座　談　会

保　守　に　つ　い　て

角田　基文[1]，後藤　敏雄[2]，蓑輪　　充[3]
山本　欣子[4]，桑折恭一郎[5]，南条　　優[6]
(以上発言順)
遠藤　　誠[7]*，戸川　隼人[8]**

司会(戸川)　今日はコンピュータの保守の問題点とRASへの期待，注文，批判といったものを，実際に保守を担当しておられる立場の方々と，ユーザの側の方々に，話し合っていただきたいと思います．

最初は保守の経験があり現在，保守員の技術指導に関係しておられる角田さん，後藤さん，蓑輪さんを中心にお願いします．角田さんからどうぞ．

角田　私どもが現在保守の中で一番困っているのは，人間をどうやって育てるかということです．小さい機械のうちは数か月で養成できてしまいますが，大きな機械になるとなかなか覚えられません．そうなると機械的にメインテナンスができるようなハードウェア（もちろんソフトもからむと思いますが）を考えてもらいたい，という心情になります．

後藤　レベルの高い人を養成するのが非常にむずかしいというご発言，まったくその通りで，計算機の1から10までを全部覚えている人は非常に少ないわけです．それで半分ぐらい知っていて，あとはオペレーションができる，という程度で保守ができればいいと思うんですが．

蓑輪　ちょっと古いたとえですが，医者の場合，患者の顔色だけ見て的確に当てるのが名医，いろいろと診断した揚句に正確な診断を下すのが良医，それでもなおかつわからないのは藪医者だというふうに言われていますが，コンピュータのサービスの面から見ると，なかなか名医というのはいないわけですね．せいぜいわれわれとしては良医を目ざすしかない．しかしそれでも現在の進歩に追いつけないので，サービスアビリティーという面でファシリティーを持たせて，藪医者でも名医と同等の仕事ができるという方向に進みつつある．たとえば，FLTとか，ログアウト・アナリシスとか……．こういうものを有効に使えば，なにもお客様に症状を聞かなくても，的確に故障を診断できる．つまり藪医者でもインターンでも名医なみになれる．問題はエンジニアのレベル以上の問題が発生したときです．そのへんまで自動化で解決されればカストマ・エンジニアというのはいらないという時代もくるのでしょうが，なかには悪口を言う人がいまして，CEというのはカストマ・エンジニアじゃなくて，故障診断を機械にまかせ，部品の交換だけをするチェンジング・エンジニアになってしまうという．しかしいずれにしても現在のコンピュータがそういう方向に動きつつあることは事実なのです．

司会　370の自動診断機能の的中率はどのぐらいでしょうか．

蓑輪　まだ詳しいデータは出ておりませんけれども，担当レベルに言わせますと，予想解析して的中するべく予想されていたトラブルが100%つかまっておるという．微妙な表現ですけれども，予想外のものが出た場合には，当るか当らないかまだケースが少ないからわからんということです．かなりの程度までわれわれは期待しておるのですが，こういったデータを収集するのは非常に期間がかかるものですから．

遠藤　しかし設計の立場から見ると，本当に頼られすぎてしまうとこわい．100%見つけることは困難です．いま蓑輪さんのほうからも，当るべきものは当っているというお話がありましたが，当らないという

1) 日立電子サービス(株)
2) 東京芝浦電気(株)
3) 日本アイビーエム(株)
4) (財)日本情報処理開発センター
5) 日本国有鉄道
6) 日産プリンス自動車販売(株)
7) (株)日立製作所
8) 京都産業大学
* 幹事
** 司会

左より遠藤氏，蓑輪氏，後藤氏，角田氏．

左より山本氏，桑折氏，南条氏．

場合があるわけですね．そうなるとわかる人間がいなくなって，ユーザさんに迷惑をかける心配があります．どういうことかというと，保守の手順を楽にすると自動診断は万能だという話が保守員が頭に叩き込まれてしまって，人間の技術が低下して，いまの RAS で引っかからないような事故が起きたときにどうしようもなくなる可能性がありますね．

角田　確かにある意味じゃそういう懸念もあると思いますね．いまの RAS のいき方ですと最終的には人間が出ていかなくてはならない．しかし高級な技術者はほんの少しいればいいということになりますので，それならば比較的やりやすい．高級技術者を多勢作るのはむずかしい話で，そのへんが，われわれとしては，非常に RAS に期待しているところです．

蓑輪　将来は高度の教育を受けた少数のエンジニアで全国をカバーすることになると思います．

インターミッテントなエラー

角田　それから，実際保守をやってましてなんとも困るのはインターミッテント（断続的）なエラーに関するものです．ときどきポロッとエラーを起す．なにもチェックに引っかからないで妙なことになってくる．現場の保存がないものですからなかなか原因がつかめない．これを解決する手段は，これからいろいろお話があると思うのですけれども，かつての機械は，そういう意味では非常にやりにくかったのです．

司会　そのへん後藤さんはどうですか．

後藤　いま角田さんのほうからインターミッテントなトラブルが起った場合のログ（記録）の取り方のお話がありましたが同感です．実際の機械ではインターミッテントなエラーは再現できない．ということは，エラーが起った直後にトラブルがなくなってしまうことなのです．しかし再発すれば，業務の最中に起った場合，それがダウンに直結してしまうので，ユーザに非常に迷惑をかける．しかし RAS の思想からすれば，リトライをし，さらに別な形でログを取りながら，なおかつ業務を進めることができる．そうすると，変な言い方になりますけれども，保守員の精神的な負担が軽減されることになるわけです．あとで行ってログを調べればよいというような形になりますからね．

蓑輪　インターミッテントなトラブルといえば，私どもも以前はたいへん苦労しました．2時間正常に動いたあとでポカッとエラーがついて，またそのあと3時間は走れるけれども，そのあとでまたエラーがつくというぐあいです．その都度止めて保守したらたいへんです．また直そうとしても適切な方法がない．確証をつかもうとすると，シンクロスコープとにらめっこするようなことにもなります．波形の落ちるのを見ているわけです．一般に余分なパルスが出るのを見つけるのは比較的楽ですけれども，一定の周期で発生してるパルスが途中でポッと消えるのをつかむのは非常にむずかしいわけです．2～3時間を1人でにらめっこするのじゃ，まばたきをする間に落ると困るから，3人ぐらいで一緒に見ると，3人の目玉でリライアビリティーが何倍になるかというような計算になってしまうわけですね．こうしてやっても，時間を使って，お客様からはクレームをいただきますし．

司会　それが370になってそういうことしなくてもすむようになったのは RAS のお陰と言ってもいいことですか．

蓑輪　ということになります．

司会　遠藤さん，インターミッテントなエラーの対策として，金物で自動的にエラーのログを取るというだけならばそんなにお金をかけずに，できるわけですね．

遠藤 そうですね．記録自身は．

司会 リトライからなにから，いろいろ面倒見なければならないけれども．

遠藤 どんな記録を取るかという問題はありますが，記録自身をとることはさほど面倒くさいことではないと思います．しかし，その記録からここだというのを探って，それをどうしたらいいかということを自動的に診断するのはかなり面倒な話です．

司会 人間にとってはどうですか．

角田 記録をとるときに，どういう記録を取るかということを最初から考えておかないと，どこがおかしいよという話にもなりませんね．そこいらあたりが，ただ記録をとるといっても設計当初より考えなければならんでしょうね．

養輪 それは重要な問題です．370 がさかんに RAS ということを言いますのは，設計の段階から RAS オリエンテッドのシステムなんですね．個々に見ると 370 の RAS 機能は，ほとんど 360 で実用化されているものばかりなのですが，370 というシステムをデザインする上で，最初からエラーをこういうふうにして解析するんだという形で，RAS を設計の基本に置いている．ログにしてもどの程度のものを取って，それを解析のプログラムにどう組み合わせるかを，設計の初期に取り入れて，RAS 機能にハードウェアを合わせて設計するという考え方です．これが 370 の一つの特徴になっていると思います．

遠藤 私としてはこう考えているんです．computer aided というのがデザインにしても教育にしてもありますね．その場合でも無能な人間をコンピュータでいくら援助しても仕事はできない．やはりコンピュータが援助する人間というものは，仕事ができて賢い人間であるということを前提にしている(笑い)．精一杯人間としてはつくすんだが，それをコンピュータで助ける．そうすればより大きなシステムであってもダウン時間が非常に短くてすむ．そういうふうに考えなくてはいけないと思います．

後藤 私は前に設計をやってまして，それから保守のほうに行きまして，3年ほど前からは保守のほうはやっていないんですけれども，以前は計算機というのは徹夜作業をするのが習わしだという感じでやっていたわけです．しかしこれからは，なにも徹夜でなくとも，というようなことになるでしょうね．完全な自動診断ができなくても，使用中に自動的にログが取れるということは非常にいいことで，コンピュータを停めてシンクロ見ている間は，ある意味ではコンピュータもしくはペリフェラルはダウンしてるわけですね．しかし，ログを取っている間はユーザは曲りなりにも業務をやっているわけですから，われわれ保守員になってみますと，使っているのと使っていない状態というのでは非常に精神的な負担が違う．だからログをとる項目の選択がたとえ最良でなくても，あるいは取り切れないものが多少あるにしても，やはり取れるものはぜひ取ってもらいたいというのが保守員の実際の気持ではないかというような感じがいたしますね．

司会 RAS と言われているものの中で，ほかのことはさておいても，まず自動的にログをとれるようにすることが大事だという結論のようですね．

ボックス RAS とシステム RAS

後藤 ところで，RAS（リライアビリティー・アベイラビリティー・サービスアビリティー）について，私自身の考えなのですが，本来，信頼性の分野におきましては，アベイラビリティーというのは保全性プラス信頼性というような考え方でいたのでして，そうするとサービスアビリティーとリライアビリティーがプラスした形でアベイラビリティーになるわけで，なにも RAS というような表現をしなくてもいいんじゃないかと思うんです．IBM さんのマニュアルを見ますとサービスアビリティーについては非常によく書かれているように感じましたが，本来はリライアビリティーを徹底的に追求した形でのアベイラビリティーのアップということを一方で考えなければいけないのではないでしょうか．

養輪 さきほどから，たびたび RAS のお話が出ましたけれども，ひと口に RAS と言いましても，ボックス RAS と，実際にお使いになる場合のトータル・システムとしてのシステム RAS の2通りのものがあると思うんです．昔からボックス RAS という考え方がありまして，設計段階でこの程度にしか壊れない基準ということを設けて，それに見合ったボックスをデザインするということをやっていました．ハードウェアのリライアビリティーは，この段階で追求されるわけです．その場合お客様からどう頼まれましても，設計基準を上回るメインテナンスというのはむずかしいわけです．これが 370 のころになりまして，システム・レベルの RAS，すなわち特定の RAS 基準で設計されたボックスを組み合わせて，全体としてのリライアビリティー，アベイラビリティー，サービスアビリティ

ーを設計するようになったわけです．その場合サービスを担当する側としてはお客様にもお願いしなければならないことがあるかも知れません．たとえばスペアユニットを何台持てばどの程度までシステム・リライアビリティーを保証できるかといったような話になると思うんです．このようなことは，最近さかんになってきたリアルタイム・オンライン・システムの場合に重要です．というのは，バッチシステムですとアベイラビリティーが高ければよかった．いったん止まっても動き出せば業務は継続できた．ところがオンラインになると一回でもダウンすると大きな問題になるので，アベイラビリティー・プラス・リライアビリティー・サービスアビリティーという考え方が必要になってきたわけです．アポロのコンピュータの場合 99.9999% のリライアビリティーを保証するには十何年かのMT-BF が必要になるという話ですから，一般に使用されるシステムでこれを保証するようなものは，ここ当分出てこんのじゃなかろうか．昼間8時間を保証することは考えられますけれども，24時間考えまして，トータルのアベイラビリティーという面から考えると総合的に落ちるわけです．リライアビリティーが要求される条件というのは，レーシングカーみたいなものですね．一定の時間，一定の距離走ればいい．その間は絶対にこわれては困る．しかしレースのあと壊れたらバラバラになってしまってもよいという要素が強いわけです．アベイラビリティーというのはタクシーについて要求される．多少の壊れはすぐ直ればいい．そのかわり長い距離を走っていれば儲かる．ところが現在のオンライン・システムでは，以上2つと，おまけに値段の安いシステムということも要求されることになるわけです．そうするとロールスロイス的な高級車を軽自動車なみの値段で作らなければいけない，ということで，これはたいへんむずかしい．最低限，そういった特殊なリライアビリティーを保証するためにはスペアが必要になることもあるし，CPU もデュプレックスという考慮が必要になることもあります．ただ複雑になればなるほどシステム全体として直しにくくなる．こういった点で，システム・レベルでの RAS の考慮が必要になり，それも，ソフトウェアとハードウェアを切離さずに，システム RAS の機能そのものが，オペレーティング・システムでコントロールされる一つのシステムになってきていると思うんです．

ユーザ側から見た保守の問題点

司会　このへんで，ユーザ側から見た保守についてお話を伺いたいと思います．

山本　私どものところは昔から各社の計算機（国産機だけですけれども）を使っているものですから，各メーカーの保守員を，いやでも比較してしまう（笑い），そのつもりはないんですけどね．先日，ある大ユーザの方と話をしてましたら，どこの機械が一番いいかという話になったわけです．そのとき，意見が一致したのは，某社の機械が一番安定していると，それはなぜかというと，保守員が非常によく教育されてるからだ，という点で意見が見事に一致したわけなんです．実は，それは常々私も考えていたんで，メインテナンスにきている保守員の人の如何がコンピュータの信頼性如何にそのまま置き替えられているケースが非常に多いわけですね．たまに保守員が交替すると非常によくわかるんです．いままでほとんどトラブルのなかった機械がものの見事にトラブルの連続になって，とても同じ機械には思えないくらいです．それで初めて前任者が非常に優秀であったということがわかるのですが……．

養輪　保守員によってシステムの信頼性が左右されるのは一面の事実ではあると思います．また，そのように評価されること自体，ありがたいことだと思うんです．しかし現状は従来のボックス・オリエンテッドのメインテナンスではどうにもならない時代にきていて，ソフトの知識も必要でしょうし，客先のアプリケーションの内容も理解していかなければやっていけないような時代にきていると思うんです．そうなるとにかくサービスの体制ですね．個々の一人一人，人間の教育も必要だけど，それを組織として如何にサポートしていくかといったようなことが広い意味での RAS 機能の一環として必要であろうという感じがします．

山本　ですが，基本的にはハードウェアが丈夫になるのが一番いいだろうと思います．保守員の教育も大変でしょうし，優秀な人がたくさんいるわけではないし，機械はどんどん数が増えますんで将来は少人数で，大きな機械でも，うまく保守ができるようになることが望ましいでしょうね．医学のほうでも最近は予防医学が重要だと言われますが，コンピュータでも運転中にロギングして予防医学的に障害を実際に起す前になんかキャッチしていただけるようなシステムにな

らないだろうかということを考えてます．特に最近の技術ではリトライとかいってなんか外に見えないところでやり直してしまうなんてことがあるようなんですけれども，そういうことになると余計，予防ということが重要になるんじゃなかろうかと思います．

実際われわれが使っていて一番問題になるのは，原因がハードなのかソフトなのかわからないというケースです．昔はマシン・コードでプログラムを組んでましたから，ハードか，自分のプログラムか二者選択でしたから簡単でしたが，最近はその間にオペレーティング・システムが入ってきていて，そのOSの中には処理プログラムと制御プログラムと両方がある．それからユーザのプログラムがある．この四つのものがからんできて非常にむずかしい．CEの方は，最近はソフトウェアの勉強もされているようですけれども，大体においてソフトウェアはあんまりご存じないし，たとえ少々知っておられても最近のコンプリケートなOSの中身でどこが悪いかなんて見つけるのはとてもとても．そのために，ユーザは昔に比べると非常にトラブルの原因を見つけるのに時間がかかるということが現実に起っております．

養輪　ハードウェアが丈夫になってエンジニアがハードウェアのサービスから解放されたものをソフトウェアに向ければソフトウェアのリライアビリティーが高度化するんじゃないかと思います．将来は保守員もソフトウェア的な感覚を持って，そちらの面でどんどん成長していかなければメインテナンスはできないでしょう．またシステムのデザイン面で，ソフトウェアのサービスに対する配慮を今後さらに強化されていかれればいい．ソフトウェアのサービス自体もカストマ・エンジニアの仕事であるということにして，ソフトウェアのリライアビリティーとかサービスアビリティーを考えていくことが必要です．

司会　それはまだ実現していないのですか．

養輪　オペレーティング・システムのメインテナンスは，すでにCEの仕事になっています．ソフトウェアのリラアイビリティーというものはむずかしいもんで，あくまでも人間が書くものですから，人間のプログラミング上のリライアビリティーは30ステップで1回のミスの可能性があるとか申しますように如何に改善してもコーディングの過程での大幅な向上は望めないでしょう．

山本　私，ちょっと考えているんですけどね，ハードウェアを組立ててロジックをテストする方式というのは割とシステマティックにできているから，そういう技術をソフトウェアのデバックに利用できないだろうか．ハードもはじめは虫がいるんでしょうね（笑い）．けれどもシステマティックに回路のテストができるようになってるので，割と早期に収れんしているのじゃないか，ソフトもなんかそういう技術を開発すべきじゃないかという気がしているんですが．

南条　ソフトというやつは簡単に直るからいいかげんに作るのでしょうね．一遍作ったら絶対に直らないようなソフトを作るとすれば，もう少し本気でやるんじゃないんですか（笑い）．ユーザが作ったプログラムとメーカーが作ったシステム・プログラムとが外見的に似ているというか，同じ手段でやっているのはおかしいんで，本当は違う手段を取るべきだと思うんですけど．

山本　まだOSが固まらない時期には，ユーザがOSの中身を読んでエラーを見つける手伝いをしないととても実用に間に合わないことがあるんです．他人の作ったプログラムを読むということはとてもたいへんなことなんですが．

遠藤　MULTICSの例を見ますと，そのへんはOSにはバグ（虫）があるんだと割り切ってますね．OSの中でも，バグの居そうな部分は少しプライオリティーを下げて動かして，バグが出てくるとすぐに尻尾つかまえられるようにしているわけです．そのへんはずい分苦労したあげくにそうしたんじゃないかと思われます．このようにハードばかりでなく，ソフトにおいてもバグが出たらその記録が取れるような手段をつけておかないといけないと思いますね．

リアルタイム・システムの場合

桑折　私はいままで主にリアルタイム・システムの設計をやりまして1時期，保守をしたこともありますが，その経験で考えてみると，デュプレックス・システムや予備機を持ってるようなシステムの場合ですと，両系ともハードがソリッドな故障を起してシステムが動かないというようなことはほとんどないんです．どういうことかというと，システムの信頼性はMTBFとMTTRで評価できますが，まずMTBFを考えてみますと，システムがダウンするという原因は，インターミッテントな障害と，ソリッドな障害と二つありますが，その比率を見ますと，インターミッテントのものが圧倒的に多いわけです．そのほかにソフトの原因によるものもありますが，インターミッテ

ントなものとかソフト自体のエラーのうちで，最終的にそのシステムをダウンさせるかさせないかというのは，普通ソフトウェアの判断でやってます．たとえば同じようなハードのエラーが起きた場合でも，あるアプリケーションではシステムをダウンさせる必要がない場合もある．しかしあるものについてはシステム・ダウンに持っていかざるを得ない場合もあるというようなことで，MTBF 自体が非常にソフトの影響を大きく受けているという感じがします．つぎに MTTR の方，つまりシステムがダウンしてから回復するまでの時間はどういうものできめられてくるかという実際のところを見てみますと，故障箇所を完全に直さないとシステムが回復しないというようなことはほとんどないわけです．ほとんどの場合には，もう一遍動かせばすぐ動くわけです．たとえばソリッドな故障が起っても，スペアの機械に切替えてやれば，それで動いてくれる．すなわちリアルタイム・システムでは障害追求とか修理よりもまず運転を再開することが優先するので，障害時間とは金物を直してる時間じゃないわけですね．ダウン時間の中身は，たとえばファイルを持つシステムではファイルの回復をやるとか，人手で回復する場合には，その回復の操作をやる時間とか，それからあとで障害追及のために必要なデータを取っておく，そのような時間がほとんどです．ただし，シングル・システムとか並列システムの中にも共通な部分がありますから，その部分が壊れた場合には，そこを直さないとシステムが動かないわけですが，そこの部分の信頼度を上げておけば，長時間のシステムダウンというのは防げるんじゃないかと思います．すなわちリアルタイム・システムでは，いかにしてシステム・ダウンに結びつけないか，またいかに回復時間を短くするかといったソフトの面での考慮がシステムの信頼性に非常に大きくきいてくると思います．

山本　そのような点でオンライン・リアルタイム・システムと TSS を比較しますと，オンライン・リアルタイムというのはソフトでも障害に関して割とカッチリやっているように思うんですけど，タイムシェアリングはなにかラフですね．しかしバッチの場合でもカード・リーダーとか，コンソール・タイプライターなんかがいかれてそのために大きなシステムが全部ストップするという大変バカバカしい経験が多いわけです．

桑折　そうですね．実際エレクトロニクスの回路の部分では，めったにソリッドな障害が出ない．ところがコンソール・タイプライターのメカのところがやられた場合には完全に動かなくなってしまうことが多い．ですからとくに長時間のシステム・ダウンというのはそういうところのほうが多いんです．さっきも言いましたように，並列システムの共通部分の障害というのは重大なんですが，原因はとくにリレーみたいなメカ的なものがどっちかというと多い．そういうところは障害の起きる確率が非常に少ないけれども，たまたま障害になった場合にはソリッドなエラーになる率が高い．そういうとこの設計の配慮はかなり重要なところじゃないかと思います．

遠藤　現在 RAS という形で，研究者とか設計者とか保守関係者が重視していることは，エレクトロニクスの部分の障害探索が非常に楽になるということですが，実際に使っておられるユーザさんから見ると，もっと泥くさい部分でもシステム上重要なことがあるということですね．

山本　とくにメカの部分になると，メインテナンスの能力だけでなくて「誠意」ですね．それが非常にきいてきますね（笑い）．

養輪　その意見にちょっと反論になるかも知れませんけれども（笑い）．最近のメカもやはりそういった昔ながらのテクニシャンを必要とするメインテナンスの方式から最近は一定の基準を作りまして，だれがやってもこういうふうにすれば必ず元のレベルに戻るといった方向に変わりつつあると思うんです．エンジニアの質によってシステムリライアビリティーが左右されるということ自体問題です．一定の教育を受けたエンジニアがやったら誰がやっても同じようなメインテナンスになっていかなければいけないと思うんですけれども．

司会　そうして救われるべきことは大体救われるけれども，そうでない障害は誰にも直せないようなことになりませんか．

養輪　そういうことに多少ジレンマもあるんですが，そいいう面については少数精鋭主義で，極端に優秀なメンバーを教育してサポートするといったような体制が可能です．また，メカの場合ですと，予防保守が比較的楽なんです．メカのインターミッテントのエラーそのものが，ソリッドのトラブルを予防するためのデーターになりますので，一定のレベルに達した時点で部品交換なり再アジャストなりの措置をすれば，そういったメカのトラブルの大半は防げると考えています．そうした上で，なお最悪の事態を考慮して，た

とえばコンソール・タイプライターがこわれたら，それを切り離してディスクに出しておくというようなソフトウェアのサポートも必要で，そういったシステムレベルの RAS という考え方で，システムの停止を最小限に押える努力が大切だと思います．

山本　しかし現状では，本来はハードが悪いんだけれども，ソフトがうまくそれをカバーしてないという問題がまだいろいろありますね．特にディスクまわりの部分で，わりとそういうのが多い．われわれソフト屋の立場からいうと，ハードというのは本来事故が起ってあたり前のようなものですから，それをソフトがどれだけカバーしているかということがむしろきめ手になるという気がします．

司会　いまのディスクまわりの問題といわれたのはどういうことですか．

山本　ディスクの VTOC (Volume Table of Contents) といいますか，ディレクトリーというんですか，要するに情報の見出し，そういうものが読めないことがよくあるんです．とくに TSS とかオンラインの場合に重大です．実際には中の情報がこわれるわけじゃないんで，ちゃんと書かれているんです．ところが見出しのところが読めないもんですから紐がちょん切れたようになって，結局は中が壊れているのと同じことになる．そういう場合，たいていディスクのハード的なエラーがからんでいるんですけど，それをソフト的にカバーするようになってないわけです．たとえばまず最初に考えられる方法は，ディレクトリーを二重にしておけばいいわけですね．そうしておけば 70% ないし 80% ぐらいは救えるだろうと思うんですよね．そういう配慮が TSS では少し欠けているんじゃないかと思います．

ソフトの問題

司会　次は南条さんどうぞ．

南条　われわれユーザ・サイドとして保守の人達に接して，一番面白くないのは縄張り争いです．ソフトの部分とハードの部分ですよね．それも，はっきり割り切ってくれればいいんですけれども，『場合によったらソフトじゃないかと思う』とか(笑い)，『ちょっと温度が高いようですけれども，これが，ことによったら影響を』なんていわれるから具合い悪い．『どうすればいいですか，空調機をもう1台入れればいいですか』というと，『そこまで言ってるわけじゃないんです』と，どうも欲求不満ですね．われわれのほうとしては，そういった問題があると困るもんですから，毎月定例的にユーザ連絡会をやっています．これは保守の部門とシステムの部門と営業から出てもらって，三者合わせてやるんですけど，あくまでも他人で片づけちゃおうというような気配が見える．ユーザ側としては，ハードとソフトを区別する必要性というのはなにもないわけです．トータルとして機能を発揮してくれればいい．ソフトとハードウェアの区別をつけてくれということは迷惑しごくな話なんですね．ところが，どこでもソフトとハードの責任区分がついてまわる．これはぜひひとつ統合指令部を作っていただいて，ユーザをあんまり迷わせないようにしてもらいたいものです．

山本　考えてみますと，ただハードとソフトという二つだけじゃなくて，ハードの方だけでもたくさんの部門にわかれてまして，CPU はわかるけれども，I/O (入出力機器) はわからないとか，普通の I/O はわかるんだけれども端末になるとわからないとか．ソフトの方もそうですね．OS 内の一部のデータ管理関係はわかるけれども，あとはわからない，ランゲージになったらますますわからないとか，そういう点の分化が最近非常に進んで必然的にそうなっちゃったんだろうと思いますけどね．だからなんでもわかる人というのは多分いないんでしょうね．なにかトラブルが起きたときの全体的な判断が1人ではできない．いままでの例の中には，特殊なケースですけど，ハード屋さんが4人，ソフト屋さんが6人ですか，それでやっと原因がわかったなんてのもありましたね．結局ハードのトラブルがからんだ時の OS の中の虫だったんですけどね．だからそこへ行くまで大変なんですよ．さっき，修理時間が数分というお話がありましたが，数分なんてとんでもない．やっぱり一週間はまるまるかかります．

桑折　結局私なんかが見てますと，新しい機種を作られる際に，まずハードの設計ができて，それが大分固まってきたときに，今度は OS の設計が始まるという具合ですが，ハードの設計の初めの段階からソフトの障害対策について考えておく必要があると思います．

ガラス張りにしたら？

南条　先程から始終お話が出ているんですけれども，なんかトラブルが出ると，すぐ『再現してくれ』という話になる．再現というのは実に難しい(笑い)．

それで『どれぐらい時間が必要ですか』というと『わからない』というので困るわけですね．また，先程から機械が壊れるとか壊れないとかいうお話もあったんですけれども，機械を絶対壊れないものだなんて思っているユーザはいないと思うんです．一番困るのは，予測できないこと，そしてそのために計画が立たないことです．この機械は十中八九は壊れますよ，とか，1カ月に1時間は必ずダウンしますよ，とか言ってくれればいいわけなんですが，ダウンはしませんという話をまず言われるのは大体メーカーの方ですよね．うちの機械は信頼性は高いと言っておいて，あとで，おそるおそる，実はやっちゃったんですけれども，という話になるんです（笑い）．ですからどうもこのへんをもう少しガラス張りにした方がいいんじゃないかという気がしますね．ダウンするならするで，わかっているダウンなら少しもこわくない．あともう一つ，何かコンピュータというものを，あまりユーザ自身にさわらせたくないというような傾向があるように思うんですけれど，簡単なところはどんどんユーザにもさわらせて，メカ的なトラブルならば簡単にユーザでも直せるような形にできないかな．たとえば，私の会社は自動車を売っていますけれども，自動車というのはユーザに『指一本触れちゃ困ります』というような言い方はしませんよ．むしろ，定期点検の講習をやったり，メカを詳しく書いたものを配ったり，応急処置用のスライドを作って見せたり，いろいろやっているわけです．とは言っても，お客さんに全部押しつけるんじゃなしに，両方一体になってお互いに愛着をもてるような方法ができないかと思います．

遠藤　今後一つのシステムで数百台とか数千台の端末がばらまかれるようになったとき，ユーザに端末の保守の一部をやってもらうというのも一つの考え方かもしれませんね．

南条　車の場合には，たとえばどこか調子が悪いというときには，エンジニアとドライバーとの間でかなり対話があるわけですよ．雨の翌日はエンジンのかげんが悪いとか，あるいは坂道のときには力不足を感じるとか，乗っているときの状況をいろいろ対話できるわけですよね．ところがコンピュータの場合ですと『何かおかしいようですからよろしくお願いします』という程度のことしか言えない．だからもっとユーザの目を活用するといいと思うんです．RAS が発達しても，人間の目，状況判断をする目というのが必要ですから．

飛行機のコクピットがありますね．天井も壁も一面にメーターがついている．あれが憧れだという人間もいるわけですよ．いっそのことコンピュータもコンソールを操縦席みたいにして，その回りをメーター類で固めて，もっと運転を楽しむ形にもっていくのも一つの方向じゃないか．それでたとえばディスクがおかしいとなったらディスク関係のメーターを見るとかいうようにやっているうちに，習慣として，こうした場合にはどこのゲージがどうなるとか，そういったことがわかるようになると思うのですが．

司会　桑折さんの所ではどうでしょう．保守を大分一緒にやられているんじゃないんですか．

桑折　そうですね．ただ，昔の計算機はシステムもそれほど複雑でもなかったので，保守員にもハードの隅まで知っているベテランがいたわけですよ．なんか障害が起きてもそれこそパターン認識ができるような形になってたんです．けれども，これが計算機が大きくなってきますと，だんだん中身がわかりにくくなってくる．ですから計算機が大形化して複雑になると，それ以上に保守のしやすさを考えなければならないと思います．

山本　ちょっと違った話になりますが，私どものセンターには機種毎にたいてい1人ないし2人の保守員が常駐してくださっているんですけれども，その人達は障害が起ったとき，できるだけ自分で直そうと思われるわけですね．自分1人ですぐ直せる自信がなかったら，早く連絡してわかる人にきてもらって直してくれたほうがありがたいと思うんですけどね．それは勤務評定されてるんでしょうか（笑い）．だから気の毒と言えば気の毒ですし，努力する気持は買うんですけどね．そのへんの兼合いというのがなかなかやっぱり難しいですね．

菱輪　確かにカストマ・エンジニアは状況を判断して障害がどのへんにあるかを切り分けて，自分のところにいるスペシャリストの最も適切な人間に最短時間で連絡できるだけのノウハウをもたなければいけないのですが，現段階ではそれが限界となってきているんじゃないかと思うんですけど．自分の手に負えない場合に，今度いったい誰を呼んでいいのか，SE を呼んだらいいのか，PSCE (Programming Support Customer Engineer) と言ってますけど，プログラミング・システムの教育を受けているカストマ・エンジニアに連絡をしたらいいのかという判断ができない．そういった切り分けのテクニックやエンジニアの普通の

ノウハウがあれば比較的簡単に判断できるようなシステムがデザインされることが望ましいんじゃないでしょうか.

今後の課題

司会　これから大事になりそうなこととか，今までのお話で欠けていることで重要な問題点はありませんか.

山本　最近大分いろんなペリフェラルが出てきたので，インターフェイスの問題がこれからいろいろ起るんじゃないかと思うんです．私の所では COM とか，あるいはマイクロ・フィルムをコンピュータでコントロールしていくとか，映像情報とか，そういう方面の計画をしてるんですが，インターフェイスの問題が出てくるような気がします．特に端末回りの信頼性が問題です．たとえば，リモートバッチ用の端末が何台かあるんですけれども，カードリーダーなんか非常に遅いくせに（1分間に100枚）けっこうジャムるんです．おまけに全然チェック機構がないんだそうです．だからスイスイはいって答が出てきて，これはカードのリードミスじゃあるまいかとあとから疑って初めてわかるような，そんな状態なんですね．そのへんの信頼性なんてのはまだまだなんですね．CPU というのはまあまあですが.

桑折　CPU というのはあんまりこわくないということは言えるんです．いま言われた端末の問題というのはメカの問題だと思うんです．たとえばドラムとディスクと比較した場合，ドラムはかなり信頼できますが，それに比べてディスクはまだこわいです．しかし今後ディスクがだんだん多く使われる傾向にありますから，とくにメカの部分の信頼性を重要視していただきたいし，そうでなければもう少し静的なものでそれに見合う機能をもったものを開発していただきたいと思うんです.

南条　自動車では封印エンジンというのがあるんですね．『このエンジンは2万キロ保証しますよ．2万キロ走ったら必ず持ってきてください．分解整備します』というぐあいです．コンピュータの保守も，そういった面があってもいいような気がしますね．今のコンピュータは保守を前提にして作られている．こわれたら直す，という前提でシステムが組まれているんですね．壊れたら捨てちまうんだぞという部分がもっと増えてもいいのではないでしょうか．一例を申し上げますと，ディスク関係で私どもで固定型のディスクを使ってるんですけど，固定型なのに蓋を開けて一枚一枚円盤を掃除できるようになってます．ちゃんと掃除する道具もついているんです．それもちょっと疑問と思うんですけど，なぜ掃除するかというと，ほこりが入るから掃除するわけで，なぜほこりが入るかというと，蓋を開けられるようになっているからですね．完全に密閉して真空にしちゃっておけば問題はないんだろうけど，そうすると，保守がつまらなくなるということでしょうね（笑い）．聞くところによるとコンピュータのハードウェアのコストは価格の 1/4 だそうですから，全部捨てて取り替えてもたいしたことはない.

山本　なんか事故起しまして，一所懸命直してくださったわけですよ．ところが使うほうはあと何分で直りますかと聞くんですよね．聞いてもしようがないとわかっていながら（笑い），保守員はカッカしてるんだろうと思うんですけどね．やっぱりユーザというのはそういうところはだめですね.

角田　確かに私どもはそう言われますと，なんとも言いようがないわけですけれども，ある程度見通しをつけるためには，全面的に保守員の腕に頼るのではむずかしい．それを機械化しますと，ある程度できるというしかけになると思うんですが，そのへんで，設計屋さんの方にあらかじめよく考えていただきたいと思っております.

司会　そんなところでよろしいですね．今日は有益なお話をいろいろお聞かせいただきまして，どうもありがとうございました.

（1972年6月9日開催，編集　戸川隼人）