラの方々の意見をお聞きいただきました。皆さん方のご意見なり、質問その他でも結構です。いらっしゃいましたらお受けしたいと思えます。

16. SE がない中で

上野((株)総合ソフトウェア研究所) ただいま、各先生方のお話になったことは非常にぐもつともでありまして、私ども 20 年間苦勞してきたことが、ほとんど今までのお話の中に盛られているわけです。全般的にみますと、どちらかという、お前も悪いけどおれも悪いような、けんか両成敗のお話がほとんどだったんじゃないか。

一つは、金井先生のおっしゃった、発注者側でも優秀な SE があってほしいというんですが、これは不可能に近いことじゃないかと思われる。これから先、技術はどんどん進歩するでしょうが、橋本先生のところのようにシステム屋さんがおられる会社は別として、普通の官庁であるとか中小企業では、利用者側の人、多少はコンピュータも知っているかもしれませんが、優秀な SE なんかににはなりっこない。せいぜい業務をよく知っているというのが精いっぱい、コンピュータの知識についても、そう深く分かっているわけじゃない。しかも新しいものについて特に分からない。

もう一つは、最初に水野先生がおっしゃったんですが、コンピュータというのは火星人だと。これに対し、メーカさんはどうしたか。昔は、これはどうせ皆さん方は分かんたらう、人から何から全部つけてあげますから使ってくださいと、分かんずに使わせたわけですね。それがいまだに尾を引いているんじゃないかという気がします。ですから、そういう点を踏まえて、どういふふうなアプローチの仕方があるのか、それをお伺いしたいと思います。

金井 私、最初にもお話をさせていただきましたように、私の携わっているのは病院が主でございます。病院というのは、おっしゃるとおり、普通の企業以上にそういった人材の確保は難しいところでございます。病院というのはライセンス社会でございますから、基本的にはほとんどすべての人がライセンスをもっている。その中で、情報処理技術者というのは、技術者の認定試験などございますけれども、地位が非常に不安定である。そういうところに一般の企業のような形で優秀な人材がなかなか集まりにくい。それを日々痛感しております。先ほど SE の確保をお願いしたいというのは、あくまで希望でございます。基本的にはそういうことができないで進んでいる場合が多い。その次の段階としましては、せめてコンピュータの知識がなくても業務に精通して、なおかつ大き

な判断が一応はできる。院内あるいは企業内の問題について右か左かというような大きなディシジョンを、ある程度説得力をもってできる人の確保が必要ではないかということです。

水野 利用者として申しあげますが、必ずしも運転まで全部つけて使えというやり方が悪いとは思わないんです。ただ、どうしてまずいかと申しますと、システムに科学技術とかノウハウが入ってまいりますと、ソフトウェアを改造、改善しなければ使っていけないような時代になってきたので、それを考えるということだと思います。とすれば、むしろ利用者側がコンピュータに対する知識がなきゃならない。

じゃ、それはどういう知識かと申しますと、また薬を例にあげて申しあげますが、薬の専門家になるのは、どこの国でも、大学程度の4年から8年、中には12年のファーマシューティカル・エデュケーションがございます。ところが、一般の人たちも薬に関係がある。薬を飲まなきゃならない、薬を使わなきゃならない、化学物質は生活の中のどこにもあるわけです。その教育をドラッグ・エデュケーションと申しまして、全く別の観点から教育しております。

そういう意味で、専門家になることではなく、それでいてかなり該博な知識を与える教育が必要なんじゃないか。医学でも患者教育が必要だとされているわけで、コンピュータでもそういう教育が必要なんじゃないか。若い人は一生コンピュータとつき合うわけですから、多少苦しくても、初めのうちに適当な教育を受けたほうがいいんじゃないかというのが私の印象です。

橋本 2点ほど。1点は、東レは関係会社が100社ぐらいありますが、そのほとんどの会社にシステム技術者はおられません。そういうところでまず言っていることは、一つは、とにかくパソコンを勉強してパソコンでやってみなさいと。とにかくメーカーの人とは絶対にそれまでつき合うなと。

2番目は、私どもはソフトウェアとして独立したわけですけども、有償のコンサルテーションを受けると。この2点を強調しております。とにかくハードをすぐ売りたいがる人とはつき合うなというのが基本でございます。

司会 今、橋本さんからおもしろい話がありました。私もソフトハウスなものですから、今、SEの話がありましたけれども、エンド・ユーザの教育ということもあると同時に、その辺のギャップを埋めるの

も、ソフトウェア産業の一つの役割じゃないかと思っております。ほかにご意見はありませんか。

17. 共通の言葉は

柳生(日本ユニシス) ハードウェアのグループ側がだいぶん批判的に……。ちょっと肩身の狭い思いをしつつあるんですが、その点については、実は私はハードウェアの立場から今ものを申しあげようとは思っておりません。といいますのは、幸か不幸か、ハードウェアが急激に安くなってまいりまして、われわれとしても、広い意味でのソフトウェアが正念場だと思っておりますので、どちらかという、ソフトウェアのベンダという立場で二、三申しあげたいんです。

利用者側と開発者ないし受注者、この間の問題として一番大きいのは、要求仕様あるいは外部仕様、基本仕様、いろんな言葉がありますけれども、要するにどういシステムをつくるのかという共通の理解、これの記述、これが非常に大事で、結局、いろんな問題というのはここから始まってここに終わるという感じをもっています。

利用者が言いたいほうだいを言って、ちっともまとまらない、あるいは全部それを聞いてるとんでもないシステムになる。実はそこまでいかないで、結局利用者が何を言いたいのかちっとも分からない。開発側が分からないのかといえますとそれだけではない。いろいろ議論をしていますと、利用者自身が自分で何を求めたいのかが分からなかったということがだんだん分かってくる、ということがかなりしばしばあるんですね。

それは言いかえますと、利用者が自分の求めたいシステムが何であるかということ、共通の言葉——水野先生のお話の中にレイマンズ・ランゲージ(素人の言葉)という言い方が出てまいりましたが、共通の言葉で表現することが難しい。しかも、表現されたものが、ただ共通であるというだけじゃなくて、明晰でないといけな。意味合いというものは明確でないといけな。明確な言葉というのは、実は自然言語とは必ずしもそぐわないわけですね。そこでやはり広い意味での数学的言語といえますか、形式言語といえますか、そういう言葉であって、しかもそれは自然言語に翻訳可能、逆翻訳可能で、情報処理の素人である利用者にも分かる、そういう言語の確立が非常に大事なんじゃないか。昔から言われていて、いろんな試みがされていますけれども、結局、今日、われわれ、標準と

まではいかなくても、非常に広く使われる言語をもっていないわけです。

プログラム言語の世界というのは、標準が確立されてきましたけれども、要求仕様を記述する言語というのは、おおむね実験室の段階にとどまっている。非常に難しいんですけども、改めてここに焦点を当てる必要があるんじゃないか。

考えてみますと、われわれの仕事は、すべて言語に始まって言語に終わっているわけです。こういう技術というのは、実は伝統的な工業製品の技術には全くなかったと思うんです。伝統的な工業製品の場合には、図面というものがある。図面というものは、いろんな意味で非常に重要な理解の媒体、あるいは表現の媒体になっていたと思うんです。ところが、図面というのは意味を自己記述していないんですね。つまり、あれを見て分かるというのは、あらかじめ人間が分かっている、暗黙の了解というものが共通にある、これが大きいわけですね。ところが、われわれの仕事の場合にはそういうものがない。すべてを言語で記述しなければいけない、表現しなければいけない。ですから、そういう意味では言語の技術である。要求仕様の段階でも、そういう言語の確立というものが非常に大事んじゃないかと私は思うんです。山本先生から、まずその辺、ご意見を伺いたいと思います。

山本 確かにそうですね。先ほど私は、言語というほど一般化しないで、モックアップとか言ったんですが……。でも、本当に言語までいってもらいたいと思います。システム開発のとき、発注側と受注側の両方に、本当にエンド・ユーザにも分かるレベルで話してもらいたいということはいつも感じるわけです。ありべき問題点とか可能なチョイスを隠さないで、こうするとこうなる、こう変えるとああなる、そこでどれがいいですかという形ですね。しかし、それが難しいんです。

こういうやり方のもう一つの問題は、見せたものがそのとおりでできるかどうか分からない。見せてしまうと、それに契約上縛られかねないとか、そういう点で、原則だけを並べた文章や表になる。今の仕様記述言語の研究というのは、私はあまり詳しくないので誤解だといけませんが、どちらかというと、それですぐ物を作るという方向に行っているように思うんですが、SE のためのエンド・ユーザとのコミュニケーションの道具になるといいなと思います。それが柳生さんのおっしゃる言語なんじゃないかな。

18. プロトタイプिंगの役割

柳生 モックアップという表現をお使いになったわけですけども、ソフトウェア危機ということが言われたのが、ご承知のとおり 1969 年、それからこれは大変だと、そのうちにああすればいいんじゃないか、こうすればいいんじゃないかと、だんだん年がたつにつれて、ああやってもうまくいかない、こうやってもうまくいかないと悲観的になって、ついに最後どうなったかといいますと、特に要求仕様については、いくら頑張っても、見るべき成果は期待できそうもない。どうするかといいますと、結局プロトタイプिंगしかない。山本先生のおっしゃったモックアップというのは、言葉でいくらお互いの理解を確認しようといっても、どうもできそうもないから、いいかげんなところで、もしかすると誤解に満ちているかもしれないけれども、作って目にも見せてくれれば、「ああ、大体こんなものだった」とか、「とんでもない、思いもよらんものをつくってくれた」とか、その辺が非常にはっきり出るだろう。

ですから、ある意味では、私は、プロトタイプングというのは、要求仕様という高い水準での言葉による確認の、言ってみれば可能性の放棄ということではないかと感じているんですけども、その辺はいかがでしょうか。

山本 なるほど。実際につくられるプロダクトというものは千変万化ですから、全部を見ようとしても見えません。ましてや、プロトタイプですか、モックアップですか、そういうものを見せたところで、その記述力はきわめて限られるから、びしっとした要求仕様と比べるとだめだというわけですね。ただ、実際、ユーザあるいは力の弱い発注者側としては、一応要求仕様を出しても、その結果が分からないんですね。それが一番の問題だと思います。私も結構長い間注文を出す仕事をしていましたけれども、それでも自分があれこれと連発している注文がどういう結果、あるいは副作用につながるかがなかなか分からない。漸く分かったころには、またシステムの基本が変わっていて、また分からなくなるということだったんですね。

そこで、最終製品のシミュレーション、すなわち、モックアップないしプロトタイプに希望をかけたわけです。私は、柳生さんのおっしゃる言語が、ユーザにもよく分かるものなら賛成です。

三宅 (NTT) 今の問題はとてもおもしろいと思い

ます。きちんとした言語で、ユーザがやりたいことを述べるというのを、ユーザに直接要求するのは酷ではないか。言語で表現できなくても、いろんなことができるようになっていて、最終的に時間がたつと言語で表現できるようになるのがむしろ普通のプロセスです。最初、ユーザというのは、そもそも自分が何を理解しているかということさえもあまりはっきりしないのが普通で、その場合には、プロトタイピングというか別の言い方ですれば、example を提示してくれるということは、多分、現在考えられる一つの非常に有効な方法ではないかと思うんです。

example を提示するということは、実はすごく危険な面も持っているわけです。一般化してないわけですから。ただ、ユーザにしてみれば、そもそも自分のしたいことを言葉できちんと表現できないわけですから、それを助けるという意味では、早い時期に、プロトタイピングなりしていくのが一番いいような気がいたします。大きなシステムをつくるんじゃないけれども、もっと小さなインタフェースを考えるのによく例に出るやつですけれども。

橋本 確かにソフトウェアの技術者は足りないんですけれども、技術者というのは利用者の言うことを抽象化して、うまくモデル化して、それをアルゴリズムに落とす、これが非常に大事だと思うんです。これは私どもだけじゃないと思うんですけれども、なかなか本当の技術者がいない。いい技術者は少数でいい。ところが、一般に SE と言われているのは技術者じゃなくて「記述者」である。これが一番問題だと思っているわけです。これならまだいいんですけれども「偽」「擬」「疑」……術者も、まますみ受られます。こういう人たちが技術者になる努力をすれば、技術者不足は解消するんじゃないかと私は考えております。

水野 また薬の例になりますが、日本では、薬学部のもっと努力の目標は、たとえば有機化学が非常に進んでいるとか、そういうことであります。

それが、カリフォルニア大学の薬学部、その薬学部の中のファームディーコース (Pharm. D.) で、薬剤師を養成していますが、そのコースのモットーはコミュニケーションです。薬学の学部で何が大切かと思ったら、学部長を初めコミュニケーションだと言っているわけです。10年も前から続いているもので、いかに患者と医師と薬剤師のコミュニケーションがなかったかという反省から、とにかくコミュニケーションということを非常に強く言っています。レイマンランゲ

ジの使用も強く言っているわけです。

それから、さっきマルチリンガルということをお願いしましたが、マルチリンガルという意味は、言葉だけでなく、あらゆる道具を使う。たとえば図面を書く。仕様書は、記号ではなくて文字で書いてある。あらゆる言葉、記号などを使って、相手に自分の意思を通じるということが大切なんじゃないか。一つの言語で何かやろうというのは不可能じゃないか。

また、われわれも経験したことです。パソコンで画面などの構成をしまして、大体こういうふう動くよと言いますと、そのあとユーザではなくて専門家がうまくつくってくれる。ソフトウェアの会社のほうで、いくつかのタイプをお持ちになってユーザに示していただけると大変ありがたいと考えております。

19. エンド・ユーザとは

浦 (慶應大) 今、水野さんがコミュニケーションが大切である、それから柳生さんが仕様記述の言語、それに対して山本さんが素人も分かる言語ならいいと言われたわけですけれども、今の SE とかマニュアルを見ても、分かる日本語が使える人がまじらないんですね。記述もできない。その手前にいるような気がするわけです。それをまず正しい日本語で分かるように伝えられる SE になるというのが前提じゃないかと思えます。

ついでにコミュニケーションのスタートとしては、心構えなんですけれども、松谷さんも、今エンド・ユーザという言葉をお使いになった。先ほど発表の武田薬品の方も、皆さんエンド・ユーザというのがお好きのようです。英語のほうは詳しくは知りませんが、「端っこ」というほかに「目標」という意味もあるんだらうと思うので、その辺があいまいなんじゃないか、日本語になった「エンド」は全くの「端っこ」になるという感じがするので、まずコミュニケーションを大切にしようとしたらば、そういった解釈も重要なことじゃないかと思っております。武田薬品さんは、エンド・ユーザシステム——EUS というので好評だったんでしょうか。

佐藤 (武田薬品) 浦先生に指摘されましたが、われわれの会社の中で、最初にプロトタイピングという形で、各ユーザにデモしたわけです。エンド・ユーザという言葉で説明会をするからと集めたんですが、特に意識してやったわけでもないんです。まず、われわれのユーザからみた場合に、エンド・ユーザという

お医者さんであったり患者さんなんですね。そのとき1回考えて、名前変えようかという話まで出たんですけども、世の中これだけ言ってるししょうがないなと。きょう浦先生に言われたから、帰って名前を変えようかなと思ってるんです。

メーカさんからみた場合に、ユーザはコンピュータ部門である。その先であるから、エンド・ユーザは、という考え方だと思います。ですから、どこを主体として考えるかということは大事な問題じゃないかなという感じはしますね。先生はエフェクティブという名前に変えた方がいいんじゃないかと言われました。こういう研究会の中で、間違った言葉を直していこうじゃないかということも大事なことではないかなと思うのです。浦先生が僕に言われたのは、そういう意味もあるのかなと理解しているわけです。

山本 今思いついたんですけれども ultimate user (究極のユーザ) というのはどうですか。

司会 実は、私、ここではエンド・ユーザという言葉を使っていますが、たまたまそれが世間に流通されている一つの言葉なので使っております。私自身はむしろオーナじゃないかと思っています。使う人は、そのシステムを自分のものとして使う。そういう意味です。実はオーナという言葉で、全部相手とも話をし、そういう言葉を習慣づけたこともございました。ちょっとはやりにくいところもあるんですが、私自身はオーナだと思っておりますし、またできればそういう言葉を普及させたいと願っております。

まだ話があると思いますけれども、時間もきましたので、最後に一言述べて終わりたいと思います。

まず、利用者の世界というのがありまして、それがなんらかのかっこうでコンピュータの世界になっていく。利用者の世界とコンピュータの世界というのは猛烈なギャップがあるということです。このギャップを解消する技術はまだ未熟です。このギャップを埋めて

いくと同時に、できたものが何かというと、好かれもするし嫌われもする、利用者の世界の仲間じゃないかということでした。

そういう意味では、エンド・ユーザというのは、システムができれば使ってやるよという話じゃないと思うんです。やっぱり自分たちの仲間なり、あるいは自分たちの世界をつくる、あるいは仲間を育てる、嫌いなものは怒って教育する、要するに批判をして直していく、そういう意味の責任なり何なりはあると思います。そういう意味では、私自身はオーナだと思っております。

片一方で、エンド・ユーザはシステム構築については、やっぱり経験は薄いと思います。開発者が有する経験というのは、ある程度偏りがあるにはしても、継続的に経験が積み重ねられます。エンド・ユーザというのは、たとえば同じ職に何年かいたとしても、そういうチャンスにぶつかるのは何年かに1回でしょう。あるいは人が変わっていく、なかなかチャンスがありません。そういう意味で、開発者はどうやって構築していくかということについてエンド・ユーザをうまくリードし、あるいはコンサルテーションを行う、できるならば有償の形というのが私の願いですけれども、役割があるんじゃないかと思います。

そして、開発者にとって、理想とするところは、コンピュータのこの字も出さずにエンド・ユーザとの間の話をして、システム構築していける人間が、本当のシステム・エンジニアじゃないかと思います。

いろんな意見がございました。それなりにどれも非常に有益な意見だったと思います。本当にいろんな意味で本音の議論が聞けたんじゃないかと思います。

時間が残念ですけれども、これからさらに懇談の中でもお互いに大いに議論していただきたいと思っています。

どうもありがとうございました。