SEAMAIL

Newsletter from Software Engineers Association

Volume 8, Number 4 September, 1993

目　次

編集部から		1
ISO 9000-3 をめぐって		2
ISO9000-3 はほんとうにソフトウェア品質の改善に役立つか?	松原 友夫	2
ソフトウェア工学の観点から見た ISO9001 および ISO9000-3 の適用の困難性についての分析	John Harauz	10
わが国情報産業のリストラクチャリング	大場 充	18
FA 機器からのぞいたソフトウェア	武田 淳男	20
あるパズル	玉井 哲雄	23
最近の Macintosh とそのユーザー会	野村 行憲	28
プログラマの本棚	山崎 利治	32
Unicode をめぐるディベイト	Ydoc People	36
Call for Papers/Participation		69
ソフトウェア・シンポジウム'94		69
Symposium on the Foundation of Software Engineering		71

ソフトウェア技術者協会
Software Engineers Asociation

ソフトウェア技術者協会(SEA)は，ソフトウェアハウス，コンピュータメーカ，計算センタ，エンドユーザ，大学，研究所など，それぞれ異なった環境に置かれているソフトウェア技術者または研究者が，そうした社会組織の壁を越えて，各自の経験や技術を自由に交流しあうための「場」として，1985年12月に設立されました．

その主な活動は，機関誌SEAMAILの発行，支部および研究分科会の運営，セミナー／ワークショップ／シンポジウムなどのイベントの開催，および内外の関係諸団体との交流です．発足当初約200人にすぎなかった会員数もその後飛躍的に増加し，現在，北は北海道から南は沖縄まで，900余名を越えるメンバーを擁するにいたりました．法人賛助会員も50社を数えます．支部は，東京以外に，関西，横浜，長野，名古屋，九州，東北の各地区で設立されており，その他の地域でも設立準備をしています．分科会は，東京，関西，名古屋で，それぞれいくつかが活動しており，その他の支部でも，月例会やフォーラムが定期的に開催されています．

「現在のソフトウェア界における最大の課題は，技術移転の促進である」といわれています．これまでわが国には，そのための適切な社会的メカニズムが欠けていたように思われます．SEAは，そうした欠落を補うべく，これからますます活発な活動を展開して行きたいと考えています．いままで日本にはなかったこの新しいプロフェッショナル・ソサイエティの発展のために，ぜひとも，あなたのお力を貸してください．

ＳＥＡＭＡＩＬ　Ｖｏｌ．８，Ｎｏ．４　　１９９３年９月３０日発行
編集人　岸田孝一
発行人　ソフトウェア技術者協会（ＳＥＡ）
〒１６０　東京都新宿区四谷３－１２　丸正ビル５Ｆ
TEL: 03-3356-1077　FAX: 03-3356-1072
印刷所　サンビルド印刷株式会社　　〒１６２　東京都新宿区築地町８番地
定価　１，０００円（禁無断転載）

編集部から

☆

今年は，台風と同じくSEAMAILも当り年です．いつもの2倍ほどに増ページしたVol.8, No.4をお届けします．

☆☆

巻頭には，松原友夫さんの力作ISO 9000-3批判論文を載せました．それに続くカナダからの意見書も，同じく松原さんの翻訳になるものです．来る10月26日(Mon)には，このホットな話題に関する月例Seminar & Forumも企画されています．ふるって御参加ください．

☆☆☆

前々号から連載されている大場さんの「SEAリストラ論」は，今回はちょっと視点を変えて，情報産業全体のリストラクチャリングを論じています．不況の折りから，会員のみなさんの周囲でもリストラの嵐が吹き荒れているものと思います．そうした実感にもとづく「異論·反論· Objection」がありましたら，どしどし編集部宛にお寄せください．

☆☆☆☆

その他，今月は4人の幹事の方々から原稿をいただきました．それぞれユニークな，そしてバラエティに富んだ読みごたえのある内容だと思います．お楽しみください．

☆☆☆☆☆

さて，そしてこの号の最後には，Ydoc(横浜支部)のみなさんがネットワーク上のmailimg listを使って展開したUnicodeに関する「朝まで生テレビ」スタイルの大激論を収録しました．次号には，やはり最近異常な盛り上がりを見せたSEA関西meiling listでのCSCW論議を載せる予定です．乞う，御期待!

☆☆☆☆☆☆

ISO9000-3 はほんとうにソフトウェア品質の改善に役立つか?

松原 友夫
(Office Peopleware)

1. はじめに

どんな規格にも問題がある．だが，それが何らかの理由で有効であると判断される状況においては，そのよし悪しに関わらず，われわれはそれに従わなくてはならない．しかし，規格を守ることとは別に，議論は大いに行う必要がある．

この小論は，ソフトウェアまたはソフトウェアを含むシステム (製品) の生産を業としている企業が，すでに制定されたISO9000シリーズの認定を受けようとする努力を，否定するものではない．ここでの目的は，カナダから提出されたISO9001とISO9000-3を批判した文書 (参考文献5) を参照しながら，この規格が生まれた背景，その基本思想に対する疑問，規格に内在する長所と欠点，それがもたらす品質への影響，などについて指摘し，あわせて，ソフトウェア開発者の視点からの提案を示すことにある．

こうした議論からは，直接的で即効性のある効果は期待できないが，運用面で規格の欠陥を補い，その効果を最大限に活かすと同時に，将来の規格の改善のために役立つものと確信する．

2. だれがISO9000-3を作ったのか?

もともとISO9000シリーズは，契約によって開発するハードウェア･システムの品質を保つための仕組みとして (これを品質システムという)，2者間の契約に盛り込むべき条件を規定した規格であり，ISO/TC 176という伝統的にハードウェアの品質専門家の集まりである標準化機関が作成してきた．ISO9001のソフトウェア版を作成するために，TC 176の中にISO TC 176/SC2/WG5が設けられたが，その委員構成も当然似たり寄ったりであったであろうことは，想像に難くない．

日本にTC 176/SC2/WG5に対応する委員会が編成されつつあったとき，委員候補者の中にソフトウェア専門家が少ないことを憂慮したISO/IEC JTC1/SC7(ソフトウェア工学関連の国際標準化担当) の国内対応委員会の委員長は，当時のSC 7の委員に，所属する企業や団体のソフトウェア専門家を，TC 176/SC2/WG5対応国内委員会に委員として参加させるよう，要請した．

当時私は，SC7に情報サービス産業協会代表という立場で参加していたので，この要請に答えて委員を出した．その結果，日本の国内委員会では，かなりソフトウェア専門家が補強されたが，参考文献1に，「純粋にソフトウェア開発に携わっている人はだれか?との質問に対して，明確に"Yes"と答えた人は少なかった．この事実から，ソフトウェア開発における品質保証を議論している人々が，必ずしもソフトウェア開発に携わったわけではないことが明らかとなり....」と述べられているように，国際委員会ではこうした補強は行われず，委員の大多数がソフトウェア開発経験のない人たち (おそらくハードウェア開発経験者) によって占められていたようである．

このように，ISO9000シリーズが，ソフトウェア工学関係の国際規格を担当するISO/IEC JTC1/SC7とは別の機関で作られたことから，おそらく大部分のソフトウェア技術者は，つい最近までそのような規格の存在さえ知らなかったであろうし，知っていたとしても自分には無縁のことと思っていたに違いない．

したがって，ISO9001 - 9003と同様に，ISO9000-3は，ソフトウェア･コミュニティでは，あまり話題になることも，注目されることもないうちに，制定された．こうした経緯から，ソフトウェア･コミュニティでこれらの規格への関心が高まったのは，それの適用を受けなければならない事態に直面してからあとのことであった．

いまや，ソフトウェアは，既存のあらゆる産業分野の製品や業務に浸透している．そのため，すでにソフトウェアは，すべての産業に共通して含まれる要素であり技術となっている．こうした背景から，複数の標準化機関がソフトウェア関連規格を独自に作成する傾向があり，ほとんどの場合，それらの間での一貫性は

保たれない．ある規格が，それを制定した特定の産業分野内で使われているのなら，問題は少なく，ソフトウェア組織は単にそれを無視すればよい．

しかし，ソフトウェア組織は，次のいくつかの理由で，ISO9000 シリーズ規格に関心を持たざるを得ない状況に置かれるようになってきた．

その第１は，これがすべての国，すべての産業をカバーする国際規格であるということ．第２は，もともと，これらの規格が EC 経済圏の統合を目指して急いで制定された経緯から，認定制度とからんで，ソフトウェアを含むシステムに関する国際間のビジネスでは，ISO9000 シリーズ規格の規定を満たすことが必須の条件と見なされるにいたったこと．第３は，ソフトウェアに制御されるハードウェア･システムの増加によって，もともと，ハードウェア･システムのために作られた ISO9001 の認定対象に，ソフトウェアが否応なしに含まれるようになったこと．第４は，ISO9001 を無理に解釈してソフトウェアに当てはめるのを避けるために，明確にソフトウェアを対象をとした規格 ISO9000-3 が，ISO9001 をベースにして作成され制定されたことである．

3. 基本的にハードウェア指向の ISO9000-3

すでに，ISO9000-3 が，ソフトウェアの開発経験を持たない委員が多くを占める委員会で，作られたという背景について述べた．

ISO9000-3 は，ISO9001 のソフトウェア版であるから，たしかに，多くの点でソフトウェアの特性が配慮されているように見える．

たとえば，いままで，いろいろなソフトウェア･ライフサイクル･モデルが提案され，使われてきたことを踏まえて，特定のライフサイクル･モデルに依存しないことを述べていること (5.1)，要求仕様の確定が品質に多大の影響を与える商用特注ソフトウェアで特に重要な，購入者の管理責任を規定していること (5.3)，契約レビューの項で，購入者が契約上の義務を果す能力を保有すべきこと (5.2.1)，購入者要求仕様についての節が新たに設けられていること (5.3)，テストおよび妥当性確認 (5.7)，構成管理 (6.1) など，ソフトウェアに適した節が設けられていることなど，数多くの配慮が見られる．

また，このような一般性の高い規格では，解釈の幅を広げることによって，容易に多少の不適合性を吸収しうるし，認定監査のプロセスでは，規格と現実のプロセスのマッピングを明確に定義することにより，規格とのバリエイションを認めている．したがって，現実の認定プロセスや規格の適用には，それほど支障をきたさないであろう．

しかし，この規格が，ISO9001 のソフトウェア版を作るという枠組みの中で作られたという経緯から，やむを得ないことではあるが，その底流には，無形物としてのソフトウェアの品質を，あたかも有形物のように，責任と義務を定め，それにしたがってプロダクトの管理を強化することによって，トップダウン的に実現しようとする意図を，明確に読み取ることができる．

4. ハードウェア指向ソフトウェア規格に対する疑問

こうしたことがなぜ問題なのか？ ソフトウェアは，工業製品として開発すべきなのだから，ハードウェアと同じように扱ってもよいのではないか？ ソフトウェア･コミュニティの中でさえ，こうした考えを持つ人は多い．表面的には，開発プロセス，品質や生産性といった開発に伴うプロパティ，使用する用語，などは共通している．ソフトウェア産業の初期にあっては，たしかにハードウェアから多くの技術を受け継いできた．

アメリカ上院の商業･科学および輸送に関する委員会が，1991 年 11 月 13 日に開催した「米国のソフトウェア産業の国際競競争力についての聴聞会」で，Laszlo Belady (Mitubishi Electric Research Laboratories, Inc.) は，証人の１人として次のように述べている：

「ソフトウェアにおいて，いま何が問題なのでしょうか？ 私は少し哲学的に述べてみたいと思います．ソフトウェアは，(これまで人類が出あった) まったく新しいもの(製品) です．それは，少し変わった性質を持っています．あるときは工業製品のように見え，またあるときは本や調理法に似ていますが，われわれは現在のすべての文化をもってしても，ソフトウェアのこの一風変わった性格，つまり法律システムのように有形でないもの，にうまく対処できないでいるのです．」(参考文献 2).

かつてハードウェア生産にかかわった人たちは，

Belady がいうように，概してソフトウェアを有形のハードウェア工業製品類似のものと見る傾向があり，事実そのように扱っても，あまり問題が起こらない製品分野が存在する．この他にも，それを本のような著作物と見る人，料理法の類と見る人，法律のようなルール集と見る人，創造的なアイディアの結晶と見る人，チームによる共同作業の成果と見る人(これはまだまともな方で，人を集めて即席でプログラマーに仕立てて顧客に派遣し，派遣費と人件費の差額を稼ぐビジネス見ている人さえいる)などなど，ソフトウェアには，実にさまざまな見方が存在する．

これらはいずれもソフトウェアの一面のみを表わしてはいるが，全体を適切にいい表わしているわけではない．これらの見方を，単にソフトウェアの一部の特徴を強調するための比喩として使うのならよいが，現実には，ソフトウェア･ビジネスの中にいる人でさえ，自分が思い込んだ見方だけがソフトウェアの特性だと勘違いしている場合がある．ソフトウェア･コミュニティでよく見られる管理者と担当者，顧客と供給者，中高年と若年間の著しい意識ギャップは，多分にこれが原因となっているように思われる．

有形物のコミュニティでは，こうしたプロダクトに対する意識ギャップは起こりにくい．なぜなら，具体物の介在が共通の理解を助け，誤解を矯正する働きを持つからである．ソフトウェアに対する最も典型的な偏見は，さきに挙げた「ハードウェアと同じように，トップダウン的に管理すればうまく行くはずだ!」という考え方であろう．この考えを持つ人には，ソフトウェア未成熟論者が多い．

これは，いままでの産業がほとんどそうであったという歴史的な事実，また，工業製品としての資質が要求されるということ，古手の管理者の多くがハードウェア出身であること，両者の経験を通しての比較論的な見方ができる人が少ないこと，さらに，現実にハードウェアの一部として埋め込まれるソフトウェアがかなりあるということ，などから当然の帰結といえよう．

事実，ソフトウェア産業の初期には，ウォーターフォール型開発プロセス，メトリックス，品質管理など，ソフトウェアはハードウェア生産から多くの思想を受け継いだ．しかし，その功罪は相半ばするといったところだろう．私がかつて9年間機械工場に勤めていた経験から，ソフトウェアをハードウェアと同様に扱えるという考えには大きな誤りがあることを，いくつかの具体例で示そう．

たとえば，ハードウェアの場合にも設計不良や設計変更による作り直しがあるが，それは例外的な扱いで，作り直しのための作業手配一件に対して赤色の伝票が一枚発行され，そのコストは仕損費として計上される．ソフトウェアでそんなことをしたら伝票発行だけで日が暮れてしまうだろう．,sp ,5

従来型のハードウェア生産では，プロセスのほとんどは加工作業であり，およその流れは，素材から部品の加工･サブアセンブリから組み立てへとほぼ段階的に進み，プロセスの結果はプロダクト上にただちに目に見える形で反映されるから，中間プロダクトを含むプロダクトのテストや，工作機械の定期的なチェックをしっかり実施するための仕組みを設けて，それを確実に管理し，例外的な問題の処理を確実に実施すれば，製品の品質は維持できる．欠陥は容易に見つかるから，欠陥品を次のプロセスに送ることはあまりない．

したがって，ハードウェア･システムでは，テストは基本的にまた伝統的にプロダクト中心であるし，またそれで十分品質を確保できることが多い．機械類の生産では，たとえば高精度の加工(超仕上げ)を要する嵌合面，力がかかり磨耗や発熱が生じる駆動部分，振動を起こす可能性のある部分といったクリティカルな部分が，視覚的に構造的に自明であるので，図面チェックやテストもまた，重点的に効率よく行える．

かつてハードウェア･システムと見なされた製品において，その中に埋め込まれているソフトウェアが次第に肥大し，ソフトウェアによって制御されるシステムに変貌したときには，このプロダクト指向のテスト依存が深刻な問題を引き起こすことがある．

たとえば，多数の放射線被曝による被害者を出して訴訟問題となった有名な医療用電子加速機 Therac-25 の事故例(参考文献 3)では，初期のハードウェア集約的でハードウェア･インタインタロック装置を持つ Therac-6 や Therac-20 でのハードウェア中心のテストにならって，その装置をソフトウェア機能に置き換えた Therac-25 でも「... 単体およびソフトウェアのテストは最小限しか行われず，大部分の労力は統合システムテストに費やされた」．つまり，これは，ソフトウェアを含むシステムの品質が，プロダクト中心では十分

保証できないことを示す例である．

ソフトウェアでは，プロダクトが無形であるために，できばえが直感的に把握できないから，開発プロセスは決して段階的には進まないし，プロダクト中心のテストだけでは，欠陥に気づかないまま，プロダクトが次々と下流のプロセスへ送られる可能性が高い．また，ソフトウェアでは，重点的なチェックがハードウェアよりはるかに困難で，それをハードウェア用語にたとえていえば，全体が「超仕上げ」のようなものなのである．

1ビット1命令のチェックもおろそかにできないから，ハードウェアではベテラン設計者のひと睨みで済む図面チェックが，ソフトウェアのコード･レビューをまじめにやれば，それを書いた時より時間がかかってしまう．現実に，アメリカで社会的なパニックを起こした1990年1月15日の9時間の電話網停止という大事故の原因は，たった3ビットのフラグビットの戻し忘れであった(参考文献4)．このほかにも，重大事故の原因が1ビットか1命令であったケースは数多い．

誤解を避けるために，ここで使っているハードウェアという言葉について，注釈を加えておきたい．ここでいうハードウェア製品とは，期間的･コスト的に製造プロセスがほとんどを占める従来型の有形工業製品を指すが，最近のハードウェアの中には，たとえばコンピュータやマイクロプロッセッサ･チップのように，これらが本来持っている複雑性のために，設計プロセスが肥大化しソフトウエア化している製品分野がある．Beladyが，前出のヒヤリングで：

> 「マイクロチップを見てください．それは，本質的にソフトウェアの設計という膨大な知的作業によって具体化されたものなのに，多くの人はそのチップを眺めて,"おお，これこそハードウェア産業の成果だ!" といっているのです!」

と述べているように，そこでは，設計者たちはコンパイラ言語でプログラムを書き，ワークステーション上でツールやネットワークを駆使して，まさにソフトウェア･プロセスを実施している．

この小論では，こうしたソフトウェア集約的なプロセスは，たとえそれがハードウェア･プロセスの一部であったとしても，それをソフトウェアとして考える．つまり，ここでいうハードウェアとは，製造集約的な製品だけを指している．

ソフトウェアに対する偏見は，主として過去の経験や，現在携わっているソフトウェアのタイプから生まれる．埋め込みソフトウェアを製品の一部として含む生産会社や，ハードウエアの品質管理に関わってきた人たちは，さきに述べたように，概してソフトウェアをハードウェア生産の延長と見る傾向が強い．たびたびヨーロッパを訪ねた経験から，私は，ヨーロッパのソフトウェアの主流は，要求仕様やインタフェイス仕様が比較的安定した埋め込み型であることを知っている．しかも，TC 176の委員の過半数はヨーロッパの人たちである．こうした背景から，TC 176/SC2/WG5の委員にソフトウェアの開発を経験した人がいたとしても，それはハードウェアの性格に比較的近い埋め込み型ソフトウェアの経験者であった可能性が高い．

ハードウェア工場の経験から出発した私は，ハードウェア生産メンタリティがもたらす数多くの深刻な問題を経験してきた．TC 176でISO9001のソフトウェア版，つまりISO9000-3の審議が開始されたときから，私は，「ハードウェアの品質管理の延長で，はたしてソフトウェアの品質がよくなるのだろうか?」という疑問を抱いていた．その後，にわかに世の中の関心がISO9000の認定に集まり，多くのソフトウェア関係者までが，ISO9000の認定をソフトウェア品質改善の決め手であると錯覚するようになると，私の懸念はしだいに高まってきた．しかし，それはあくまで私個人の意見であり，実証する機会もないまま心の内に秘めておいた．

5. カナダから提出された批判文書(根底に潜む問題

私と同じ疑問を抱く人が他にもいることを知ったのは，今年の5月に東京で開催されたISO/IEC JTC1/SC7の総会に出席し，そこに提出されたある公式文書を手にしたときであった．それは，カナダがSC7国際標準化会議に公式に提出した「ISO9000およびISO9000-3に対する批判文書」である．提出者のPeter Voldnerと作成者J. Harauzは，これをSEAMAILのために翻訳紹介したい，という私の申し出を正式文書で快く承諾してくれたので，その全文をこの論文の後半に翻訳掲載した(参考文献5)．なお，Peter Voldnerは，SC7からISO/TC 176/SC2/WG5への連絡委員である．

実際にISO9001とISO9000-3を読めば，カナダの文書が指摘している問題点，たとえば，特定のライフサイクル·モデルに依存しないといいながら，現実には古典的なライフサイクル·モデルに依存していること，継続的な改善を指向していないこと，全員参加による自発的な改善を指向していないこと，評価の枠組みが存在しないこと，品質システムの手直しが十分考慮されていないこと，および，組織内の管理に指向していて顧客を含んだ視点に欠けること，などにすぐに気づくであろう．

さらに，これらの規格文書をよく読むと，規格の根底に潜むいくつかの思想が読み取れる．

その第1は，一部で購入者責任(これは日本からの提案で採用された)や共同レビューに言及しているものの，基本的には，システムの品質確保は供給者の問題だと考えているように見えることである．

対象が埋め込み型のソフトウェアならそう考えるのもわかるし，ソフトウェアが購入者との直接的な接点を持たないシステムでは，購入者責任をうたうのはむしろ不適切であろう(これについては，ある認定監査人は，購入者を社内の発注部門に拡張解釈できるという考えを示したようである)．しかし，「要求仕様をまとめる過程で購入者と供給者が共に学ぶ(参考文献6)」，つまり，開発しながら共同して要求仕様を決めて行くタイプのシステムでは，長期にわたる(分析の開始から開発の完了，システムの保守にいたるまでの)両者間の緊密なコミュニケーションによってのみ，システムの品質が確保されるのである．

第2は，トップダウンの責任体制にもとづく管理の締めつけによって，品質を確保しようとしているように見えることである．これは，生産段階がはっきり分割でき，責任範囲が限定しやすいハードウェア生産では適切であるし，そこでは伝統的なやり方である．もちろん，ソフトウェアにおいても，経営者責任の項目にあるようなトップダウン·コミットメントは重要であるし，ある程度の統制的な管理も必要である．しかし，生産段階や責任範囲が不分明でできばも見えにくく，結果が作業者の心理状態に左右されやすいソフトウェアでは，作業者個人やチームの自発的な協調やルールの遵守意識といったボトムアップの努力がなければ，十分な品質は確保できないし，かれらにやる気を起こさせるには，作業の自由度は制限するよりも，むしろ，それを拡大した上で自らを律するやり方のほうがうまく行く．管理強化を促進する傾向のあるこの規格は，文書上に書かれることと真実の乖離をもたらすだけでなく，ソフトウェア技術者のやる気を阻害する危険がある．

第3は，この規格が，暗に膨大な公式文書の作成を要求していることである．ハードウェアの場合，生産のためには図面，部品表，手配伝票があればよい．管理のための文書は，通常すでに確立された専任の現場を支援するスタッフ部門が作成するので，設計，製造といった本来の仕事を阻害することはない．しかし，ソフトウェアの場合には，開発作業そのものが膨大な論理的な文書作成作業であるから，その上に文書作成を強いることは，大きな負担になるだけでなく，開発作業自体をおろそかにする危険がある．かといって，これを専任管理部署にやらせると，コスト高になるのことが避けられない．

6. ソフトウェアをソフトウェアとして見る

では答えはどこにあるのか? これまで述べてきたように，われわれはソフトウェアが「無形の工業製品である」という厄介な特性を持つということを，もっともっと理解した上で，その改善について考えなければならない．いいかえれば，ソフトウェアをハードウェアの延長としてでなく，ソフトウェアそのものとして見つめ直さなければならない．そのために最初にやるべきことは，ソフトウェアの特性を知ることである．

1985年1月に，ミュンヘンのISO/IEC JTC1/SC7/WG3で，ソフトウェア品質特性についての実質的な議論が始まった．関連資料としていくつかの文書が提出されたが，議論は，事実上白紙の状態からスタートした．考えられる品質特性を1つずつ紙の小片に書き，それらをグルーピングしながら壁に張りつけて行った．

こうしてできあがった最初の文書をもとに，6年間におよぶ世界のソフトウェアのエキスパートたちの議論の積み重ねの成果として，1991年12月に国際規格として出版されたのが，カナダの文書にもしばしば引用されているISO/IEC 9126: Information Technology — Software Product Evaluation — Quality Characteristics and Guidelines for Their Use (参考文献6)である．

そこに規定されているトップレベルのソフトウェア品質特性は，次の６つである：

－機能性 (Functionality)
－信頼性 (Reliability)
－可用性 (Usability)
－効率性 (Efficiency)
－保守性 (Maintanability)
－移植性 (Portability)

この規格には，その他に，ソフトウェア品質特性の利用の仕方として，ソフトウェア品質の見方とソフトウェアの評価プロセス･モデルが示されている．現在では，この規格が１つの確固とした枠組みとして使われており，これをもとにして，ソフトウェアの品質関連規格は，品質の副特性分解や，ソフトウェア･プロダクトの評価など，さまざまな方向へと展開されつつある．しかし，これはマクロなソフトウェア品質評価という視点から見たものであって，問題をソフトウェア品質評価に限っても，より実践的な問題，たとえば，どのような性質をどんな方法で測定しランクづけしたらよいか? といった問題は，今後に残されている．

ハードウェアとソフトウェアの改善のポイントは，見かけは似ていてもその重点がまったく異なる．再びBeladyの証言を引用しよう (参考文献1)．

「問題はソフトウェアの無形性という本質であり，そのために，ソフトウェアを１ステップづつ作るときに，測り知れない困難を伴うのです.」

ソフトウェアの開発は，それにどんなに新しい技術が適用されようと，本質的に，人間が細かいステップを１つ１つ書いたり，テストしたりしなければならない知的な作業である．しかも，それは大勢の人の共同作業に頼らなければならない．ソフトウェア開発においては，個人の能力･意識･やる気･気分，あるいはチームの結束といったことがらが，品質や生産性を大きく左右する．したがって，人間の教育，たとえば顧客と開発者，管理者と担当者といった人間どうしのコミュニケーションや，開発者が作業する環境などが，きわめて重要になる．つまり，ソフトウェアの場合，「プロダクトの品質を改善するためには，まずプロセスを改善する必要がある」ということになる．

ソフトウェアが無形物であることから，完成品のイメージが描きにくく，しばしばあいまいな要求仕様，不適切な要求仕様，または仕様の誤解を生じる．有形物の生産では，モノが段階的に形を成していくから，ウォーターフォールの生産過程が自然であるが，プロセスが縦横に飛躍し，実際に走らせるまで欠陥が潜在する傾向があるソフトウェアに対しては，それはなじまない．

ハードウェアでは直感的に目に見えるために省略できることが，ソフトウェアの場合には，省略できないばかりでなく，ハードウェアよりはるかに多角的な視点から見た上で総合評価しないと，実際にどうなっているかが見えてこない．ソフトウェアを目に見えるようにするためには，プロダクトとプロセスの両面からのメトリックス制度を確立するという，きわめて大きな努力が必要となる．

人が１つ１つのステップを共同で作り上げる作業では，個々の作業者の自覚や自主性に訴えるやり方の方が，トップダウンの管理強化よりもはるかに効き目がある．ソフトウェアにおいて，プロダクトよりプロセスが重視される理由は，ここにもある．もちろん，ソフトウェア･プロセスの組織的な管理を完全に否定するわけではないが，それは上意下達的な管理をむしろ最小限にして，できる限りルールをツールに組み込むと同時に，自主的な管理に委ねるという形で実現すべきである．

品質システムをこの視点から考えると，理想的な品質システムは，開発プロセスを統合ツール環境の下で行えるようにし，それを通して自らのプロセスの品質レベルを自らが知り，自らが改善するような仕組みであろう．理想とまで行かなくても，前述のようなプロセスを主体としたメトリックス制度は，ソフトウェアでは必須である．しかし，ISO9000-3規格では，品質の測定 (6.4) について述べてはいるものの，それはプロダクトについてのみである．

ソフトウェアで有効なのは，人間的要因とプロセスと可視性に着目した，もっと総合的で基本的で血の通った対策である．したがって，継続的改善，全員参加による品質管理，品質システムそれ自体の手直しによる改善，および顧客からの視点といったことを含む総合的な品質改善策であるTQCは，カナダの文書が指摘しているように，明らかにハードウェアとは異なる優先度で，ソフトウェアの品質システムに組み込む必要がある．また，ソフトウェア技術者間で，いま見

つけたばかりの欠陥を題材にして動機的原因を追求し，再発防止策を考えるという形の原因分析も，品質システムに組み込むべき重要な要素である．

カナダの文書が指摘しているように，ハードウェアとソフトウェアによってシステムが構成される以上，両者の品質システムの一貫性とハーモナイズは必須である．いままで，両者の違いを強調してきたが，前述のように，ハードウェア的な見方ができるということは，ソフトウェアにそれとの共通点が多いことを示す．たとえば，どちらにも機能があり構造がある．品質や生産性の考え方も共通点が多い．ツールの効果的な利用は，むしろハードウェアから学んだものである．ソフトウェアをソフトウェアとして見つめ直すということの中には，本質的に類似しているところと異なるところの正しい認識が含まれる．そこまでさかのぼらないと，両者の品質システムを統合することはできないだろう．

すでに，根底に潜む問題に関して述べたように，ソフトウェアが頭脳労働集約的であることから，いままでハードウェアで成功してきた上からの管理中心のやり方は，ソフトウェアの場合には，はかえって人間の品質改善意欲を削ぐことになりかねないという問題を内在している．これから必要なことは，人間的要因の視点から見た品質改善策を考えることである．その考えを推し進めていくと，はたして規格によってソフトウェアの品質が改善されるだろうか? という根本な疑問に行き着く．

私は，個人的には，近い将来，「すべての規格がツールにインプリメントされ，ツールを使って仕事をすれば，自然に，規格を守って仕事をしていることになる」というような環境が出てくることを期待している．それが実現すれば，それこそソフトウェア開発者にふさわしい品質システムということができよう．

いま，われわれはソフトウェア品質改善に苦しんでいるが，少し見方を変えると，明るい未来展望が開けてくる．ハードウェアの労力やコストの配分は，特に歴史のある確立した産業分野では，製造段階に重点がある．それと全く異なるのがソフトウェアであって，そこでは，ハードウェアでいう製造はほとんどゼロに近く，大部分の工程は，ハードウェアでは手作業が主体で改善が遅れている設計作業の品質改善やあ機械化に取り組んでいる．

ハードウェア生産至上主義者は，ソフトウェア開発を未成熟だといって軽蔑するが，かれらは，われわれが設計品質の改善という最も困難な仕事に取り組んでいることに気づいていない．われわれが，もし設計プロセスでの困難な問題を解決すれば，それはハードウェア設計の改善にも役立つはずである．これは決して架空の夢物語ではなく，すでにそうしたことが，コンピュータ，マイクロチップ，またはそのたのエレクトロニクス機器の設計において実現されているし，そこでは，一部にソフトウェアから借りてきた技術も使われている．約1年半ほど前，IEEEの機関誌IEEE Spectrumの Concurrent Engineering 特集に，ハードウェアの開発期間を劇的に短縮した例が紹介されていた．驚いたことに，そこで必須の環境やツールとして紹介されていたのは，何とわれわれにとってはなじみの深いネットワークとCASEツール (その例ではStateMate) の利用であった．

7. 正しい道か，回り道か?

コンピューターに制御されたシステムが社会の基盤となった今日，ソフトウェアの品質改善は，われわれソフトウェア開発に携わる者が，自らの努力で取り組まなければならない最も重要な緊急の課題である．それは，基本的にソフトウェア·コミュニティに生きているわれわれ自身の努力によって獲得すべきものであって，他の産業でのやり方の借用や強制によって獲得されるべきものではない．仮に，他の産業でのやり方を用いなければならない場合でも，それらがソフトウェアの本質的な性格について，豊富な経験とそれに支えられた優れた研究にもとづく適切な解釈が与えられたものであれば，ソフトウェア産業にとっても意味があるだろう．しかし，もしこれが的外れなものであれば，品質は多少は改善されるかも知れないが，コストの増大を招き，あるいは最悪のケースでは，それが原因で混乱が生じ，結果として品質が低下する場合があるかもしれない．

品質改善のために，不適切な基準やルールがまかり通るということは，有形物を生産する産業では実証が容易であるために，ほとんど起こりえない．しかし，残念なことに，ソフトウェアは作業のほとんどを人の頭脳労働に頼る無形のプロダクトであるため，理論的あるいは実証的な根拠のない仮説にもとづいた基準が作られ，効果が実証されないままそれを適用すると

いうおかしな現象がしばしば起こる．たとえば，ある顧客の作業基準には，Goto を強制したり，決して読まれることのないドキュメントの大量作成を要求するようなものが，現実に存在する．

こういった現象を，取るに足らない問題としてかたづける訳にはいかない．歴史的に見ても，われわれはソフトウェアが無形物であるという単純な事実の重大性を見過ごしてきたために，たとえばウォーターフォール型プロセスのように，有形物の生産方式を単純に継承してきたことの誤りに長い間気づかなかった，という大きな誤りを犯した．ようやく最近になって，われわれはプロセスや人間的要因の重要性に気づき，それにもとづいたソフトウェアにふさわしいアプローチを考えるようになったのである．

しかし，ISO9001 および ISO9000-3 品質システム規格のソフトウェア組織への適用は，どうやら再び同じような大きな回り道をわれわれに強制するかのように見える．私は，この規格がソフトウェアの品質を改善するための有効な策だとは決して思わない．むしろ，問題を隠蔽する恐れさえあると考える．われわれは，これまで，ソフトウェア管理上の数値がよい組織が，ひどい品質のソフトウェアを出荷してきた例を数多く見てきた．無形のソフトウェアでは，よい管理数値を示すように装うことは，ハードウェアよりはるかに容易である．それがソフトウェアの本質なのである．

8. おわりに

ISO9000-3 規格は，すでに国際規格として承認されているので，ここに挙げたようなかなり強烈な批判があっても，国際的な取り引きを行っている企業は認定を受けなければならなくなるであろう．わずかに救われるのは，この規格は ISO9001 - 9003 とは異なり，手引き書としての位置づけであって，規定を強制する書き方にはなっていない (規格文面では shall でなく should が用いられている)．しかし，運用面ではこの規格が厳格に適用される恐れは十分ある．

私があえてカナダからの批判文書を翻訳し，私の意見を加えて公表した理由は，第 1 に，もし ISO9000-3 の遵守を強制されそうになった場合に，その適用を緩和したり，手直ししたり，または適用を拒否する理由を提供することにある．私個人としては，SC7 が進めている規格が完成するまで，ISO9000-3 は参考規格の位置づけであって欲しいと願っている．第 2 には，この機会にソフトウェア技術者の立場からソフトウェアの品質改善にはどんな規格が望ましいか，どんな実践的アプローチが必要かについて考え，よりソフトウェアに適した案を提案するきっかけになればと思ったからである．この小論がそのために役立てば幸いである．

参考文献

1. 飯塚悦功編，ソフトウェアの品質保証 ISO9000-3 対訳と解説，日本規格協会，1992，pp..147
2. Competitiveness of the U.S. Software Industry: Hearing before the Comm ittee on Commerce, Science, and Transportation United States Senate, U.S. Government Printing Office, Nov. 13, 1991.
3. Nancy G. Leveson, Clark S. Turner, An Investigation of the Therac - 25 Accidents, IEEE Computer, Vol.26, No.7 (July 1993), pp.18-41.
4. Vulnerability exposed in AT&T's 9-hour glitch, IEEE The Institute, Vol.14, No.3 (March 1990).
5. Harauz, John, CAC-JTC1SC7 / CAC-TC176 DRAFT CANADIAN POSITION PAPER, A nalysis of Implementation Difficulties with ISO9001 and ISO9000-3 Stand ards for Software Engineering, Revision 0.5, ISO/IEC JTC1/SC7/WG7 N1108, April 28, 1993.
6. Bill Curtis, Herb Krasner, and Neil Iscoe, A Field Study of the Softwa re Design Process for Large Systems, Communication of ACM, Vol.31, No.11 (November 1988).
7. ISO/IEC 9126: Information Technology - Software Product Evaluation - Q uality Characteristics and Guidelines for Their Use, Dec.15,1991.

ソフトウェア工学の観点からみた ISO 9001 および ISO 9000 － 3 の適用の困難性についての分析

John Harauz
(Ontario Hydro)

－ Revision 0.5: April 28,199 3 －

この文書は，SC7 関連の規格と ISO 9000 シリーズの規格の役割についての活発な議論をうながすために，カナダが JTC1/SC7 に提出したものである．

ソフトウェアの品質システムと認証に関して，現在ある種の混乱が存在する．規格の作成者は，この分野における TC176 と SC7 の相互に関連する役割を，より明確に理解する必要がある．産業界は，ソフトウェアの品質に関して，相互矛盾のない規格および総合的な視点を示すことを望んでいる．

この文書は，カナダの国家諮問委員会の TC176 と JTC1/SC7 関係の 2 人の委員長によって，共同で承認された．この共同の承認は，カナダのソフトウェア産業が，ソフトウェアの品質に関するより高度な一貫性と総合性の実現に，重大な関心を持っていることを示すものである．カナダは，開発および保守プロセスの期間におけるソフトウェア品質要求について，SC7 がより一貫した見解を規定し配布することにより，先導的な立場を取り得ると考え，またそうすべきであるということを提言する．

現在，SC7 では，WG7 におけるライフサイクル･プロセスの標準化，WG10 におけるプロセス･アセスメントの作業など，重要な規格文書の審議が行われている．しかし，ソフトウェア工学が，ISO 9000 との関係において，それらがどういう視点からの役割を持つかを示すために，より全体的な構造を明らかにする作業が必要である．これらの規格は，最低要求標準を示すという意味で支持されているが，ISO 9000 － 3 と同様の意図を持つ代替案が存在する．

P. Voldner

I. 分析結果

1. はじめに

この分析結果は，ISO 9000 および ISO 9000 － 3 規格を実施する際の困難性と，ソフトウェア工学的な品質システムを実施することの有用性に重点を置いて述べている．この分析結果は，主として Ontario Hydro の"Trillium Model" (参考文献 9)，IBM Germany での実施経験 (参考文献 6)，英国での実施経験 (参考文献 7)，および「国際規格とカナダ国家規格の適用から不合理な要素を除去しハーモナイズして欲しい」という (1993 年 2 月 26 日のカナダ･ソフトウェア品質フォーラムにおいて確認された) 緊急の要請にもとづいている．

急速な技術の変化，プロダクトの複雑性の増大，激しく変化する世界的な要因，コンピュータ技術利用の普及，および品質に対する考え方の進化によって，より実践的で理路整然とした，そしてより一貫した標準が，緊急に必要とされるようになってきた．コスト的に効率のよい解決策を得るために，過去において実施してきたやり方よりもはるかに詳細に，プロセスとプロダクトを分析し，評価することが，必要となってきた．また，システムに直接かかわるエンドユーザーからの要求を含んだ総合性も，考慮する必要が生じてきた．現在，多くの組織は，規模の縮小を迫られつつあるが，それでもなお，ユーザーの要求を満たし競合力の強い高品質プロダクトを生産することが，市場から求められている．そして，プロダクトを運用する場の重大性に見合った，コ

スト的に効率のよいやり方でプロダクトを供給することもまた，求められている．

ソフトウェア工学が，開発プロセスの基本的実施事項の遵守を通して，成熟した学問分野に進化し始めたのは，ごく最近のことであり，ソフトウェアにまつわる複雑性を克服するために，ソフトウェア工学は，他の学問分野から各種の技法を「借りて」こなければなければならなかった．数多くの，そして変化に富むソフトウェア工学関連規格を，SC7 が作成しつつあることが，こうした複雑性の証拠である．組織の品質システムを評価するための詳細な基準は，ソフトウェア品質を評価する基準と重複するので，ソフトウェア工学のコンセプトと，品質システムのコンセプトを，スケールアップまたはスケールダウンしても矛盾がないようにすることは，緊急の課題であり，それをやらなければ，規格の重複，規格の相互矛盾，および不適切な規格，といった状態が今後も続くであろう．

この分析結果は，ISO 9000 規格の構造と適用範囲が，急速に変化しつつあるソフトウェア技術と歩調を合わせた実践的な規格の作成をすでに妨げており，今後も妨げ続けるかも知れないという根本的な欠陥があることを示している．心にとどめるべき根本的な困難性は，ISO 9000 規格にもとづいた適切な品質管理システムが，それ自体では，ソフトウェア品質の改善を保証する条件を結果として十分満たしていないことである．少し範囲が違うが，同様の根本的な欠陥が，ISO 9000 規格を反映して作られたカナダ規格 CSA Z299 および CSA Q396 にも，存在する．

ここに，ISO 9000 規格が，品質のコンセプトや技術の進化によって，時代遅れになってしまったのではないか? また，ISO 9000 規格を品質システムに組み込んだ結果，特に小規模の組織では，もはやコスト的に効率のよいやり方は考えられなくなってしまったのではないか? という根本的な疑問を提出する．

2. ISO 9001 と ISO 9000 － 3 との関係

ISO 9000 シリーズは，従来からの，ハードウェア生産で用いられてきた品質管理のやり方を，規格文書にしたものである．その結果として，ソフトウェアおよび情報を主要なコンポーネントとして含む複雑なコンピュータ·システム製品を開発し，適格性を与え，保守する仕事に品質システムを組み込むためには，規格の解釈が必要となる．ISO 9000 － 3 は，ソフトウェアを開発し，供給し，保守する組織が適用できるように，ISO 9001 を解釈した規格である．

しかし，設計と製造段階については，ISO 9000 － 3 は，ISO 9001 とは基本的に異なったやり方で解釈した．こうした根本的な相違によって，組織が基礎とする品質システムの規格に依存した，異なる品質保証活動が必要となる．また，ISO 9000 － 3 は，IBM Germany での品質システムの実施体験が示すように，プロトタイピングやソフトウェア再利用といった技法と相容れない古典的なライフサイクル·モデルを，暗に前提としている．IBM Germany では，ISO 9001 にもとづいて品質システムを実施したが，なおかつ，ソフトウェア用の特別な解釈用手引書を作って，ISO 9000 － 3 との対照を示さなければならなかった．このことは，ISO 9000 － 3 が，ソフトウェア開発をカバーするために，ISO 9001 を「無理に当てはめ」たものであることを意味する．

ISO 9000 － 3 のみにもとづく品質プログラムの確立は十分ではないが，その理由は，ソフトウェア工学が，システム工学やハードウェア工学とともに，全体品質のシステム的な解決策を実現するために，総合的な観点から考慮しなければならないからである．現実に，もし，ある人が，ISO 9000 － 3 にもとづいてソフトウェア工学を実施したとしても，別のある人は，特にハードウェア工学向きにつくられた ISO 9001 にもとづいて，別の品質システムを実施しなければならないだろう．IBM Germany は，単一の統合された品質システムが直接的に ISO 9001 に対応しなければならないと考えたし，また，Ontario Hydro を含むカナダの企業も，同様に考えている．

オーストラリアは，ISO 9001 に反するソフトウェア品質プログラムのための国独自の規格およびガイドライン (参考文献 1 および 2) を作成したが，それは，ISO 9001 にもとづくハードウェアへの要求と並行して適用できるようになっているので，ISO 9000 － 3 の適用をバイパスすることができる．

異なる国々のいくつかの企業は，ISO 9001 との一貫性がないという理由で ISO 9000 － 3 の適用を避けている．その中の主な企業は，Electricite de France および Merlin Gerin of France (ここでは IEC 880 および AQAP 規格を適用している) である．

最後に，ISO 9001 の基本的な解釈は，ISO 9001 についてのセミナー，および監査人用チエックリストに，あるものを用いなければならないことになる．これはまた，ISO 9001 にもとづく品質システムの実施が，国または監査人固有の経験や解釈によって，多様に変化する可能性があることを意味している．

3. ISO 9001 および総合的品質管理 (TQM)

3.1. 継続的改善

ISO 9001 に準拠した品質システムは，企業の品質上の方針という目的を達成するために，組織の構造，プロセスの構造，責任分掌，手続きおよび資源へ組み込むものとして規定されている．それには，明示的にはシステム･レベル，暗示的にはプロダクトおよびプロセス･レベルでの包括的な監査システムによって，きっかけが与えられるようなフィードバック機構を含んでいる．これは，大部分がプロダクト監査，および従来型の顧客対供給者の契約関係における火消し作業のような，苦情解決を指向した手法である．

TQM の中核をなす組織全体の継続的な改善という視点は，ISO 9000 シリーズでは触れていないが，これは，この規格の重大な欠陥である．これが，われわれが"Trillium Model"の作業を開始した基本的な理由である．"Trillium Model"は，TQM モードにおけるプロダクトの視点を伴った品質成長モデルを指向している．TQM プログラムは，組織全体の競合力を高め，マーケット･シェアを増大させることを目的としている．これは，実証的なやり方を用いて無駄(すなわちやる気のない従業員は無駄な資源と見做される) を排除することにより達成される．

,供給者のソフトウェア工学を実施する能力を評価する手段として，独立監査から自己総合評価への重要なパラダイム･シフトが起こっている．総合評価は，独立監査がもたらすマイナスの観点を排除するため，またより客観的に行うために，供給者と共同で行われる．外部調達を指向した監査は，基本的に被監査人が監査に合格しようとする動機に限界がある．

"Trillim Model"，ソフトウェア工学研究所 (SEI) の能力成熟度モデル (CMM)，ImproveIT などにもとづいて提起され，採択された SC7 におけるプロセス･アセスメントの作業の一部として，監査人用チェックリストと類似のプロセス総合評価チェックリストが，ISO において標準化されつつある．

3.2. 全員参加による品質管理

ISO 9001 は，独立した品質保証およびコントロール機能を暗に要求する．最近の傾向は，組織内の中央集権的な品質保証機能から，TQM のコンセプトが示すように，組織のあらゆるレベルでの全員参加による品質の制御，保証，および実施へと向かっている．

IBM Germany は，最初に TQM のコンセプトの実施を規定し，その後それを ISO 9001 の構造上に展開した．このプロセスを実施している間に，かれらは，TQM の開発段階と独立した取り組み方と，ISO 9000 － 3 の開発段階に依存した要求との間に，相互矛盾があることを発見した．かれらはまた，全員参加による品質システムの実施は，従業員が継続的にプロセスや手順に挑戦することを可能にするものであり，革新や改善を確実に許容し促進するためには，それがどうしても必要であることを発見した．

3.3. 評価

IBM Germany での品質プログラムの実施は，研究所が行ったビジネス･プロセスの詳細な分析にもとづくものであった．その後，すべての外部との依存関係と派生物を含むプロセスのネットワークの記述によって，プロセス構造が規定された．プロセスは，さらにサブプロセスとタスク (アクティビティ) に分割された．それに続いて，専任者がプロセスとサブプロセスのオーナとして割り当てられたが，その仕事には，プロセスの最適

化のオーナシップ，および派生物と顧客から受領したものの内容と品質のインタフェイス定義が含まれる．個々のタスクの開始と終了基準を，タスクの詳細と妥当性確認のためのメカニズムと同様に規定したIBMのパラダイムにしたがって，すべてのプロセス，サブプロセス，およびタスクの文書化が実施された．このタスクの最少単位は重要で，このために，タスクやアクティビティは，管理上のオーバヘッドなしに，容易に変更することができる．プロセス最適化の取り組みの主要部分は，タスクの品質，サブプロセスの品質，品質データ，および顧客満足度の分析と評価からなる．

SC7は，評価プロセスが品質向上の要となることから，それが，高品質ソフトウェアを入手あるいは開発するために，最も重要なプロセスであると認識している．SC7/WG6は，ISO/IEC 9126に規定した一組の品質特性と評価プロセス･モデルにもとづいて，プロセスとプロダクトの双方に関連するソフトウェア品質評価の一般的な枠組みを規定した．この基本的なプロセスは，IBM Germanyの品質システムの実施，およびTQMに固有のものである．この評価の枠組みに該当する部分は，ISO 9001には存在しないが，現在SC7/WG6で規定作業中の一般的手引き(参考文献11)は，評価の対象を，ソフトウェア以外を対象とした品質システムに容易に拡張できるようになっている．

3.4. 品質システムの手直し

コストにからむ問題の一部として，ISO 9001が,品質システムの手直しを適切に考慮していないことが挙げられる．ISO 9002およびISO 9003は，製造業の状況においては，設計，製造，および最終審査が分離された作業であるとの認識にもとづいて，手直しを許している．この前提は，複雑なコンピューター･システムのための品質システムには，当てはまらない．このようなシステムでは，手直しは，プロダクトの各段階への適用にもとづかなければならないが，稼働中のプロダクトから，設計を分離して取り出すことは不可能である．このことは，監査人用チェックリストは，特定企業の必要性に応じて手直しされた品質システムの実施を認めるためには，評価基準の新たな解釈を用意しなければならいことを意味し，これは，結果として，認可プロセスのコスト増を招くだろう．

IBM Germanの経験によると，いくつかの企業が第三者パーティーによる認可に抵抗している．これらの企業は，ISO 9001または独立QA機能の観点に立脚した企業の品質マニュアルを持たなくても，適切なプロダクトの品質保証タスクや手順を現実に実施している，と主張している．このことは，こうした企業にISO 9001の資格認定を与えることはできない，ということを意味するのか? そしてさらに，そのことが，供給者からの高品質である可能性の高いプロダクトを拒否する格好の理由づけになりはしないか? このことは，供給者の能力やソフトウェア･プロダクトを総合評価し，資格認定する基準を見直しする必要があることを示している．

3.5. 品質保証および顧客からの視点

再確認したい最後の基本的な問題は,「品質保証」が有益であるといる直感を与えることに失敗したときは,「品質保証」は一体何を意味すのか? ということである．この場合の「品質保証」の実施は，文章化または暗に示されたユーザーの要求にプロダクトが合致するかどうかを検証するという，その基本的な意味から外れているように思える.「品質保証」を，単に第三者パーティーが実施する手順を遵守するための監査からなり，検証作業を次第に強化していくだけのものだ，と受け取っているようだ.「品質保証」は，そのプロダクトが顧客またはユーザーのすべての要求を満足していることを実際に示すために実施されるすべての検証，妥当性の確認，監査，および総合評価作業の全体を意味するものとして，表現されるべきである．

顧客視点の強調が欠けている点は，ISO 9000シリーズ規格の重大な欠陥である．I SO 9000の状況において，品質は仕様に対する適合を意味する．ここには，単にある顧客とその供給者という一対一の関係の存在しか感じられない．返品数を減らすことはコスト低減になるが，一方，高すぎる品質はコストを増大させる．品質システムにおいては，過剰品質は品質不十分と同様によくないことなのである．

TQMの状況においては，品質は，無駄の削減および(顧客の期待を優先した)顧客の満足度を意味する．このことは，サービスが不要になることに関係がある最少の欠陥で，正しい製品を最低のライフサイクル･コスト

で期限通りに納入することを意味する．顧客視点から見た全体としての「プロダクト」には，ハードウェア，ソフトウェア，文書類，教育，および支援サービスが含まれる．顧客からの視点はTQMの中心部分である．TQMは，事前行動モードでの顧客に指向した取り組みである．品質システムにおいて，ユーザーや顧客を巻き込んで解決策を考えることは，TQMを成功に導くために「必ずやらねばならないこと」である．顧客とのよい関係は，信頼にもとづいて確立され維持されなければならない．

ISO 9001およびISO 9000－3の欠陥の要約

(1) ISO 9000－3は，ISO 9001をソフトウェア用に解釈したものである．だが，ISO 9000－3規格にしたがったとしても，低品質のソフトウェアしか得られない可能性がある．ISO 9000-3は市場からの要求に答えて作成されたが，その時点ではSC7に対応する作業グループが存在しなかった．したがって，ISO 9000－3の適用可能性は，それが存在する現時点で見直す必要がある．

品質プログラムの実施が適切であるかどうかを判断するためには，監査人用のチェックリストの形式(すなわちTickIT)で，基準を追加する必要がある．

それぞれの監査人およびそれぞれの国で，ISO 9000－3にもとづいて信認される供給者のレベルが異なる．

(2) ISO 9000－3とISO 9001は，お互いに矛盾に満ちていることが指摘されている．このため，IBM Germanyは，最初にISO 9001を適用し，必要なときだけISO 9000－3を適用した．

このことは，ISO 9000－3がソフトウェア開発をカバーするために，ISO 9001を「無理に当て嵌め」たものであることを示す．

(3) ISO 9001は，ハードウェアとソフトウェアで構成されるシステムの品質要求を適切に分割していないので，その各々について，今後一貫した規格を作成することは可能である．これらはお互いに独立ではない．

補助的なハードウェアとソフトウェアの２つの品質規格は，システム全体の品質に対して，一貫性と関連性を持っていなければならない．

スエーデンの参照モデルは，システム開発の要求を，企業という状況の下で規定している(参考文献12)．これは，IBM GermanyがTQMを実施する際に準拠したパラダイムと類似していると考えられる．

(4) ISO 9000－3は，ISO 9001からレベルを１つ削除したが，これによって新たな解釈が必要になったために，わかりにくくなっている．現在進行中のSC7での作業と努力によって，ISO 9000－3は将来不要になるだろう．

IBM Germanyは，ISO 9000－3とISO 9001のライフサイクル･モデルが異なっているため，それら双方の条件を満たすためには，２つの品質プログラムを実施する必要があることを示した．ISO 9000－3(およびCSA Q 396)は，古典的なソフトウェア･ライフサイクル･モデルを強制している．SC7のライフサイクル･モデルは，そのような制約から独立である．

また，IBM Germanyは，TQMコンセプトに合致する品質プログラムの実施では，ISO 9001の条件を満たすことができないことを指摘した．Malcolm Baldridge AwardおよびEuropean Quality Awardの受賞条件を満たすことは，自動的にISO 9001の条件を満たすことにはならない．このことは，ISO 9001が現在の品質改善の実践からかけ離れていることを示すものである．

TQMプログラムを実施するためには，初めに実施のコンセプトを規定し，その後でISO 9001の遵守事項をマッピングによって確認しなければならない．

(5) CSA Q396およびISO 9000－3は，中央集権的なQA組織(たとえば独立品質機能)を広め，またはその必要を示唆している．この指摘は，IBM Germanyが支持している．これは，現在の傾向が，中央集権的な独立QA組織から分散する傾向にあるという真の必要性を満たしていないことを，再度指摘するものである．

(6) ISO 9001およびISO 9000－3は，「評価プロセス」を適切にカバーしていない．

プロダクトおよびプロセスの評価は，ソフトウェア工学では基本的なプロセスであり，TQMの重要な要素

である．

現在のSC7での作業は，ソフトウェアの評価の枠組みを，品質特性，メトリックス，プロダクト，プロセス，資源，および稼働環境でのプロダクトの影響，といった観点から規定している．この枠組みは，ソフトウェアについて規定しているが，それはソフトウェア以外を対象とする評価にも手引きとして使用できる．

この考え方は，ImproveIT, SEIプロセス成熟度モデル，およびTrilliumにもとづいたプロセス･アセスメントの標準化の方向によって支持されている．

IBM Germanyの品質プログラムにおける，詳細に規定された入出力，オーナー，およびプロセスの最適化基準を伴ったタスクとアクティビティの実施は，SC7/WG6の評価モデルでその妥当性が確認されたので，現在審議中の一般的手引き(参考文献11)に採用された．

検査(Verification)プロセスとしての開発作業実施中の評価とアセスメントは，独立監査機能を不必要性にしている．

(7) ISO 9001は，事業上の要求に合わせて修正できるような柔軟でコスト的に効率のよい品質システムの枠組みを提供していない．

TC 176における新たな作業は，今後システム，ハードウェア，ソフトウェア，およびサービスを扱おうとしている．このことは，それが十分上位のレベルでなされるのならよいが，しかし，そこでは，大規模企業と小規模の企業の違い，および重大なシステムとそうでないシステムへの適用の違いについて，はっきりした言及がなされるべきである．

(8) ISO 9001およびCSA Z299シリーズでは，品質システム実施上の要求事項を減らすという意味から，設計を別扱いにしている．このことは，次の場合は適切でない．

- きれいに分割して構成する必要のあるシステム
- ソフトウェアが埋め込まれたハードウェア(ASICSはハードウェア内のソフトウェアである)
- ハードウェア･プラットフォームとソフトウェア･プロダクトのシステムを統合する設計者

(9) CSA Q396は，重大なシステムとそうでないシステムへの適用の違いについて，およびこれらの場合における要求事項の修正について述べているので，ISO 9000－3より幾分ましである．どちらの規格も，ソフトウェア工学のプロセス要求またはプロダクト要求という観点からは不適切である．

規格が「適切に」フォローされているかどうかを検査するためには，解釈と第三者パーティによる検査が必要である．ソフトウェア･プロセスとプロダクトのための最低限の要求は，「適切である」という言葉に対して，客観的な意味を与えるやり方で規定されなければならない．なぜなら，それは，「すぐれた」ソフトウェア･プロセスを規定する上で，きわめて重要なこっとがらだからである．

(10) これらの規格に存在する根本的な矛盾点は，これらが実際にはシステム工学を念頭に置いて，システム統合とシステムの妥当性の確認を要求していることである．しかし，ハードウェア工学とソフトウェア工学とのインタフェイスとしてのシステム工学は，不適切であり首尾一貫性がない．

(11) IBM Germanyは，ISO 9001が，解釈の手引きなしには，ソフトウェア工学にもとづく品質システムの枠組みとして，直接実施できないことを指摘した．

実施に当たって問題があった領域は，次の通りである．

- 文書の管理
- 品質の監査
- 品質の記録
- 統計的な手法

(12) ISO 9001は，TQMコンセプトの観点から見ると限定された規格である．IBM Germanyは，革新的な考えを殺さないために，品質保証または品質管理(コントロール)ではなく，全員参加による品質管理(マネージメ

ント) システムにもとづく品質システムを実施した.

(13) ISO 9000 は, 従来型の顧客対供給者の契約関係という観点からのみ, 顧客に注目している. TQM の中心的な考え方である顧客対供給者のパートナとしての関係という観点は, ISO 9000 では重視されていない.

(14) ISO 9000 規格は, TQM の中心的なコンセプトである継続的な改善という視点には触れていない.

これが強調しているのは, 被監査人のモティベーションを基本的に制約する包括的な監査プログラムであるが, 監査に代わるもっと優れた技法と手法が存在する.

提案および勧告

(1) SC7 および／または TC 176 は, 異なるレベルで問題を別々に割り当て, 次のことに関係するインタフェイスを明確に規定するメカニズムを, 見いださなければならない.

- 開発, 調達, 保守, 生産のための異なる品質システムについて
- 管理と組織的な問題について
- 評価システムについて
- システム工学, ハードウェア工学, およびソフトウェア工学について
- 構成管理および変更制御に関する問題について

このことによって, SC7 の作業は, TC 176 の作業の下位に, 矛盾なくスケールダウンされることになるであろう. TC 176 は, 一般的なことだけを規定し, その他のところに特定したことについての規定は, 避けるべきである. こうしたことは, 双方の組織のビジネス·プランで取り扱うべきである. 一貫性を保つためには, 双方が緊密に連絡をとるべきである.

TC 176 と SC7 の間のインタフェイスは, プロダクト·プランの段階で確立できるだろう.

(2) TC 176 は, 複雑なシステム開発から保守まで, および他の産業での開発から製造までの, 品質プログラムのシステマティックな分割のための要求事項と, 品質プログラムの進化について言及した一般性の高いレベルでの, 統一的な企業全体の品質システムの要求事項を規定すべきである.

規格は, 開発者, 保守者, ユーザー, 評価者, 管理者, 購買者, 供給者, 生産者, 統合者, オペレータなどに共通する問題についての, 適切で首尾一貫した統合を確実にするために, いろいろな異なる見方を許容しなければならない.

こうした要求事項は, 複雑なシステムを対象とするシステム, ハードウェア, およびソフトウェア工学のコンセプトを統一したものであるべきだし, いろいろな異なるライフサイクル·モデル, たとえばプロトタイピングおよび再利用, を許容するものであるべきだ. また, これらは, 特注ソフトウェア, 既開発または市販ソフトウェア製品, 埋め込みソフトウェアを伴わないハードウェア, および埋め込みソフトウェアを伴うハードウェアといった, 相異なるものの混合から成るシステムをカバーすべきである.

要求事項は, QA 組織における変化, アセスメントに対するパラダイム·シフト, 品質保証のための新しいモデル, などを許容するものあるべきだ.

これらは, 品質プログラムを, コスト的に効率のよいやり方で実施することを可能にするために, おそらく企業の品質についての基本方針, 企業規模, および企業の事業運営方針といったことにもとづいた, 要求事項の修正基準を規定すべきである.

ハードウェアとソフトウェアに対して一貫したやり方で適用できるような, 適切なシステム定義を規定すべきである.

(3) TC 176 は, ISO 9000 － 3 を更新するための計画を, 注意深く見直すべきである. ISO 9000 － 3 は, システム的な解決策が要求されるところでは欠陥があり, またISO 9001 の手引きとしての役割は, 現在の SC7 のライフサイクル·プロセスの作業完了によって, それに置き換えられるであろう. しかし, TC 176 は, SC7 での規格が完成するまで, 解釈の支援を引き続き行うべきであろう.

(4) TC 176 は，プロセスとプロダクトの属性の評価を許容する SC7/WG6 の一般的手引きに示された評価モデルから，上方向にスケールを拡大した一般的な品質評価モデルを，かれらの作業の中に組み込むべきである．そうしておけば，TQM や継続的改善モデルを容易に実施できるようになるだろう．

(5) カナダの見解としては，カテゴリ別の評価基準の制約と，重複についての適切な許容範囲メカニズムを伴う調達の状況の下で，ISO 9000 規格をコスト的に効率のよいやり方で使用するためには，ポリシー文書と標準化されたチェックリストを規定して，それを組み込むべきである．実際的でコスト的に効率のよい修正可能なチェックリストを作成するベースとして，適当な様式は，おそらく TickIT と Trillum のチェックリストをマージするのがよいだろう．

このチェックリストは，ISO 9000 － 3 と ISO 9001 間の矛盾についての解釈に言及すべきである．

また，このチェックリストは，品質システムだけでは高品質プロダクトを保証するものではなく，その他の基準での評価についての手引きを与える必要がある，という問題を許容すべきである．

このチェックリストは，SC7 のプロセス･アセスメントの作業におけるカナダの寄書が，統一されたチェックリストとして役立つ可能性がある．

参考文献

1. Standards Australia, Australian Standard AS 3563.1, "Software quality management system, Part 1: Requirements", 1991.
2. Standards Australia, Australian Standard AS 3563.2, "Software quality management system, Part 2: Implementation Guide", 1991.
3. Canadian Standards Association, National Standard of Canada CAN/CSA-Q9000.3-92 (ISO 9000-3:1991) "Quality Management and Quality Assurance Standards - Part 3: Guideline for the Application of ISO 9001 to the Development, Supply and Maintenance of Software", February, 1992.
4. Canadian Standards Association, "Plus 8807: Comparison and Analysis of ISO 9000 and CSA Z299 Standards", March 1990.
5. L. D. Eicher, "Quality Management in the '90s: The ISO 9000 phenomena", Quality Forum, Volume 18, Number 2, pp. 74-79, June 1992.
6. M. Donie and W. Dette, "ISO 9000 Implementation in Software Development; Concept, Approach, and Experiences, IBM Germany.
7. G. M. Jennings, "ISO 9001/9002 - Use, Misuse and Abuse", Quality Forum, Volume 18, Number 1, pp. 33-35, March 1992.
8. TickIT, "Guide to Software Quality Management System Construction and Certification using EN 29001", Issue 2.0, 28 February 1992.
9. Bell Canada, "TRILLIUM: Telecom Software Product Development Capability Assessment Model", Draft 2.2, July 1992.
10. ISO/IEC(JTC1)-SC7/WG7 Committee Draft, "Information Technology - Software Life-Cycle Processes", 12 February 1993.
11. ISO/IEC(JTC1)-SC7/WG6 Working Draft 6/N-218, "Information Technology - Software Product Evaluation - General Guide", Version 4, March 1993.
12. SIS - Standardiseringskommission i Sverige, SIS Teknisk rapport TR 321, "Systems development reference model", September 1989.

わが国情報産業のリストラクチャリング

大場 充
(日本IBM)

情報システム需要は，組織規模の2乗に比例する．

システムの規模が大きくなると，その要素の数もそれに比例して増加する．要素の数が増加すると，要素間で交換される情報の量は，要素数の2乗に比例して増加する．これは，ネットワーク型の組織(システム)では，組織のなかの自分以外のすべての要素と情報交換しないと，調整ができなくなるためである．いま，要素数をnとすれば，その情報交換の量Iは，

$$I = n \times (n - 1)$$

で与えられる．すなわち，nの2乗に比例する．

要素間の調整を最小限にする完全な階層型の組織でも，情報交換の量は，組織の規模とともに増加する．ただし，その増加率は，ネットワーク型の組織に比較して，小さなものになる．要素数がnのとき，その情報交換の量Iは，

$$I = c \times \log n$$

で与えられ，要素数の対数に比例する．極端な場合，組織が10倍になれば，情報量は2倍になる．そのような階層型の組織の典型は，軍隊組織である．階層型の組織では，情報のフィードバックがかかりにくく，組織に機動性がなくなる．このため，現代の企業は，特にアメリカでは，ネットワーク型に近い構造になっている．

1980年代の半ばからさかんになった企業のダウンサイジングによって，米国巨大企業の規模は，1970年代のスケールメリットを目指した時代の規模のほぼ1/2になりつつある．また，1980年代の後半には，「スモール·イズ·ビューティフル」の流れに乗って，小規模な企業が数多く出現した．このため，米国企業の平均的規模は，従業員数で，1/3から1/4ぐらいに縮小したと予想される．

IBMやGMのような巨大企業も，いまや実態は小さな独立組織(企業)の集合体である．このダウンサイジングの潮流の影響で，米国の情報処理需要は，組織が階層型からネットワーク型に移行したことを加味しても，従来のほぼ1/2から1/3に縮小したと予想される．これが，今日の米国コンピュータ産業の構造不況(情報不況)のほんとうの原因である．

わが国では，企業のダウンサイジングは，今日にいたるまで，米国ほど表面化していない．これは，日本型経営の影響もあろうが，主としてバブル景気で，企業はダウンサイジングの必要性を感じていなかったことによるものであろう．その証拠として，1993年に入って，不況が本格化してきたことから，真剣にダウンサイジングを検討し始める企業も出てきている．日本IBMや富士通がそのよい例である．

ただし，日本IBMのリストラクチャリングも，最近のダウンサイジングも，もとをただせば，米国IBMからの余波で始まったもので，日本IBMの経営陣が主体的に始めたものではない．日本IBMの椎名会長は，米国IBMからのダウンサイジング勧告に対し，『日本の特殊事情』を理由に最後まで抵抗したと聞いている．かれの抵抗の結果，日本IBMのダウンサイジングの規模は，米国IBMの主張を20パーセント下回ったものとなった．

この事例からも理解できるように，日本の経営者の現状認識はきわめて甘かった．マクロ経済がどのように変化しつつあるのかを，まったく理解していなかったとしかいいようがない．これからの日本企業の経営者には，グローバル(世界的)なレベルでのマクロ経済の理解が要求されるようになろう．

かりに，そのような経営者が数多く出現したとすると，日本の企業にも真剣にダウンサイジングを考える経営者が出現する．現在の日本のダウンサイジングは，そのような戦略的なものではなく，後手に回った不景気対策の観が強い．ただ，その場合でも，米国の場合とは違って，すべての企業がタウンサイジングとはならないであろう．たとえば，ダウンサイジングとリストラクチャリングを，バランスをとって組み合わせる例が多くなろう．

そうなると，規模の差こそあれ，日本でも本格的な情報不況が発生する．各企業の情報処理需要は，1991年をベースラインとして，その半分ぐらいにまで縮小するのではないか？ ただしその場合，企業間での情報交換の需要が新たに発生するので，ネットワーク構築や整備に関する情報システム需要は増大する．また，ダウンサイジングやリストラクチャリングの手段として，機械化が必要になるケースも多くなろう．

そのような場合には，従来人手で行ってきた事務処理を，機械化する必要がでる．ただし，この機械化は，従来のような単純作業の機械化ではなく，従来はできなかった高度な知的作業の支援という側面が強くなる．したがって，従来の情報処理需要とは『質の異なる』情報処理需要が発生する．それでも，需要の量的な減少は，大きなものになるであろう．

では，日本の情報処理産業およびその技術者に与えられた課題は何か？ それは，『質的な変化』にほかならない．量から質への転換である．これからの需要が見込まれるネットワーク構築や新しいアプリケーションの開発も，『付加価値の高いサービス』の例だといえよう．他の企業にはない『ノウハウ』をベースにしたビジネスの展開が，成功のカギになる．そのため，『ノウハウ』の効率的な蓄積が，企業の競争力を左右するであろう．

つまり，『専門化』である．

また，単一の企業のノウハウでは解決できる問題がなくなるので，複数の企業がそれぞれのノウハウを持ち寄って問題を解決することが必要になる．そうなると，企業のネットワーク (物理的なネットワークではない!) が，企業の重要な知的資産となる．

各企業が専門化するのにともなって，技術者の専門化が要求されるようになろう．従来には考えられなかったような専門性も必要になってくる．ある意味では，『プログラム』から『システム』への移行である．『プログラミング』自体に対する需要よりも，その他の専門性に対する需要の方が強くなる．

これは，個々の技術者に対して，多方面にわたる知識·技術が要求されてくるという意味ではない．その程度の専門性では解決できないような専門性が要求されるのである．つまり，複数の超専門家が，それぞれの専門性を生かして協同作業を行なう必要性がでてくる．

チームの学際化である．

メンバーには，いわゆる『スーパー·プログラマ』もいるだろうが，コンピュータ·ネットワークの専門家もいる．心理学者もいる．社会学者もいる．数学者もいる．公認会計士もいる．そのようなきわめて多様性のあるチームでなければ，解決できない問題が多くなる．

『専門性』とは何か？ 突きつめれば，それは『ある特定分野についての深い知識とノウハウの蓄積』である．知識だけでは，問題は解決できない．どんな理論がどんなときに有効かを，経験にもとづづいて判断できなければ役に立たない．その意味で，知識よりもノウハウが重要になる．『ノウハウの質がサービスの質を決定づける』からである．

つまり，経験の質と量である．このことは，『ノウハウのある企業にしかノウハウはたまらない』ことを意味する．ノウハウのある有名企業には，客がくる．客の問題を解決することで，新たな経験を得る．その経験がまた，ノウハウになる．

技術者についても，同じことがいえる．ノウハウをもった優秀な技術者ほど，ノウハウをためやすくなる．ほんとうの意味での『専門化』である．

こんな時代がきっと来る．SEA はそのとき，ほんとうの意味での技術者集団として機能するか？ そうなるように改革して行くのが，われわれの使命ではないだろうか?

FA機器からのぞいたソフトウェア

武田 淳男
(安川電機)

1. FA 邑 (むら) にようこそ!

みなさんは,「PC」という略号を見て,何だと思いますか? 大抵の人は,Personal Computer(パソコン)だと答えるでしょう.しかし,FA (Factory Automatin)の世界では,Programable Controller を意味するのです.アメリカでは PLC(L: Logic が入る) と略され,PC と区別されています.日本ではなぜ? 「コントローラはロジックだけじゃない.うちの方がパソコンより早くから使っていた」という至極もっともな理由から,規格にまでなって,1つの邑を形成しています.お互い干渉がなかったらよかったのですが,最近では,厚かましくもパソコンがどんどん FA 邑に侵略してきて,いろいろな混乱が起ってきています.

かくいう私も,40 歳を過ぎてから,この邑にノコノコ足を踏み入れた素人ですが,最初は素人なりに計算機の本を手当たり次第に読んでみました.乱読の結果,計算機がわかったわけではありませんが,まず感じたことは,この世界はいささかもコンピュータ·サイエンス (CS) の恩恵にあずっていない,ということでした.どうしてでしょう?

2. プレディクタブルなんてクソくらえ!

FA のソフトウェアで,まず問題になるのが時間です.時間の大変なところは,それが物理そのもので,コンピュータ·サイエンスが現実世界を抽象化して,論理の世界で扱うものであるとすれば,非常に相性が悪いということです.もちろん,時制論理のように,時間の順序関係までは扱える技法もありますが,絶対的な処理時間そのものが問題になるようなところは,どうにも抽象化できません.

実時間がわずらわしいのは,具体的にやってみないと分からないところです.このため,トップダウン設計が簡単には持ち込めない.実現性を見てから,初めて要求とのネゴシエーションが行なわれる場合もある.FA 機器は機能表現で展開されることが多いのですが,時間内に計算が終りそうもない場合に,機能を組み立てる方法論まで変わってくる.

一般に,制御サイクル内に計算を終らなければ,その系は不安定になります.オーバフローで値が飛んで不安定になるのも,時間内に計算が終らないで不安定になるのも,不安定になるということでは一緒です.ならば,少しでも計算時間を減らすために,「オーバフロー·チェックなどやめてしまえ!」という世界です.

そんなわけで,確実に時間内に計算を終了するためには,プレディクタビリティ (予測可能性) という概念が重要になってきます.それは,実行前に処理時間が予測できることを意味している.ということは,データ,タスク,チャネルなど,すべてに関してダイナミックスが使えないことになる.再帰なんていう便利な表現も使えない.

こういう予測可能性の阻害要因は,たくさん存在します.ざっと列挙してみただけで,次の通りです:

言語	ダイナミック·バインディング ダイナミック·プロセス 再帰
OS	ページング スワッピング メモリ割り付け ガーベッジ·コレクション
ハードウェア	キャッシュ DMAサイクルの割り込み 並列処理におけるバス獲得

多分もっとあるでしょうが,これを見ていると,何だかコンピュータ·サイエンスの半分くらいは,まったく役に立たないんじゃないか? という気になってくる.こんなのを避けて通っていたら,技術的進歩から完全にとり残されてしまう.プレディクタビリティなんて構っちゃいられない.キャッシュはヒット率があるから,旧態依然とした CPU を使う? No! やっぱり,"Faster is Better!" ですよ.

"Faster is Better" かつ「安く」となると,アセンブリ言語の登場です.「アセンブラなんぞ古い.いまではコンパイラも賢くなってきてるから (ホントかな?),アセンブラで下手に書くよりずっと速いよ!」といっても,信奉者を説得することはできません.となると,CPU を変えるごとにプログラム書き直さなければな

らないから，再利用なんていう流行語はどこかへフットンで行ってしまいます．

3. モグラ叩き?!

ひとくちにFA機器のソフトウェアといっても，大から小まで千差万別ですが，成長製品の代表的プログラムの成長率は3年で2倍くらいの線を行っています．8ビットのMPUが出てから，もう20年近くたちますから，最初のプログラム量にくらべて，100倍くらいになっています．その間，マシンは16ビットから32ビットになり，RISCまで登場してきて，さすがにそういうものを使う場面では，アセンブラではどうにも手に負えなくなり，ある部分を除いてCが本流になってきました．

最初のプログラムに，あとから追加·追加でプログラムが大きくなるとなると，どうしたらよいか? 当然，いままでのプログラムにフラグを立てて，新しい機能をもぐり込ませる．それをくりかえすと，あっちこっちに旗が立ったプログラムが，そこら中を這い廻り出します．そうすれば，これまた当然，そう，這い廻って潜っているうちはいいけど，どこかでピョコッと顔を出す．「コリャァまずい」と頭を叩いて押し込めば，その反動で，別のところからまた顔がピョッと出てくる．それを叩くとまた別のところが..... これを称して「モグラ叩き」という．こういうバグ-じゃなかったモグラ)は，そのプログラムを捨ててしまわないかぎり，永久に死滅しません．

いろいろな開発プロジェクトのやり方を見ていると，プログラム量の増大が，結果としてそうなっているのと，最初からそうなると予想して取り組むのとでは，出来の上で大きな差になってあらわれる．そこで，教訓：

複雑なシステムは，よくしようと思ってもよくならない．しかし，よくしようと思わなかったら，絶対によくならない．

4. 複雑さの三悪

なぜ，これほどまでにプログラムが大きくなって行くのでしょうか? それには，DRAMに代表される半導体の進歩が大きく影響しているでしょう．容量は3年で4倍にふえても，チップ単価は変わらない．それに合わせてプログラムを大きくして行かなければ，機器のプライスを維持して行くことができない．そして，プログラムを大きくするためには；

- 自己機能を上げる．
- 周辺(環境)をシステムに取り込む．
- 上位／下位機能を取り込む．
- 使いやすいようにヒューマンインタフェイスを充実する．

というぐあいに，回りをどんどん飲み込んで統合化して行く．そういう風に拡張して行かないかぎり，製品が陳腐化して寿命がきてしまう．しかも平均的な伸びを維持するかぎり，価格には転化できない．20年前の8ビット50KBのFA機器と，いまの32ビット2MBの機器が同じ値段なのですから．生産現場では，ろくにソフトウェア工学の成果も採り入れていないくせに，生産性向上に対するプレッシャーが人一倍強い．

わからなければわからないほど，きびしいことがいえる．

FA機器のソフトウェアで，さらに，プログラムの複雑さ(汚さといった方がピッタリ!)を増大させている要素は次の3つです；

- マニュアル操作
- 例外処理
- ヒューマン·インタフェイス

マニュアル操作というのは，一種のヒューマン·インタフェイスかも知れませんが，どういう場面にも対応できるようにと，いろいろな状態からマニュアル操作に切替えることが要求されるわけです．そうやって眺めてみると，FA現場で対応しなければならない状態というのが，意外に多い．このマニュアル操作が，自動化の自然な流れの中にさまざまな場面で強引に割り込んでくるものですから，プログラムが汚くなることはなはだしい．しかも，複雑なプロセスを自動化するところまで来ているから，マニュアル操作の手続きが簡略化できないときている．

それでも，やっと組み込んだこの苦労が報いられたら，それはそれでいいのですが，最近の機器は信頼性が上がっているものだから，めったなことでは故障が起こらない．つまり，マニュアル操作の機会はグンと減っています．システムが複雑になっているせいもあって，そのうちにいざ事故があったとき，マニュアル運転できる人がいない(!)という事態になる．いや，笑いごとでなくて，これは悲劇です．もっと悲劇なのは，たまたま起こったところのマニュアル運転にバグが潜んでいた場合です．何年も経って起こるから，作った人もわからなくなっている．

それで，賢明なユーザのなかには，完全に別系統のコントローラを2系統持って，マニュアル操作をなくしてしまうところも出てきています．自動化した方が製品の品質もいいし.... というわけです．

もう，FA は人間ワザを越えている!

次の要因は例外処理で，これは例の Goto 論争のときにも，なくすことができないものの代表として挙げられたくらいですから，厄介なことでは筋金入りです．次のヒューマンインタフェイスともつながる話ですが，量的なことで説明すると，ある一連の運転で必要な操作画面は，通常 5 ないし 10 あれば足ります．仮に 10 画面として，一連の操作からはずれる場合に持っておかなければならない画面が，通常の 1 画面当たり 10 枚くらい張りつく．したがって，全体ではすぐ 100 画面は作らなければならなくなる，それに故障診断などの副次的な処理が加わると....!?

上の例からもわかるように，ヒューマン･インタフェイスも複雑さを増す大きな要因です．とくに，中途半端に AI がブームを起こして，「いままでは人間が機械に使われていたが，これからは機械のほうが人間に近づいて行かなければならない」なんて，もっともらしいことをいわれるものだから，この方面の注文がいろいろきびしくなってきています．それらを全部聞いて凝りだしたら，キリがない．しかし，一方では，機器性能の面では差がなくなってきているような商品では，いかに使いやすくするかが，勝負どころになってきている．とくに，人手不足や合理化の影響で，機器はふえたのにメインテナンス要員は減らされる傾向があり，故障箇所の特定など，熟練していない保守員でも短時間で容易に対処できることが要求されています．

5. ソフトウェアはいつも最後

FA 機器は，ハードウェアとソフトウェア両方を一緒に開発することが多いのです．最終的にモノ (実機) で動作を確認するのは，ハードウェアができあがってからなので，プログラムを動かすのが一番最後になる．ということは，ハードウェア開発を含めて，その前までの計画遅れを全部背負い込むことになります．結果として，納期遅れが明らかになり，みんなから注目され出したときに「バタグルッて」いる姿を見せるのは，ソフトウェア開発者です．つまり；

問題はすべて最後の人の責任になる．

開発にさいして，デバッグ中に機器に不具合が起こったとき，ハードウェアのせいか，それともソフトウェアが悪いのかは，すぐには明らかにならないことが多いのですが，ハードウェア屋さんのほうは，「われわれはちゃんと作ったよ」とかなんとかいって，なかなかつき合ってくれません．

厭な仕事にはだれも積極的に近づかない．

切迫した状態であればあるほど，みんなの足は遠のきますから，状況はますますひどくなります．そうなると，ソフト屋のほうでハードウェアの悪いことを証明して，首に縄つけて引っ張ってこなければならない．そんな手間をかけていたら，ますます納期は遅れてしまいます．

自分の問題とわかるまで，自分の問題にするな!
自分の問題かどうかは他人にはなかなかわからない．
複雑な問題になると自分にもわからない．

こんな風に並べて行くと，いつまで経ってもキリがないぞ? だんだん愚痴っぽくなってきたから，もうこの辺でやめておこう．

6. さぁ，どうする！はまたの機会に

SEA であるからには，こういう状況を少しでも改善して行くヒントを出さなければいけないのではないだろうか? 計算機を使えば，策はいろいろ考えられるのだが，むずかしいのは，「すべてが人の問題」に帰着することです．そういう話を始めるとまた長くなるので，紙面を改めた方がよいでしょう．今回はひとまずここまで．

そうそう，最後に，忘れないようにお定まりのテロップを流しておかなくては．

ここに書いてあることは事実を拡張したフィクションであり，特定の人物･団体には一切関係ありません．

あるパズル

玉井 哲雄
(筑波大学)

1. ことの始まり

この間，テレビでパスル特集という番組を長時間やっていて，そのごく一部を見たのだが，そこで次のような面白い問題が出された．

次の□に 1 から 9 までの数を重複なく入れて，等式を成り立つようにせよというのである．

$$\frac{\square}{\square\square} + \frac{\square}{\square\square} + \frac{\square}{\square\square} = 1$$

その場で 30 分ぐらい考えてみたが，答は見つからなかった．暇とやる気のある方は，この先を読むのをやめて考えてみていただきたい．

少し試してみると，次のようなことが分かる．

1. 多くの組み合わせは，1 を下回る．たとえば，もっとも大きそうな 9／14＋8／25＋7／36 でやっと 1.16 程度である (これが最大値かどうかは明らかでない)．だから，1, 2, 3, 4 のような小さな数字は，なるべく分母にくるようにしなければならないし，7, 8, 9 のような大きな数字，はなるべく分子に持ってくる必要がある．

2. 3 つの分数を通分した結果は，比較的簡単な分母を持つはずである．そのためには，各分数が約分によって簡単化され，さらにそれらの分母どうしに公約数があるようなケースが想定される．そう考えるとやっかいなのは 5 と 7 である．とくに 5 は，分子になるとすると，5 を約数に持つ分母が考えられないし，また分母の下位の桁におくとすると，5 が分母の素因数として残って，それは他の 2 つの項の分母には含まれない約数となる，というように，使い方が限定される．

この考察から，5 を分母の上位桁にもってきた 9／54 や 7／56 を含む場合を考えたが，うまくいかない．かなり近かったのが 9／12＋3／48＋7／56 だが，これは 15／16 となる．この辺であきらめた．このパズルは，視聴者からのファックスによる回答を受けつけるというものだったが，あとで家人に聞いたところによると，20 分で正しい答を送ってきた人がいたそうだ．ただ，解答そのものは家の者も聞いていない．

2. プログラムを作ってみることにする

その後もこの問題が気になっていたので，しばらくしてから，プログラムを書いて答を出そうという気になった．ここでやっと，多少はSEAMAIL らしい話になるわけだ．

プログラムといっても，典型的な玩具のプログラムである．要求仕様だのシステム設計だのという話ではない．次のような道具立てがあれば，後は簡単にできることはすぐわかる．

(1) 1 ～ 9 の順列を次々と生成する仕組み
(2) 有理数の計算

既存のものをなるべく活用しようというのが，現在のソフトウェア工学の精神であろう．有理数を扱える手近な言語に Common Lisp があるので，それを使うことに決めた．Mathematica のような数式処理系を使っても

いいのだが，正直なところあまり使ったことがないので今回は敬遠した．

それでも，順列の生成に便利な機能が組み込みであるなら，Mathematica を使ってみる気になるところだったが，ちょっと調べた限りでは，ぴったりのものがない．たとえば (a b c) というリストが与えられて，その全順列をリストとして返す関数といったものはあるのだが，それではとても役に立たない (遅延評価でもあれば別だが)．きっとどこかには，適当なライブラリがあるに違いないが.....

そこで順列生成は本から探すことにした．手近なところで見ると，野下浩平さんの「基本的算法 (岩波情報科学講座)」に，2つの有名なアルゴリズムが載っている．ただ，アルゴリズムの書き方は再帰的で，全部の順列を出力するものであり，当面の目的にはかなり変換する必要がある．もう1つ思いだしたのが，Dijkstra の *Discipline of Programming* にあるアルゴリズムで，これを Pascal プログラムにしたのが，土居範久さんの Pascal の教科書に載っている．これは，野下さんの本にある方法より効率は悪いが，1つずつ新しい順列を生成する点が，以下で述べるようなプログラムの構成の中で使うには便利である．生成する順序は，辞書式順序の昇順である (つまり9桁の整数と見たとき，だんだん大きくなる)．

プログラムの基本構造を，次のように考えた．

```
順列を初期化する．
以下を繰り返す．
    有理数式に変換･評価．
    ＝1なら結果を書き出して，繰り返しを抜ける．
    次の順列を生成．
```

この構造は素直だと思うが，順列生成のモジュールは，1回の呼び出しで新しい順列を1つだけ生成するものを想定している．もし，次々と順列を生成するようなモジュールを想定するなら (つまり，野下さんの本にあるアルゴリズムを，そのまま素直に実現したプログラムを想定するなら)，コルーチンにするか，逆にそちらを上位へ持ってきて，新しい順列が得られるたびに，有理数式を評価し検査することになろう．

次は，基本となるデータ構造である．といっても，問題となるのは1から9までの数からなる順列をどう表現するか，という点だけである．このデータ構造は，順列を生成する部分で変更され，有理数式を計算する部分で参照される．単純に大きさ9の1次元配列にすればよいだろう．

3. プログラム

ここまで方針が決まれば，プログラムにするのは単純作業である．実際，1時間ぐらいで作成し結果も出た．そんな簡単なプログラムでも，自分の書いたものを人前に見せるのは恥ずかしいという気持ちがある．しかし，せっかくの機会だから，恥をさらして諸兄姉のご批判を仰ぐとしよう．

まず，1--9 の順列をしまう配列を digits という名前とし，その大きさ N とともに大域変数とする．配列の初期値を与えるのは，後の便宜のため関数にしておく．

```
(defvar digits (make-array 9))
(defvar N 9)
(defun init-digits ()
  (setq digits #(1 2 3 4 5 6 7 8 9)))
```

プログラムの全体の構造は，すでに示した通りで，それを関数にしたのが次である．

```
(defun find-solution ()
  (init-digits)
  (dotimes (i (factorial 9))
```

```
        (when (= (add-three-rationals) 1)
              (print-result)
              (return))
        (next-permutation)))
```

ここで繰り返しの回数を9! までとしているのは，ほとんど無限ループといっているようなものだが，プログラムに虫があって，ほんとうに無限ループに入ってしまったり，万一問題の不備で答がなかったりした場合の，最後の歯止めのつもりである．階乗の計算は，教科書の再帰関数のところには必ず出てくるが，この場合，そのような教科書的なプログラムでいいだろう．たとえば，

```
(defun factorial (n)
  (if (zerop n) 1 (* n (factorial (1- n)))))
```

これをあえて逐次型に書き直そうとすると，案外間違えたりする．むしろ，末尾再帰のプログラムは最近のコンパイラでは自動的に最適化してくれるから，任せた方がよい．

問題の式を計算するところは，Common Lisp では有理数計算を勝手にやってくれるので，楽である．□／□□の計算は，次のようにすればよい．

```
(defun a-by-bc (a b c)
  (/ a (+ (* 10 b) c)))
```

これを使って3つの有理数の和を求めるところは，

```
(defun add-three-rationals ()
  (let ((sum 0))
        do ((i 0 (+ 3 i))) ((> i 6) sum)
          (incf sum (a-by-bc (aref digits (+ i 2))
                             (aref digits (+ i 0))
                             (aref digits (+ i 1)))))))
```

となる．ここで d_0／d_1d_2でなく，わざわざ d_2／d_0d_1を計算するようにしているのは，初期値として(1 2 3 4 5 6 7 8 9) を与えているが，すでに述べた理由により，1は分母の上位の桁に来る可能性が高いので，探索でそのケースを優先するようにしたものである．順列が辞書式昇順に出てくるので，1の位置は一番最後に動く．これ以外には，探索を高速化するための工夫は何もしていない．

結果の出力は，単純である．

```
(defun print-result ()
  (format t "~d/~d~d + ~d/~d~d + ~d/~d~d = 1"
        (aref digits 2) (aref digits 0) (aref digits 1)
        (aref digits 5) (aref digits 3) (aref digits 4)
        (aref digits 8) (aref digits 6) (aref digits 7)))
```

さて，順列生成は参照したPascal プログラムを単純に焼き直した．そして，テストしたらエラーが出た．今回のプログラムで出たほとんど唯一のエラーである．Common Lisp ではCと同じように，配列の添字は0から始まることになっている．Pascal では，添字の範囲は任意にとれるが，元のプログラムでは1 からn までになっていた．それをうっかりそのまま引き移して，エラーを出したものである．それを手直しするため，新たにN1という大域変数を定義しておく．

```
(defvar N1 (1- N))
```

これを用いて，順列生成のプログラムは次のようになる．

```
(defun next-permutation ()
```

```
(let ((i (1- N1)) (j N1))
  (loop (when (< (aref digits i) (aref digits (1+ i)))
              (return))
        (decf i))
  (loop (when (> (aref digits j) (aref digits i))
              (return))
        (decf j))
  (swap i j)
  (incf i) (setq j N1)
  (loop (when (>= i j) (return))
        (swap i j)
        (incf i) (decf j))))

(defun swap (i j)
  (let ((x (aref digits j)))
    (setf (aref digits j) (aref digits i))
    (setf (aref digits i) x)))
```

これですべてである．

4. 結果

実行してみた．待つことしばし．次のような答が出た．

```
>(find-solution)
9/12 + 5/34 + 7/68 = 1
NIL
```

これには，うなってしまった．確かに 5 と 7 がポイントだったが，1 つの分数の中だけで比較的簡単な形に約分することしか考えなかったのが，手落ちである．それにしても，分母に 17 を素因数として持つものを持ってくる可能性など，考えもしなかった．

この他に解はないのだろうか．よくできた問題であると信じると，多分この他には解がないのだろう (項の入れ換えという自明の別解は考えないとして)．もちろん，このプログラムを解が見つかったところで止めないで，最後までループを回せばいいわけだが，そこまでやる気はしなかった．

なお，使用した計算機はいまや旧式の SUN3/60．処理系は KCL．コンパイルして，解が見つかるまでの時間を測ったところ，次のようであった．

```
>(time (find-solution))
9/12 + 5/34 + 7/68 = 1
real time : 119.283 secs
run time  : 102.767 secs
NIL
```

参考文献

[1] 土居範久：PASCAL 入門，培風館，1985.
[2] 野下浩平，高岡忠雄，町田元：基本的算法 (岩波講座情報科学 10)，岩波書店，1983.
[3] 湯浅太一，萩谷昌己：Common Lisp 入門，岩波書店，1986.

[付記]

原稿を送った後で，このプログラムの欠陥に気づいた．欠陥といってもプログラムの仕様をきちんと示していないので，このプログラムの動作にあった仕様を想定すれば，問題はないともいえる．仕様が，

$$\frac{\square}{\square\square} + \frac{\square}{\square\square} + \frac{\square}{\square\square} = 1$$

の□に，1～9の自然数を重複なく入れて，等式を成り立たせるような組合せがあれば，そのひとつを出力する．

というものだとすると，解がない場合については何もいっていないから，このプログラムで構わないだろう．

しかし，本文中で少し余計なことを書いた．いわく，「ここで繰り返しの回数を9!までとしているのは,.... プログラムに虫があってほんとうに無限ループに入ってしまったり，万一問題の不備で答がなかったりした場合の，最後の歯止めのつもりである」．あるいは，「もちろん，このプログラムを解が見つかったところで止めないで，最後までループを回せばいいわけだが,....」これがまずい．

もし繰り返しを途中でぬけず，最後まで，つまり9!まで回ったとしよう．最後の順列は(9 8 7 6 5 4 3 2 1)で，これに対しadd-three-rationalsを計算し，結果が1になるかどうか判定する．そこまではよい．その後，この状態でnext-permutationを実行する．ところが，土居さんの教科書にあったプログラムを機械的に移植したこの手続きでは，このような辞書式順序で最大値になるような順列の次を求めることは想定していない．無理に実行すると，内部変数のiが-1となって，配列の添字の範囲から外れてしまう．

かといって，繰り返しを9!－1まで実行するのでは，(9 8 .... 1)のケースを確かめないことになる．だから，繰り返しの終了の判定を頭でやらずに，next-permutationを呼ぶ前で行なうようにするか，next-permutationの仕様を変えて，辞書式順序の最大値が来ても，エラーにならないようにするしかないだろう．

たとえば，辞書式順序の最大値が来たら初期値（辞書式の最小値）に戻るようにするに変更すると，

```
(defun next-permutation ()
  (let ((i (1- N1)) (j N1))
    (loop (when (or (minusp i)
                    (< (aref digits i) (aref digits (1+ i))))
                (return))
          (decf i))
    (when (minusp i)
          (init-digits) (return-from next-permutation))
    (loop (when (> (aref digits j) (aref digits i))
                (return))
          (decf j))
    (swap i j)
    (incf i) (setq j N1)
    (loop (when (>= i j) (return))
          (swap i j)
          (incf i) (decf j))))
```

この他にも，きっとおかしいところがあるだろう．プログラムを実際に読んで下さる方がどのくらいいるか分からないが，なんでもご指摘いただけるとありがたい．

最近のMacintosh とそのユーザー会

野村行憲
（岩手電子計算センター）

■はじめに

トースターのユーザー会というものが存在するのだろうか？

ユーザー会という組織が存在するのはコンピュータの世界だけだろうか？

コンピュータを「道具」として見るのならば、リョービの大工道具ユーザー会や、ヘンケルの包丁のユーザー会が存在しないのに、コンピュータのユーザー会があることに、疑問はないだろうか？

半面、車やバイクにはユーザー会が存在している。ロータスのオーナークラブ、ハーレーのオーナーズクラブがある。彼らのクラブでは、会員同士がそれぞれが保有するマシンの状態や改造の品評会や、部品交換、さらには団体でツーリングするといった活動をしているらしい。

さて、コンピュータ（パソコン）のユーザー会はというと、どうも今までの範疇（同じ趣味を持つ人達の集まり）では語れない「何か」が存在するように思う。今回は、盛岡（岩手県）で Macintosh のユーザー会を立ち上げ、活動を維持してきた経験から、このあたりを中心として考えていることを纏めたいと思う。そして、Macintoshの久々の新しいアーキテクチャーとして登場した AVシリーズ（AVとは言ってもxxxxビデオのことではない）や、AppleScript あたりの所感も述べてみたい。

■人生を狂わせた？ Macintosh Plus の登場

1986年春、あの Macintosh に最新型の、しかも漢字Tlak1.0 が搭載され日本語が使える Plus が登場した。モトローラ製 68000 CPUを 8MHz で駆動し、1MB のRAM と、800KB の FDD 1 Driveだけを装備したこの小さなマシンは、麗容なデザインだけではなく、人を虜にする不思議な魅力があった。価格もそれまでにない低価格の、648.000円ということで、1986年７月にLisa の頃から憧れていたマウスとマルチウインドウを使う環境を、遂に自分のものにすることができた。以来、彼は人生のいろいろな局面で大きな影響を与えることになる。

■Macintosh User's Group の黎明期（幸せの時代）

私が盛岡で Macintosh のユーザー会を立ち上げたのは、1988年4月20日のことでした。当時は今ほど Macintosh がメジャーではなく、圧倒的な国民機のユーザーの影で、ひっそりと限られた一部の目覚めた人達が（私を含め）ハードウエアとしては、きょう体と性能の割に飛び切り高価なモノクロ・マシンを、そのOS環境の先進性に驚嘆しながら見入っていた時代だった。

それゆえ、設立の呼掛けに呼応して集まった人達には、ある種の同胞意識（連帯感）が感じられ、すぐに共通の価値観を持った仲間として打ち解け合うことができた。

話題を理解し助言できる相手が居ることの喜びは、何も Macintosh に限ったことではあるまい。今まで孤軍奮闘していた（当時はMacintoshの情報誌が少なく、英語か、せいぜいローマ字のBBSが貴重な情報源であり、新しい発見や、疑問点などがあっても周りには誰も相手になる人が居なかった）彼らは、堰を切ったように嬉々として話に弾んだものであった。

従って、当時のユーザー会の集まりでは、自分が発見した（少なくとも本人はそう思っている）ことや、新しい（と思われる）情報の交換が主なもので、誰も知らなかった情報を提供した人は、まるで鬼の首でも取ったように勝ち誇り、皆は尊敬の眼差しで見ると言う状況であった。

■Macintosh User's Group の現在（惑いの時代）

幸か不幸か Macintosh のユーザーは２次曲線的な伸びを示している。この原因にはエバンジェリストの存在、流行、性能の向上など色々な面があるが、価格が下がったことも大きい。当然、入会希望者も多くなったが、当初は考えられなかったようなユーザーも増えている。曰く「パソコンを買いたくて調べたら、マックの評判が良いので購入した。ついてはマックで何ができるか教えていただきたい」とか、「マックは購入したばかりなので右も左も分かりません。とにかく何でもいいですから教えてください。」。何をかいわんやである。パソコンの低価格戦争の背景にはサポート料を含まないというのが最近の流行である。しかしこのツケをユーザー会が払うというのは納得が行かない。かくしてヴェテランユーザは足が遠のき、初心者だけが集まるという結果になる。会の世話をしている人達は、活動の意義を見失っている。もはやヴォランティアの限界が見えてきている。

■Macintosh User's Group の未来（成熟の時代）

冒頭に述べたように、コンピュータが「道具」であるとすれば、本来ユーザー会など必要の無いものだ。一般に認知され、使い方が確定されていればの話ではあるが。ユーザー会を必要とするものが、とんでもないユーザー・インターフェイスの説明、できの悪いマニュアルの補足、サポートの悪い販売店の肩代わりや、トラブルを抱えたユーザの駆け込み寺としてであれば、会を運営するスタッフの救いは何も無い。これらのことは本来メーカーやデーラーが行うべきことであり、その費用は当然、受益者が負担するべきものである。

もう一つの存在意義である、趣味の会であれば別の話である。「切磋琢磨」とか、「価値観の共有」などが存在する会こそ今後のユーザー会の有るべき姿なのではないだろうか？それには或る程度のレベルを持つことが参加する必要条件になるだろうし、若干の閉鎖的な雰囲気があるかもしれないが、それに打ち勝って初めて参加の喜びがあると思う。駆け込み寺的機能も必要ではある。しかし、hog には参加の資格は無い。参加者は常に対等の立場であり、互助互恵の組織であるべきだ。このようなユーザー会が理想だと思う。

■最近の関心

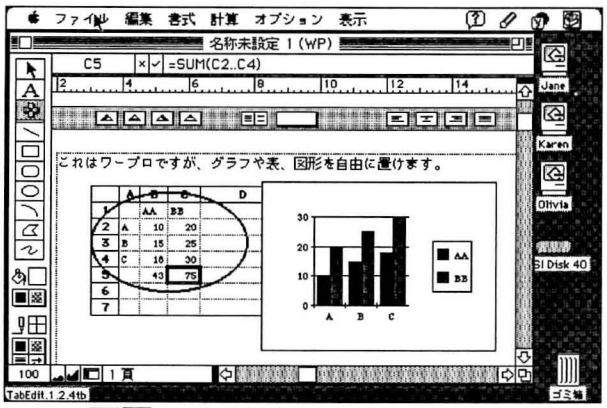
話を変えよう。

例会などで最近関心が高いのは、クラリス・ワークスなどのいわゆる統合ソフト。日本語版の登場がブームのきっかけであろう。御存知のように統合ソフトというのは、ワープロ、表計算（スプレッドシート）、データベース、作図、通信といった、ビジネスの大部分の局面をカバーする機能が一本のソフトに統合されたものを指す。

最近２回のユーザー会の例会では、Claris 社製クラリス・ワークス 1.0 と、Symantec社製グレート・ワークス 2.0 のデモ版（いずれも日本語版）を実際に使用して、集まった会員で評価をしてみた。

その結果、それぞれ一長一短があるものの、全体的な評価では、クラリス・ワークスの方に軍配が上がった。評価されたポイントは全体的に動作が早いこと、ワープロやグラフィクスの中でも表計算が出来るなど操作の自由度が高い点だ。特にグラフィクス機能は強力で、今までの、グラフィクス・ソフトとは全くイメージが異なるものになっている。どちらかというとDTP ソフトのように複数のテキスト・フィールドをチェインして、グラフィクスとの配置を効果的に行ったりということが、簡単にできてしまうことも凄い。しかも、この中にさえも、好きなところに好きなサイズでスプレッドシートが開け、その数値を使ったグラフもまた、好きなとことに好みの形で配置することができる。このことをメーカーである Claris では、「シームレスな統合による使いやすさ」と表現している。（画面-1）

この一枚のシート上に散りばめられた、文字枠、図形、表とそれに連動したグラフは、オブジェクトとして捕らえることができる。それらのオブジェクトにメッセージを送って一つのドキュメントを完成するという作業は、オブジェクト・オリエンテッドなものである。



画面-1 クラリス・ワークスの
シームレスな環境

■先進的だが受け入れられない Appleの新技術

AVの話しの前に、前述の統合ソフトに関連してSystem7　の新機能の話をしたい。特にSystem7 で登場した「IAC:Inter Aprication Communication」と、「Publish/Subscribe（漢字Talk7では発行／引用）」を利用することによって、アプリケーションの単機能化／矮小化を図るという考え方は、統合ソフトの登場で否定されたかに見える。この背景にはシンプルで低価格なソフトを自由に組み合わせて、ユーザーにとって最適な構成ができるという、利用者の立場に立った考え方が、商業的見地から見て魅力の無いものだったという現実の所為であろう。更に、そういった考え方が無く、Macintosh の GUI を模倣した（Macintoshファンの立場から見ると）だけのWindows や、DOS-PC への販路を重要視するソフトヴェンダーとの考え方のズレと見ることもできる。

一方、Cut & Paste の拡張（Live Paste）として新たにEdition Managerを導入して実現された「発行／引用」の部分は、たいがいのソフトが対応した。これは、他のソフトで作られたデータを、自社のソフトでも利用できる方が、競争力が有るとの判断が働いたと見える。

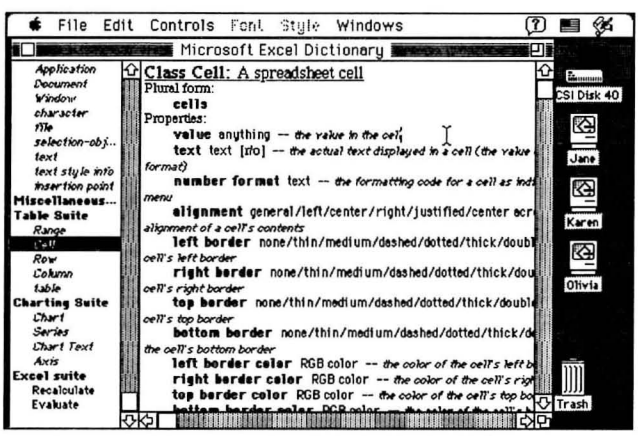
こういった中で、今年春に登場した AppleScript は、果たしてどう受け入れられて行くのだろうか？Macintosh の世界では望まれた機能ではあるが、これもまた、同じて轍を踏まないように期待する。

■AppleScriptのアドバンテージ

AppleScripは、アプリケーションなどをコントロールする（それだけではないが）マクロ言語であるといえる。Macintoshではユーザが行う操作を自動化したり、操作を記録して自動実行するという機能は、以前から求められていた機能ではあるが、実現されていなかった。（以前、MacroRecorderなどで一部実現されたこともあったし、QuicKeyなどの製品もあったが...）操作マクロというとMS-DOSの .BAT ファイルと比較されるが、全く別のものであり、遥かに高度な考え方に基づくもので、これはさすがに Apple だと思わせるものであろう。それは、アプリケーションなどをオブジェクトと捕らえ、それにメッセージを送って動作させるという、オブジェクト思考に基づくものであるからだ。

対応しているアプリケーションはまだ少ないが、Microsoft Excel 4.0では既に対応しているので、その一部を紹介しよう。Script Editorで辞書を開く機能を使い、MS-Excelの辞書を開いてみる。そうすると、Excelというアプリケーション・オブジェクトが受け取れるメッセージクラス（サブクラスもある）とそのメッセージ内容が、表示される。更にExcelのセルが受け取れるメッセージを表示してみた。（画面-2）

このようにAppleScriptを書こうとする人が、そのオブジェクトが受け取れるメッセージを知らなくても、アプリケーションが持っている辞書を参照しながらスクリプト（マクロ）を記述できるのも特徴だ。



画面-2 AppleScriptの辞書
（MS-Excel のセル）

さらに、このスクリプトは内部では中間言語で表現されているので、それらを表示する形式を、英語や日本語と自由に選ぶことができる。Appleではこれらの形式をDialectと呼んでいる。

■更に新しいAVシリーズ

この原稿を書いている（キィを叩いている）中に飛び込んできたのが、10月に発売されるAVシリーズを発表したというニュースだった。この中で特に「AV Technology」と呼ばれる新技術は、通信、ビデオ、音声の技術を融合したものであり、一台のデスクトップマシンだけで、電子メイル、ファクス、ヴォイスメイル、からハンドフリーの電話（スピーカーフォン）、更には留守番電話にもなるというもの。（Fax Modemなどの周辺機器を接続せずに）このAVシリーズのmacintoshを利用すると、高価な装置を使わずに電子会議システムを簡単に実現できる。

さらに、コンピュータの操作をマイクに向かって音声で指示することができる。（オフィスがやかましくなるのは困りものだが）

これらは、Macintoshに初めて搭載されたAT&T3210 Digital Signal Processor (DSP) と、GeoPortと呼ぶ通信アーキテクチャのなせる技だ。これを利用するには、GeoPortテレコム・アダプタ（定価18,000円）を接続するだけ。

■クロス・プラットホーム

最後にSYMANTEC社のBEDROCKを簡単に紹介する。これは、Windows と Macintosh で動作するプログラムを開発するためのフレームワーク（スケルトン）、クラス・ライブラリー、ツールで構成されるクロス・プラットホームである。言語としては Full ANSI-standard の C++ で記述し、そのソース・コードは、Windows か Macintosh でコンパイルするだけで、それぞれの OS で実行することができるというものだ。（図-1）

動作環境は Windows 3.1以降、Windows NT と Macintosh System 7.0 以降。果たしてTYMANTEC社の思惑どおり、土台として定着するか興味津々ではある。

仙台でのソフトウエア・シンポジウム'93の基調講演で伺ったMicroKernelと考え合わせると、ハードウエアや OS に依存しない環境が実現されるのも遠いことではないように思う。これもソフトウエア工学の産物か？

■SEAとの関わり

こういった内容の話は、残念ながら地域のユーザーグループでは会話できる相手が居ない。さらにソフトウエア工学的な話をしようとしても無理な相談である。このような地方の孤独なSEは結構いるのではないかと思う。この唯一の解決策がソフトウエア技術者協会に加入し、そのイベントに参加することだ。

ソフトウエア技術者協会では、数多い会員の中から価値観を共有する人を捜し出し、会話することができる。これを利用しない手はない。

こんなことばかりしていて、ビジネスに繋がるのだろうか？しかし、興味は尽きない....

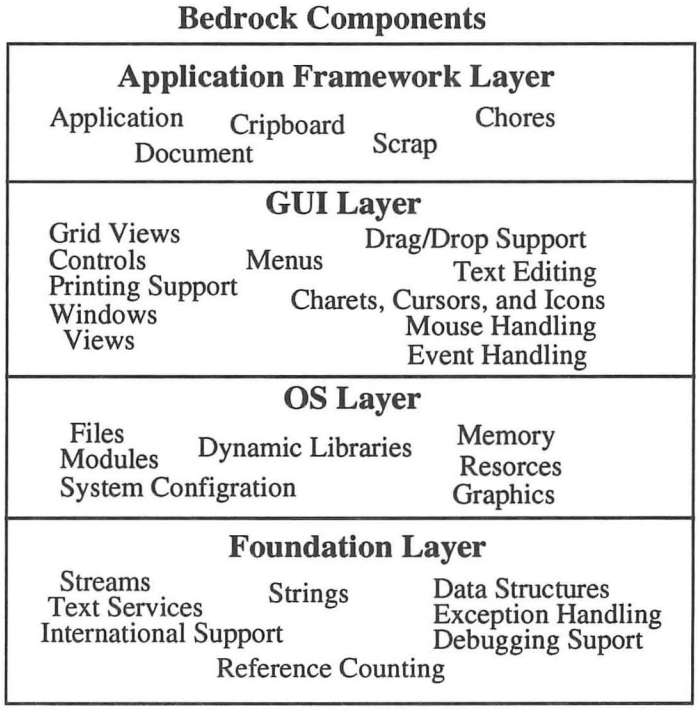


図-1 BEDROCK のComponents階層

プログラマの本棚

山崎 利治
(日本ユニシス)

Seamail になにを書けばいいかと考え倦んでいたとき，松原友夫さんの新刊の訳書が目に留まり，面白くて一気に読んでしまった．E. ヨードンの「ソフトウエア管理の落し穴：アメリカの事例に学ぶ」である．原題は "Decline and Fall of the American Programmer" (v. 酒匂寛，きまぐれブックレビュー．Seamail 7,11-12,1993) で，こちらの表題からは，アメリカのプログラマはローマ帝国ほどに栄華を極めたのだったかと考えたりもするが，いま，この本の内容には触れない．読みやすく面白いから，まだ読んでいない方にお薦めしたいとだけいっておく．

ところで問題はぼくの原稿で，実は，ヨードンのこの本がなにを書けばいいかに対して暗示を与えてくれたのである．本の最後に「付録B プログラマの本棚」があって，そこに87冊もの本を挙げ，あなたが読んでいないなら，時代遅れにならないように読みなさいというのである．こんなことならぼくだっていえる．つまり，ヨードンの顰に倣おうというわけである．87冊を持ち出すのは草臥れるから10冊に止めよう．

[1] R.Bird and P.Wadler, **An Introduction to Functional Programming**. Prntice-Hall, 1988. [武市正人訳，関数プログラミング．近代科学社，1991. xiii+317pp]

関数型のプログラム言語といえば，日本やアメリカではLisp系が全盛を誇っているが，イギリスでは，非Lisp系の数多くの関数型言語を研究開発してきた伝統からか，MLやMirandaが盛んに利用されているようである．MLについては，佐原伸さんがSeamail (8,2.,1993/7) に解説してくださっているが，バードとワドラーの2人によるこの本は，Mirandaを念頭においた関数型プログラミングの入門書である．それも厳密な科学としてのプログラミングを説く点で，命令型プログラミングに対するN. ヴィルトの「系統的プログラミング入門」に相当するといえるものである．

言語Mirandaは，純粋の関数型言語－関数定義と関数引用だけによってプログラムし，LispやMLがもっているような命令型機能は全くもっていない－であるが，パターン照合，高階関数，リストの内包表記，遅延評価による無限リスト (ストリーム)，多相の強い型付け，抽象データ型などを許す強力な言語である．これらについての詳細はこの本によって学んでほしいが，このような概念が命令型言語によるプログラムの仕様記述にも極めて有用なので，プログラミング入門はまず関数型で行うべきであるとさえ思う．というのも，MLにせよ，Mirandaにせよ，処理系の実現が，研究的な試作ではなく，十分実用になるまでになっているからである．

[2] D.Gries, **The Science of Programming**. Springer-Verlag, 1981. [筧捷彦訳，プログラミングの科学．培風館．1991. xiv+365pp]

問「命令型言語による正しいプログラムはどう作ればいいか?」に対する答えの一つがこの本である．

ほとんどすべての実用プログラムがFortran, Cobol, PL/I, Cなどの命令型言語によって作られている．ところでこの型の言語は厄介な性質，つまり，クワインのいう参照の不透明さ (referential opacity) をもっていて，そのためにプログラム言語以外の参照の透明な言語で仕様を作成し，プログラムが仕様を満たすという正当性の検証が必要になったわけである．

作ったプログラムの検証は骨がおれるから，はじめから正しいプログラムを書けばいいと考えたダイクストラは，この教義を例示すべく「プログラミング原論」を著わした．これがヨードンも挙げている E.W.Dijkstra, **A Discipline of Programming**. Prentice-Hall, 1976. [浦昭二·土居範久·原田賢一訳，サイエンス社，1983] である．

ホーアのようなひとはこれを「プログラミングの知的指導原理の発展における傑出した業績の一つ」と称えるが，世俗では，概ね，高踏的で晦渋であるとし，初心者には薦めないという．そこで使徒グリースが布教のために教義をやさしく解説しようとし，できたものが「プログラミングの科学」である．教祖ダイクスト

ラは序文を寄せてつぎのようにいっている.

「これはダヴィッド·グリース教授のようなひとに書いてほしかった教科書である. そしてグリース教授に匹敵するひとはいないから正に教授に書いてほしかったのである..... ここ10年の間に用語「プログラム」の潜在的意味は大きく変わった. われわれが10年前に書いた"プログラム"も今日われわれが書く"プログラム"もともに計算機上で実行する. しかし共通点はこれだけである. この表面的な共通点以外にはおなじ用語でこの両者を呼ぶのは混乱を惹起するものであるというほどに根本的に相違していて, "旧プログラム"と"新プログラム"との差は, たとえば, 予想と証明済みの定理の差, あるいは, 数学的事実についての前科学的知識と前提から厳密に演繹された帰結との差ほどに深く大きな差なのである....」と.

内容は3部から構成されている.

I部は命題計算と述語計算の入門である. プログラマにとってこれらが必要だというのは, 主題がなんであれ厳密な議論のためには正しい推論形式としての論理がすべての基本だからである. 本来なら数理論理学を素直に学べばいいのだが, 周到なグリースは実務上不可欠な部分だけに留める.

II部はIII部で用いるプログラム言語·ダイクストラによるふた付き命令(guarded command)を最弱前件(weakest precondition)によって定義する.

III部は本題のプログラミング教育, つまり, ダイクストラ計算の実習である. 問題の仕様を前件Pと後件Qを与える形で書く. PやQは入出力変数を含んだ論理式である. そこでPを満たすどんな入力に対しても必ず停止して, そのときにQを満たすプログラムを作成する. この作成方法を小さいが自明でないさまざまな例題によって教示し, プログラムの導出に有用ないくつかの戦略を提示する.

Pascal, PL/I, Fortranなどのプログラム言語ではどうするかについても丁寧に述べ, さらには, 注釈のつけかたにまで言及している. 付録に, バッカス記法, 集合·列·整数·実数·関係·関数など数学的事項を要領よく解説·補足して, グリースは「読んで判らないとはいわせないぞ」といっている.

[3] T.H.Cormen, C.E.Leiserson and R.L.Rivest, **Introduction to Algorithms**. MIT Press, 1990 . xvii+1028pp

プログラミングは課題を計算機向きに定式化し, それを解く算法を考案しコード化する作業である. 巨大プログラムは素人では作れないからプログラム作成方法は極めて大切だが定式化した課題の解法である算法は, それが案出できなければ全くプログラム不能なのでもっと大切である. 計算機の出現は今世紀後半であったが, それ以降計算機向きの計算方法がさまざまな課題に対して研究開発されてきた. したがってもうすでに算法に関して学ぶべき多くの知見が蓄積されているわけである. エイホ·ホップクロフト·ウルマンやクヌースの有名すぎる古典や和書も, それこそ, 汗牛充棟である.

ここに挙げたべつの三人組による新しい算法入門はB5変形の1045ページもの浩瀚な本である. しかし, この膨大さに怯む必要はない. ふつう入門書であれば大部さは懇切丁寧を意味し, 読みやすさと判りやすさに繋がっているものである. ただし, 持ち運びは無理だから紙装の本を2冊買って(42米ドル程度だろう), 1冊をばらばらに分解して読み, 1冊を便覧としてそのまま本棚におけばいい. ぼくたちはプログラム書きが商売だから2冊買う投資は決して惜しくはないだろう. ところでなぜこの本か? 網羅的であって, なお, 判りやすく, 厳密であり, 要領よく古典的材料を料理し, ならし計算量解析(amortized computational complexity analysis)や並列計算などの最近の成果も取り入れているからである.

[4] A.V.Aho, R.Sethi and J.D.Ullman, **Compilers, Principles, Techniques, and Tools**. Addison -Wesley, 1986. [原田賢一訳, コンパイラ原理·技法·ツールI, II. サイエンス社, 1990.xii+916+LIX pp]

コンパイラ作成理論は計算機の出現とともに研究されはじめ, 現在では計算科学のうちでももっとも成熟したものの1つになっている. この理論なしにはコンパイラは作れないし, 逆にこの理論を学べばだれにでもコンパイラが作れるわけである(ほとんど嘘だが).

プログラミングの現場で実際に直接コンパイラの作成や保守に従事するひとは多くないだろう. そうであっても, 仕様記述·プログラミング·利用の手引きの

作成など，つまり，ぼくたちの仕事はすべて文字列処理だからコンパイラ作成理論が直接に役立つわけである．事実CASEツールの作成にはこの分野の知識が不可欠である．いまではもうlexーyaccなどを使って，字句解析や構文解析のルーチンは文法を与えて自動生成ができるようになっている．しかし，例外処理を含む実用ツールを作ろうとすれば本格的にこの理論を弁えていなければならない．

すべての技術者が微積分を常識としたように，すべての職業プログラマはこの理論を身に付けていたい．プログラマの知的訓練の題材としてもこれは格好のものといえるからとくにそう思う．

[5] J.C.Cleaveland, **An Introduction to Data Types**. Addison-Wesley, 1986. [小林光夫訳，データ型序説．共立出版，1990．ix+301pp]

プログラミングにおけるデータ型の効用を明快に説いたのは，ヨードンも挙げているO.J.Dahl, E.W.Dijkstra and C.A.R.Hoare, **Structured Programming**. Prentice-Hall, 1972. [野下浩平·川合慧·武市正人訳，構造化プログラミング．サイエンス社，1975]のホーアによる「データ構造化序論」であった．事実，コンパイラが行う型検査によってプログラムの虫取りを経験したひとも多いはずである．

データ型について組織的に学んでおくことはプログラミングだけではなく仕様記述のためにも有用である．まして，最近は算体主導型（オブジェクト指向）プログラミング全盛である．ここで抽象データ型や多相型(polymorphic type)の概念が果たす役割は大きい(cf. S. Danforth and C. Tomlinson, **Type Theories and Object -Oriented Programming**. ACM Computing Surveys. 20,1. 29ー72, 1988. [柴山悦哉訳，型理論とオブジェクト指向プログラミング．コンピュータサイエンス

bit別冊．共立出版，1980.5]．これらの諸概念を豊富な例示によって丁寧に解説している教科書がこの本である．あえてプログラマの必読書の一冊としたわけである．

[6] C.A.R.Hoare, **Communicating Sequential Processes**. Prentice-Hall, 1985. [吉田信博訳，ホーアCSPモデルの理論．丸善，1992．xxii+246pp]

複数のプロセスが相互に情報の授受を行ないながら同時並行的に動作するプログラムを並行プログラムというが，並行プログラムにおけるプロセスはプログラミングの視点に立てば，プログラミングの動的側面に着目した強力なモジュール化機構を提供するものである．

プロセス間の共通資源とそれを操作する手続きとをまとめて抽象データ型としたモニタなどを考えたホーアは，ダイクストラのふた付き命令を基礎に並行プログラムのためのCSP（通信する逐次プロセス群）を提案した．プロセスは実際強力なモジュールであるが，その経時的な挙動は極めて把握しにくいものでもある．ミルナーのCSP（通信システム計算）などに刺戟を受けたホーアは，ダイクストラによる正しいプログラムの作り方である，いわゆるダイクストラ計算に倣い，正しい並行プログラムを作る枠組，いわばホーア計算を確立する．

これをやさしく解説したものがこの本である．ホーアはかってダイクストラの「プログラミング原論」を誉め称えた．今度はダイクストラの番である．「この待ちに待った本は，なにしろホーアのはじめての本だから，印刷インクが乾く前にすでに古典になってしまっている....」．並行プログラムは実時間システム作成のためには不可避であるが，この本を学んで，電話機，目覚時計，自動販売機，ポケット電卓などの仕様記述を試みるといいだろう．

[7] 井田哲雄，**計算モデルの基礎理論**．（岩波講座ソフトウエア科学１２）岩波，1991．xix+386pp

この本の２つの章ー関数モデル（ラムダ計算）と論理モデルーは，それぞれ関数型と論理型のプログラミングの基礎である．項書換モデルや代数モデルの章も仕様作成に大きなヒントを与えるから貴重であるが，プログラマとして，MLやMirandaまたPrologなどによるプログラミング能力は必須だからとくにはじめの二章は必読である．岩波のこの講座は独習を意図した教育的配慮の行き届いたものだから読みやすさはいうまでもない．

[8] 米田信夫·野下公平，**ALGOL60講義**．共立出版，1979．iii+135pp

プログラム言語ALGOL60の詳説である．1963年1月号のCACMに公表されたALGOL60報告書（仕様書）17ページを片言隻句も疎かにしないで徹底し

て読み解釈する．ここでぼくたちは textual criticism を学び，仕様とはこのように読むものであることを知る．襟を正して米田·野下両師の教えを受けなければならないが，ぼくたちが日頃書いている仕様に思い至れば，むしろ慄然とする体験となる．命令型プログラム言語の本質を学ぶとともに仕様記述に対する反省がえられればーー LOTOS や VDMー SL の仕様書とこの ALGOL60 報告書とを比較せよーー古い言語を見直すだけのものではないことを知るに違いない．

[9] 増永良文，**リレーショナルデータベース入門**．サイエンス社，1991．viii+213pp

データベース管理系，たとえば，Oracle とその問い合わせ言語 SQL などを用いて極めて簡便に事務計算システムが作れるのは同慶の至りである．こんな仕事に従事している仲間で実は関係データベースに関する基本知識に欠けるひとがいたのでこの本を挙げたいと思ったのである．つまり，正規形や不整合（anomaly）についての常識を欠くので実世界を眺めてそこから直接に見えるレコード型からどんなデータベースを作ればいいかが判らないのである．そこでとんでもなく大きなレコード型をデータベースにしてしまったりしている．版を重ねているデートやウルマンの本も翻訳されていて，もちろん，これらの熟読を薦めたいが，増永先生のこの本は電車のなかでも読めるから必読の一冊である．

[10] 山田真市，**情報処理の科学**．朝倉書店，1984．xii+390pp

「物質やエネルギーとともに，情報は，今日の科学技術における一つの基本概念」であり，「情報処理の応用分野は急速に拡大，多様化，高度化し，多岐にわたる情報処理の工学的科学的研究が積極的に進められている.... 反面，情報処理の将来に問題もある.... コンピュータの論理素子は電子工学上の限界に近づいているが存在する.... システム構成規模の面でも超 LSI や大規模ソフトウエアなど集積度や信頼性の向上には複雑さに起因する難しさがある」という現状認識のもとで，「役立つ情報処理の科学と技術を真に創造しようとするなら，現存のコンピュータや情報処理技術をあるがままに受け入れるだけに留まってはならない．まず目前の応用技術の仔細から一歩しりぞいて，情報処理のいろいろな基本概念や理論の意味内容とその発展の系譜をふりかえって，それぞれのアイデアを再吟味してみる必要がある」との立場で，

I 計算機械の科学
II 命令型プログラムと計算量の科学
III 関数型プログラムとプログラムの信頼性の科学
IV プログラムの意味論

を展開する．著者の学識は広く深く，百科事典的な内容を格調高くしかも判りやすく述べている．さらに学ぶひとのために 199 編に及ぶ文献表を付けている．ぼくはこの著者と長期間職場をともにし，武芸百般を習うことができた．それはいい思い出であるが，折角の教示も身につかなかった腑甲斐無さに恥じ入るだけである．

以上の 10 冊は，できるだけ日本語で読め，自己完結的なものを選んだつもりである．またごく基礎的なものばかりである．しかし，これらすべてを読了すれば，職業野球でいえば，二軍選手にはなれるだろう．それでもこれらの本がすーっと読めないかも知れない．時間もかかるだろう．仲間と一緒に読んだり，いいコーチが必要だったりするかも知れない．それというのもぼくたちの仕事の難しさの反映でしかない．日本のプログラマが衰亡しないためにおたがいに頑張りたいもである．

Unicode をめぐるディベイト

－ Ydoc mailing list 上でのやりとりの記録 －

SEA 横浜支部 (Ydoc) では，毎月最終金曜日の夕方から例会を持ち，そのあと中華街へ繰り出すのが常になっていますが，去る 6 月の 2 次会では，文字コードの国際標準化をめぐって大激論となり，以後 1ヶ月以上にわたって，コンピュータ･ネットワーク (Ydoc mailing list) 上で，活発な意見の応酬が続けられました．仕掛け人は，野中さん (Apple)，受けて立ったのは，藤野さん (FXIS)，桧原さん (NEC)，稲葉さん (Ricoh)．それに，はるかコロラドの酒匂さん (SRA)，その他の方々が茶々を入れて，とくに，8 月中旬のにぎやかさときたら...!

野次馬としての編集子の目には，たいへん面白く，また，いわゆる S/N 比 (Signal/ Noise Ratio) も高い有意義な討論であるように感じられました，そこで，関係者の方々の了解をえて，ここにその詳細な記録を掲載させていただくことにしました．なお，編集の都合上，一部の文章をカットしたり，用字･用語などを手直ししたりしてあります．したがって，文責はすべて編集部にあります．

まだネットワーク文化に親しんでいない方にとっては，なんともクレージイなやりとりの連続にみえるかも知れません．さすがに曲者ぞろいの Ydoc だけあって，ときどき本筋からはずれて飛び出す話題もなかなかユニークです．ちょっと長いですが，まあ，気軽にお楽しみください．

****

Date: Wed, 30 Jun 93 13:31:41
From: nonaka

先日の Ydoc では，みなさんお疲れさまでした....

また，2 次会では，標準化の話から Unicode の方へ話が脱線しました．私が Unicode の肩を持ったところから，大騒ぎとなり，周りのお客さんに迷惑がかかったのではないかと心配しています．来月は，断わられるかも知れない．

Date: Wed, 30 Jun 93 15:59:51
From: pv

たまにはよいのでは? 面白いディベイトの陣営だったと思いますよ，しかし檜原さんと私との連合では，ノナカちゃんの不利は一目瞭然．ああいう手合いのギロンは論理より気合い，古今東西のもっともらしい話をさも根拠がありそうに引き合いに出すことにかけては，相手が一枚上手だったよね．(^.^)

Date: Thu, 01 Jul 93 10:54:19
From: hihara
---

pv> ああいう手合いのギロンは論理より気合い，

はは.... 議論の進め方にも主張にもいえてたような?

Date: Thu, 01 Jul 93 11:28:35
From: inaba

先週は行けなかった稲葉です．

しまった，そういう面白いのがあったんなら行くんだった．:-(もっとも Unicode に関する議論を日本語でやっても，世間の情勢を変更できるだけの影響力は持てませんけどね．

Date: Wed, 28 Jul 93 12:37:28
From: nonaka

みなさん，いかがおすごしでしょうか? またまた，Ydoc の日が近づいてまいりました．

ところで，ひょっとすると，今月も 2 次会で Unicode の話が出るかも知れないので (あっ，別にやりたいわけじゃないよ)，簡単に私の考えをまとめておきます．

私の主張は，「同じ形の文字に 1 つのコードを割り当てるというアイデアは十分合理的だ」ということです．

Unicode で行われた現実の Unification の質が，どうしようもなくひどいものかどうかは，とりあえず別の問題として考えています．これが重要な問題であることは認識しています．実際問題として，CJK の Unification を有効かつ問題なく行うのは神わざに等しいほどむずかしいので，それを根拠に Unification を実現不可能とする主張は尊重すべきだと考えますが，とりあえず，その問題は切り離しておきましょう．ここはまだ調べてないし，調査の結果，私が Unification 批判の立場に回る可能性は大いにありますから．

なんだその程度か，と思った方ごめんなさい．

さて，「日本と中国と朝鮮では同じ字でも，文化的背景が異なるのだから，意味は異なる．それらの文字コードを，Unify するのはけしからん」という主張は，以下の 2 つの点で的外れであると，私は思っています．

(1) 文字コードは，意味を表わさない．単に描画する時に，アウトライン，ビットマップなどのテーブルルックアップを行うためのインデックスである．だから，描画したとき，まったく同じ形になるなら，それらに同じコードが割り当てられても，何も問題は生じない．

(2) 漢字は，そもそも輸入品であり，その後日本で独自の発達を遂げたことを考えたとしても，中国や朝鮮のそれらと，音，意味，形，など共通点は多い．たとえば，私が KangaTool の件で台湾へ出張したとき，会話は通じなかったが中華料理のメニューはある程度理解可能であったことを証拠として挙げたい．したがって，たとえば「上」という文字は，これらの国々の間で「同一の文字」と考えるほうが自然である．

それから，「実際に Unicode がどうやって日本の情報処理産業に浸透して行くか?」という件については，"シフト･ユニコード

普及論" という予想をしています．それはまた，別の機会に....

Date: Wed, 28 Jul 93 16:35:15
From: inaba

また同じ問題の蒸し返しだとは思いながら，日本語だからついついリプライしてしまいます．

nonaka> 簡単に私の考えを ....

ということなので，私の意見も書いておきます．Unicode の話が出るのなら Ydoc に行こうかな? どうも一次会のテーマだけ見て出欠をきめるのはよくないなあ．

nonaka>「同じ形の文字に ....

ここまでは，私も賛成です．つまり「ユニコード」には賛成ということ．

nonaka> # なんだその程度か，と思った方ごめんなさい．

どうせこの程度だろうと思ってましたもん．

nonaka> さて，「日本と中国と .... けしからん」

Anti-Unicode (とまとめられていますが，このいいかた自身おかしいと思う) の中で，こういうナイーブな反論をする人は，いまや少ないと思います．

各論に入る前に，今回の「似而非ユニコード」化について，日頃僕がうさんくさいと思っているのは，ラテン文字に対するユニコード化と非ラテン文字に対するユニコード化の程度が違いすぎることです．

この観点から野中さんの (一見もっともに見える) 非難に対する反論を書いて見ます．

nonaka> (1) 文字コードは，意味を表わさない ....

この問題には 2 つの反論があります．

(1.1) 文字コードは意味を表わさないか? 描画のためのインデックスだとすると，同一の文字を使うことが慣例であった大文字のオーとローマ数字のゼロに 2 つのコードを表わす (印刷に比べて非常に新しい) 慣習と矛盾します．本来，文字コードは「たまに」描画に使われることがあっても，本来はデータ処理の観点から分類されるべきでしょう．描画のためのインデックスという範囲を越えている例は日本語のひらかな，かたかなにおける「へ」にもあるし，KS にも存在していたはずです．これは，もととはいえデータベース屋さんとしては，ごくごくあたりまえの発想です．

(1.2) 100 歩ゆずって，文字コードが意味を表わさないとして..... それなら，ドイツ語のエスツエットとギリシャ文字の小文字のベータに代表される「同じ形をした文字」もユニコード化してほしい．もう過去に忘れる程の議論の中で，一番多かったこの意見に対する反論は (なんと，形が違うではなく)「この 2 つの文字は起源が違う」でした．片方では今回の CJK には多くの起源が違うのに，ユニファイされた文字があるのにですよ．こっちは，いまの仕事のプリンタ屋の発想です．

nonaka> (2) 漢字は，そもそも輸入品であり ....

文字というものは，そのもとをたどれば，別に日本の漢字に限らず「そもそも輸入品」であるものがほとんどです．この観点からいえば「その後中部ヨーロッパで独自の発達を遂げたことを考えたとしても」ラテン文字ではギリシャ文字と完全にユニファイするべきでしょう．

今回の「似而非ユニコード」というのは，いままでに述べた観点からすると，明らかに「ダブル·スタンダード」だと思うのですが，違いますか? 1 バイト文字と多バイト文字に対して同じやり方が適応されるのでない「似而非ユニコード」が，「文化的侵略」といわれるのは仕方がないのでは?

[PS] しかし，マイクロソフトも捨てようとしてるのに「似而非ユニコード」にしがみつかざるをえないハードメーカには，未来はない． :-)

Date: Thu, 29 Jul 93 14:51:14
From: nonaka

文字コード描画インデックス説は，私の誤りですね．取り下げます．

さて，

(1) 日本の漢字の集合のなかには CJK でユニファイ可能な (比較的大きな) 部分集合が存在する．

という主張には，稲葉さんは基本的に賛成だと判断してよろしいのですね? 私はこれを支持しています．稲葉さんの主張は，

(2) アスキー，ヨーロッパ·ラテンも，ちゃんとユニファイする．

という前提条件つきならね，というようにも読めますが，ひょっとすると全体が反語になっていて，「ユニファイなんてくだらないから捨ててしまった方がいい」という主張なのかな? 私はこの点については，どっちでもいい(無関心) といったところかな?

さて稲葉さんの問題提起は以下の通りだと思います：

(3) CJK 漢字コードのユニファイに関し，Unicode コンソシアム内で差別的扱いが存在する．

具体的な Unicode の内容については判断を避けたかったのですが，どうもそれではすまないようなので，見解を述べます．

実際のコード割り当ての作業に入れば，ダブルどころでなく，無数のスタンダードが出てくるのが，人の世の常であると思います．どうしても漢字を 5 万字，ユニコードに入れたかった人たちが，「差別的扱を受けた，文化侵略である」と騒ぎたくなる気持ちは理解できないわけではありませんが，利害関係のない第 3 者からみれば，「あるプロジェクトにおいて案 A が採用され，案 B は不採用になった」というだけの現象に見えるでしょう．この場合，B 案の提案者のとるべき態度としては，まず，なぜ不採用になったかを反省するのが正しいのではないでしょうか?

もちろん，「文化侵略だ!!!」と騒ぐのは，危機感をあおり，世間の関心を高め，巻き返しを図るための高度な戦略の一環であるのかもしれませんが，それは，私はあまりカッコよくないと思うわけです．

Date: Thu, 29 Jul 93 16:32:18
From: pv
---

short follow を!

「利害関係のない第３者」とは?

nonaka> まず .... 反省するのが正しい ....

「表意文字」の文明を持たない/ちゃんと理解していない/使っていない人が「判断」に関与してるから問題なのだよ.「反野中論」はそのうち書きます，いまは時間がない.

Date: Thu, 29 Jul 93 18:11:18
From: hihara

---

nonaka> (1) 文字コードは，意味を表わさない.

これが最大のセマンティック･ギャップでは?「そもそも文字コードにセマンティックなどない」といってしまえばそれまでですが ....

私は，いま提案されている多国語処理方式としては，TRON 文字コード体系が一番妥当だと思います．文字セットとそれにともなう操作も一緒に体系化するという点で ....

ただし，あのプロジェクトは実行力がともなわないのが難点ですね．Micro Soft の AtWork なんて，はるか昔に提案された MTRON のほんのサブセットなのですが，前者がいまだに概念でとどまっているのに対し，MS は着実にやるでしょうね.

ところで，さらに主題を離れてしまう余談ですが，先日「TRON が失敗したのは，みんなでやらないからだ」という話が出ましたね．しかし，TRON プロジェクトの中で，ITRON と CTRON は成功した，といえると思います.

なぜなら，いま市場に出ているリアルタイム OS の中では ITRON に準拠しているものが大半です．また，CTRON は NTT の調達条件となり，まっさきにタンデム･コンピュータが製品発表を行いました．これに対し，BTRON はまだマイナーだし，MTRON は概念段階を抜けていません.

ここで面白いのは，TRON プロジェクトの中で，前二者が「みんなで作った」規格であり，後二者がプロジェクトリーダの主張が強く出ているものだ，ということです．だから，TRON は一言では語れない．BTRON が広まらないのは，この OS をパーソナルメディアと共同開発した某 M 社が，自分のところのハード以外に移植するのを禁止しているのが非常に大きいと思うのですがね.

Date: Thu, 29 Jul 93 21:13:31
From: inaba

「それなりに」まじめに反論がきたとは，さすが Ydoc ですね.

nonaka> という主張には，稲葉さんは基本的に賛成 ....

そうです．CJK (いまのところ漢字をコード化しているのはこの4ヶ国しかない) でのコード化には重複が多く，それ自身はユニファイできると思います．ただし，技術的に CJK をユニファイ可能ということと，ユニファイすべきだということは違います．文字のコード化なんて characters for information interchange という意味からやるのでなければ意味はないはずで，そういう意味では；

nonaka> 下らないから捨ててしまった方がいい ....

と取られるかもしれません．反語のつもりはさらさらないですが...

nonaka> 私はこれ (1) を支持しています.

しつこいですが，(1) の命題には技術的可能性しか含まれていません．技術的および経済的妥当性も含めての支持ですか?

実はこの問題は次の (2) および (3) の２つの命題ともかかわるのですが，こちらに関しての私の意見は：

－ ユニフィケーションという概念を導入するのなら，それを１つの文字セットの中のすべてのキャラクタに適用せよ.

というものです．そういう意味で，命題の (3) は，おそらくそういうものが存在しているのでしょうけど，あまり重要な問題ではなく (だってこれをいいだしたら，文字セットの中のごく一部のページだけが１バイトであらわされていることだって，「差別的扱い」ですから．:-)

むしろ，この命題から出てくるのは；

[1] ユニフィケーションするのなら CJK それぞれの中に存在する非ユニファイ文字ペア (ununified character pairs) も統合化すべきである.

[2] その考え方は multi-byte characters page 以外にも適応すべきだ.

という結論です.

ただ，この意見はすかさず；

[x] characters for information interchange という観点からは ununified character pairs は unify できない.

という問題を起こすので，今のところユニフィケーションは非現実的だろうと思っているのです.

nonaka> 理解できないわけでは....

そんな，「理解できない」ってことを，わざわざ強調してもらわなくてもけっこうです．:-) Unicode ともとの ISO 2022 からの延長とを比べて「差別的扱を受けた，文化侵略である」と主張するのは，non chinese character users にもいるというあたりまえのことすら，想像の外なわけでしょうから.... ただ，Unicode に対して Microsoft が「それ以外の文字セットの使用を禁止したものではない」という発言をせざるを得なくなった理由は，CJK が原因ではなく，インド政府の圧力によるものだということを忘れないでください．まさか，インドで漢字が使われているって詭弁に走るんじゃないだろうな? :-)

nonaka> 利害関係のない第３者からみれば....

これは利害関係のない人が発言したら考えましょう.

nonaka> この場合，B 案の提案者のとるべき態度....

"Unicode" が "One of character sets" になったときに「A 案の提案者」がどのような態度を取るかを見ていましょう．どうせ，ここ２～３年のレンジだからね.

nonaka> あまりカッコよくない....

文化の問題をカッコよさで計られる筋合いはありません．まあ，この問題は；

－ 小選挙区制 ($+\alpha$) を唱える人は「改革派」そうじゃなければ「守旧派」.

－ クジラを食べるのは「残酷,野蛮」，スズメを食べるのは

「おとがめなし」.

といった問題と一緒で,「理屈に対しては,詭弁で逃げる」のが常道みたいですね.

Date: Fri, 30 Jul 93 01:23:39
From: inaba

さーてと,そろそろビールの酔いもさめたし(仕事は明日までなので恒例の打ち上げだったのです),まじめにUnicodeの話をもう少し書きましょう.

もともとISO 2022からの,文字コードのユーザ(multi-languageのDBMSとプリンタの開発者)としては,前回のメイルにも書いたように

- 同じに見える「文字」には同じコードを割りふらんかい!
- 人間ならコンテキストによって識別できるとはいえ,コンピュータにそんなことを期待するんでない. :-(

という2つの相矛盾する立場を持ち続けてきました.また,前いた会社がフランス語系であった(上司とも片言くらいはフランス語でしゃべってた)ところから,Latin-1ができる前のLatin character codeは非常に英語に偏っている(まあASCIIなら当然なんだが)ようにしか思えませんでした.

これに関しては,他の文字コードの標準化活動にそれなりに参加している人も同じような,精神的な紆余曲折(野中さんの方がもうちょっとうまくいえるでしょう)をたどってきてるようです.で,まあやっと最近SPDLグループとfontグループも含めて,「文字コードとは何か?」とか「グリフとは何か?」とかを定義せにゃあいかんということになりつつあるのです.

"The Unicode"に関しては,こうした背景を無視してJTC1/TC97(最近名前が変わったような気がするなあ,いま掃除中だからファイルがない)での作業に途中から割り込んで,ISO 10646を改悪してしまったのが問題だと思っています.だから,「アメリカ人はUnicode 派」と単純にいえないような現状になってしまったのだと思います.

実は,野中さんにすかさず取り下げられてしまったけど,「文字コード描画インデックス説」こそが,"The Unicode"を最も有効に使えるdomainだと思います.で,これでありがたいのは,文字をディスプレイしなければいけない会社で,一時期は"The Unicode"でキマりみたいになってたわけ.

ところがアプリケーション屋(といっても僕が付き合いのあるのはデータベース屋に限られますが)からすると,もはや文字コードなんて1つにしてくれなくってもかまわない.どうせWindowsでいうところのEmbeddingが出てくれば,browserが勝手にデータを処理するわけで,複数の文字コードのページを扱うのなんてEmbedding そのものの処理ですから.

実はOLEにおけるEmbeddingとISO-2022の文字の指示の仕方というのが,モデルとしたらほとんど同じだって,だれかMSの中の人も気がついたみたいで,最近では"Unicode is one of character sets."って発言に近くなってきてるワケです.

プリンタ屋からすると,わけのわからん文字コードをいろいろサポートするより,バサっと一つの文字セットにしてくれた方がありがたかったんだけど,ここ1年くらいは「font定義も標準化できるんだから,複数文字セットの処理はむずしくねえや」って方針を変えてきているのです.これでやっと,文字コードグループとSPDLグループがそれなりに仲よくなりかけたとこだったのに,また最初からやりなおしなんですよ. :-(

まあ,技術論争とはほとんど関係ないことではありますが,この点からは私は"The Unicode"問題に関しては,「熱心な観戦者」でずっといようと思っています.まあ,ATMなのかTrueTypeなのかと同じレベルですから.

ただし,この論争が技術論争と関係がないからこそ,技術論争にみせかけた

nonaka> 反省するのが正しいのでは....

といういい方には賛成できません.これをいいだせば,アーキテクチャのよし悪しを議論しているときに,「32 bitsではintelアーキテクチャが最もよい.なぜならもっとも売れているから」ってことになるわけ.まあ,たしかに僕はMacよりはIBMコンパチの方が「よい」と思うけどね. :-)

さて,野中さん,marketingの論争をします? それとも,技術論争の方がお好み? 文化論争でもいいよ. :-)

いまや本籍は企画セクション,副業が研究者. :-(

Date: Fri, 30 Jul 93 01:36:36
From: inaba

Unicodeから話がTRONに行きましたね.ちなみに今日掃除をしていたら,TRONキーボードがたくさん出てきてしまった.これを機会に,何人かは夏休み明けからSunでTRONごっこ(あのキーボードの方が慣れてる奴は相当いるのであった)でしょう. :-)

hihara> TRON文字コード体系が一番妥当....

これは僕も賛成.判り安いし,いくらでも拡張できる.ただし,この文字コードともとのISO-10626とは考え方は同じ(操作法という概念が明確になってませんが,後者では)だと思います,私は.

hihara> Microsoftの AtWorkなんて....

ゴメンなさい,AtWorkの片棒かつぎ(の会社の社員)です.

hihara> ITRONとCTRONは成功した....

ただし,CTRONはNTTの名もない(僕が忘れただけ)OSにカッコいい「冠」がほしいから,TRONファミリに入れたもんですよねぇ.ITRONに関しては文句はありませんが....

hihara> TRONは一言では語れない.

Ydocはソフトウェア屋の集りだから話が出ないのもわかるけど,おそらくTRONプロジェクト最大の汚点はTRONチップでしょうね.本来,「きれいから早い」はずのアーキテクチャが,ハード屋から「あれは遅いから使いもんにならん」とされてしまいましたからね.もう少しまじめにやれば,どうにかなった可能性もあるんだけどねぇ. :-(

Date: Fri, 30 Jul 93 19:28:50
From: hihara

---

inaba> 夏休み明けからSunでTRONごっこ....

うーん，遊びではなく，私は毎日会社で使っています．

inaba> 名もない (僕が忘れただけ) OS に....

そう，俗にいう NTT のタダノリですが，まあ，うまくいったのでいいんじゃないでしょうか?

inaba>あれは遅いから....

あれ? 最近出た Gmicro/500 は 60MHz で 130MIPS ですよ．でも，この話をうちのデバイス部門の同僚に話したら，「歩留まりがほとんどないだろ?」の一言でした．:-)

Date: Mon, 02 Aug 93 12:36:31
From: nonaka

稲葉さん，だめですよ，折角楽しみにまってたのに金曜日こないんだもの．:-)

さてさて，藤野さんや Dr.K と話してみて，私の考えが正確につたわっていないことが判明したので，少し詳しく説明します．そもそも，私が Unicode に関して，うさん臭く思っていたのは：

(1) CJK のユニファイに関する，異常とも思えるアレルギー
(2)「文化侵略」という言葉の持つ被害妄想的なひびき

の 2 点で，技術的な話とはまったく関係なかったのであります．(1) に関しては，このような，ナイーブなアレルギーを持つ人が私の攻撃対象でありました．反 CJK ユニファイ＝「美しい日本の文化と言葉を守りましょう」的国粋主義と捉えていたわけ．私は「文化はどんどん破壊せよ」と思ってますから．

あくまでも，アイデアとして面白いと思っただけで，現時点で，CJK ユニファイに経済的利益がないという点については，まったく同感です．

次に (2) に関して

inaba> .... といういい方には賛成できません．

これも逆にとられてしまった．最初に「文化侵略」という言葉を見たとき，私は「鬼畜米英」が「神国日本」を侵略しようとしているという風に，主語と目的語を補って理解しました．米の自由化論争の例もあるので，これがデフォルトの解釈としては，正しいような気もするが:-)，まあ百歩ゆずって，

－ 「大国」が「弱小国」を，あるいは，「主流派」が「反主流派」を侵略しようとしている．これは正義に反するのだ！
我々は，Unicode の崇高な理念を守るため，断固たたかうゾー!

という意味だとしましょう．で，私自身のような (利害のない，無責任な) 第 3 者からみると，この論争は，もはや技術論争ではなく，「コード･スペースの縄張りを争うための殴りあい」にしか見えないわけです．だとすると，殴った奴も悪いが，殴られた方も悪い．「もっと身体を鍛えて頑張らんかい!」といいたかったわけです．「B が反省しろ!」とはこういう意味なのでありました．

Date: Mon, 9 Aug 93 19:30:47
From: pv

藤野です，今度は medium follow．

nonaka> 私の考えが正確に伝わっていない....

これは勘違いだよ．野中ちゃんの「現象論的反駁」の意図は初めから十分理解していましたよ．だから，あなたが pretend するようにそれに呼応した「その counter partner として stereotype 的」な論争をしていたわけだし....

----->> そうですよね，檜原さん? (^^)

nonaka> このような，ナイーブなアレルギー....

このあたりに関しては，すでに 100 年くらい前から日本では日本語 (いわゆる国語審議会とか) で論争が行なわれてきましたよね，古くは森鴎外の名文があるし，戦後でも，正字正仮名を援護した福田恒存の美文といってよい反駁も出版されているし....

だから私としては，yet another problem であるこの種の問題についての論争に，新しい展開があるとはまったく期待などしていないわけ! 結局，理ではなく power game になるだけだから．

少なくとも鴎外と恒存の主張を越える「合理的」な主張が「効率最優先主義陣営」からなされるまでは，私としては同じ「土俵」には上がるつもりもないし，何者にも期待していないの (きっぱり)! (@.@)

nonaka>「文化はどんどん破壊せよ!」

個人の主張は自由です，それぞれですから．でもね，私はこの手の議論で「文化」という言葉は使わないよ．「文明」というべきでしょ? 「文化」なんぞは壊したきゃ勝手にどうぞ，でも「文明」は壊せないのよね．廃れるだけ．どのくらいの人が，意識してこのような言葉の使いわけをしているだろうか? この点については「敏感」でありたいと思う．嫌いなの，「鈍感」って．生理的にかな？

で，私はこのような「差別」に「快感」を憶えるエピキュリアンなので，それができないような「枠組」を効率という名のもとに押しつける輩には，すべて例外なく，自動的に反対なの．わかる?

nonaka> 断固たたかうゾー

そうじゃない場合も上の例で理解できたかな？ こういう身勝手な解釈はスノブ非難と同根だよね，歴史のある国の知識階級に対するヤキモチですよ．:-)

nonaka> B が反省しろとはこういう意味....

このあたりの意味不明．

総じて稲葉ちゃんの論旨が明解かつ論理的，かつ文章としても味わいがあるのに較べ，野中ちゃんのは悪文だよ，ハッキリいって．(__)

その主張を信じる/信じないは別にして，論点をもう少しきちんと整理して説明してもらえると，もう少しましな議論ができそうでこちらとしても嬉しいのですが..... そう，文明論でも技術論でも．いかが?

Date: Tue, 10 Aug 93 09:54:06
From: hihara

---

pv> そうですよね，檜原さん? (^^);

うっ，こういう感じでふられると回答に困りますが，けっこ

ういくつかのレイヤの議論があるにもかかわらず，議論のレイヤがすれちがっている感じはしますね．

pv> 結局，理ではなく....

そうですね．相手によって技術的な反論か，政策的な反論か，感情的な反論かが変わってきますね．場合によっては歴史的な観点が組み込まれるというところかな?

pv>「文明」は壊せないのよね.....

このあたりは何となくわかりますね．いま，われわれのやっていることは，ルネサンスのようなことではなくて，ローマ文明の過程を現代版に直したようなものだと日頃思っています．

Date: Tue, 10 Aug 93 19:16:39
From: nonaka

藤野さんの注文：

pv> 論点をもう少しきちんと整理して....

もういちど書きます．私の主張は次の２点：

(1) 日本で使用されている「上」と，中国で使用されている「上」．この２つの文字は，ほぼ同じ文字である．だから，同じ文字コードを割り当てるのは合理的である．もし，これを実行すると，藤野さんの表現したい世界に，どのように影響するのでしょうか?

(2) パワーゲームには，勝たなければいけない．これは人生論 (観) ですね．

以上です．

Date: Tue, 10 Aug 93 20:20:32
From: inaba

いろいろ (先週が夏休みだったというのも含めて :-) あって遅くなりました．

野中さんが論点を明確にしてくれたので，そこに関する反論を「まず」書きましょう．

nonaka> ほぼ同じ文字である．だから....

ふむ，「技術論」ですね．前の意見からもわかると思いますが，私はこの考えには賛成です．

しかし，もとの会合でも問題になったことですが，「ほぼ同じ文字」というものをどう判定するかが問題です．この議論を続けるのなら，野中さんがどういう観点で同一性を判定するか根拠を示してください．そうじゃなければ，この主張自身の「合理性」だって判定できないじゃないですか? 同一性の判定条件なしに，同じコードを割り当てることの合理性を判断するということは，「観念論」か「ひとりよがり」にすぎないと思います．

nonaka>もし，これを実行すると,....

少なくとも，私の表現したい (文字デザインの) 世界においては「同一だとみなされた文字」をひとくくりにデザインしなければならないので，どこまでが書体デザインに許された範囲かを明確にしてくれないと困ります．

一番ひどい例をあげれば，「土」と「士」を同一と判断する (あくまでわかりやすくするための例ですからね!) ような文字コードに対応するフォントを作るのなら，「二本の平行線の相互の長さの差は同一性判断には使用しない」という基準が必要になり，その世界でデザインする必要があります．

nonaka> (2) パワーゲームには，勝たなければ....

これも賛成です．

ところで，パワーゲームの典型としての戦争なんですが，この前の戦争で日本は負けて，(アメリカと) ソ連は勝ったんですよね．たしかに「戦争」という「小」パワーゲームには負けましたが，その後の「大」パワーゲームには日本は勝ったと思ってます．

野中さんの主張にある「パワーゲーム」とは，上の例でいうとどちらにあたるんでしょうねぇ? Anti "The" Unicode 派が負けたとすれば，ISO 10626 への採用の拒否が否決されたということだけでしょうね．Windows うんちゃらとか Mac うんちゃらが採用したというのは，勝ち負け以前の問題でしょ? どうせいまだって shift-JIS を使ってるんだから．

ということで，今回の議論のみに対する反論をまとめると：

(1) Unified code を作るには同じ文字コードを割り当てることのできる文字の相似について，規則 (原理) がなければいけない．さもないと合理的な文字割り当てができない．いやなことに逆のルール (同じ文字コードを割り当てできない場合) はあるんだよな．

(2) ちょっとずれるけど「人間万事サイオウが馬 (これくらい変換できろよなあ!)」かな?

です．次回の野中さんからの反論では，ぜひとも，このまとめから一歩進んだ論が含まれていることを期待します．その中には，ぜひとも，ギリシャ文字のベータとドイツ文字のエスツェットがなぜ同じ文字コードを割り振られないかの説明も入れてね．:-) やっぱ，議論してる時は，自分の意見だけをいいっぱなしでは何の実りもありませんから．

前回はいろいろあっていけませんでしたが，次回の Ydoc は極力行くようにします．でも僕はウナギがいいな! :-) もし必要なら ISO 10626 も持って行ってもいいけど，重いんだよなあ．

Date: Tue, 10 Aug 93 20:47:15
From: pv

基本的に野中ちゃんの論旨は以下の価値観 (並び，優先順位) を前提としているようですね．

- identical な entity には identical な code を付与することは合理的である．
- 何ごとも勝たなければいけない．

nonaka> この２つの文字は，ほぼ同じ文字である．

私はここにすでににいくつかの問題を見ます，

- まず identical と判断する基準
- entity の何をもって identical 比較の対象とするか? 表現 (appearance) か，「意味」か，あるいは?

で，私の立場は，

- 判断基準：ある言語 (文明) を構成する社会を判断の境界とし，その境界内における対象entity の概念的位置づけが相

対的に一致する時，これを identical と判断する．

- 比較対象：意味 (境界内における相対的概念位置).

であるべきだと考えます，これを「合理的」という言葉を使って表現してもよいのですが，私と野中ちゃんとではその布置が違っているので，ここではその表現は使いません．

nonaka> 同じ文字コードを割り当てるのは合理的....

「同じ code をふる」本来の意味は何なのでしょう?

- identify の容易さ
- 単なる便宜 (共有のための?)

さらに，付与される code 自体はいかにあるべきでしょうか?

- ある code 体系はどのような配慮から案出されるのか?
- その配慮は十分 reasonable と何をもとに評価するのか?
- code 体系間のその優劣は?

などなど，「合理的」という言葉で片づけられしまっていますが，実は配慮されていない factor が多すぎるように思えます．

nonaka> もし，これを実行すると,....

中国語の「上」と，日本語の「上」の微妙な差異が，(たぶんまったく) 表現できなくなってしまうでしょう，要は，

上 [反方向＝下，同意＝高，情感＝高揚...]
oid[属性と値の pair，...]

という表現 (encode method) で identical なら「同じ」とする方法が，言語 (文明) を表現する上で，拡張性や認識論的な側面から適していると考えているわけです，重要なのはここで単純に「上」という entity の code のみ見ていては identify できなくて，そこにあらわれる「属性」項や「値」項のそれぞれも再帰的に identical であることが要求されます，すなわち，その言語 (文明) domain 全般にわたっての相対的位置を，この code は表現することになります．

当然，中国語と日本語とでは domain が違いますから，この code method で行けば一致するという case はほとんどないと考えるのが当然だといえます．

まして，時間とともに domain の相対的な位置関係は変化 (拡張) して行きますから，そのような変化の中においても「同一性」を保証するには，対象を直接記述するのではなく，「対象の周囲」を relief 的に記述するしかないのでは，と考えています．

```
    野中ちゃん                私
certain entity          ?still magic?
    |                       |
    | han-ji encode         | new encoding (now)
    |                       |
    v                       v
表象としての漢字        oid[...]
    |                       +「漢字」appearance
    | yet another encode    +「発音」entity
    |                       +...
    v
???????
```

という違いですね，何を「基」として見ているかの違いかな?

nonaka> (2) パワーゲームには，勝たなければ....

ですから，私は「表現能力の劣るやり方を，それもそちらの勝手な＜価値観＞を拠り所に押しつけないで!」，といってるだけです．

私は redundancy を尊重するので，強制しないかぎりにおいては何者の存在も許します．「価値観」の相違自体も含めてね．

Date: Tue, 10 Aug 93 21:29:07
From: inaba

野中さんの反論がくるまえに先走りするのもいかがかとは思うのですが，pv 氏のも面白いから....

pv> 比較対象：意味 (境界内における相対的概念位置)

「理想的」文字コードの議論をする時ならいざしらず，現実の (つまりは工学的な) 文字コードを議論するときには，一応過去の文字コードのルールを前提にした方がいいと思います．もちろん理想的文字コード議論もそれはそれで面白いものですが...

「工学的」文字コード (たとえば ASCII とか IRA no. 5 とか) においては，後にのべる例外を除いて appearance で比較をするという合意がありました．この辺は Prentice だったかから出ている,"History of coded character set (だったかな?)" を見てもらうのが一番いいと思うのですが，たとえば ASCII においては (overlapping を許して)

```
c <backspace> "      で，c-tremma を
c <backspace>        で，c-accennt を
```

という風にあらわせるというのが，文字コード設計に含まれています．だから，フォントをデザインするときにも <"> <'> <`> <^> の 4 文字はアクセント記号にも使えるようにすることが「望ましい」とされていました．当然 non-overlapping の場合は，こういうことは考えなくてもいいです．

いま述べた基準に反するものは <1> と <l> および <0> と <O> です．昔のタイプライタを使ったことのある人ならよくご存知のように，これらはもともと同じ活字で印刷されるものでした．ただコンピュータの世界では，それではあんまりなので (ほんとうは telex の世界で困ったんですが) ASCII では，これらに関しては semantic 優先になっています．

だから，フォントをデザインするときも (Courier などが典型ですが) これら 2 つは特に区別がつくようにする慣習になっています．普通のディスプレイ用のフォントがほとんどの文字を sans-serif で設計し,<l> や <1> は serif にする理由の一つがこれです．

また,<0> と <O> の ambiguity が ISO キーボードの配列において，zero の shift に文字を割り当てなかった理由のうちの 1 つ (おそらく最大のもの) です．

まあ pv 氏の書かれている

pv> redundancy を尊重するので,....

こそが，IS-2022 および IS-10626 の原案の基本となっている考えですし，僕の理想とする unified code もこの延長にしかありえません．

The Unicode に対して，技術上で私がもっとも承諾しがたいのは，「何とじてでも文字コードは 16bits に入れるんや，そのためにはわてらがあまり使わん文字は寄せ集めてしまえ!」とい

う考えです．これは僕がそう感じているというのではなく，公式にCJKミーティングでいわれたセリフですからね．:-(

Date: Wed, 11 Aug 93 11:41:05
From: inaba

偶然にも，電子協が出版した「未来の文字コード体系に私達は不安をもっています」という非常にわかりやすい入門書が回覧されてきたので，必要ならコピーを作ってYdocの時に持っていこうと思うのですがいかが?

ざっと読んだかぎり，ゲラ刷りの時にくらべてより冷静な非難がふえて，説得力も増していると思います．

Date: Wed, 11 Aug 93 14:04:33
From: pv

やっと本論に入れてうれしい，でも，ほんとうはこんなことしている暇はないのだが....

inaba>「理想的」文字コードの議論を....

実は私が問いたいのは，まさしくここにMr.Inabaが述べているように，「あるimplicitな価値観による対象domainの絞り込み」自体の正当性なのです(さぁ，わたしの土俵だぞ!).

詳細を述べると大変なので，誤解されることを恐れずにいえば，19世紀の産業革命当時に確立された意味で，の「工学的」あるいは「科学的」という言葉の持つ意味や価値観を，そろそろ脱却すべき時にきていると私は認識しています．

これはもちろん，それらをまったく破棄しろといっているわけではありません，私は「強制」は嫌いだから．そうではなくて，時代に合わせて定義しなおすことに非常な重要性があります．

で，primarilyにはvisionally/ideal modelを考慮すべきで，それを現実世界におけるさまざまなconstraintsを考慮しtrade offすることで，いわゆる「工学的」modelがdecuctionされるべきだと信じています．

ひるがえって，現在の「工学的」codeの持つ現実的利点も，私は一応認めています．しかし，いつまでもそれに拘泥している「合理的」理由は，しだいに乏しくなってきているのではないかと感じています．稲葉さんは，いまのstandardizationやcode化に関して，そのような疑問は感じませんか？

inaba>「工学的」文字コードにおいては,....

そうなのです，historical back groundが実は前提としてあって，それにもとづいた「合理性」なのです．すでにencodeという概念自体が西洋科学文明の，つまりは「表音文字(本当の厳密な表音ではないが)」文明のdomainを前提としています．

さらに，これは表意文字でも同様ですが，「情報伝達」手段として2次元空間までしかなかった時代ですので，codeはおのずからそのappearanceと表裏一体とならざるを得ませんでした．そうでないと，具体的には紙などに記した時のambiguityが無視できないほど大きくなってしまうので....

しかし，いまの時代，virtuallyにはmulti-dimensionalなexpressionが可能なのですから，そのabilityを駆使したcode設計を行なうのは十分に「科学的」だし，「工学的」でしょう．当然，2-dimensinalなinformation expressionはそのsubsetとしてhandling可能です，もちろん，multi-dimensional informationの持ついくつかのadditionalなattributesが変換された時に落ちてしまうのは，しかたがありません．そのさいにambiguityをできるだけ小さくするような配慮は，当初のcode化の段階に1つのconstraintとしてとりあげてもかまいませんが，あくまでもそれはsecondaryという位置づけにおいてです．

まぁ，ほんとうにこれを「どのdegreeまでやるか?」は，たとえば印刷屋さんたちからはかなりクレームがあるでしょうし，一方，文学者や考古学者は，表現能力の向上という観点からよりfineなものを要求するかも知れません．

どのdegreeに落ち着くかは経済性や各国の利害関係，その他もろもろの要因が関係してくるでしょうが，reasonableであるとのconsensusはとれると思います．で，しつこいようですが，これはreasonableなものであって，mandatoryあるいはone and onlyなものではなく，時代の変化のもとで，さらに見直されて行くというのが，望ましいprocessではないでしょうか?

inaba> たとえばASCIIにおいては ....

2-dimensionalなappearanceにおいて非常に単純で限られた数のshapeしか持たない表音文字文明においては，このようなambiguity解消のためにad-hocなextensionの問題は常についてまわりますね．

appearanceにもとづくambiguityは，表意文字でも単純なshapeのもの，たとえば「口(カタカナ)」と「口(漢字のmouth)」と「国(の略字で中の玉がないもの)」や，カタカナと平仮名の「へ」,「一(漢字のone)」と「ー(長音記号)」などにも散見されます．

しかし，表意文字の強みはatomによるcombinationを基本としたsystematicな構成にあります．上述のaccent notationとは比較にならない範囲で，複雑なものがかなりsimpleなruleで実現可能となっています．

ですから，私は表音文字のencode methodについての合理性判断基準と，表意文字のそれとは大きく異なっていると思います(実はすでに前のmailに述べたmodelでは，表音文字のencode ruleは包含されてしまっている).

inaba> 昔のタイプライタを使ったことの ....

このようなhistorical な変遷をもっとdynamicに行なうのを躊躇する理由はないというのが，わたしの主張です．(^^)

inaba> だから，フォントをデザインするときも ....

現実の実装で「工夫」するのはかまいません，ただしglobalにnucleusとして扱われるべきentity自体に変なrestrictionを及ぼすことは排除されるべきでしょう．

inaba> 僕の理想とするunified codeも ....

基本的にはこれが大切ね．

まぁ，わたしはいつまでたってもきっと文句をいうでしょうが..... :-)

inaba> 私がもっとも承諾しがたいのは,....

「合理的」精神の持ち主としては，「いいかげんにしろ!」といいたいですよね．

Date: Tue, 10 Aug 93 23:30:58 -0600
From: sakoh
---

inaba> 電子協が出版した....

ぜひ読みたいと思います．お手数ではありますが，もしYdocでSRAの人に会う機会がございましたら，その人にお渡しいただけますか? そちらから社内便で送ってもらいますので .....

— 酒匂寛 @ 美国·漂礫郷

Date: Wed, 11 Aug 93 14:45:22
From: sahara
---

nonaka> だから，同じ文字コードを....

これは実用的には，はっきりダメですね．「上」という字の場合はよいかもしれませんが，もっと複雑な漢字の場合，中国がその漢字を略語にしてしまえば，日本と中国で同じコードが別の漢字に割り当てられることになって，不都合がおきます．しかも，中国はどんどん略字をふやそうとしているのだから...

Date: Wed, 11 Aug 93 00:28:42 -0600
From: sakoh
---

pv> 誤解されることを恐れずに ....

きっと誤解しているでしょうけど... :-)

pv> で，primarily には ....

私の理解では，この最後のパラグラフは現在の(そしておそらくは近代以降の)「科学的」「工学的」の言葉の定義そのものだと思うのですが? いわずもがなですが，ideal model を模索するのが「科学」で，trade off をするのが「工学」ですよね．では藤野さんが脱却したい「科学的」「工学的」価値観ってなんでしょう?

純粋に質問ですよ!

pv> そうではなくて時代に合わせて ....

もし現在「時代に合わせて定義し直すこと」がなされていないとするなら，それは「科学的」でも「工学的」でもない態度というだけじゃないでしょうか?

言葉に関する問題にすりかえただけかもしれませんが，そもそも言葉に端を発する話題なので ...

Date: Wed, 11 Aug 93 16:27:49
From: mari
---

inaba> 電子協が出版した....

わたしも読みたいのですが，何枚ぐらいの資料ですか? 今度のYdocには行けるかどうかはわからないので，コピーを一部送ってくださると助かるのですが，お願いできますか?

Date: Wed, 11 Aug 93 16:28:47
From: inaba
---

pv> やっと本論に入れてうれしい．

ふっふっふ，いよいよpv氏との喧嘩だぞ，ああ楽し! :-)

まず「標準化」というものの規範 (principles for standarization) について，一応簡単にのべておきましょう．これでpv氏の立場と僕の立場がなぜ異なるかが，かなり明確になると思いますから．モトネタを持ってたんですが，引越しで見つかりません．まあ，どこの会社でも(メーカなら)標準化の講習で配られるものですから，お手元から出してもらえればよろしい．コメントは文字コードのときに適用してみた僕の解釈です．

(0) 標準は文書化されていなければいけない．

あまりにあたりまえですが，実は文字コードではこれすら問題になります．DIS-10626 (rev.2) を見てもらえばわかるように，CJKに関しては，明確な意味で文書化されているとはいえません．だって，unifyされたけど，他の国に対応する文字がないところは空いてますからねぇ．

(1) 使用域明確化の原理

その標準の適用範囲をきめなければいけない．IS-626の場合は，「コード化したラテン文字を情報交換に使う」というのが使用域です．

(2) 汎用性の原理

(1) の原理にしたがって定義された使用域においては，標準は汎用性を持たなければいけない．IS-626でいえば「ラテン文字を情報交換に使うかぎり，これだけ使えば十分」という風になっていないといけないわけです．

(3) 経済性の原理

ある目的を達成するためには，標準は他の手法と比べていちじるしく経済性が劣っていてはいけない．一番経済性がすぐれている必要がないところに注意．IS-626なら「ラテン文字を送るのに2バイトはいらない」というのや，「9 bitsにすると，紙テープが文字あたり1コラムで終わらない」というのが経済性の原理になります．

他にもいくつかあるんだけど，忘れました．ちなみに(0)に含まれるものには，単位系はSIを使う方に移行しろだとかいうのもあり，特に国際標準化では，英語で書けだとかいうのや，参加国のうちに利益不利益があってはいけないということも「明文化」されています．

次に，前回のmailにおける僕の立場ですが，あくまで野中さんへの反論という観点から「Apple (& MicroSoft) の提案しているUnicodeのdefact standard化とそれをもとにしたDIS-10626の改定作業」がいかに不合理かについて書きました．

つまり，この時点で僕の議論のドメインは，「工学的文字コード」が現存し，それは今後も必要であるという立場に立ったことになります．だから

pv> 実は私が問いたいのは,...

は，前回の僕の議論とは根本的に噛みあいません．現在の工学的標準の確立方法に対する反論やコメントはまた別途書く(それより酒を飲みながらしゃべった方がいいなあ，でもこれだと横浜に宿でも取らないと何時間になるかわからん :-) として，このメイルではあくまでもとの議論だけに絞ります(pv氏ごめんね)．

ただ1つだけコメントをしておくとすれば，この「工学的標準」自身も55年体制の崩壊みたいなもんで，ISO流とIEC流，

そして CCITT 式で微妙な (ほんとうはそれ以上の) 差があり，体制自身が内部崩壊に近づいているのが現実です．すでに CCITT なる組織はないしね．:-)

pv> つまりは「表音文字 ....

最近では「表音文字」といわずに "poor character set" 反対を "rich character set" ということが多いですね，余談ですが.....

本論に戻れば，「ある言語で情報を交換するのに必要な文字が有限個ある」というのはコード化の大前提です．そういう意味では JIS X0208 を作成した時点でボタンを掛けちがったという面はあります．でも僕の立場でいけば，その上で今後どうするかを議論しています．

pv> 2 次元空間までしかなかった....

2 次元空間というのは「平面」のことですか? 平面を情報伝達手段として使うということと meaning <-> appearance の関係とのつながりがわかりません．

pv> multi-dimensional な ....

multi-dimensional にすると，何が変わりますか?

pv> これを「どの degree までやるか」は,....

IS の立場でいっても，この問題は分けて考えています．つまり，別のドメインなんだから別の representation を使うという，ごくあたりまえの方法ですね．たとえば JIS でいえば X0206 (だっけ?) のフォーマッティング法や，NAPLPS に代表される C0 の拡張，もっとミーハーにいえば SPDL そのさきの Babel プロジェクトに代表される hyper rich character set は別の標準 (もしくは案) ですね．

pv> ad-hoc な extension の問題 ....

これは ASCII においては extension ではなく，コード設計時に含まれる概念なんです．だから別のドメインなら別の考え方をすることは，何の問題もありません．

pv> しかし表意文字の強みは ....

これがあるからこそ，本来は X0208 を作ったときに，別のコード系の検討をも開始すべきではなかったかと僕は思っています．

pv>「合理的」と判断する評価基準 ....

ただ，僕は "A" と「あ」と「亜」のそれぞれを「文字」という一つのジャンルに入れる以上，X0208 および IS-2022 があるということは「合理的」だと思っています．

pv> 躊躇する理由はないのでは?

躊躇はしてません．日本でそのドメインにおける研究が活発でないだけです．

pv> 変な restriction を及ぼす ....

フォントハンドリングは，ASCII においては規格外の実装といいきれるものではありません．これは，モザイクキャラクタがフォントまで定義しているのと似ています．任意の文字コードにおいてフォントハンドリングを考慮することというのなら restriction ですが，いくつかの文字コードにおいてフォントハンドリングまで決めているのは，変な restriction だとはおもいません．

pv>「いいかげんにしろ」といいたい....

まあ，あとは Apple と R.M.Stallman くらいですから．どうしても，Unicode 1 つにしたがっているのは．:-)

Date: Wed, 11 Aug 93 18:13:56
From: nonaka

---

1 つずついきましょう．まず，稲葉さん．

(1) A と B が同じ文字かどうかを，どうやって判定するか？

もちろん，これには議論が必要でしょうが，これを提供することが，私の主張にとって本質的とは考えていません．その根拠は，たとえ日本国内だけで考えても，X208 での異体字の扱いにもあるように，(1) の問題は発生しているからです．

「CJK 共通の character encoding を定める」という行為は，「日本で JIS を制定する」という行為」の自然な延長線上にあるというのが，私の主張の本質と考えてください．

(2) ベータ，エスツェット問題

私は残念ながら，これらの文字が，「ほぼ同じ」なのかどうか判定できません．専門家におまかせしたいと思います．

蛇足ですが，わたしは，The Unicode で行われた実際の character encoding についての言及を，いまのところ避けています．ただし，The Unicode の basic design にある；

> — The Unicode standard avoids dupulication of characters by unifying them across languages: characters that are equivalent in form, usage and essential porperties are given a single code.

は，工学的に合理的であると評価しています．たとえ，そもそもの動機が不純であるとしても．

藤野さんへの反論は，ちょっと待っててください．

パワーゲーム論もしばらくお休みにしましょう．争点がぼけてしまいますから．

Date: Wed, 11 Aug 93 18:30:17
From: nonaka

---

sahara> 実用的には，はっきり駄目ですね．

しばらくは，実用性の話は，待っていただけませんか？私も，CJK ユニファイには，経済的にペイしないと思ってますので．

Date: Wed, 11 Aug 93 19:52:32
From: inaba

話が徐々に明確化されてきたので，(pv 氏の好きな観点は現在のところ戦線膠着ということで待ってね)「合理性」問題のみにしぼって考えてみましょう．

野中さんの引用を (大量に) 含むので，長い (いつもの話か :-) です．先に主張だけをいうと，

> — 「同じ形の文字に，1 つのコードを割り当てる」というアイデアが工学的に合理的であるかどうかを判定するには，同じ形の文字の定義もしくは，同じ形であると判定する手段が必要である．現時点では，この必須条件が満たされていないので，命題が合理的であるとは主張できない．

です．合理性に関しては前提が狂えばどんな命題だって真になることは，デカルトまでさかのぼらなくっても「工学者の常識」だと思いますけど．:-)

この主張に対して pro-(the) Unicode 派はいつも,「個別の文字に関して議論をしても判断はできないので, 同じ形の文字かどうかは議論しない」とのステレオタイプな思考停止 (野中さんのは典型ですね) なさいますが, もし前提の合理性を判定しないのなら, 以下の命題は工学的に合理的になってしまいます.

－ (タイムマシンは時間軸に関してのフィードバックがかけられるから) 時間軸上でネガティブ・フィードバックをかけてエネルギーを (安定的に) 増幅できる.

これは, 前提のタイムマシンのところさえのぞけば, 現実のオペアンプそのものだから, pro-Unicode 的ないいかたをすると十分「合理的」ですね. :-)

ただ, この問題に深く入ると CJK ミーティングで行った作業そのものになるので, 上の主張では「判定する手段」でもいいという風に, ちょっとゆるめています. この辺から野中発言をおっかけてみましょう...

nonaka> Date: Wed, 28 Jul 93 12:37:28
nonaka> 私の主張は,「同じ形の文字に 1 つのコードを....

これがオリジナルの article であり, 同じところで

nonaka> 日本と中国と朝鮮では同じ字....

ともおっしゃっています. この中で上の (というか今回の論の中心である)「同じ形の文字」に関しては,

nonaka> Date: Tue, 10 Aug 93 19:16:39
nonaka> ....「上」. この 2 つの文字は, ほぼ同じ....

と, 断言なさっています. ところが同じ人が (おそらくパターン認識能力は日によってかわらないと思うのですが :-)

nonaka> Date: Wed, 11 Aug 93 18:13:56

では,

nonaka> ベータ, エスツェット問題
nonaka> 専門家におまかせしたいと思います.

とおっしゃっています. パターンとしては, 何もめちゃくちゃ複雑なものではない (おそらく「上」とそんなに差がない) にもかかわらず, どうしてこっちは「判定できません」とおっしゃるのでしょう. もしこちらの主張を続けるのなら,「上」に関しては判定できるという過程も, 嘘という方が近いんではないでしょうか?

万が一, これら 2 つの文字をご覧になったことがないといけませんからいっておきますと, Mac では, ベータはシンボルセットの 142 に, エスツェットは標準テキスト文字の 373 にアサインされています. :-)

また, もし野中さんの真意が「私は日本語 (の文字) および中国語 (の文字) に関しては専門家である (ベータエスツェット問題では判定を専門家にまかされていますから) が, ギリシャ語およびドイツ語に関しては専門家でない」というところにあるのだとすれば, (ついでに韓国語に関してもそうでしょうから), Unicode の中の全文字に対する専門家が存在する (そうじゃないと, ほぼ同じかどうかは判定できないと主張なさっているのですから) という仮定が現実的かどうか? という問題に帰着します.

さて, 野中さんのオリジナルにある主張その 1 への反論はここまでにして, 次に「(ベータ, エスツェットのように起源がまったく異なるものはさておいて) 漢字は, そもそも輸入品であり, その後日本で独自の発達を遂げたことを考えたとしても, 中国や朝鮮のそれらと, 音, 意味, 形, など共通点は多い (ので, CJK のみを unify することは合理的である)」という (カッコの中はコンテクストによる, 稲葉の追加です. ここがムチャだというなら指摘してね!) 主張に対して考えてみます.

野中さんは, この主張の他に

nonaka> Date: Wed, 11 Aug 93 18:13:56
nonaka> － The Unicode standard avoids ....
nonaka> は, 工学的に合理的である....

ともおっしゃっています. 僕の感覚では, ある規範が工学的に合理的であり, 評価に足るものになるためには「その規範自身が独立して合理的であり, その規範にのっとって作成された系がその規範のおかげで効率的に作成された」ときだけだと思っています.

たとえば,

設計コストを下げるために, 大きいコンピュータから小さいコンピュータまでプログラマから同じように見えるようにするでぇ!

という規範に対し,

－ IBM 360/20 から 95 ができた.

というのは, 規範の合理性をみとめますが,

論理プログラムにしてプログラムを書きやすくするでぇ!

という規範に対し,

－ cut だらけの prolog だけで OS を書いた.

のは, 規範自身の合理性を疑います.

だから, この The Unicode の basic design に対してもやはり各論に入らないと, 規範自身が工学的に合理的かどうかは判定したくないという気分です. 当然,「くうりくうろん」としてならいくらでも「合理的な」規範は作れますが, 工学とは「手を動かしてナンボ」の世界のはずですから...

ということで, またもや各論 (なので専門家でない野中さんは読まないでよろしい! :-)) になってしまいますが,

(1) この principle は CJK にのみ言及していない, もし野中説 (漢字は中国 origin や!) を受け入れるとすれば, ギリシャ文字に起源を持つラテン・アルファベットも unify すべきである.
(2) この principle によれば, 意味が違う文字は unify してはいけない. ところが, 日本語の「湯」は中国語のよく似た (僕からすれば形は一緒の) 文字と unify されている. こっちはこれが一例じゃないからね.

という風に, もとの規範にのっとらない例が, あまたあります. これらのことから私は規範自身の合理性を疑っているのです.

以上が「同じ形の文字」という単純な言葉に起因する unification のウサンくささに対する, 明確化であります.

最後に今回の主題とは若干ずれますが,

nonaka> 言葉の持つ被害妄想的なひびき

に関してですが, The Unicode に関して「チェスの駒は dingbat だが将棋の駒は dingbat になっていない」という現実は,「被害妄想」という言葉でカタがつくんでしょうかね? :-)

Date: Thu, 12 Aug 93 00:22:49
From: pv
---

sakoh> ぜひ読みたいと思います.

わたしも欲しい (社内に尋ねればあるとは思うが), では 8 月は稲葉さんによる「character set encoding scheme 概論」にしましょうか, 都合はどうでしょうか? 8/27 の 6 時ですが.

で, 他の人の要望は無視して「鰻」にします, 稲葉さんの好みを優先して :-)

Date: Thu, 12 Aug 93 01:16:44
From: pv
---

inaba> ふっふっふ, いよいよ pv 氏との喧嘩だぞ!

おっと, 藪蛇だったかな? :-)

ちょっと舌足らずでしたので, まずいなぁ. (湯治から帰ってから) もう少し練って, また書きますので, しばらく休戦とさせてください.

ただちょっと conflict してるので教えて! IS の number に暗いので.

inaba> DIS-10626 (rev.2) を見てもらえば....

これって ISO-10646 のこと?

inaba> その上で今後どうするかを....

いまさらですが JIS X0208 (第 1, 第 2 水準), そして X0212 (補助漢字) とか破綻してますよ. まぁ,「当用漢字」とか, そもそも漢字の利用文字数が制限できると考えていた連中の不条理さに疑問を持たずに, 文字集合をその範疇で策定しただなんて....

encoding と collation の関係も一貫性がないし...., :-(

inaba> JIS でいえば X0206(だっけ?)....

えっ, X206 は知らないなぁ, 何のこと?

またここでいう「別の domain」の意味がちょっとよくわからない, 対象 application domain が違うってこと?

inaba> X0208 および IS-2022 がある....

すみません, IS-2022 って何でしたっけ?

Date: Thu, 12 Aug 93 01:44:04
From: pv
---

sakoh> きっと誤解しているでしょうけど... :-)

いえ, こちらが舌足らずなのがいけないのでしょう.

sakoh> では藤野さんが....?

うっ. 墓穴を掘ったかな?

一般には「普遍性」とか「連続性」とかが暗黙の評価基準ですよね,「科学」の? で, これら強迫観念を離れた「真の ideal model」を考えることが許されないような,「科学的」常識が浸透してしまっている, という意味です.

で,「工学的」も基本的には上述の「科学」の範を越えないことが暗黙の前提になってしまっていますよね? 私は「便宜」こそ「工学」の命ではないかと思うので, 工学的にはもっと divercity を優先したいと思っているのです (もうちょっとうまく表現します, 次回に).

sakoh> もし現在「時代に合わせて....

一番の問題は「見直し」の基準に「客観」性が必ず必要とされることですね, だから「客観の存在しない科学」をいうことが本質的に困難になってしまっています. :-(

sakoh> # 言葉に端を発する話題なので ...

私がいいたいのは, real world において先祖が「言葉 (発話)」や「表記 (表意文字)」を案出したのと同じように, いま (そして未来) の「computation world」に適した symbol (それが何かはまだわかってませんが, digital では表現不可能かも知れません) をわれわれも案出すべき時にきているのではないか? ということですね.

思っていることを伝えるのはむずかしい....

Date: Thu, 12 Aug 93 09:50:44
From: hihara

けっこう高次の概念の議論が続いていますので, 別の観点から, ハードウェア屋さんからの (コウモリのようだ :-)) 疑問を 1 つ.

ASCII に含まれているような文字は 7bit ですむのに, 16bit の記憶領域を常時持ち歩くのはどうか? また, 漢字のようにまじめにやると16bit では足りないものを「何らかの判断基準」で 16bit 空間に押し込めるのはどうか?

文字コードを何 bit で表わすかというのは, 副次的な問題だと私は思っています. だから, クラスにするなら unsigned int と決めないで, char なのか unsigned char なのか, unsigned int なのか, long int なのかはテンプレートにすべきだ, というのが私の考えです.

その意味だけでも, エスケープ·シーケンスで文字セットを変えるという方法は意味があると思います. そして, こういう方法をとれば, 各文字セットの規格化は各文字セットをよく理解した人たちに作業をまかせる, という分業ができることになるんですよね.

そして, 文字セットは文字コードばかりではなく,「何らかの」属性とともにまとめられるべきだと思っています.

文字セットを決めるためには, こういった上位の枠組みを決めることをそろそろするべきで, いまだに「文字コード」を決めれば標準化できると思っているのは, 四半期毎の決算報告が身に染み付いている人たちが井の中の蛙になってしまっているからではないか, というのが個人的な感想です.

Date: Thu, 12 Aug 93 10:01:16
From: hihara
---

inaba> にもかかわらず, どうしてこっちは....

この「判断する」過程の存在を無視しているから, 結果が曖昧になるのですよね. 決めた過程がトレースできない標準化はそもそも議論を呼ぶ源泉を含んでいるわけです.

Date: Thu, 12 Aug 93 10:12:46
From: nonaka
---

inaba>「工学者の常識」だと思いますけど :-)

同意します.

inaba> 嘘という方が近いんでは....

まあ,そのとおりです.私の判定は嘘に近いです:-) 当然,漢字の同一性の判定についても,専門家の判定をお願いしたい.

inaba> 全文字に対する専門家が....

全文字である必要はないでしょう.私が,主張しているのは,あくまでも CJK の範囲に限ってますので,CJK に精通した専門家 (あるいは専門家の集合による討議) の判定にまかせるということです.

inaba> 僕の感覚では....

純粋な質問ですが,X208 における,異体字の扱は,稲葉さんの感覚では合理的なのでしょうか? あるいは非合理的なのでしょうか?

私の主張している" CJK ユニファイは合理的" という主張で用いた" 合理的" という言葉は,「X208 が持っている程度の合理性」という意味です.もっとよい言葉があればいいのですが,はしょってしまいました.「X208 は,まったくもって非合理的」ということであれば,それはそれで,1 つの見識であると思います.

Date: Thu, 12 Aug 93 10:32:37
From: nonaka
---

hihara>「判断する」過程の存在を無視....

これに対する回答は,先に示したつもりですが,ひとつ質問.

(1) 判断のルールに,明確な基準が存在し,
(2) 結果が曖昧でなく,
(3) 議論を呼ばなかった

標準化の例を,お示しいただけますか.ぜひその過程を研究してみたいと思いますので.

それなら,標準なんて決める必要ないのではないかい?

Date: Thu, 12 Aug 93 12:43:45
From: nonaka
---

inaba> ということで,またもや各論....

ご主張よくわかりました.

少し私の方も補足します.私の考え (あるいは妄想) は,The Unicode の "unify across languages" という文に,刺激を受けたところから始まりはしましたが,現在は,「日中朝共通漢字コード体系" の実現可能性はいかに?」(もちろんその経済的動機もふくめ) という問題意識へ変化しています.この時点で The Unicode とはまったく乖離しています.そもそも,私が Unicode というキーワードを持ち出したのがいけなかったのでしょう.

とりあえず,稲葉さんから「技術的可能性」に関して支持がありましたので,私は満足です.ということで,このスレッドに関しては終わりにしたいのですが,どうでしょう?

さて次は,パワーゲーム論にしましょうか?

それとも,藤野さんの構想に挑戦しようかな? :-)

それとも,仕事に精を出すかな?

Date: Thu, 12 Aug 93 20:05:49
From: inaba
---

ydoc に限らず,ソフトウェア屋さんと話をしていると,どうも「美しい標準」なるものに日頃触れていないために,標準を重視しないという慣習があるような気がします.と過激にはじめるとまた議論を呼ぶから面白い. :-)

nonaka> それなら,標準なんて決める必要ない....

最初の標準であるネジは,長い年月こそかかっていますがそうなっています.さすがにウィットワースの昔に戻るのもなんですから,ユニファイ以降のネジに関して簡単にいえば;

(1) ある太さ (呼び径という) のネジは,公差を持ち,その範囲内にはいらなければいけない.それぞれの公差の中にあれば,同じ精度のネジ穴にきちんと入る.
(2) それぞれの公差 (これを級といいます) に入っているかどうかを判定するための標準がある.その標準で使用する計測機はもちろん別の標準できまっていて,最後の最後には (日本でいえば) 計量法によって定められる,標準の長さ,重さ,時間,そして電流の量に帰着する.
(3) (これはむしろ議論を呼ぶことが問題ではなく,複数の提案の中で最も合理的なものが示せればよいのでしょうから) ウィットワースのネジ山の角度 55 度から,ユニファイおよびメトリック 60 度になったのは,
   - 60 度の治具の方が,作成が楽で,精度を向上させることができる
   - 55 度と 60 度とでさまざまな素材でネジを作り,その強度を比較したところ,60 度の方が極度に強度のおちるものがなかった.ほとんどの材質では,60 度の方が高かった.

となっています.それでも標準を決める必要があるのは,こういう判断を毎回技術者がしなくてもいいようにです.こういうのを私は「工学的に合理的」と呼びたいです.

他のことがらに対する返事はまたあとで.

Date: Thu, 12 Aug 93 21:27:29
From: mary

脱線しますが,,,

inaba> ソフトウェア屋さんと話をしていると,....

ムッ.「美しいか」どうかの問題じゃなくって,技術が進んじゃって「決めても決めてもチンプ化しちゃう」ところに問題があるんだと思うんだけれどなぁ....

それとも,チンプ化するようなものは,美しくない? それだったら,この世界,しばらくは標準は美しくなりえない....

ネジと比べるのはちょっと....

「9mm は技術的に可能で,しかも××するためには 11mm じゃ駄目なのにぃ.... 標準なんか嫌いだぁ!」と,power game に負けた知人は涙ながらに語った (私じゃないよ! :-).

Date: Thu, 12 Aug 93 21:59:35
From: inaba
---

pv> わたしも欲しい.

とのことですので, もし可能ならば「リコーの誇るフルカラーコピア」で20部くらい作ります. もし手配がつかなかったら, 安物の白黒になるかも知れないけどね. 部数はこんなもんでいいですね.

pv> では8月は稲葉さんに....

えーっと, 僕の方はかまいませんが, しゃべる内容がちょっと問題かもしれません. というのは, ことこの分野に関してはいろいろと聞きかじったことは多いのですが, いまだに自分のオリジナリティのある貢献をしていないからです. ということで,「どっかのだれかのウケウリ」を集めたものでよければ, しゃべらせていただきます.

タイトルですが, あまりかしこまったものは何なので「標準化の光と陰 － ネジと文字コード」あたりの方がいいでしょうかね? (よっぽどこっちのほうがかしこまってるか? :-)) どっちにしろ, 文字コードだけでなく, 工業標準化一般の話もした方が, 今後のydoc MLでの議論の足しにもなるかと思います.

pv氏へ折角喧嘩を売ろうとしたのに, 急いでいて相当ウソを書いてしまいました. 一番大きいのはIS-10626で, これはもちろんIS-10646の間違いです.

罪滅ぼしと, あまりにいろんな数字を使いすぎた(これも急いでいたからだと, 思って勘弁してやってください. 誠にすみません)ことの反省をこめて, この辺に関係する標準のタイトルを一覧にしておきます.

ちなみに, 僕も明日から月曜まで(ちょっと遅い)夏休みを取るので, pvさんも心おきなく湯治を楽しんできてください.

検索しやすいようにJISの順にします. 関連するISOの番号からは見にくいですが...

---

*JIS X0124 単位記号の情報交換用表記方法 (Representation of Unit Symbols for Information Interchange)

ASCIIとかにはギリシャ文字がないので, その世界でどうやって単位記号をあらわすかの標準(マイクロをuでよく現すのはこの標準から来ている).

>ISO 2955 Information Processing -- Representations of SI and other units for use in systems with limited character sets

こっちの名前の方が正直ですね.

*JIS X0201(C6220) 情報交換用符号 (Code for Information Interchange)

ASCIIの日本語版と思ってほとんど間違いないもの.

[適用範囲] この規格は, 情報処理及びデータ伝送を行うシステム間で情報交換に用いる符号について規定する.

>ISO 646 7-bit coded character set for information processing interchange

昔は"6 and 7 bit"でした. 昔は"currency sign"とかいうわけのわからんものが入ってたけど, この前ついにASCIIと完全に一緒になったものを規定して, 各国で変えてもいいところが規定されている.

>ISO 4873 Information processing － 8-bit coded character set for information interchange

ISO 646も新しい名前はこうなっているはず.

*JIS X0202(C6228) 情報交換用符号の拡張法

Code Extension Techniques for Use with the Code for Information Interchange

<ESC> $@ で漢字になるとかを決めているモトネタ. 昔ミスプリがあって, 日本語の終端文字がスウェーデン語になってたなんて話もあります.

>ISO 2022 Information processing － ISO 7-bit and 8-bit coded character sets － Code extension techniques

>ISO 2375 Data processing - Procedure for registration of escape sequences

"@"がJIS-78で"B"がJIS-83だとかいうのはこのルールにしたがって登録されます. 実務はECMA (European Computer Manufacturing Association だったかな?) がやってます.

余談ですが, ECMAは標準化委員会の中では最も鷹揚で, 規格書は「タダ」でくれます. そのうち, internetでも配布されるはずです. 他の委員会は規格書の印刷が最大の収入源なんで, 一時やったけどやめちゃった.

>JIS X0203(C6221) 情報交換用および数値制御機械用符号の紙テープ上での表現 (Implementation of Code for Information Interchange and Numerical Control of Machines on Paper Tape)

この他, 無数の表現が規格化されています(紙の上と頭の中にどう表現するかは除いて :-() が, むしろ注意して欲しいのは「情報交換用」以外の「JIS文字コード」があることです.

>JIS B6311 数値制御機械用符号

別にX0201と差はないけど, 前にいったように「違う応用には違うコードがあってもいい」という例です. だから, 国語研だったかでやっている「漢字をヘンとツクリで現わす」コードを標準化したって別にいいんですよ, 必要なら.

*JIS X0207(C6225) 情報交換用漢字符号系のための制御文字符号 (Code of the Control Character Set for Japanese Graphic Characters for Information Interchange)

文字を上付きにしたり合成したりするためのコントロール. 問題はこれが制御文字のための区画に入っていないのだなあ :-(いいわけは解説にあります.

>ISO 6429 Additional Control Functions for Character Imaging Devices

これにはルビがないということから, 上の規格ができたのですが...

*JIS X0208(C6226) 情報交換用漢字符号系 (Code of the japanese Graphic Character Set for Information) Interchange

説明はいらんね.

>GB 2312

>GB 13000

>KS C5601
>CNS 11643
　CJK の内の C と K のコードです．KS は最近英語版が出たはずで，今取り寄せ中．これで辞書を引き引き読む必要がなくなる．:-)
*JIS X0212 補助漢字符号
　これも説明はいいですよね．

あーしんどかった! この他にテレテキストのとかあるけど，当面いいですよね．

Date: Thu, 12 Aug 93 22:14:52
From: inaba

稲葉です (あー忙しい)．

檜原さん曰く；

hihara> コウモリのようだ．:-)

僕も「ソフトウェア事業部」の中では一応ハードがわかることになってるので，常日頃「人生よきコウモリであるべし」を座右の銘にしております．:-)

hihara> ASCII に含まれて....

これは標準の説明の時にもいった「経済性の原理」からも相当重要視されます．だから

hihara> 副次的な問題だと....

という観点から，

hihara> エスケープ･シーケンスで....

に叛旗を翻してエスケープ･シーケンス撲滅に走った DIS-10646 (ほー，今度はきちんと書けた!) の最初のバージョンでも "compaction method" という (非常にわかりやすい方法で) 1 バイトから 4 バイトの固定長さと，可変長さへの圧縮を規格に入れています．しかも，圧縮はプロファイルとして扱われますから，双方の合意のもとで最初どういう圧縮モードからはじめるかを決めておくことができます．だからいまの "EUC" というのも (たしか) DIS-10646 で書けるとか，フレキシブルでわかりやすいシステムになって「いました．:-(」.

hihara> そして，こういう方法をとれば,....

この間から野中さんに対していっているのはまさしくこの問題で，僕自身は「各文字セットを越えてよく理解することはできるのだろうか?」という点に非常に疑問をいだいているのです．まあ DIS-10646 (日本は和田先生だったかな?) のメンバーもその危惧は抱いていたので，当初考えていた unification を時期尚早として排除したのです．ソフト屋なら，モジュール設計っていうごくあたりまえのことのはずなんですけどね．

hihara> まとめられるべきだと....

もう少し説明してくださいますか? これだけだとイエスともノーとも僕の意見をいえない．

> こういった上位の枠組みを....

上位の枠組というのが「何等かの属性を含むコード」という意味なら，すでにその方面の標準化は相当進んでいます．たとえば SPDL, SGML, DSSSL はすべて，何らかの属性を ISO-646 などに代表されるものに付加する形で進んでいるのです．だからこそ，こういう上位の標準に参加している身としては，今回の "The Unicode" 問題に関心を払わざるえないのです．

Date: Thu, 12 Aug 93 22:56:49
From: inaba

(またもや) 稲葉です．

nonaka> 専門家の判定をお願いしたい．

僕の問題提起は「専門家とはだれか?」です．まあ日本語に関しては一応私もそのうちに入れて欲しい (単に 30 年ほど使ったというだけの理由ですが) と思っています．

>全文字である必要はないでしょう．

CJK に限定する野中さんの立場と「文字は unification できるはずだ」という僕の立場の差ですね．また，野中さんの立場は，当初「Unicode に対するヒステリックな非難」ということから,"The Unicode" における unification の是非を技術的に論じていたはずですが，現在の意見はあくまで「CJK の文字の unification の可能性」に変わってきていますね．

後者に関しては，「専門家にまかせることができるのであれば可能」というのが僕の立場です．また，同時にその専門家はおそらく人間ではないだろうというのが「理想 Unicode 論者」である，僕の考えです．

これはもはや工学より科学の立場 (わたしのホームグラウンド) ですが，一応物理では「計ることのできないものは存在しない」という立場を取りますから，「すべての人に対して識別のできない文字の組 (この悪い例としてベータとエスツェットにこだわっただけです．当然僕はすべての人を代弁はできませんけどね) は文字として差がない」という命題を信じています．

工学の立場では，上の命題を実証するのに時間がかかりすぎるだろうから，次善の策として「各論を併記するしかないだろう」という主張になります．

nonaka> 純粋な質問ですが,....

yes or no なら「非合理」です．ただしこれは X0208 自身の非合理性というより，その前提となる「当用漢字」というシステムの非合理性に起因するところが大きいです．

一番僕が非合理だと思っているところは，「当用漢字にあるものはシンニューが点 1 つ，そうじゃないものは点 2 つ」というところですね．もともと審議会で，「使わない文字は字形を決めんでもいいだろう」と安直に考えたところが気に食わないわけ．

つまり，X0208 というものは (非合理であるとしても) ある国語を制定する authority の方針にしたがって，ある適用範囲の中での解を求めたという観点では十分合理性を持っていると思っています．まあフランスのアカデミーほどの合理性を国語審議会が持っているとは思ってませんが，(近代) 言語というものは何らかのオーソリティが作りあげるものだと，私は思っています．

nonaka> X208 が持っている程度の合理性

そういう意味で，なんら言語のオーソリティによる裏打ちのない "The Unicode" には X0208 並の合理性を認める気にはなれ

ません.

ちなみに, ここでいうオーソリティは別に「長いもの」である必要はなく, (アメリカ) 英語における Noah Webster のような個人でもなんもかまいません.「ヨーロッパ言語学に毒されすぎだ」と pv 氏あたりからイチャモンがきそうですが, ヨーロッパ言語における綴り字法の確立と同じレベルに漢字の書き方を考えたいというのが, 僕の立場です.

nonaka> それはそれで, 1つの見識....

ということで, 限定的とはいえ X0208 は合理的,"The Unicode" はまったくもって非合理と主張する私の立場はおわかりいただけましたでしょうか? :-)

nonaka>「日中朝共通漢字コード体系」....

その問題に対して, 私も非常に興味を持つからこそ "unification" の問題と "The Unicode" の問題の差を強く指摘したい (そうしないと, お題目だけのよさから, 中も見ずに "The Unicode" はいいという人ばかりふえるから) といつも思っているのです.

nonaka>「技術的可能性」に関しては, 支持が....

次の議題も面白いから, これでけっこうですが, 私が支持しているのは「理論的可能性」ですね. unification の実現度と「核融合発電」の実現度をほぼ同じレベルに考えていますからね.

余談ばかりで申し訳ないですが, 私は「核融合発電」は実現すると思っています. ただし, 実現するまでにある程度大規模な事故が起きないと主張する輩はあまりにウサンクサイと思っています. あんなに規模の大きいシステムが事故らないわけがない. :-) 当然ですがここでいう「実現」とはエネルギー収支の黒字化という意味です. そうじゃなきゃ, 太陽電池を置いといて, 水爆一つブッぱなせば, 今日にでも発電できる.

Date: Thu, 12 Aug 93 23:14:10
From: inaba

(最後の) 稲葉です

野中さん曰く;

nonaka> # さて次は, パワーゲーム論にしましょうか?

ですが, mary 曰く;

mary> power game に負けた知人は....

とのことですから, ぜひとも「標準化という名のパワーゲーム」とでも題して一度 Ydoc でしゃべってもらうのもいいんじゃないかと思います. まあ今月の2次会 (ギリシャ料理とかね. :-)) をこれに使ってもよいのではないかと思いますし... それなら僕も「国際標準化で1票を10票にする方法」とかで貢献できるしね.

一応 mary のにもついて行くとして,

mary> 決めても決めてもチンプ化しちゃう...

ざっと読んだとき「決めても決めなくてもチンプ化しちゃう」と読んでしまった. :-) まあ, コンピュータの標準における「チンプ化」というのは認めますよ. でもねぇ, RS-232C と称するあまりによく使われる言葉を見ちゃうと, コンピュータ屋は標準なんてどうでもいいと思ってるんじゃないかと思ってしまいます. ミリネジの RS-232C なんてお笑いですぜ. :-)

mary> それだったら, この世界....

僕に取っては陳腐化の問題もさることながら, コンピュータ関係では標準に「公差」という概念と「検査」という概念がほとんど導入されていないことの方が問題だと思います.

たとえば, 言語標準でいえば, それぞれの実装がスーパセットになっていることから, 折角の標準があまりに使えないもんになってしまっている. すべからくコンパイラには- standard というフラグがあって, これをつけると標準にない仕様は全部 warning 出してくれるようになってるべきだと思います.

mary> ネジと比べるのはちょっと....

だって「例を出せ」ってことでしたからねぇ.「こんぱちびりてぃ」って言葉もネジから出てるんだし, すべからく標準と名がつくものが守るべき基準はそんなに変わらないんじゃないかな?

ということで, 他の方のご意見をぜひともお待ちしております.

Date: Fri, 13 Aug 93 00:55:07
From: inaba
---

mari> 何枚ぐらいの資料ですか?

16 ページくらいの薄いものです. もし sra 誰かがくるのなら, 何冊か渡してもそんなにかさばらないと思います.

それとも...

mari> コピーを一部ほど送って....

やはりこれは, カラースキャナで読んだものをメイルで送れという意味でしょうか? :-)

こっちはそんなに手間じゃないので, それでもいいよ. 酒匂さんもそうしましょうか? :-)

Date: Fri, 13 Aug 93 02:15:03
From: inaba
---

pv 氏がいぬまの何とやらになってしまいますが, 一応東京脱出の前に書いておきます. しかし一方でパワーポリティックス, 他方でこれでは大変! 適当に議論の priority を決めた方がいいかも知れません.

pv> 暗黙の評価基準ですよね,「科学」の.

むしろ「(西洋) 哲学の」というべきかもしれません. たしか論理学における直感論理はこの辺に反駁したんじゃなかったでしたっけ.

pv>「工学的」も基本的には上述の「科学」の....

科学にのっとった工学というのは最近の (ここ 200 年くらいの :-) 主流ではありますが, 工学の出始め (J. Watt 等) の時には, 普遍性や連続性という点からはそこまで科学には隷属していなかったと思います. 今でも「材料工学」などの世界では「僕がやったらうまく行った (んだから仕方ねーだろ)」とか,「昨日はうまくいった」とかいうレポートはないわけではなく, これに普遍性を適用したい理由はただひたすら「便宜 (量産化)」のためだという事実はあります.

事実の指摘でない私の意見としては，こうした「工学」が「科学」の呪縛から離れることは，もっと他の分野でもあってよいだろうと思っています．どうせ工学にとって科学なんて公式集のうちの一つのはずですから．

ただし，このさいに，工学が(哲学から生まれた)科学とくっついて，それなりの普遍性や連続性をメソッドに取り入れた理由の１つである練金術などの(表面的な)失敗における反省は，考慮に入れる必要があると思います．

これに対して科学(特に物理ですが)というものは，「巨人の肩の上に乗る」学問なので「普遍的」であるべきものだと思っています．言葉を科学から物理に変えてしまって申しわけない(他の分野はそんなに知らないから)のですが，この世界では

－ ある地点で起きたことは，たとえ地の底，水の中，宇宙のはてでも成立する．

－ 今日起きたことは，昨日も明日も起きる．

という普遍性がないと，先人の努力をサブルーチンとして使えないから困るのです．一番目立つので例にあげますが，高エネルギー物理学では「モノを割っていったらどうなるの?」という，ギリシャ時代から人類みんなが一度は抱いた疑問を尖鋭化して「クォークがいくつかあって」とか「テキサスにドーナツを作って」とかいうところに問題が行ってしまったわけです．これが，「昨日は原子になったけど，今日は割れない」とか，「東京ではチャームがあるけど，アンドロメダではチャームはない」とかいう仮定が正しそうだとすれば，(モノを割ってやろうという)好奇心をそがれてしまうと思います．

まあここまで工学と科学を対極に置く考えは，日本の専門教育の影響に依るところが大きいのかもしれません(アメリカではCSってのは数多あってもCEってのはないですよね)が，偶然日本ではコンピュータの研究を理学部でやったり工学部でやったりしてるので，この偶然をもっといかせればいいのになと思っています．

ということで話を情報{科|工}学に持っていけば，たとえばアルゴリズムの研究を(この話にすればmaryは少なくとも食いつくだろう :-)

[理学的] Gaussが1+2+...+10を計算するときに，早い方法を思いついた．じゃあどこまで早くなるんだろう(という好奇心)．

[工学的] この構造計算を1時間でできれば，模型で試験するより安くあがる(というソロバン)．

という観点から行うくらい，研究の動機に差があるんじゃあないかなあ?と思います．

この観点からすると，pv氏の

pv>「computation world」に適したsymbol

は，そのsymbolの創出によって従来に比べて何か効率がよくなれば(「適した」という言葉からそのニュアンスを感じたのですが)十分「工学的」だと思います．つまりpv氏のこの命題に関しては，なんら科学の裏付けがなくても(おそらくできちゃうと思いますけど)案出することが「非工学的」というレッテルが張られることはないと思います．

ただし，「じゃあその案出につきあえ」といわれたら「他にもっと面白いことあるもん」といって，お断りしますけどね．:-)

今回の「文字」の話に近いところでいえば，書体によって人間の読むスピード(そして正しく理解できるスピード)が違いすぎるって事実は，僕にとって上の命題より圧倒的に興味を引きます．

Date: Fri, 13 Aug 93 09:27:59
From: hihara

---

nonaka> 標準化の例を....

民生品のことはよく知らないのですが，人工衛星の一連の設計標準はこのようになっています．(少ななくともそう努力されています)なぜなら，数十年にわたって使用することがわかっているので，状況に応じて内容の再検討が可能でなければならないからです．

「なんとしても16bitに収めたい」人たちは，どれくらいのスパンでものをみているのですか?

Date: Fri, 13 Aug 1993 10:16:24
From: nonaka

---

inaba>「専門家とはだれか?」

わたしは”CJKの各専門家達の討議による合意”は可能ではないかと，楽観的に，思っています．まあ，ここから先は，実際やってみるしかないわけですが，この面倒臭い作業をおこなうための動機づけ，いまのところがはっきりしてません．

ここまでが，技術的見解．

日本語の専門家のみによる，日中漢字の文化侵略的ユニファイを行う．どこかの会社がそのコード採用したプリンタを中国で売り出す．その圧倒的高性能･低価格で市場を支配し，その結果中国に，そのコードが定着する．

これが，パワーゲーム的見解．後者の展開となっても，これは必ずしも中国人民にとって，不幸なできごとであるとは考えていません．

inaba> 現在の意見は.... 変わってきていますね．

うーん．変わってるつもりは，ないのですが，「反The Unicode派」に対する理解は，とっても深まりました．

Date: Thu, 12 Aug 93 21:45:15 -0600
From: sakoh

---

inaba> 酒匂さんもそうしましょうか?

やはりこれは，カラースキャナで読み込んだドキュメントをOCRでしかるべき文字コードに変換して，メイルで送っていただけるという意味でしょうか? :-) コピーマシンに英和辞典が装備できるRICOHさんの技術なら朝飯前ですよね．それだとBoulderまで社内リンクでとどきますし，たかだか16ページ位ならデータ量も知れていますのでかまいませんよ．:-)．

....と冗談はともかく，次善の策は何でしょう? Faxでしょうか? でも国際電話をお願いするのもあんまりですしね．まあ特に急いではいませんので，機会がありましたら．

余談ですが，現在AT&Tを用いて日本に電話をかけると１分

46 セントで，接続時間 10 分以降 1 分 44 セントになります．少々クレージーな価格競争が続いています．

Date: Fri, 13 Aug 1993 14:35:58
From: nonaka
---

hihara> 分業ができる....

別に，切り替えシーケンスの導入が分業の必須条件ではないですよね？ N バイトのコードスペースを適当に区切って，各ワーキンググループに割り当てても，分業は可能だから．

分業が必要だとは，最初から思っています．だからこそ，私がベータ/エスツェット問題の判定をする必要はないと思ってました (もちろん能力もないですが)．Tamil, Thai, Bengali, Gurmukhi など，恐ろしげな文字セットは山ほどありますからね．

ただ，漢字文化 (文明?) 圏に身を置くものとしては，分業を行う 1 つの単位として，日中朝の共同作業は可能ではないか，と思っているのですが，その理論的可能性について，檜原さんはいかがお考えですか?

原理的に，日本，中国，朝鮮の，相互理解は不可能？ それとも現実的でない？ etc.

Date: Fri, 13 Aug 1993 18:05:16
From: nonaka
---

inaba> 標準を重視しないという慣習がある....

これは，私に関してはあたっていますね．私が標準という言葉から連想するのは；

IBM PC/AT
PC9800
MS-DOS

といったところですから．ISO の投票がどうの，こうのというのは問題外の外でありましたし，稲葉さんがしきりに「ユニファイルール適用の一貫性」を重要視するのが，理解できなかった．藤野さんのように霞を食べて生きている人とも，話が通じないのは当然ですね．:-)

さて，美しい標準の例として，ねじと人工衛星があげられましたが，たとえば，

- 有力企業が自社に有利なように規格をねじ曲げる．
- 米国から，非関税障壁だとクレームがついて，政治的に決着がなされる．

というようなことは，ほんとうになかったのでしょうか？ なかったとすれば，いかにしてそれは成し遂げられたのでしょうか？ その手法は，国際文字コードを決める活動に適用可能でしょうか?

Date: Fri, 13 Aug 1993 18:51:54
From: nonaka
---

pv> ある言語 (文明) を構成する社会を....

質問ですが，なぜ言語というドメインを導入するのでしょうか？ もはや，藤野さんの表現方法は，言語などという枠組みにとらわれる必要は，まったくないように思われるのですが?

日本語や中国語というドメイン設定を認めるなら；

関西 vs 関東 domain
Apple 社員 vs Xerox 社員 domain
漢字 vs Roman domain
地球人 vs 火星人 domain

何でもありでよいのではないでしょうか?

Date: Fri, 13 Aug 1993 18:59:49
From: nonaka

もう 1 点気になるところをば...

pv> ですから，私は「押しつけないで....

藤野さんの考える「強制される，押しつけられる」という状況は，もう少し具体的にいうと，どのような場面なのでしょうか？ たとえば，現在私たちは Internet の上でお話ししているので，JIS コードの範囲内の文字しか使えないという制限があるわけですが，これは強制されている状態なのでしょうか?

Date: Fri, 13 Aug 93 20:32:01
From: nonaka

野中です．いろいろうるさくてすみません．

いま，職場の同僚との話合いの中で指摘されたのですが...

- たとえば，16bit より大きな空間を必要とする文字文明をわれわれが持っているとするならば，なぜそのことを強く主張する文字コード体系を，われわれが先頭をきって ISO へ提案できなかったのでしょうか？ あるいは，マーケットを支配できなかったのでしょうか?
- The Unicode へ文句をいうだけでよいのでしょうか?
- われわれのふがいなさを反省する必要はないのでしょうか?
- The Unicode よりすぐれた体系を提案する必要はないのでしょうか?

もちろん，私自身の自戒の意味も含めて...

Date: Fri, 13 Aug 93 21:34:18
From: hiro-f

深瀬 @ IIJ です．一連の議論を聞いていて，思いだしたことをコメントしたいと思います．

nonaka> われわれが先頭をきって ISO へ提案....

数年前に，10646 を押そうという研究会に参加していたことがあります．10646 はそのままでは国際投票の場で Unicode 派に負けてしまったのですが，その時，中華人民共和国も 10646 に反対投票を投じたことを，意外に思った記憶があります．あとでわかったことですが，当時，中華人民共和国には 32 プレーンを割り当てる案でしたが，かれらのいい分は，中国は 200(?) 以上の民族がいて，とても 32 プレーンにはおさまらない，ということでした．結果的に Unicode 派を利することになってしまった訳です．

日本の National_Body の受取り方は，東京にある某帝国大学の先生に御伺いを立てたところ，「まぁーいいんじゃないの」と

いうことで....:-(

先頭は切ってはいないかも知れませんが，提案はされました．ここらあたりの事情を知っている人は少なからずいると思いますが，Ydoc では知らされていないのかも知れません．

noonaka> ふがいなさを反省する必要は....

もちろん，みんな反省していると思います．中国，韓国の人も交えて議論した結果，投票で負けたのですから....

北京まで出かけて中国科学技術院と議論したり，韓国の KAIST/ETRI の人たちとも議論しましたし，シンガポール，マレーシアの人たちとも意見交換をした結果が DIS(Draft_International_Standard) にも最初はできなかったわけですから....

マーケットを支配すればよいという問題ではない気がします．私は，Shifted-JIS の Code-Set の決定に関与した経験から，よい案がマーケットを必ずしも支配しないことをよく知っています．Code-Set は，その時代の技術/経済効果のバランスの上に成り立つモノだと思っています．決めた時はバランスがよくても，コンピュータのように技術革新が激しい分野では，すぐに非難を浴びることになります．いまは，2000 年までの技術革新を織り込んで考える必要があると思っています．

Date: Sun, 15 Aug 93 05:19:25
From: hihara
---

nonaka> その理論的可能性について....

いまが日曜日の午前 5 時だからというわけではないのですが，どうもこの「理論的」というのがひっかかります．何を持って「理論」というのでしょうかね?

僕なりに「理論的」にいうのならば，大阪弁でさえ必ずしも関東弁と一致しないのだから，ましてや外国語がユニファイできるとは思えません．

なんだか，「理論」というのを軽々しく使ってほしくないなぁ．ほんとうに「理論」といえるものはディラックの相対論的量子力学のように美しいものなのだと思いますよ．

時代により，国により異なって行く漢字をどうやって「理論的」に扱うのですか? 中国の略字の増大のどこが「理論的」なのか? とか，日本でも戦後の漢字の簡略化のどこが「理論的」なのか? とか，いろいろ疑問がわくんですよね．

#何で UNIX のバージョンアップって 2 ～ 3 時間で終わらないの?

Date: Sun, 15 Aug 93 05:30:08
From: hihara
---

nonaka> われわれが先頭をきって ISO へ提案....

これって現状の認識に合っていませんよ．私の支持する TRON 文字体系を含む BTRON は，マイクロソフトがカーラ·ヒルズさんに申し入れて，「非関税障壁」ということで CEC の標準案と一緒に葬り去られてしまったのは，周知の事実だったと思うんですけどね．

Date: Sun, 15 Aug 93 05:42:30
From: hihara
---

inaba> もう少し説明してくださいますか?

僕の考えている「何らかの」属性は，藤野さんのように高尚なものではなく，検索·置換などのメソッドです．

#もともと，TRON プロジェクトは，徹底的にオブジェクト指向にしてしまったら，効率が落ちてリアルタイム性がなくなる，というポリシーなのだ．:-) いかにも，ハード屋さん向きでしょ．

Date: Sun, 15 Aug 1993 09:09:54
From: nonaka
---

hiro-f> 思いだしたことをコメント....

大変参考になりました．長老の :-) ありがたい意見が出ましたので，ここらで軽くまとめを．

(1) CJK ユニファイの可能性について

現実的には，不可能．クラス NP に属すという表現がいいのかな? 先日，和田先生が fj に書かれていた，" 無謀な試み" という表現は，実に的を得ている．

(2) 文化侵略はあったのか?

これは，相対的なもの．稲葉さんが文化侵略的と指摘した事実は；

(a) ユニファイ基準適用の一貫性の欠如．
(b) 16bit 空間に文字を詰め込むことを優先し，必要欠くべからざる文字が，切り捨てられた．
(c) チェスの駒はあるのに，将棋の駒はない

といったことだったと思いますが，これらはいずれも，私の感覚では，侵略にはあたりません．とくに，議論を経た投票という手続きがとられたことを考えると．しかしながら，侵略を感じた人が存在したしたことは事実であるようです．したがって，「妄想」ではない「被害」が存在したのも，これまた事実．よって，「被害妄想である」という部分は撤回します (なんか，政府の答弁みたい!)．

Date: Sun, 15 Aug 93 17:00:43
From: nakakita

コード関係は勉強してないので，だまって聞かせていただいていたのですが，

nonaka> The Unicode よりすぐれたれた体系を提案....

私もそう思います．画面で見える「上」がたかが，14 ドットとか 24 ドットなら，私は何コードでもかまわないのですが，漢字だけを取り上げても，書き順とか，毛筆の世界になってくると，国が違えば文字といってもいろいろな要素が必要になってくるように思います．Unicode のコードも表わそうとしている文字の単なるある 1 つの要素だけの気がしてしまいます．もっと大きな枠を提案して行かなければいけないと思います．

Date: Sun, 15 Aug 93 20:17:54
From: pv
---

inaba> ふっふっふ，いよいよ pv 氏と....

体力的に自信がないものでどうぞお手柔らかに．:-)

初めにわたしの主張がどのような stance からなされているのかを述べます．ちょっと抽象的ですが，心に留めておいて下さい．principle は；

(1) 人間 (活動) の本質は「さらなる複雑性の追求」にある．

(2) 評価の第一基準は，その「追求」を可能たらしめる程度にある．

です．ここからわれわれ人類がもう過去の simple さや independency に逆戻りできない，ということがわかりますね．もう 1 つは computation の影響力で；

(3) 情報化により，われわれは好むと好まざるとにかかわらず，予想もしないような範囲で相互に影響しあうようになっている．

(4) 何かを決定することの「影響範囲」を a-priori に把握すること自体が不可能になってきている．特に「情報」の domain で．

という認識があります．「文明論」的立場かな? つまり情報化のお蔭で，domain の arbitrary な boundary の設定はもはや不可能だ，というのが私の主張の骨子です．

inaba> 標準は文書化されていなければいけない．

たとえば，ambiguity は，こういった用語に，もうすでに存在します．そもそも「文書化」されている/いないの判断について，それがつねに universal に可能かどうかを，いつ，だれが保証できるのでしょうか? 「文書化」は「文書化可能」な対象のみを前提とする，という boundary の設定を暗黙のうちに認めなくては，成立しません．

で，わたしは「character set」は，すでに，この範疇に当てはまる対象ではなくなっている，と考えているわけです．その理由は，「character set」という用語によって示される概念が，背景としてきている「文字文明」によって異なっているように見えるからです．ですから，標準化の対象である「character set」自体の共通な「記述」も不可能にならざるをえないのではないか? と主張しているわけです．そして，今後，情報処理分野では，さらにこのような「逸脱物」が対象として取り扱われて行くようになるでしょう．これは，上述の principle (1) からの本質的な帰結です．

inaba> (1) 使用域明確化の原理 inaba> (2) 汎用性の原理 inaba> (3) 経済性の原理

ここに述べられた標準化の scheme では，標準化の「影響の範囲」について厳密に考察されているとはいいがたいのでは? つまり principle (3) & (4) の視点からは，「標準化」活動の「独立性 (影響伝搬)」が a-priori に判断可能でない場合が多々出てきているので，その枠組自体を問い直す時期に来ているのでないか? と思っています．

inaba> 参加国のうちに利益不利益があっては....

ここまで情報化による globalization が進むと，「参加国」の「利益／不利益」とかも，非常に naive な問題になってしまいますね．その影響の及ぶ boundary はもう a-priori ではないでしょう．

いわゆる従来の「工業製品」についての guidance を延長して，「情報」分野の構成要素についても apply しようとすること自体に無理がきているのではないでしょうか?

inaba> 前回の mail における私の立場ですが,....

この考えは私も理解しています，暫定的にその「範疇」でわたしもこうやって皆さんと「意思疎通」させてもらっていますし.... (_)

しかしわたしの主張が問題としているのは，まさしくこの「噛み合わない」点にこそ存在してまして....

つまり，そのような「domain の絞り込み」自体の正当性に疑義が生じてきている，ということなのですよ．

つまり「工学的 code」の設定は「工学的 domain」のみにしか影響しない，ということが通用しないのではないか? といっているのです．これは「情報処理」がすでに学際的な位置にあって，princple (1) の追求において「言語活動」と同等の役割を果たしつつある，という認識からの帰結です．

たとえば，すでに computer 処理は文科系の学問でも行なわれています．当然，教育にも使われますし，日常の communication にも使われてしまうでしょう．そのおのおのの domain に「適切な」情報の re-presentation を行なえばすむといっても，「情報処理」自体の役割が glue みたいなものですから，その意味での「情報表現」の均質化への tendency はつねに存在してしまいます．

で，character set の (場合のように) arbitrary な「切り出し」は，結果として，さまざまな他の domain の知的活動に対する restriction となってしまう恐れがある．したがって，もっと他分野との連係をとるべきだろうし，もっともっと慎重であってよいのではないか? と考えているわけです．たとえば，その背景をなす文明 (way-of-thinking) と切り離して扱うこと自体がすでに問題かな? とも考えていますが．

では，どうするか? 文明 (way-of-thinking) を同じくする人々の範疇で，適当に representation の工夫が行なえる程度の「帯域」を与える．かつ transfer のため (だけの，したがってかなりの属性が落ちるぞ，ということ) の「便宜的 expression」も用意する．というようにしてはどうでしょうか? 当然，「帯域」は open ended ですよね? 適時に見直され，まったく architecture の違う体系が，その都度施行される，ということを排除しない．「標準」とかいうと「偉そう」なので，「暫定 (追加) 措置」とか呼んでね．

つまり，短期的な「経済性」よりは，princple (2) による評価軸を尊重するわけですね，これからは．diversity の尊重が本質で，「標準化」は「どうしようもない奴にしょうがないけどこれだけでも伝えるか!」ってとき以外には使わない，というような意識を皆が持つようになれば，しめたものだけど.... (^,^)

inaba> でも私の立場でいけば,....

もちろん，そういう approach もあるでしょう．でも，「ある言語で情報を交換するのに必要な文字が有限個ある」っていう命題自体を問い直すことも可能性としては追求したいな，折角，computer という tool が手に入ったのだから．

princple (1) の追求に「必要な」言語や文字がもっと出現してもよいわけで，たとえばいま，Glingon 語でしたっけ，けっこうマジにやられてるでしょう？ 私は，そのような activity をもっとやってもよいように思ってるわけで，そのさいに，「標準化」という activity の結果が「制約」になりつつあるように感じているのですよ，cognitive に．

inaba> 2 次元空間というのは....?

基本的に表音文字しか atom として持たない場合，意味の表現 (単位) には，それらの combination を用いざるを得ません．立体的に組み合わせてもよいのですが，でも一瞥での認識容易性 (読む場合のこと) の問題から，まぁ大体，2 次元以下の combination になるでしょう．再現性 (書く場合のこと) の容易さなどを考えると，1 次元的並びになるかな？ Latin 文字はこれね．atom 自体 (α から Ω など) はそれのみでの識別が必要だから，2 次元的な相違をそれぞれに持たせざるをえない．で，伝達の道具としては，情報の受け手にどれだけ pricise に entity を rememberance させうるか？ が問題となるわけで，たとえば delimiter (空白のこと) の存在なんかも，それがないと ambiguity が高過ぎるわけです，意味の表現単位を切り出すのに．

多分，Greek あるいはそれ以前の Sumale (? spelling unknown) など，早くから「少ない atom で universal な表現を可能とする」やり方を追求していたのでしょう．code という概念にすでにそれが伺えますし．あるいは「再現性」をより重要視していたのかも知れません．

inaba> multi-dimensional にすると何が変わりますか?

表意文字の発展経過をみると，「対象の模写」→「抽象化」→ある「漢字」というふうになっています．模写の段階で 3 次元以上での効果的な表現 (保存／伝達) 手段が当時はなかったので，結局 2 次元による表現になったのだと思います．

ここからは当然の進展で，ある「漢字」を atom として扱うことで，その 2 次元的な combination によって，さらに複雑な「漢字」を表現することが可能になります．組み合わせの rule は次第に複雑化し，たとえば，現実に「存在する表象の組み合わせ」による新たな (意味的) 表象の産出にとどまらず，口頭表現の便宜のために，漢字の構成要素の一部に「発音」を表わす漢字を組み合わせたりするようになっています．

また，delimiter を必要としないというのも特徴的ですね．こういう characteristics を切り離して，同一の「character set」という概念範疇で取り扱ってしまってよいのか？ が疑問なのですよ．

で，computarized されたわれわれは，従来と同じ「認識容易性」や「再現性」の枠 (つまり 2-dimensional という制約) にいつまでも拘泥している必要性もないだろうから，たとえば「漢字」における構成的な表現方法を multi-dimensional に拡張した「表現」手段を模索してもよいのではないか？ と考えているわけです．

chess の駒のように 3 次元表現していたら.... と考えるとたのしいですよね，そうすれば，たとえば「context」とか「nuance」とか，もっとさまざまな情報を entity と一緒に「直接」的に取り扱えるようになるでしょう．

そう，text-based e-mail においてさまざまな「意匠」すなわち :-) とか

(^.^) とかが生まれたように．

inaba> IS の立場でいっても，この問題は....

たしかにそうです．ですが，上に述べたように，このような「学際的領域」での activity は，本質的に現状の，たとえば「character set の標準化」といった結果のinfluence から独立ではありえないという点で，「文明 (哲学?)」的側面からの配慮が必要だし，標準化の対象の絞り込みにしても，工学関係者だけで arbitrary に行なうことはもはや不適切ではないだろうか？ と主張しているのですよ．

inaba> ただ，私は "A" と「あ」と「亜」....

漢字は character よりも word に近い側面を多々持っていますよね，概念的には．じゃ，word の標準化ができるか？ それは何を意味するかということを同時に考えてみてはどうでしょうか?

inaba> そのドメインにおける研究が活発でない....

これはおっしゃる通り．

ibnaba> 変な restriction だとは思いません．

この意味ではおっしゃる通り．

とりあえず，今日はここまで．

Date: Mon, 16 Aug 93 11:40:18
From: pv

---

nonaka> 本質的とは考えていません．

ここにも本質的な問題が存在しますね，「表意文字」文明はこのような「表現上の tolerance」を許容するのが特徴です．つまり，表音文字の構成単位が code と呼べるのに対し，それよりは tag，つまり「受け手」がある entity を憶い起こす rememberance の役割という意味合いが強いですから．

つまり「客体」としての han-ji character そのものについての precise な記述が追求すべき本質なのではなく，「書き手」が意図していた entity と「同じ (analogically に)」ものを「受け手」に「想起」させる「きっかけ」としての役割が重視されるべきだと思います．

X208 の時代の computer と現在のそれとでは，大きく状況が違ってきていますから，「当用漢字」などという最早意味のない枠組の廃棄とともに，character set の「本質的な役割」から computer 上における「表現」形式を検討してもよいと思いますね，まぁ，そのためにはかなりの人文/社会科学など様々な分野の人の協力を必要とするでしょうが．

nonaka> 自然な延長線上にある．

ですからこの前提は私には受け入れ難いわけです．:-)

nonaka> 工学的に合理的であると評価.....

では usage や essential properties という言葉で表わされる対象が，異なる文明においても「抽出可能」でかつ「比較可能 (equivalent)」であることを，まず，証明すべきではないのでしょうかね?

わたしは unify の本質的な不可能性をここに「感じて」しまいます．もし，可能であるとするなら，character というものに

対する解釈自体に,「恣意性」を排除した「限定された」定義を与えて欲しいな.

Date: Mon, 16 Aug 1993 11:54:33
From: nonaka
---

hihara> カーラ・ヒルズさんに申し入れて....

つまり,その(美しい)BTRONが,おばさん一人によって葬り去られてしまったことへの反省は必要ではないかと思いませんか?

Date: Mon, 16 Aug 93 12:38:03
From: pv
---

ibnaba> pv氏の好きな観点は.... 待ってね.

湯治から復帰しましたので,また「揉んで」やってくださいませ.(__)

inaba> 嘘という方が近いんでは....

Mr. Inabaに座布団1枚!! :-)

inaba> (1) この principle は ....
inaba> (2) この principle によれば....

その他に日本固有の問題ですが,芸:ウン,「香り草」のことがあります.

たとえば陰暦11月を「芸生(ウン・セイ)」,「芸夫(ウン・プ)」で草苅り男のことを指す漢字の「ウン」がありますが,これは「芸:ゲイ(いわゆる藝の「誤った」略字)」とは違うものです.しかし日本では「藝(本字)」の本来の略字である「草冠+ケモノ偏+丸+云」や,藝の同字の「草冠+ケモノ偏+丸」が無視され,本字の「芸」を本字「藝」の「略字」としてしまっています.このような誤った「略字」や「俗字」化によるconflictの洗いだしと,それをどう解消するか,はstraightforwardには行なえないでしょう.

inaba> これらのことから私は....

これはまったく同感.

inaba> チェスの駒はdingbatだが....

なるほど,これは「手前味噌」すぎますねぇ.しかし,rookやknightやbishopでdingbatしたいので,これは許そうかな? LaTeXでの\footnotemarkで使いたいもの(^,^)

Date: Mon, 16 Aug 93 12:48:24
From: pv
---

hihara> 決めた過程がトレースできない標準化....

槍原さんのおっしゃる通り,「工学」というのならcomprehensiveであると同時にtracableであるべきですよね.

Date: Mon, 16 Aug 93 12:58:57
From: pv
---

hihara> コウモリのようだ. :-)

わたしはもっと過激に「粘菌」を目指しています. :-)

hihara> テンプレートにすべきだ....

なるほど,actual elementのreifyをぎりぎりまで遅らすわけですね.

hihara> 井の中の蛙になってしまって....

ほんとうに.もう少し広い視野を養って欲しいと思いますよね.

Date: Mon, 16 Aug 1993 13:09:31
From: nonaka
---

hihara>「理論的」というのがひっかかります.

ひっかかるのは,きっと午前5時だからでしょう. :-)

私が聞きたかったのは,"理論"という言葉の使い方にあるのではなく,「大阪文明,関東文明に属するわれわれ2人の人間が,当用漢字という文字セットを共有できるならば,日本文明と中国文明の間でも文字セットの共有は可能であろう」という命題を槍原さんはどう思いますか? ということなのであります.

まあ,大阪文明と関東文明の距離と,日本文明と中国文明の距離は,あまりにも違い過ぎ,本質的な差がある.ということでも,私はかまいませんが....

Date: Mon, 16 Aug 93 13:39:44
From: pv
---

nonaka> 専門家の判定をお願いしたい.

こういういい方がまずいんだって!「専門家」ってなにさ? たとえばAcade'mie Franc,aiseのような「由緒と見識」ある人々の集まりに該当するものが,日本にあるかしら?

まぁ,断腸亭とか観潮楼,森銑三とかと同程度の知見を有した人なら任せてもよいか? 現在なら,丸谷才一とか井上ひさし,筒井康隆なんかも,入れといて損はないかな?

でも,やはり,他人任せにしないで,おのおのが自分たちの言語や文明にもっと関心を持たないとね! computerという新しい世界に,それぞれの文明の可能性を展開する役割を担ってるわけだから.

nonaka> CJKに精通した専門家....

これに関しては「江戸時代」の儒学者連中がもっとも適任だったと思います,中国は清朝の時代にかなりの経典を焼いてますし,さらに文化大革命の影響もありますので,あまり(historicalな側面からは)適任とは思えませんね.

あと,たとえば瀬戸内海に面した地域では古くから朝鮮の影響を受けたと思われる(最近は古事記や日本書紀の記述も朝鮮語読みして新たな解釈が可能となったりしているけど)語幹の言葉などがありますから,単にunificationすべきか?/できるか?という問題を離れて,JKの共同作業は,非常に有意義だと思います.

nonaka> それはそれで1つの見識....

ここらへんから野中ちゃんと私のway-of-thinkingの隔たりがあるわけよね.

Date: Mon, 16 Aug 93 13:42:00
From: pv
---

nonaka> それなら，標準なんて決める必要ない....

逆に質問したいな，ノナカちゃんは「標準」というものをどういうふうに位置づけてるの?

Date: Mon, 16 Aug 93 14:52:02
From: pv
---

nonaka> 私の考え (あるいは妄想) は....

すでに述べたように,「できる/できない」という次元だけで考えるのは「片手落ち」です，影響範囲や computation の意義から問い直し，その上で「上記の妄想」 :-) 自体の是非を検討されることを要望してやみません.

実は，わたしも「CJK unification」が不可能だとか，許さないとかいってるわけではないのですよね，非常に限られた portion では，今日に至るもほとんど原典そのままの用法以外に usage や form の variation が存在しないものがありますし，それらの unification はあってもよいかな? と感じたりもしています.

しかし，中国や韓国での略字化，あるいは Hangul 文字化などはかなり早急に進められていて，いま (表意文字の connote するという特性を危険に晒して) 拙速の「形状一致」を求めることに，はたしてどの程度の意義があるのか? 疑問も少なくありません.

nonaka> このスレッドにに関しては終わりに...

まぁ!

nonaka> 次は，パワーゲーム論にしましょうか?

活躍を期待しております. (__)

Date: Mon, 16 Aug 1993 15:13:59
From: nonaka
---

pv> すでに述べたように....

もちろん,「できる/できない」という次元だけで考えるつもりは，ありません. 単に，サブ問題に分割して問題点をあきらかにしようとしただけです. ユニファイの問題点が，どのレベルにあるのかが，はっきりすれば，問題解決の糸口も見つけやすいでしょうから.

pv> .... といってるわけではないのですよね.

了解です. :-)

Date: Mon, 16 Aug 93 15:33:10
From: pv

8/27 ですが,「鰻」ではなく「寿司」にしました，江戸前の「焼き穴子」を頼んでありますので，夏の「鰻」より格段にいけます，関東では鱧の代わりに穴子が一般的ですが，中でも特に東京湾ものは絶品です.

inaba> 相当ウソを書いてしまいました.

たかが ML での「論争」ですから，気になさらずに.

inaba> 関係する標準のタイトルを一覧に...

いたみいります (__)

inaba>「違う応用には違うコードがあってもいい」

基本的には同意します，が,

inaba> .... を標準化したって別にいいんですよ,

もちろん，それはそれでやるべきでしょうが，たとえば「numerical control」では broard な information interchange (say, interaction) や，広範な意味表現能力とか，あるいは connotation の可能性などを前提とはしていない分野ですよね? ですが，X208 とかは「日常」の interaction と同程度の範囲の representation 能力 (を実現するに足る構成要素) を対象としています. ですから，私の主張する「影響」の及ぼす範囲という観点からは，この 2 つは比較にならないでしょう. あるいは適用 domain の boundary の設定が意味がないほど general な対象なのだと思います.

だから「慎重」であってもらいたいといってるのですよ.

inaba> KS C5601

KS はけっこう頻繁に更新がされてるようですね，また実装のほうの catch up もけっこう早いと聞いています.

inaba> あーしんどかった!

非常に助かりました，いやぁ，ご苦労さまでした. 感謝!!!

Date: Mon, 16 Aug 93 15:55:37
From: pv
---

inaba> DIS-10646 のメンバーもその危惧は....

非常に normal な approach ですね，経過についての minutes には目を通してないので不案内ですが，short range では mutual benefit を考慮した規格といえたと思います.

inaba> すでにその方面の標準化は相当進んで....

まぁ，CALS や EDI (ですよね?) などの information interchange が actual に機能している国々からは，より複雑な構成の entity をそのまま送りたい，という要求は強いでしょうからね. 日本は infra の networking がまだまだだから「蚊帳」の外か :-(

Date: Mon, 16 Aug 1993 16:33:21
From: nonaka

また少し整理を...

使用できる漢字の制限について

(A) そもそも，漢字は有限という考えは不合理 (藤野さん)
(B) X0208 には限定的合理性がある (稲葉さん)
(C) 漢字を制限するという考えに何の疑問を感じたことがない (野中)

という 3 つの立場が出てきたように思います. 檜原さんは，どれに属しますか?

さて，(A) は別スレッドで扱うとして，better Unicode のために，ちょっと思考実験をしてみたいと思います. The Unicode では，CJK あわせて約 20,000 字に収められましたが，ここで Shift Unicode なるものを考えます. これでは，日本語に約 7000 字弱 (いまの第 2 水準まで) を割り当てます. ここに，X0208 を適当にシフトして収めます.

これは，稲葉さんの立場からは，限定的合理性を持つと考えてもよろしいですか?

Date: Mon, 16 Aug 93 16:47:16
From: pv
---

inaba> かれらとわれわれでは判定基準が....

視覚による識別力の差はかなりありますね，OSF の I18N 担当の知人(日本人)もいってましたが，output の確認で，アメリカ人には，漢字はおろか「かな」でさえ，ほとんどその相違が識別できないようですから．

で，人間の total な cognitive processing との関係/影響なども考慮しないといけないな，と思ってるのですよ，このあたりは，例の Gibson の "The Ecological Approach to Visual Perception" に参考になる面白い実験結果が載っています．日本語訳は「生態学的視覚論」というタイトルでサイエンス社から出ていますが，原書の方が 100 倍くらいわかりやすい，日本語が厳密な論理的記述に「適さない」という典型的な実例といえましょう．

inaba> これはもはや工学より科学の立場....

なるほど．面白いですね．私は科学には暗いのですが，たとえば「3 体問題」などは最近の物理学では「観測不能」ではなかったでしたっけ，違ったかな? 「計ることができる」ための条件は何なのでしょう? あ，これらは純粋に質問ですよ．Mr.Inaba の土俵に登ろうなんて無謀なことはしません．(^^)

inaba> yes or no なら「非合理」です．

全面的に同意．

inaba> もともと審議会で....

まぁ，構成メンバもひどいし，決定迄の scheme も「不条理」でしたから．

inaba> つまり，X0208 というものは....

ふむふむ，ここらあたりから違ってくるわけね，私の立場と．

inaba> 綴り字法の確立と同じレベルに....

いや，それはわたしも異論ありませんよ，それどころか近いですね，「ヨーロッパ言語における綴り字法の確立と同じレベル」というのも問題ないです，多分．相違は「漢字の書き方」そのものより，「漢字の構成法」を extrapolate した symbol 構成を追求しようとしてるところかな?

まぁ，あとは Academie Francase 程度の「言葉」に対する care というか maintenance というかをまず行なうべきだ，というところかな? 要は Shakespear の Tempest や Sonnet は digital 化されても origin のままで味わいたい．同様に菅茶山や亀田鵬斎の漢詩も「常用漢字」でなくそのまま味わいたいし，参照したりしたいと願っているのですよ．

inaba> 問題の差を強く指摘したい....

これはよくわかりました．

inaba>「核融合発電」は実現する....

えっ，事故を含んでも「採算」があうの? ほんと?

Date: Mon, 16 Aug 93 16:57:37
From: pv
---

inaba> 今月の 2 次会 (ギリシャ料理とかね :-)....

残念ながら．8 月は 2 次会なしのいきなり「会食」なの．スピーカはだから食えないのよ．みんなが食べてる間，しゃべるのだから :-)

inaba> たとえば，言語標準でいえば,....

これがまぁ，言語の本質のように思ってますが．programming language といえどその呪縛からは逃げられないわけで....

inaba> だって「例を出せ」ってことでしたから....

いろいろと参考になります．賢くなれたかな?

Date: Mon, 16 Aug 1993 17:21:08
From: nonaka
---

pv> こういういい方がまずいんだって．

まあ，これも分業あるいはモジュール化ということで，ご勘弁を．ステレオタイプの思考停止というご批判も，甘んじて受けます．:-)

Date: Mon, 16 Aug 93 18:29:47
From: pv
---

inaba> 適当に議論の priority を....

おおせの通りで．取りあえず 1 round したら決めます．

inaba> むしろ「(西洋)哲学の」というべき....

もちろん，哲学も科学の影響を受けてますから (相互作用かな?)．観測の結果，それまでの paradigm にもとづく予測とは異なった現象がいくつか発見された場合に，次の paradigm を産むのは結局「直観」のように見えますが....

あまり際だった paradaigm 変革ではありませんが，たとえば宇宙における dark matter の存在予測とか．

inaba>「僕がやったらうまく行った....

そうそう，分子生物学などでも同様の現象があるらしいですね，「かれがやると必ず colony が全滅する」とかね (^^)

例の kimera rat とかも実験の担当者が「成功しろ」って念じるとうまく行き，そうでないとまったく発芽しないとかあると聞いています．

inaba> 事実の指摘でない私の意見としては,....

おっしゃる通りで．

inaba> 科学 (特に物理ですが) というものは....

フ～む，これは micro level でも macro level でもいえるのでしょうか?

inaba> 高エネルギー物理学では....

「モノを割っていったらどうなるの」を追求したいという dynamics の裏には，「単純で唯一の真理がある筈」という強迫観念が存在するように思えます．

たとえば，ある化学溶液があって，そのうちのある構成要素の濃度は時間の経過とともにある特定の閾値で catastrophic に 2 つの方向に jump (急激に増える/減る) するとしましょう．どちらに jump するかは「確率論的」にはつねに等しいはずですが，実際は「同じ」溶液ではどちらか一方への jump しか生じない，という現象があります．

ひとたび，この jump の方向が決まってしまうと，その溶液

は必ずその決まった「方向」にしか jump しないようになり，あたかもどちらの「系」から来たのかを「記憶」している，としか考えられないような状況が生じます．

同じ「組成」でも「どこから来たか」によって異なる振舞いをする，つまり時間軸について「不可逆」な現象についてはどう考えるのでしょうか?

inaba> 日本の専門教育の影響....

賛成です，もっと他分野に興味を持って diversity を豊にしてもらいたいし，学際的領域で相互にもっと力になれるはずでしょうから．

inaba> 研究の動機に差が....

なるほど．

inaba> レッテルが張られることはない....

やればよいだけってか! (^^)

inaba>「.... につきあえ」といわれたら...

わかります．:-)

inaba>「文字」の話に近いところでいえば....

例の Gibson の実験/研究などを見ると，たしかに興味深いですね．さらに，最近の神経生理学における研究からは，たとえば，

> — 単純ないし複雑な幾何学的形態をとり，一様あるいは斑紋状の spectrum 構成をもった多くの配置が，相互に区別できない色彩経験を生み出す．(中略) 網膜の活動を有機体の外的な物理的刺激に関係づける代わりに，主体の色彩経験に関連づけたらどうだろうか．(中略) いい換えれば，新しい方法では神経システムの活動を，外界でなく神経自体によって規定されるものとして扱わねばならない．したがって，外界は神経システムに内的に規定された活動がひきおこされるにあたっては，引金の役割しか果たしていない．私達はこれを厳密に遂行し，このような方法によって実際に観察者の色彩空間全体をつくり出せることを示した．[Maturana, H.R., G. Uribe, S. Frenk 1968]

とあるように，「知覚は外的現実の把握ではなく，むしろ外的現実の特徴化だとみなすべき」という方向に移ってきています．

つまり，神経システムは「知覚」と「幻覚」を区別しない，できないという「事実 (としての consensus)」ができつつあるようです．こういったことがらと，「表意文字」文明圏の人の「識別」能力との関係などを調べたいと思っているのですが....

Date: Mon, 16 Aug 93 18:39:34
From: pv
---

nonaka> CJK の各専門家達の討議による合意

丸谷才一とか，筒井康隆とかを入れば私は許します．:-)

nonaka> 不幸なできごとであるとは....

ここが問題．北米先住民族の悲哀とか，ちゃんと勉強してからいってね．軽々しく「不幸なできごとであるとは考えていません」というのは，ノナカちゃん的宇宙でだけならかまいませんが，現実的世界では，やはり問題になると思うよ．

nonaka> 変わってるつもりはないのですが....

性格を疑りたくなってきたなぁ，もう! (^^)

Date: Mon, 16 Aug 93 18:46:55
From: pv
---

nonaka> 話が通じないのは当然ですね

そうやって人は段々賢くなるのだよ，まぁ，「盲蛇に怖じず」で何でも思った通りを発言することの価値 (番勇?) は認めますけどね．

霞もたまには食べてみたら．意外といけるよ (^.^)

Date: Mon, 16 Aug 93 19:01:16
From: pv
---

nonaka> なぜ，言語というドメインを....

いわゆる locale の切替とどう程度の意味であった方が便利．

nonaka> なんでもありでよいのでは...

前回はいい忘れましたが，[...] 内の属性/値 pair を arbitrary に省略 (universal に matching 可能とすることで) した general な superset を考えることができます．同様に specialize された subset も可能ですから，たとえば，関東/関西 domain などはこういった subset として扱えるでしょう，多分．

漢字/ roman あたりまではできるかも．究極の superset が「地球」domain かな? でも抽象化されたところが多過ぎて実用にはならないな．

だって，5 以上の数は「たくさん」としか表わす術がないとか，「旅行」という概念がないとか，いろいろな言語がありますからね．

Date: Mon, 16 Aug 93 19:32:43
From: pv
---

nonaka> これは強制されている状態なのでしょうか?

というか，「当用漢字」という「不条理」をそのまま疑問も抱かずに X208 のような標準にしてしまうことだよ，たしかに cost の問題もあったけど，「画数の多い字は書くのに不便」とか「覚えるのが大変だから略字化する」という「不条理」をそのまま引き継がなくてもよいでしょう．

systenatic に扱うには「正字/正仮名」が一番で，画数の多さもかえって remeberance としての意味あいでは有効だし，工学的に評価して較べてみても coherence はすぐれているはず．

で一度，情報交換の element として使われると，それは cancer のように広がるからね．一種の fascio 的な働きをするでしょ? 私は適宜，好みの (同意の)「漢字」を宛字したいのだけど，たとえば「隣：リン」と「鄰」は完全に同字なので，どっちを使ってもよいのですよ，ほんとうは．

ところが「第 2 水準」とかに置かれることになればどうしたって「鄰」は「例外的」に扱われることになる．実際，「リン」や「となり」と入力して「鄰」が素直に出るシステムなんかない．

こうして「知らず知らず」のうちに利用可能な「表現手段」に制約がかかるのだよ．これを fascio といわずして何を指す．で，私はすべての fascio 的なものには，断固反対します，もう

生理的だねこれは.

というわけで，見識が疑われるよ，fascio の片棒担ぎは.

Date: Mon, 16 Aug 93 19:41:39
From: pv
---

nonaka> .... できなかったのでしょうか?

簡単です，globalization に伴い，「経済性」という指針が暗黙の primarily な価値観/評価基準となったからですね.

nonaka> 文句ををいうだけでよいのでしょうか?

というか，世界規模では何がいま最大かつ最優先の課題かというと，「macro 経済」の initiative なのですよ，あるいは controlability かな? だから「拝金主義的」価値観が大手を振り，効率化の名のもとに「文明」fascio が起きているのだと思います.

これは冗談ではないよ.

Date: Mon, 16 Aug 93 19:51:44
From: pv
---

hiro-f> 結果的に Unicode 派を利することに....

なるほど，そういう経過があったのですか.

hiro-f> 日本の National_Body の受取り方は....

そうなのですよね，無関心派が多いから.

hiro-f> Ydoc では知らされていないのかも....

たしか SEA の Forum でも一度取り上げられましたよね．経過の詳細に暗いので，そのうちに教示いただければ幸いです (_o_)

hiro-f> もちろん，みんな反省している....

変なたとえですが「相撲とり」が「boxing」の rule で戦うようなものでしょうか.

hiro-f> 2000 年までの技術革新を織り込んで....

同感．時間軸上での変遷にともなう拡張容易性とかも，「経済性」の評価の重要なfactor にすべきだと思っているのですが，そのような配慮は現状 (の国際標準の activity) では，ほとんどなされていないようですね.

Date: Tue, 17 Aug 93 09:40:12
From: hihara
---

nonaka> おばさん 1 人によって....

どーも腑に落ちない点を 2 つ.

(1) いくら個人主義のアメリカでも，ヒルズさんの言動の裏に何人ものスタッフ，陳情者がいることは明白．彼女の動きはその時点でのアメリカの動向の代表でもあったわけです.

(2) 上の文章は (美しい) という皮肉が混じっていますね．日本人には同じ日本人の業績を認めない人がかなりいる．野中さんがアメリカのコンピュータを持ち上げている間にも，合理的なアップルの社員は，とっくに BTRON の真似をして分割式のキーボードを出してしまったのよ.

Date: Tue, 17 Aug 93 09:53:52
From: hihara
---

pv>「工学」というのなら....

このレベルでは藤野さんと意見が合いますね．現実問題として，100% comprehensive な規格は難しい，というのと，規格には四半期以上の寿命があってしかるべきだ，(つまり変化する過程を認めなければいけない) という，泥臭い背景があります.

Date: Tue, 17 Aug 93 10:00:04
From: hihara

あまり浮き世離れした議論は好きではないのですが....

nonaka> 大阪文明，関東文明....

言葉の捉え方が違うかもしれませんが，大阪と東京の違いは「文化」ですよ．文明は共有しています.

Date: Tue, 17 Aug 93 10:10:02
From: hihara
---

nonaka> また少し整理を...

これ，全然整理されてないよ．だって，執拗に「コードさえ決めればよい」と主張し続けているだけだもの.

nonaka> また思考実験をして....

コード空間を増やせばよい，という問題はとっくに議論の中では寿命が尽きたと思っていたのですが....

Date: Tue, 17 Aug 1993 11:20:14
From: nonaka
---

hihara> あまり浮き世離れした議論は....

ならば，まあやめておきましょう.

hiharar> 大阪と東京の違いは「文化」です.

これには異議がありますが，しばらく保留しておきましょう.

Date: Tue, 17 Aug 1993 11:39:43
From: nonaka
---

hihara> (1) いくら個人主義のアメリカでも....

そんなこと，十分わかってるつもりなのですが...

別に，おばさんが 1 人だとか 100 人だとか，アメリカは軍隊を持っていて，核ミサイルがあるとかを問題にしたいのじゃないですよ．「葬り去られた」(かどうかは，私は関知しないが) という事実をどうとらえ，次に生かすか? ということですね.

hihara> (2) 上の文章は (美しい) という皮肉が....

私の坂村氏に対する評価はとても高いですよ．TRON は，その最初の頃から，商業的成功を強く意図していましたよね．坂村さんは，「アメリカに負けた」というような低レベルのところで思考停止せず，ちゃんとその事実を反省し，きっと次のシナリオを考え，実行しているだろうと思いますから.

Date: Tue, 17 Aug 1993 11:56:59
From: nonaka
---

hihara> 主張し続けているだけ....

主張をしてるつもりはさらさらないので，もう一度整理しなおしてみます．

(A) 文字コードの枠組みを根本から考え直す．
(B) 現在の JIS 等の限定的合理性を認め，マイナーチェンジとしての改良を考える．

これではいかが? 両方独立に議論可能ですね．

私としては，仕事上の都合 (次の漢字 TALK をどうするかってことですが) が，かなり切羽詰まっているので，(B) のテーマで，もう少しみなさんの意見を聞きたかったのですが，それには興味がない (あるいは疲れた) ということなら，やめときますが...

Date: Tue, 17 Aug 93 11:58:47
From: "Dr.K"
---

nonaka> 私の坂村氏に対する評価は....

「商業的成功」というのは，平たくいうと「オレこんなに金もうけしたんダゼ!」ということのように思えるのですが，これだとそこらの悪徳不動産会社の社長とか変わらんような気がする．

いいものを広く普及させる >>> 人類への貢献というのならわかるのですが....

Date: Tue, 17 Aug 1993 12:01:48
From: nonaka

Dr.K> いいものを広く普及させる....

もちろん，そのつもりです．

Date: Tue, 17 Aug 93 12:01:48
From: pv

ちょっとギロンの雲行きがあやしくなってきてますね．

nonaka> そんなこと，充分わかってるつもり....

ノナカちゃんは power game の「勝ち負け」に関連した各自の reflection を期待 (問題と) したいといってるのですね?

nonaka> 坂村さんは,.....

それはそれで「１つの」選択でしょう，で，そのような approach が別に唯一であるわけでもないし，人はそれぞれですから，たとえば私みたいに power game そのものを「相手」にしないという立場もあるわけだ，霞喰いながら．:-)

で，ノナカちゃんは結局，何をいおうとしているの?

Date: Tue, 17 Aug 1993 12:30:09
From: nonaka
---

pv> ちょっとギロンの雲行きがあやしく....

はあ，どうも檜原氏には，テニス部の後輩という意識が強く，負けたくないという意地が働いてしまいます．以後気をつけます．

pv> power game の「勝ち負け」に関連した....

そうですが，もう少しカッコよくいえば，Dr.K のいう人類への貢献かな?

pv> で，ノナカちゃんは結局....

ウチの社長のマイケル･スピンドラー氏のスピーチで感動した言葉があります．

— 世の中を動かしているものは２つある．お金と情熱だ．どちらが欠けてもいけない．

売れない CASE ツールのセールスマンを２年もやると，やはり，後遺症があるみたいですね．少し，お金にこだわり過ぎかも知れない．でもね，やっぱり「XEROX がもっとちゃんとマーケッティングしてれば，いまの計算機環境はもっとよくなっていたはずだ」という反省は必要だと思います．

Date: Tue, 17 Aug 93 12:44:18
From: pv
---

nonaka> では，もう一度整理しなおして....

(B) の意図がよくわかりません，前節を認めたとして，minor change を必要とする対象とその理由が具体的に思いあたらないので．具体的な問題を示してもらえれば「思考実験」の rat になってもいいよ．:-)

nonaka> 私としては，仕事上の都合....

前の mail の「Shift Unicode 〜」の意味もよくわからなかったので，もう少し詳しく説明してもらえますか? Unicode の目指した unification は置いといて，等価な representation capability を持つ character set をまったく別に考えようということなのかしら?

Date: Tue, 17 Aug 93 12:53:38
From: inaba

休んでいる間に相当たまったので，まずは事実に関係することに対するイチャモンだけを書きます．ちょっと走り書きなんで，あらっぽいです．

pv> 従来の「工業製品」についての guidance を....

これに関しては，今回の議論で僕が前提としている「(工業) 標準としてのコード」という範疇からは出てしまいますね．新たに「情報」は「工業」の範疇のみでよいか? という議論をすること自身は，意味があるとは思いますが，それなら ISO-10646 も "The Unicode" も当面は関係ない (というかそれらを参考にしてもいいから，独立して決めるべきだというのが僕の立場)．

pv> 少ない atom で universal な表現を....

poor character set 文化ってのは，筆という万能の筆記具を持ちえなかったから出たという仮説を読んだことがあります．だとすると，「少ない atom」主義や，「どんどん割って行く」主義というのが，単純に筆記具の差から出たということになります．

pv> delimiter を必要としない....

英語で書くとわかりにくい (ことをねらっている?) けど，日

本語で文字といった場合に "A" や「亜」を指すことには同意いただけると思います．とすると，同じ概念範疇で取り扱うことに (工業) 標準としては問題がなかろうというのが私の立場です．

pv> じゃ，word の標準化ができるか?

逆に "colour" や "color" や「色」に同じ「語」という概念を map してそれをコード化するということも，純技術的には興味深いことだとは思いますし，そうすることで何等かのメリットがあればそれが工業標準になったって不思議はない．

pv> たとえば瀬戸内海に面した地域では....

そういう意見もありますが，この点に関しては豊田有恒氏やいま週刊ポストに連載されている (タイトルを忘れた) ものなどもお読みいただければ幸いです．僕にすれば「古事記朝鮮語説」の方が "The Unicode" なんかよりよっぽど文化侵略にしか思えないのですが...

pv> たとえば「3 体問題....

もしこれが「天体の 3 体問題」を指しているのなら答えは NO です．あくまで，5 次の方程式が解析的にとけないのと同じく，一般解は解析的に求められないだけの話です．

pv> 要は Shakespear の Tempest や....

僕も online で「大乗仏典」を読んだり，「日本書紀」を読んだりしたいという希望はありますね．

pv> 事故を含んでも「採算」があうの?

第一議的に考えている「採算」は「エネルギーコスト」です．経済コストは考えていません．

pv> dark matter の存在予測とか....

dark matter ってそんな大きなパラダイム･シフトですかねぇ．ものがないとつじつまがあわないから仮想的にモノを考えるって，ごく自然ですが....

pv> フ〜む，これは micro level でも....

と思います．逆に成立しないとすれば，どんな例があるでしょう．

pv> 強迫観念が存在する...

これを脅迫観念というかどうかは疑問ですし，「唯一」というのは数え方の問題 (Maxwell の方程式は 4 つですからね) かと思いますが，考えれば考えるほどものは単純になるというのは，「この手の」物理にはつきまとうでしょうね．でもこれは科学の特性かというと疑問です．地球物理なんかは違うもん．

pv>「当用漢字」という「不条理」を....

工業標準がよって立つルールを考えると「悪法といえども法なり」というのは私は正しいと思います．

pv> というか，世界規模では何がいま...

でもって，工業標準ってのはそのための道具ですから．だから，固有の文化だとかなんとかいわれても，アメリカ人がインチを使うのは許さない．

Date: Tue, 17 Aug 93 13:38:55
From: pv

---

nonaka> テニス部の後輩という意識が....

別にいけないといってるわけじゃないので．:-)

そういう 2 人の関係がわかれば，「なんだ，じゃれあってるのか!」って納得できますから (^^)

nonaka> 人類への貢献かな?

だとするなら「どの側面から」貢献したいのか? どうなれば「貢献した」ことになるのかを前以って示し (それ自体は各自の思い込みだから特に議論の対象にはならないので) た方がいいな．こっちもノナカちゃんの stance が推測できるから．

nonaka> 世の中を動かしているものは 2 つある．

なるほど，stock holder には「頼りになる奴」と思える人らしい言葉ですね．しかし，どう感動したの?

nonaka> XEROX がもっとちゃんと....

それはいえてますね，1960 末に PARC の環境を open にしてたら unix も windows もいまごろ必要なかった．compaction method など実装は別問題として，character set も世界中での利用を前提として extensible だし....

余談ですが，Xerox PARC に隣接して FX PARC を去年開設したのですが，その時に office environment や facilities はどうあればよいのかを検討したのです，内部で (私は傍観者)．研究者への enquete などもやって，factor の priority づけまでやりました．PARC の連中も「そこまでやるか?」って驚いてたくらい．

参考に Xerx PARC の environment を調査したりもしたのですが，驚いたのは researcher booth ごとに当時 (60 年代) ですでに concent が 26 個 (!) も用意されていたことです．いまの intelligent office でも，けっこう table tap で誤魔化してるでしょ，それが個人に 26 個ですよ，30 年前で．

こういうのって，やはり「人類への貢献」を意識してないとできないなって，つくづく思いましたよ．だって 26 個要るんだっていう説得できる理由なんかどうアガイタって無理だよ，当時に．いまだって日本じゃほとんど無理なんじゃないかな?

結局，PARC は commercially には失敗だけど，influence という意味では，いまでも still shining だね．もちろん，だからといって Xerox の商売下手をそのままにしてよいってことではないが．

私が働きかけてる/やろうとしてるのは，上記のような time span での (人類に対する) rational を意識しつつ，business 的にもそれを意味のあるものにする，というとこかな? つまりは mutual benefit なのよ．

Date: Tue, 17 Aug 93 14:45:01
From: inaba

---

inaba> この点に関しては豊田有恒氏や....

に関してですが，さきほど神田まで行って適当な本をさがしてみました．専門的なものもいろいろありますが，それなりに理論的で，従来の研究の仕方を包括したものだという点で，次の書をお勧めします．

新･朝鮮語で万葉集は解読できない
安本 美典
JICC (ISBN4-7966-0183-X) 1,100 円

まあ，筆者が筆者なんで，だまされることなく批判的に読む分においては，いま手に入る本としては一番よいと思います．

なんで，朝鮮の人が古事記を古代朝鮮語で解釈したがるかについては，上でもちょっと書いた「逆転の日本史(だったかな?)」という，井沢某(忘れた)が週刊ポストに連載しているのに，なかなか合理的な意見が載っていました．そろそろ一巻目が単行本になるはず．

Date: Tue, 17 Aug 93 15:02:08
From: inaba

一応これで最後のつもりなので，前から思っている意見も最後につけておきます．

nonaka> 私としては，仕事上の都合....

「コロモの下から鎧がちらり」やな．

pv> .... の意味もよく分からないので,....

僕の理解するところ，「現行の JIS-X0208 の文字をすべて含むが，各文字のコーディングは(一定の規則によって)変更された文字コード」を "Shift Unicode" といいたいのだろうと思いますが... でもって，この「一定の規則」が偶然にも "The Unicode" のエンコーディングと一緒なんでしょ，どうせ．

僕としては，いまの X0208 のコーディングになんらかのルールがあるとは思えないから，「勝手にやれば!」ですね．まあ，RMS にも見捨てられたような弱小 OS の上で何がおきようと，人類の未来に影響はないから 10646 自身の話とは違って，わざわざ Ydoc で話をするほどの内容もないと思います．

ただし，その結果 Windows 軍団とコーディングが違うことになって，コンパチビリティの問題からそのコーディングが生き残らなくっても「パワーゲームに負けた」にすぎないんだから，恨みっこなしよ．:-)

ということで，ここんとこずーっと持っているギモン．

－ 文字コードの unify を考えている欧米人は，なぜその前に OLE (や Apple Event でしたっけ) の中での，複数アプリケーションのコーディングを統一しないんだろう???

「ここには，Excel のデータが埋め込まれている」とか「ここには Word のデータへのポインタが入っている」なんてやるより，「すべてのアプリケーションのデータ構造は unification された」ってほうが，効率がいいと思うんですけどねぇ．それをすると，アプリケーションの発展を妨げるってんなら，文字コードも一緒っていうのは，僕からすればあまりに自然なんだがなあ．

Date: Tue, 17 Aug 93 16:27:22
From: pv

予想外の展開を見せていますが，勉強(わたしの)になるので，門外漢にもかかわらず，ちょっと突っ込みます．

inaba> この点に関しては....

豊田有恒って，あの SF 書いてた人? それとも別人かしら?

「週刊ポスト」はまったくといってよいくらい読まないので，知りませんでした，だって何か gravure page が低俗っぽいじゃないですか．何となく恥ずかしくて見ないようにしてましたが，いやぁ偏見はいけませんね．さっそく，購読してみましょう．

inaba> 次の書をお勧めします．

御教示，恐縮です．さっそく見てみることにします．

inaba> まあ，筆者が筆者なんで....

了解．

inaba> そろそろ一巻目が単行本になるはず．

それは期待できますね．ですが，たとえば；

－ 万葉仮名における漢字使用に，甲類／乙類という２種の区別があり，その理由が解明困難であった．ところが古代朝鮮語の専門家から，当時記録に関わった人達の出身地の方言の相違(百済系／新羅系)に２系統あることに因るのではないか，との指摘がされている．

とか，

－ 山彦と豊玉姫の間に産まれたウガヤフキアエズノミコトは，産室の屋根を鵜の羽で葺き終らぬうちに産まれたので，ウガヤフキアエズと名付けたとあるが，古代南部朝鮮に「伽耶」という文化国があり，それは古代韓国語で「上伽耶(ウカヤ)」および「下伽耶(アラカヤ)」と呼ばれた兄弟国であったという．

という指摘，さらに日本国内にも「伽耶」に因んだ地名が数えきれないほど多くあること，などから両国の「関係」がまったくないというのはどうかな? と思っているのですが....

「解釈」はいろいろあっても面白かろう，とノンキな....

Date: Tue, 17 Aug 93 16:41:41
From: pv

わたしもここらで終りにします，

inaba> 僕の理解するところ,....

はっはっは，ノナカちゃんの手のうちはすっかり見透かされてるね．(^^)

inaba> わざわざ Ydoc で話をするほどの....

なるほど．10646 自体の current status はどうなの，使用促進の普及段階かな?

inaba> 恨みっこなしよ．

これはノナカちゃんの「人生哲学」ですから，よもや反故にはしないでしょう．:-)

inaba> ここんとこずーっと持っているギモン．

同意，要は基本的な data structure に関してだけでも合意がとれて，object oriented になっててくれてればねぇ....

inaba> 僕からすればあまりに自然なんだがなあ．

なるほど．納得．

Date: Tue, 17 Aug 93 17:28:32
From: inaba

あーあ，やめようやめようと思っているのに...

pv> 豊田有恒って，あの....

そうです.「ヤマトタケル」とか書いてる人. この人のあまり学術的でない奴 (だから面白い) に聖徳太子や天武天皇の話なんてのもあります.

pv>「週刊ポスト」はまったく....

ポストはこの他にも「マエストロ列伝」とか「票田のトラクタ」とか面白い記事が多いですよ. 昔の「男の料理」もよかったし....

pv> 万葉仮名における漢字使用に....

言語学的にいえば, 日本ほど母音が少ない国は少ないですよね. 別に二系統あったことを前提にしなくても, 母音が多かったコジツケはできると思います.

まあそれ以上の問題は (井沢氏の論に詳しくありますが) 古代朝鮮語の資料が少なすぎることですね. なんて話をこの間ある所でしていたら「そうだ, 日本の朝鮮占領がすべての原因だ」ですと :-(

過去を知っていることはえらいんだけど, なんでもかんでもそのせいにするという態度もねぇ...

pv> 山彦と豊玉姫の間に産まれた....

これの真偽を問わず (だって調べないとわかんないもん) さきほどあげた本を読んでもらえば,「偶然の一致」と「必然の一致」を区別しようという態度 (成功したかどうかは別としてね) がどういうものかはわかると思います.

アラスジだけいえば,「(コジツケで) 万葉集が朝鮮語で読めた」と同じ手法を取れば,「万葉集が英語で読めた」とかいうのが主題です. 実はこれ (の改訂前の本) に対する反論が抱腹絶倒なんだけどね. :-)

Date: Tue, 17 Aug 1993 17:50:49
From: nonaka
---

inaba> 僕の理解するところ....

違う違う! The Unicode のエンコーディングは, 完全に無視でいいと思ってます. やっぱりあれは, 病的ですよ. ただ;

(a) 1 バイト･カタカナと 2 バイト･カタカナのユニファイ.
(b) 2 バイトの非漢字部分 (1 とか A) と, いわゆるアスキーとのユニファイ.
(c) 漢字は今のままの並び順でどこかに収める.

程度をやれば, そこそこよくなるのではないかと思っています. コード･スペースもだいぶ空くから, 藤野さんや稲葉さんが入れたい漢字も, とりあえず収容できます. アラビックやタイとかは, なくなるけど....

inaba> 恨みっこなしよ.

できれば, 日本のマイクロソフトとは, 反 The Unicode で手を組みたいと考えていましたが...

でも, 何といってもベースの米国製 OS やアプリケーションが, 1 文字 16 bit になってくれるのはありがたい. だから, Windows 軍団が脱落したのは残念. あとは, Shift JIS の 2 バイト目だけ見ると何なのかわからん, という問題が解決されるのもうれしいわけです.

お言葉にしたがって, 勝手にやらせていただきます. :-)

Date: Tue, 17 Aug 93 20:39:42
From: inaba

あれ?!

nonaka> 違う違う!

これが "Shifted Unicode" なんですか???

たしかにユニファイこそしてるものの, これを「シフトした」っていったんじゃ, The Unicode 屋が怒るんじゃあないの?

まあ, どっちにしろ先走りすぎました, ゴメンなさい. しかし...

nonaka> 日本のマイクロソフトとは....

になるんじゃあないかなあ, このコーディングの仕方だと?

nonaka> あとは, Shift JIS の....

でも, 制御コードをよく考えないと, graphic-char か control-char かわからんという問題が発生しますよ.

nonaka> コードスペースもだいぶ空くから....

むしろ, このあたりを入れるときは, over-lapping を許すかどうかをコードの中に入れておかないとダメっていうほうが, 影響が大きいのではないでしょうか?

nonaka> 勝手にやらせて....

内部コードなんてそんなもんですから.... ああ, それと何でもいいから名前をつけてあげてくださいね.「半角」「全角」にしても JIS-X0202 とか X0208 といいにくいから出たんで, やっぱり ASCII やら Latin-1 のようなカッコいい名前をつけてあげないと, 浮かばれませんから.

Shift JIS だって, 内部コードで使ってるだけなら, メクジラ立てようとは思わない. なんせエディタなんて作らないからね. :-)

Date: Tue, 17 Aug 1993 23:50:28
From: nonaka
---

pv> 北米先住民族の悲哀とか, ちゃんと勉強して....

市場での自由競争と武力的侵略を同列に扱うのは, 乱暴だとは思いますが, あえてこの土俵で反論するなら, 私のいいたいのは「侵略された時どうするかぐらいは, ふだんから考えておけ!」ということですね.

戦うのか? 降伏するのか? 逃げるのか?

実際, われわれは, Microsoft に支配されてしまっているのですから.

Date: Tue, 17 Aug 1993 23:50:58
From: nonaka

Unicode に関する議論に関わったみなさん;

長い間お疲れさまでした. だいぶ, けだるい雰囲気がただよってきました. そろそろ, 第 1 ラウンドは終了でよろしいかと思います. さて, ご承知とは思いますが, 岸田編集長から, これまでのメールのやり取りをまとめて SEAMAIL に報告するよう要請がありました. Ydoc のクレイジーなようすを, そのまま

伝えたいので，なるべく手を加えないようにして欲しい，とのことです．

私も個人的にはその方針に賛成なので，無修正でそのまま，発言順にメールを並べるだけにしたいと思います．とはいえ，どうしてもここだけは削除したいという希望もあると思います．のちほど，ファイルを発言者の方へメールしますので，校正をお願いいたします．

最後に一言という方のため，あと２～３日待って，締切にしたいと思います．

議論も，とくに結論らしいところまでは到達できなかったので，一連の議論での感想や疑問点等をまとめておくと，第２ラウンドの方向性もはっきりしてよいのではないかと思います．こちらのほうは，各自８月末までに編集部へ直接送るということでいかがでしょう？ これは，このメーリングリストには出さず，SEAMAIL が発行された時のお楽しみ，というのがよろしいのではないでしょうか?

観戦者の方々もぜひご感想をおよせください．

Date: Wed, 18 Aug 93 01:33:26
From: pv

---

nonaka> 自由競争と武力的侵略を同列に扱うのは....

そうか，わたしの書き方も悪かったね，舌足らずでしたから．「武力的侵略」もそうだが，それ以前に，そのようにならざるを得なかった経過だよね，要は欧州民族がかれらの文明とは「異なる」文明の存在に気づかなかったことが原因にあるのですよ．武力的侵略になったのはその結果だから．

具体的には，北米先住民族には「土地の所有」という概念がなかった．さらにかれらは「大地」を母としていたので，大地を「耕す」ということさえしない．母なる大地をいじるなんて思いもよらなかったのでしょうね．

一方，移ってきた「白人」たちは「土地を開拓」し，耕す．そして，自分たちが耕したその「土地を所有」するわけだ．

さて，harvest の時期になると先住民がきて，勝手に実ったものを採る．先住民にすれば大地の実りはすべて母からの贈り物で，共有すべきものだから，これは当然な行為なわけですが，土地の「所有者」であり「耕した」白人からみれば「とんでもない奴ら」となるわけだ．「文明以前の連中だ」ってなわけで，脅したり，教化したりと，まぁ後はおおむねみなさん御承知の通り．

で，欧州からの移民である「白人」たちは，かれらから見れば「哀れで貧しく無知」な先住民を「文明人」にしてやろうと，自分たちの「文明」を先住民に押しつけた．その結果がいまの「北米先住民族の悲哀」なのだよ．民族の文明がまったく否定されてしまったわけで，identity を失った民族がどうなるか?「文明」の「押売リ」がどのような結末を招くか? その悲しむべき典型でしょう．

ところが ecology が注目されるにしたがい，先住民族の life style が見直されはじめた，自然と協調していたからね，かれらは．で，西洋文明は最近になってやっと「北米先住民族に文明」が存在していたことに気づくことができた．だからこそ，Canada における先住民族自治州の承認 (ほとんど独立) や，米国政府の先住民族に対する地位向上/保証といったことが生じてきているのです．

いずれにしても，両文明にとって tragedy だったわけですよ，これは．そして，いまだに終章を迎えてないわけで..... :-(

nonaka> 戦うのか? 降伏するのか? 逃げるのか?

私は民族主義者ではないし，optimist だから，「文明」には本来優劣はないのだと思ってます．前にも述べた通り，diversity の追求が人間の活きる (根源的) 理由だと思ってるからね．で，diversity の追求を妨げる activity は結局は人類に tragedy をもたらすだけだと思ってる，long range でね．

だから「戦う/降伏/逃げる」とかは，私には意味をなさないね，ただ diversity の追求の妨げになりそうな activity に気づいたら，断固，「反対」の声を上げますけど．

ついでに，「鈍感」さは fascio を「増長」させる「触媒」のようなものだからね，だから「生理的」に嫌いなのだと思う，多分．

nonaka> われわれは Microsoft に支配されて...

結局，その「つけ」はいつか払わなくてはならなくなると思う，人類全部がね．人間は本質的には賢くなってないよ，自分たちの未来を考えることがてんでできてないもの．

以上がわたしの「妄想」かな? :-)

Date: Wed, 18 Aug 93 01:44:05
From: pv

---

nonaka> Unicode に関する議論に関わったみなさん

やれやれ．yet another 恥を晒すわけだ．:-(

ほんとうに勉強になりました，いろいろな意見が聴けて．まぁ，Mr.Inaba みたいな debate についての「確信犯」がいたお蔭かな? :-)

nonaka> 感想や疑問点等をまとめておくと....

蟻地獄だね，これじゃ! (;.;)

nonaka> 観戦者の方々もぜひご感想を....

観戦料を取っておけばよかったなぁ．ブツブツ...

Date: Thu, 19 Aug 93 10:19:47
From: hihara

---

nonaka> 私の坂村氏に対する評価は....

坂村先生には直接話を聞きましたが，なかなか，こんなにエレガントにはいっていないようです．

Date: Fri, 20 Aug 1993 14:28:49
From: nonaka

---

hinara> 坂村先生には直接話を....

なにはともあれ，頑張ってほしいものです (これは皮肉じゃないよ)．日本人が日本人の仕事を認めないという風潮は，たしかにありますね．このさい，シリコンバレーに会社を作って逆上陸を目指すというのも，１つの考え方であるようにも思い

ます．:-)

さて，編集のため，一連のメールを読み返していますが，檜原さんの質問に答えてなかったので最後に補足しておきます．

－ 「どうしても 16bit に文字を収めたい人」は，どのくらいのスパンでモノをみているのか?

わたしは，16bit 固定長案の支持派の 1 人ですが，どのくらい先まで陳腐化しないか? ということはあまり考えませんでした．それより，いますぐ，つまり，MS-DOS や MacOS の次のメジャーリリース等で採用可能な体系が欲しいと思ってました．US のソフトそのままでも，バックスペース 1 回で，漢字がちゃんと消えてくれるとうれしい．まあ，その程度の利益が欲しいだけだから，あまり，移行のコストが高いのは許せないわけですね．だから，本家 Unicode のエンコーディングに反対なのは，

－ フォント作りなおすのが面倒
－ シフト JIS とのフィルターが面倒

といった理由によります．でも，16bit 固定長は捨てがたいですなあ．別に 32bit でもいいわけですが，一挙に 4 倍というのもなかなか抵抗が多いでしょうね．

Date: Fri, 20 Aug 93 16:30:11
From: nonaka

もう，ほんとうに最後ですが，質問だけ答えておこうと思います．

pv>「標準」というものをどういうふうに....

「# それなら、標準なんて決める必要ないのではないかい?」という行がよけいでしたね．「標準」そのものの位置づけは、檜原さんとそんなに変わらないと思います．しいていえば，私のほうが，デファクト･スタンダード的なものを強く意識しているような気がしますが....

Date: Fri, 20 Aug 93 17:59:18
From: nonaka

これも答えておきたいと思います．(最後, 最後と言いながら...)

pv>「どの側面から」貢献したいのか?

「よい品質のものをできるだけ安く提供する」でしょうか?

だから，中国市場が某社のプリンタであふれるのも，(自由競争の結果なら) 正義ではないか? というのが私の思い込みです．

pv> しかし，どう感動したの?

私の説明不足ですね．たしかに，これだけじゃわからない．

なにごとをするにもやはり実行力 (財政的基盤) は大切ではないかと思います．同様に，「いくら美しい規格でも，それにのっとった製品が手軽にみんなの手が届くような価格で出てこなければ意味がない」というのが私の行動原理です．

The Unicode に関していえば，たしかに実際のエンコーディングはひどいものだし，非難されるべきでしょう (文化侵略という言葉を使うのには反対ですが)．ただ，プロジェクトとしての力強さ (強引さというべきか?) にかぎれば，多少は学ぶべき点もあるのではないか? と個人的には思います．対抗してゆく上でもね! そんなわけで，シフト･ユニコードというのが，私なりに考えた The Unicode への対抗策のつもりなわけです．「どこまで理想を追及するか?」と「どこまで現実的な妥協をするか?」のバランスは，なかなかむずかしいということなのでしょう．

Date: Fri, 20 Aug 93 18:24:10
From: nonaka

野中です．ほんとに最後にします．

inaba> The Unicode 屋が怒るんじゃあないの?

稲葉さんの怒りと相殺してもらいましょう．:-)

inaba> むしろ，このあたりを入れるときは....

実は，完全に The Unicode のマッピング無視ではなくて，[0000-04FF] あたりまでは，The Unicode を尊重．それから上は日本で勝手に決める，ということでいいのではないかと思います．ちゃんと切り替え方式の提案をしたほうがいいのでしょうが，別にメカニズムを決めなくて," 外国へデータを送るときは，[04FF] 以下の文字を使いましょうね」という，運用上の約束で逃げても，実際のデータ交換には，ほとんど支障は無い，と踏んでいるのですが，いかがでしょうか?

あと，もちろんタイ語やチベット語のテキストを送って来られたら，それは漢字に化けてしまいますが，きっとそれで困る人は，日本に 5 人ぐらいしかいない．これは，差別ではないぞ!

****

Unicode は文化侵略なのか?

野中 哲

この一連の議論のルーツは，6 月の Ydoc の 2 次会にありました．Ydoc の 2 次会は，横浜の中華街で行われるのが習わしです．この月は，1 次会でのスピーカーが，国際標準の制定に長年携わってこられた松原さんであったため，話題も，自然に標準化そして Unicode の方へ向かったように思います．Unicode に関して，私は，雑誌やネットニュースでの解説や議論に多少目を通していた程度で，具体的なエンコーディングの内容まではほとんど知らない状態でしたが，いくつか気になっている点がありました．さて，議論の詳細は，ここに採録された ML の記録をご覧いただけばわかる通りですが，現時点での議論の成果を私なりにまとめると，次のようになります．

1. 中国･朝鮮･日本で共通の漢字コード制定は可能か?

現時点では，これはほぼ不可能な作業と考えてよさそうです．ただ，それは，各国の間の文化的背景のが異なるという理由によるわけではなく，統一的な文字コード体系を推進するための十分な経済的動機がないことが，その原因であるように思われます．

2. 漢字の使用制限は合理的か?

" 漢字の使用制限はは非合理的" という主張がなされました．

私はこの点に関しては，

(1) 情報交換を効率的に行うために，文字の制限を行うことは合理的．
(2) すでに存在しており，また広く受容されているので，その事実を尊重する．

という考えです．

すべての新しい技術は，旧来の文化，文明を衰退させる働きを持っています．これは，文化の盛衰というごく自然な現象ではないでしょうか? 情報処理の効率化という観点から，文字の体系を機械的に扱いやすいように組み替えるのも，また1つの文化の進化の形態であると思います．

3. Unicode における漢字とアスキー，ラテン文字の差別的扱いについて

Unicode では，漢字のユニフィケーションがかなり強引に行われ，欧米の文字セットに関するそれはあまり熱心に行われていないというのは，事実のようです．これは，欧米の日本に対する差別的扱いです．しかし，この点を問題と考えるならば，同じように「現在の JIS コードには，漢字や，かな文字に混ざって，キリル文字が入っているが，ハングル文字は含まれていない」という在日朝鮮人の人々にとって差別的な事実も，問題にするのが筋というべきでしょう．こうした差別的な扱いは，純粋に工学的なトレードオフの結果なのではないでしょうか? 漢字を必要とする人々は，コンピュータの世界では，歴史的にもマーケット的にも，現時点では少数派です．したがって，16bit 空間の利点を優先するか? 漢字をすべてきちんと扱うのを優先するか? のトレードオフを考えれば，前者を選択し漢字の扱いを軽んじるのは，当然の結論といえるでしょう．

4. われわれは Unicode を押しつけられたのか?

私は，押しつけられたとは考えていませんが，もし，そのように受け取り，それが好ましくない状態であると考えるならば，今後どう戦って行くのかを考える必要があるでしょう．Microsoft が強大な力を持つなら，それに見合う力をわれわれが持てるように考えるべきなのか? あるいは，その強大な力を未方につけるように努力すべきなのか? 感情的な反発のレベルにとどまらず，次のステップへの戦略を練る時期にきていると思います．

論 文 募 集

ソフトウェア･シンポジウム '94

1994 年 6 月 15 日 (水) ～ 17 日 (金)
於： 金森ホール (北海道函館市)

主 催：
ソフトウェア技術者協会 (SEA)
協 賛（予定）
日本ソフトウェア科学会　情報処理学会
情報サービス産業協会　北海道ソフトウェア協会

ソフトウェア･シンポジウムは，ソフトウェアハウス，コンピュータメーカ，計算センタ，エンドユーザ，大学，研究所など，さまざまな場で活躍している技術者，管理者および研究者が一堂に会し，ソフトウェア技術に関する多面的な経験や知識を交流するユニークで貴重な場を提供してきました．迎えて第14 回となる来年のシンポジウムは，初めて本州から離れ，会場を北海道の函館に移して開催します．

毎年，現場に立脚した質の高い論文が集まるこのシンポジウムのレベルは，回を重ねる毎に着実に上がっております．今回は，会場の関係もあり，単一トラックでセッションを構成します．そのため採録可能な論文は 15 編程度と少なくなる見通しで，例年にも増して高いレベルの論文採否が予想されます．そこで，従来からの当シンポジウムならではの現場に立脚した論文を尊重する意味で，採録の審査を，(1) 経験報告と (2) 技術論文に分けて行うことにしました．

なお，従来通り最も優秀な論文に贈られます最優秀論文賞 (服部賞) は，シンポジウム開催時に選定します．その際は，経験報告，技術論文の区分によらず，全論文を一律に選考審査の対象とします．経験報告，技術論文とも，日本語の他，英語による投稿も可とします．ただし，発表は日本語に限ります．

応募論文テーマとしては，たとえば;

・ 分析／設計技法　・ 品質管理　・ オブジェクト指向　・ 開発環境　・ ネットワーク
・ プロセス管理　・ CASE　・ 新時代のパラダイム　・ プロジェクト管理　・ 人間的要因

などさまざまなものが考えられますが，必ずしもこれらにとらわれる必要はありません．

応募要領

応募論文は未発表のものに限ります．これまで，概要のみによる応募も可とすることがありましたが，今回は本論文による投稿のみを認めますので，御注意下さい．また，他への二重投稿は御遠慮下さい．様式は自由ですが，A4 サイズで 5 ～ 10 ページ程度を目安とします．応募論文は，別掲のカバーシートに必要事項を記入の上，**1993 年 12 月 17 日**までに 2 人のプログラム委員長のいずれか宛に，郵送 (この場合は 5 部) ないし電子メールでお送り下さい．電子メールの場合は，roff, TeX, LaTeX, PostScript, 単テキストのいずれの形式でも構いません．プログラム委員会で内容の審査を行い，結果を 2 月下旬までに，応募者全員に通知します．その他不明な点がありましたら，プログラム委員長まで御問い合わせ下さい．

主要スケジュール (Important Date) :　1993 年 12 月 17 日　論文応募〆切
1994 年 2 月 25 日　論文採否通知送付

シンポジウム･スタッフ

実行委員長
杉田義明 (日本 NCD)
プログラム委員長
玉井哲雄 (筑波大)
渡邊 雄一 (アスキー)
プログラム委員
青山幹雄 (富士通)
荒木啓二郎 (奈良先端大)
伊藤昌夫 (MASC)
大塚理恵 (RSK)
大場充 (日本 IBM)
兼子毅 (東京大)
岸田孝一 (SRA)
久野靖 (筑波大)
熊谷章 (PFU 上海)
元治景朝 (さくら KCS)
佐伯元司 (東工大)
坂本啓司 (オムロン)
佐藤千明 (長野県協同電算)
塩谷和範 (SRA)
武田淳男 (安川電機)
高橋光裕 (電力中研)
中野秀男 (大阪大)
布川博士 (東北大)
野口正浩 (新日鉄)
野呂昌満 (南山大)
藤野晃延 (FXIS)
二木厚吉 (北陸先端大)
細谷僚一 (NTT)
堀江進 (NES)
松本健一 (奈良先端大)
山崎利治 (日本ユニシス)

ソフトウェア･シンポジウム '94

論文応募カバー･シート

氏名 (筆頭著者) : ＿＿＿＿＿＿＿＿＿＿＿＿ (ふりがな) ＿＿＿＿＿＿＿＿＿＿＿＿

所属 (会社, 大学, 組織) : ＿＿＿＿＿＿＿＿＿＿＿＿

部門 (学部) : ＿＿＿＿＿＿＿＿＿＿＿＿ 役　職 : ＿＿＿＿＿＿＿＿＿＿＿＿

住　所 : (〒　　　) ＿＿＿＿＿＿＿＿＿＿＿＿

TEL: [　　－　　－　　内線　　] FAX: [　　－　　－　　]

E-Mail: ＿＿＿＿＿＿＿＿＿＿＿＿

論文タイトル: ＿＿＿＿＿＿＿＿＿＿＿＿

共同執筆者 (もしいれば):

氏名: ＿＿＿＿＿＿＿＿＿＿＿＿ 所属: ＿＿＿＿＿＿＿＿＿＿＿＿

氏名: ＿＿＿＿＿＿＿＿＿＿＿＿ 所属: ＿＿＿＿＿＿＿＿＿＿＿＿

氏名: ＿＿＿＿＿＿＿＿＿＿＿＿ 所属: ＿＿＿＿＿＿＿＿＿＿＿＿

氏名: ＿＿＿＿＿＿＿＿＿＿＿＿ 所属: ＿＿＿＿＿＿＿＿＿＿＿＿

論文概要:

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

応募カテゴリ:(いずれかにチェック)

☐ : 経験報告　　☐ : 技術論文

キーワード:(最低 2 つ、最大 5 つのキーワード挙げて下さい)

1 : ＿＿＿＿＿＿＿＿＿＿＿＿

2 : ＿＿＿＿＿＿＿＿＿＿＿＿

3 : ＿＿＿＿＿＿＿＿＿＿＿＿

4 : ＿＿＿＿＿＿＿＿＿＿＿＿

5 : ＿＿＿＿＿＿＿＿＿＿＿＿

------------------------ 送　付　先 ------------------------

〒 112 東京都文京区大塚 3-29-1
筑波大学大学院 経営システム科学 助教授
玉井 哲雄
TEL: 03-3942-6869
E-MAIL: tamai@gssm.otsuka.tsukuba.ac.jp

または

〒 215 神奈川県川崎市麻生区南黒川 12-1
(株) アスキー 管理本部 ＯＡ推進部
渡邉 雄一
TEL: 044-989-9435
E-MAIL: yuu-w@ascii.co.jp

********** 応募締め切り： 1993 年 12 月 17 日 (金) **********

PRELIMINARY PROGRAM

ACM SIGSOFT '93: Symposium on the *Foundations of Software Engineering*

Sponsored by ACM SIGSOFT

Los Angeles, California USA
7-10 December 1993

ACM SIGSOFT '93 focuses on innovative research results that identify and contribute to the foundations of software engineering. The intent is to help establish a viable software engineering discipline. The conference consists of a one day tutorial program followed by a two and one-half day technical program. Turing Award winner C.A.R. Hoare will give Tuesday afternoon's tutorial as well as the opening keynote address.

Tutorial Program, 7 December 1993

(Registration Monday 5:00-8:00PM and Tuesday 7:30AM-2:00PM)

Tutorial AM1 8:30AM-12:30PM, *Architectures for Software Systems*
David Garlan and Mary Shaw, Carnegie Mellon University

As the size of software systems increases, the algorithms and data structures of the computation no longer constitute the major design problems. When systems are constructed from many components, the organization of the overall system -- the software architecture -- presents a new set of design problems. This level of design has been addressed in a number of ways including informal diagrams and descriptive terms, module interconnection languages, templates and frameworks for systems that serve the needs of specific domains, and formal models of component integration mechanisms. This tutorial provides an introduction to the emerging field of software architecture. It considers a number of common architectural styles upon which many systems are currently based and shows how different styles can be combined in a single design. It presents case studies to illustrate how architectural representations can improve our understanding of complex software systems, allow software engineers to evaluate existing systems in terms of their underlying architectures, and design new systems in principled ways using well-founded architectural paradigms. Finally, it surveys some of the outstanding problems in the field, and considers a few of the promising research directions.

Tutorial AM2 8:30AM-12:30PM, *Software Development Using Formal Specifications (Written in Larch)*
John Guttag, Massachusetts Institute of Technology

Specifications can provide precise and easy to read module-level documentation of software interfaces. If used properly, they can have a profound effect on the way software is produced, audited, and maintained. The difficulty lies in writing specifications that are at once accurate, precise and comprehensible. This tutorial deals with how to write such specifications in Larch. Larch is a family of specification languages, each with a precise syntax and semantics. This allows one to write specifications with an unambiguous meaning, and to use software tools to help check specifications. Emphasis will be placed on the pragmatic aspects of using Larch in the context of developing programs in the programming language C.

Tutorial AM3 8:30AM-12:30PM, *Software Testing: Foundations, Current Techniques, and Advanced Technology*
Debra J. Richardson, University of California, Irvine

The need to produce high-assurance software-intensive systems is becoming more paramount as computers are being introduced into increasingly more critical applications. Testing is one means of contributing to improved software quality, which is one of the greatest challenges facing American industry today. This tutorial will focus on available testing technology for assuring software quality. Underlying testing theory will be presented to provide a foundation for evaluating testing technology, and several new approaches will be discussed. Issues of integrating complementary techniques to achieve a comprehensive testing process will also be addressed. The intent of this tutorial is to equip software engineering researchers and software engineers with an understanding of testing foundations and technology to enable them to promote software testing from an ad hoc, labor intensive, error prone activity to a managed, disciplined, and technology-supported process. This requires strategies to properly select techniques to achieve desired goals, to integrate complementary techniques synergistically, and to transition technology into software development environments. Emphasis will be on advanced software testing concepts and techniques that can make a contribution toward achieving high-assurance software products.

Tutorial PM 1:45PM-5:45PM, *CSP Principles and Experiences*
C.A.R. Hoare, University of Oxford

Ever since I failed to deliver a promised operating system some thirty years ago, I have been fascinated by the principles of distributed and parallel programming. I conjectured that a proper understanding of parallelism would suggest solutions to many structural problems of sequential programming as well. I now believe more than this. The principles of parallelism illuminate and unify many other topics in Computing Science, from hardware design to object oriented programming. Furthermore, they provide assistance to more intuitive human design skills just at the crucial interface where theory meets practice. I would like to share with my audience some of the fruits of my experience, and some of my hopes for the future. It would help if my audience had some preliminary acquaintance with the first two chapters of my book, *Communicating Sequential Processes* (Prentice Hall 1985). I hope my tutorial will provide guidance to practitioners on how to identify and exploit useful results of research in computing science; and also help researchers to identify directions and develop methods of research that will prove even more useful in future.

Technical Program, 8-10 December 1993

Registration Tuesday 5:00-8:00PM and Wednesday 7:00AM-)

	Wednesday, 8 December 1993	Thursday, 9 December 1993	Friday, 10 December 1993
8:30AM	Introductory Remarks (8:45AM)		Test Case Generation by Means of Learning Techniques, F. Bergadano (Catania)
9:00AM	Keynote Address: Algebra and Models, C.A.R. Hoare (Oxford)	Keynote Address: Tracking the Human Factor in Software Development [tentative title], M.B. Rosson (IBM TJ Watson Research Laboratory)	Software Measure Specification, D.A. Gustafson, J.T. Tan & P. Weaver (Kansas State)
9:30AM			Partition Testing, Stratified Sampling, and Cluster Analysis A. Podgurski & C. Yang (Case Western Reserve)
10:00AM			Break
10:30AM	Break	Break	Signature Matching: A Key to Reuse, A.M. Zaremski & J.M. Wing (CMU)
11:00AM	Using Style to Understand Descriptions of Software Architectures, G. Abowd, R. Allen & D. Garlan (CMU)	Targeting Safety-Related Errors During Software Requirements Analysis, R.R. Lutz (JPL/CalTech)	Scalable Software Libraries, D. Batory, V. Singhal & J. Thomas (UT Austin)
11:30AM	Mechanisms for Generic Process Support, R. Balzer & K. Narayanaswamy (USC/ISI)	Does Every Inspection Need a Meeting? L.G. Votta, Jr. (AT&T Bell Labs)	*Conference ends to allow time to get to LAX for afternoon flights.*
12:00N	Lunch	Lunch	
1:30PM	Fine-Grained Revision Control for Collaborative Software Development, B. Magnusson, U. Asklund & S. Minor (Lund)	Panel: Technology Transfer, Susan Gerhart (chair)	
2:00PM	Direct Update of Dataflow Representations for a Meaning-Preserving Program Restructuring Tool, W.G. Griswold (UCSD)		
2:30PM	PLEIADES: An Object Management System for Software Engineering Environments, P. Tarr & L.A. Clarke (UMass)		
3:00PM	Break	Break	
3:30PM	Towards Increased Productivity of Algorithm Implementation, J. Cai & R. Paige (NYU)	Enhancing Compositional Reachability Analysis with Context Constraints, S.C. Cheung & J. Kramer (Imperial)	
4:00PM	A Practical Approach to Software Engineering using Z and the Refinement Calculus, K.R. Wood (Oxford)	Proving Compositional Properties: An Agent-Oriented Programming Approach, J.P. Bahsoun, S. Merz & C. Servieres (IRIT)	
4:30PM	Deriving Modular Designs from Formal Specifications, D. Carrington, D. Duke, I. Hayes & J. Welsh (Queensland)	A Logical Approach to Data Structures, R. Turpin (SES)	

Wed. evening reception co-sponsored by the LA Chapter of the ACM

Conference Committee

General Chair
Barry Boehm, University of Southern California

Program Chair
David Notkin, University of Washington

Tutorials Chair
Dewayne Perry, AT&T Bell Laboratories

Local Arrangements Chair
Hal Hart, TRW

Program Committee

Lori Clarke, University of Massachusetts
John Gannon, University of Maryland
David Garlan, Carnegie Mellon University
Susan Gerhart, RICIS
Ross Jeffery, University of New South Wales
Nancy Leveson, University of Washington
Harold Ossher, IBM TJ Watson Research
Tom Reps, University of Wisconsin
Tetsuo Tamai, University of Tsukuba
David Wile, USC/Information Sciences Institute
Michal Young, Purdue University

Conference and hotel registration forms are on the next page.

Registration Form
SIGSOFT'93 CONFERENCE

Holiday Inn Crowne Plaza,
Redondo Beach, California USA
7-10 December 1993

Name: _____

Affiliation: _____

Address: _____

_____

Phone: (_____)_____

E-Mail _____

Conference (8-10 Dec.)	Early Fee (by Nov. 1)	Late Fee (after Nov. 1)
ACM or SIGSOFT Member*	$300	$340
Non-Member	$350	$390
Student	$100	$100

Conference fees include proceedings, Wednesday and Thursday luncheons, Wednesday evening's reception, and refreshment breaks. Student fees do not include luncheons and the reception.

Tutorials (7 Dec.)	AM or PM		AM and PM	
	Early	Late	Early	Late
ACM or SIGSOFT Member*	$170	$210	$270	$310
Non-Member	$200	$240	$320	$360
Student	$120	$160	$220	$260

*ACM or SIGSOFT Membership Number: _____

Please circle the TUTORIALS you are paying for (at most one "AM"):

AM1 "Software Architectures" by Garlan/Shaw
AM2 "Formal Specifications" by Guttag
AM3 "Tesuno" by Richardson
PM "CSP" by Hoare

Tutorial fees include handouts, luncheon, and refreshment breaks.

SIGSOFT Annual Membership $50
(may be used immediately to qualify for members' rates for the conference and/or tutorials)

Total Amount Enclosed: US$ _____

Please circle the fees you are paying

Mail completed Registration Form and check payable to SIGSOFT '93 to:

SIGSOFT '93 c/o Michael P. Walsh
14960 Victory Blvd.-Unit 101
Van Nuys CA 91411-1805 USA
email: mp_walsh@acm.org

Misc

Hotel Reservation Form
SIGSOFT'93 CONFERENCE

The conference will be held at the Holiday Inn Crowne Plaza in Redondo Beach. 200 rooms have been blocked, so it is important that you make your reservation early; this block of rooms is only guaranteed until Nov. 22, after which reservations at SIGSOFT'93's rates are subject to availability. Quoted room rates apply from *Friday, December 3, through Sunday, December 12*, to accommodate the Saturday night stayovers that are often required for the most economical air fares. All participants are responsible for making their own hotel reservations by mailing this form by Nov. 22 directly to:

Holiday Inn Crowne Plaza, Redondo Beach
300 N. Harbor Drive
Redondo Beach, CA 90277-2552 USA

[Or call 310/318-8888 and mention "SIGSOFT'93" to get our group rate]

Please circle accommodations and rate you are requesting:

Single Occupancy $87
Double Occupancy $87
(All rates subject to 10% taxes)

Name: _____

Organization: _____

Address: _____

_____

Phone: (_____)_____

Arrival Date (3PM check-in): _____

Departure Date (noon check-out): _____

This reservation will be held only until 3PM on the day of arrival unless guaranteed by credit card or check for a deposit equal to one night's rate plus tax:

Credit Card (Circle one): AX DC CB VISA MC

Credit Card Number: _____

Expiration Date: _____

LOCATION: The Holiday Inn Crowne Plaza in Redondo Beach is part of the Redondo Pier/King's Harbor Marina complex. The Pier and all its shops and restaurants are a 5-minute walk south. Approximately 7 miles south of Los Angeles International Airport (LAX), the Holiday Inn Crowne Plaza is an easy 15-20 minute drive from LAX at most hours. Downtown LA is 30 minutes to the northeast. Redondo Beach itself is a prosperous town of 65,000 people, incorporating the best of both its late 19th century beginnings and its mid-20th century transformation into a great resort area. While Redondo Beach is within an hour of every major business center and tourist attraction in Los Angeles, it enjoys a certain serenity and isolation because no freeway runs through the city. Redondo Beach is also only 45 minutes from Knott's Berry Farm or the Queen Mary/Spruce Goose, an hour from Universal Studios and Disneyland, and two hours from San Diego.

LOCAL TRANSPORTATION: It takes about 15 minutes by rental car or taxi from LAX to the Holiday Inn Crowne Plaza in Redondo Beach. The most economical local transportation is one of the numerous shuttle vans serving LAX. We recommend "SuperShuttle" at ~$12; use SuperShuttle's courtesy phone on the Ground Transportation Board at any luggage claim area, and a blue van will pick you up at the designated curb area 5-15 minutes later. For driving directions, please call the hotel at (310)318-8888.

FURTHER LOCAL ARRANGEMENTS INFORMATION: Contact Hal Hart, 310/812-0661, or "halhart@AJPO.SEI.CMU.EDU".

ソフトウェア技術者協会
〒160 東京都新宿区四谷3-12 丸正ビル5F
TEL.03-3356-1077　FAX.03-3356-1072