



SEAMAIL

Newsletter from Software Engineers Association

Volume 4, Number 1 1989

目次

事務局から		1
第2回日中ソフトウェア・シンポジウムから		2
第2回日中ソフトウェア・シンポジウム 総括報告	熊谷 章	2
日中ソフトウェア・シンポジウム 参加報告	佐原 伸	12
日中ソフトウェア・シンポジウム 第4セッション・レポート	中野 秀男	18
中日軟件検討会 '88 感想文	林 香	20
第2回中日シンポジウム参加記	藤野 晃延	22
日中ソフトウェアシンポジウム 参加レポート	片山 禎昭	24
第2回日中ソフトウェアシンポジウム	梶山 雄三	26
第2回日中ソフトウェアシンポジウム レポート	水谷 時雄	27
第2回日中ソフトウェア・シンポジウムの感想	久保 宏	28
オブジェクト指向を越えるもの	大木 幹雄	30
システム作りの自動化支援について	小林 俊明	32
処理速度への挑戦	横山 博司	35
地方衛星都市におけるソフトウェア産業労働者の実体	匿名希望の地方会員	41

ソフトウェア技術者協会（SEA）は、ソフトウェア・エンジニアの、ソフトウェア・エンジニアによる、ソフトウェア・エンジニアのための団体であり、これまでに日本になかった新しいタイプのプロフェッショナル・ソサイエティたることを目指して、1985年12月20日に設立されました。

現在のソフトウェア技術が抱える最大の課題は、ソフトウェア・エンジニアリング研究の最前線（ステイト・オブ・アート）と、その実践状況（ステイト・オブ・プラクティス）との間に横たわる大きなギャップを埋めることだといわれています。ソフトウェア技術の特徴は、他の工学諸分野の技術にくらべて属人性がきわめて強い点にあります。したがって、そうしたテクノロジー・トランスファの成否の鍵は、研究者や技術者が、既存の社会組織の壁を越えて、相互の交流を効果的に行うためのメカニズムが確立できるか否かにかかっています。SEAは、ソフトウェア・ハウス、計算センタ、システム・ハウス、コンピュータ・メカ、一般ユーザ、大学、研究所など、さまざまな職場で働く人々が、技術的・人間的交流を行うための自由な〈場〉であることを目指しています。

SEAの具体的な活動としては、特定のテーマに関する研究分科会（SIG）や地方支部の運営、月刊機関誌（SEAMAIL）の発行、各種のセミナー、ワークショップ、シンポジウムなどのイベントの開催、既存の学会や業界団体の活動への協力、また、さまざまな国際交流の促進等があげられます。

なおSEAは、個人参加を原則とする専門家団体です。その運営は、つねに中立かつ技術オリエンテッドな視点に立って行われ、特定の企業や組織あるいは業界の利益を代表することはありません。

代表幹事： 岸田孝一

常任幹事： 白井義美 久保宏志 熊谷章 佐藤千明 藤野晃延 松原友夫 吉村鉄太郎

幹事： 青島茂 稲田博 岡田正志 落水浩一郎 片山禎昭 川北秀夫 杉田義明 鈴木弘 武田知久 田中慎一郎 長井剛一郎 中園順三 中野秀男 西尾出 野村敏次 野村行憲 針谷明 深瀬弘恭 藤本司郎 村井進 盛田政敏

会計監事： 辻淳二 吉村成弘

常任委員長： 白井義美（技術研究） 久保宏志（企画総務） 藤野晃延（会誌編集） 杉田義明（セミナー・ワークショップ）

分科会世話人 環境分科会(SIGENV)： 田中慎一郎 渡邊雄一

管理分科会(SIGMAN)： 相沢圭一 川北秀夫 芝原雄二 野々下幸治

教育分科会(SIGEDU)： 大浦洋一 杉田義明 中園順三

再利用分科会(SIGREUSE)： 青島茂 阿倍正平 村井進

ネットワーク分科会(SIGNET)： 青島茂 野中哲

法的保護分科会(SIGSPL)： 能登末之

CAI分科会(SIGCAI)： 大木幹雄 寺嶋裕一 中谷多哉子 中西昌武

ドキュメント分科会(SIGDOC)： 田中慎一郎 野辺良一

支部世話人 関西支部： 白井義美 盛田政敏

横浜支部： 熊谷章 林香 藤野晃延 松下和隆

長野支部： 小林貞幸 佐藤千明 細野広水

名古屋支部： 岩田康 鈴木智 西村亨

九州支部： 植村正伸 小田七生 藤本良子 平尾一浩 松本初美 中島泰彦 能見巧 後藤芳美

SEAMAIL 編集グループ： 岸田孝一 佐原伸 芝原雄二 関崎邦夫 田中慎一郎 中村昭雄 長井修治 成沢知子 野辺良一 藤野晃延 渡邊雄一

SEAMAIL Vol. 4, No. 1 平成元年5月15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）

〒102 東京都千代田区隼町2-12 藤和半蔵門コープビル505

印刷所 サンビルト印刷株式会社 〒162 東京都新宿区築地町8番地

特価 1,000円（禁転載）

事務局から

1. 手が腫れる

事務所の窓から見える皇居の緑がまぶしいこの頃です(ご存知ないかたもいると思いますが、事務所の窓から皇居が見えるのです)。こんな都心に位置しながらも、バス通りに面していないので、騒音もここまでは(5F)上がってきません。四季の移り変わりは目に見え、ベランダにやってくる鳥も、連休あけの休養たっぷりとはばかり、心なしかイキイキとしています(残念ながら事務局は連休がとれず、慢性疲労になりつつあります)。飛行船ものんびり浮かんでいるし、気合いをいれなおして、5月の風を入れて、サァー今日も仕事ダ!

と、またしても苦情とも皮肉とも思われる会員さんからの電話が入ります。「SEAMAIL が来ないけれど、どうしたわけ?----」

こういときはすかさず、

「---原稿がなかなか集まらなくて--- 何か書いて下さると助かるのですが」と半分おどして、半分お願い----

とはいうものの、受話器を置いたたんに途端に、事務局も電話の主と同じ気持ちになります。そこで一首、

春すぎて 夏来にけらし 桃妙の
表紙すてふ 天の筆

比較的新しい会員さんには、前の号を手にしてからの長さが、不安とシビレの切れる頃あいでしょうか。また、古くからの会員さんには、また船便シンドロームが始まったか、とあきれられているところでしょうか。編集長と前編集長(ともにまた海外出張中)と関係者のお尻を叩き続け、そのためか、手のタレントにと誘われた程の我が手も無惨にはれしまいましたので、これ以上編集を待ってられないので、とりあえず今手元にある材料をまとめて出すことにしました。

大変遅くなりましたが、気分一新して(?), 桜いろの4-1号をお届けします。

2. 日中シンポジウム

昨年10月、中国杭州にて開かれた、第2回日中ソフトウェア・シンポジウムの、日本側プログラム委員長熊谷さんに総括を書いていただきました。また、参加メンバー数名の帰国後の感想をあわせて載せました。第1回目の時より、距離(相互理解)が近くなったように思いますが、どうでしょうか。この秋には大阪で第3回が予定されています。中国語に強い人いませんか?

この他に、現地で書かれた各自の日記は整理中です。どうぞ首を鶴などにしないようにしてお待ち下さい。

3. Well come to Essay

4人の方が貴重な時間をさいて書いてくださいました。その原稿も勿論そうですが、添付されている手紙を読むと、嘘ではなく胸がアツクなります。最近、人をアツクさせていない方! 気軽にFaxで何行でも構いませんので送ってください。私はいつでもまっています。

4. 会員総会

当初、5月に予定されていた会員総会は、3月の幹事会で承認され、6月のシンポジウム第一日目終了後に決まりました。詳細は別便にてご案内致します。

5. 次号の予告!?

エッ、次号の予告もするんですか? デハ、編集長になりわかりまして、責任をもって次号以降の予告を..

次の号は、3-10号で予告しているようにSIGENVの特集になる..はず..です...よね?? 昨年11月に神戸で開催された環境ワークショップのまとめも順調(?)に編集が進んでいると聞いていますので、それほど間を置かず次号が出せるのではないかと事務局も大変期待しています。

また、同じ号でEssay「お尋ねします」を書かれた野村さんですが、男の自立度(?)アンケートの結果が、まとめの段階にはいったと聞いています。世のご主人の奥さま依存度がどうであるか、ミーハーの当方としても興味深いことです。期待しましょう。

会員一口アンケートもだいぶ集まりました。ありがとうございます。これも近々にまとめて載せたいと思っています。これからもいろいろなど意見をおよせください。

6. エトセトラ etc.

転勤や移動etc.のご連絡は、早めをお願いします。最近、迷子の郵便が増えてきています。

朝夕の涼しさは、油断できません。皆様ご自愛くださいませよう。今年の軽かった花粉症のおかげで、例年より春先に仕事はかどった事務局からおおくりしました。

第2回

日中ソフトウェア・シンポジウム

総括報告

熊谷章

PFU

0. 内容

若干遅くなったが、昨年の秋に中国・杭州で開催された第2回日中ソフトウェア・シンポジウムおよびその関連行事について、簡単にその概要を報告する。内容は、(1) 日程、(2) 参加メンバ、(3) ソフトウェア・シンポジウム、(4) 大学訪問、(5) 企業訪問、である。また、最後に、今回のシンポジウムおよびツアーに関する個人的な感想を述べさせていただく。

1. 日程

今回のシンポジウム・ツアーの日程は、およそ次のような強行スケジュールであった：

- 10月2日(日) 成田発・上海着
- 3日(月) 上海発・杭州着 杭州観光
- 4日(火) シンポジウム第1日目
- 5日(水) " 2日目
- 6日(木) " 3日目(紹興観光)
- 7日(金) " 4日目 杭州発 上海着
- 8日(土) 上海計算機工場, 復旦大学, 上海交通大学訪問
- 9日(日) 上海発・武漢着 武漢大学訪問
- 10日(月) 日中技術交流会(於 武漢大学)
- 11日(火) 武漢発・北京着
- 12日(水) 中国軟件中心, 北京航空航天大学, 北京大学訪問
- 13日(木) 休日
- 14日(金) 北京発・上海着 返礼パーティ
- 15日(土) 上海発・成田着

2. 参加メンバ

(日本側)

- プログラム委員長 熊谷章(PFU)
- プログラム委員 落水浩一郎(静岡大学)
- 久保宏志(富士通)
- 岸田孝一(SRA)
- 野村敏次(日本電子計算)
- 盛田政敏(KCS)
- 一般参加メンバ 石井 悟(エーエスピー)

梶山雄三（日立製作所）
 片山禎昭（日本システム）
 片山卓也（東京工業大学）
 斎藤信男（慶応義塾大学）
 佐原 伸（S R A）
 下田博次（フリージャーナリスト）
 杉田義明（S R A）
 中野秀男（大阪大学）
 林 香（S R A）
 藤野晃延（富士ゼロックス情報システム）
 水谷時雄（日本システムサイエンス）
 宮成 功（富士通）
 宮本 勲（ハワイ大学）
 方 学芬（静岡大学）
 楊 啓延（S R A）
 通訳 合計 21名

（中国側）

実行委員長 朱 三元（上海軟件技術開発中心）
 プログラム委員長 楊 芙清（北京大学）
 プログラム委員 周 伯生（北京航空航天大学）
 賈 耀良（北京信息工程学院）
 何 克清（武漢大学）
 郝 克剛（西北大学）
 張 然（復旦大学）
 尤 晋元（上海交通大学）
 黄 嘉啓（上海計算技術研究所）
 一般参加メンバ 祝 繼康（上海軟件技術開発中心）ほか37名
 合計 47名

3. シンポジウム**3.1 プログラム概要**

第1日（チェアマン 熊谷 章）

開会の辞：朱 三元

熊谷 章

基調講演：楊 芙清：ソフトウェア工学のための基礎的環境について

落水浩一郎：日本の大学におけるソフト工学研究の現状

朱 三元：上海における計算機利用の調査研究

野村敏次：日本におけるソフト開発環境の実態

歓迎レセプション

第2日 4つの分科会に分かれた並列セッション

第1グループ：設計プロセスの分析とモデリング

(チェアマン 片山卓也・王林泉)

- 発表者 梶山雄三：オンラインアプリケーションの設計プロセス
 陶 増楽：MIS開発のためのラビッドプロトタイピング
 片山卓也：ソフトウェア設計記述のフォーマルモデル
 朱 銘：ラビッドプロトタイピングの基礎としての抽象データ型
 落水浩一郎：設計プロセスモデリングの制御構造
 俞 時権：PADとLCP方法論の比較

第2グループ：設計方法論と設計ツール

(チェアマン 熊谷章・克剛)

- 発表者 宮成 功：構造化設計ツール YACII
 顧 明：PUBを用いたソフト設計のオブジェクト管理システム
 宮本 勲：設計プロセスと方法論の計算機による支援
 周 伯生：巨大／複雑なソフトウェアシステムの統合化技術
 斎藤信男：オブジェクト指向に基づいた分散プロセッシング
 王 吉：コースウェア開発のための支援ツール

第3グループ：ドキュメンテーションと設計の再利用

(チェアマン：盛田政敏・張然)

- 発表者 林 香：巨大アプリケーション開発の設計書作成支援
 張 然：自動プログラミングに基づく実行可能仕様の再利用
 水谷時雄：アプリケーション開発における部品の利用
 唐 稚松：XYZを用いた並列処理と知識の表現
 佐原 伸：CASE ツールにおける再利用機能
 王 令宏：C言語向けのインクリメンタルプログラミング環境

第4グループ：設計の品質管理

(チェアマン：久保宏志・何克清)

- 発表者 片山禎昭：ソフトウェア開発のQCアプローチ
 周 慶隆：SQM のアプリケーション
 岸田孝一：設計プロセスのモニタリング環境
 周 曉梅：ソフトウェアの信頼性モデル
 中野秀男：設計フェーズのアルゴリズムとデータ構造の評価
 眞 梅：MIS 開発へのソフトウェア工学の適用

各グループ共に次のプログラムで発表および討論を行った：

- 午前 (9 : 0 0 ~ 1 2 : 0 0) 発表と質疑応答
 午後 (1 4 : 0 0 ~ 1 7 : 3 0) 自己紹介と討論
 夜 (1 9 : 0 0 ~ 2 1 : 0 0) 討論

第3日 全員で紹興へ観光し、自由討論を通じて相互の親善を深める。

紹興市街見学と魯迅記念館および蘭亭訪問

第4日 各グループの討論結果の発表 (8 : 0 0 ~ 1 0 : 0 0)

第1グループ： 片山卓也

第2グループ： 熊谷 章

第3グループ： 佐原 伸

第4グループ： 中野秀男

クロージングセッション（10:00～11:00）

朱 三元： 今回のシンポジウムのまとめ

楊 芙清： 今回の感想と今後への期待

鐘 錫昌： 日中の今後の活動へのメッセージ

岸田孝一： 第3回へ向けてのメッセージ

3.2 シンポジウム全体の印象

第1回のシンポジウムと比較すると、次の点が大きく変わったといえよう。

- ・参加メンバ：質と量ともに飛躍的に向上した。
 - － 前回：上海地区 + SEAの有志。
 - － 今回：中国全土からの著名な計算機科学者の大部分が参加。
日本の産学から公募し、主要大学／企業が参加。
- ・内容：質と量ともに向上し実質的な討論ができるようになった。並列セッションでは具体的なテーマについて日中の立場から比較検討ができた。お互いの現在の置かれている状況が理解できた。
- ・形態：同一のホテルに宿泊し合宿形式が実現できた。フォーマルとインフォーマル両方のコミュニケーションができた。これには、日程の中に1日近郊へ旅行したことが無用の用として働いている。
- ・発展：シンポジウムの出会いを契機に、種々の大学や研究所と交流ができるようになった。

以下、各イベント毎に寸評してみる。

(1)基調講演： 両国ともに力が籠った発表であったが、ただ、内容が技術の最前線の中味に欠けていたのが残念であった。野村氏のアンケート調査は評判がよかったが、「SEAという特殊な団体が母集団である」という補足が必要だと感じた。朱さんの仕事は、アンケートの集計結果より、コード化などのその過程の方が大変だったと知る。楊先生の話は論理的に美しすぎた。BETAシステムは実在するのだろうか。落水先生の話は、情報処理学会の内部に閉じていたぶんだけ迫力を欠いた。コンテキストに非依存なので、面白味に欠けた嫌いがあつた。しかし、参加者を色々な面から刺激した効果は抜群であった。

(2)グループ討論： 4セッションの並列トラックは、効果があつたように思える。10名から20名の人数で自己紹介を含めた討論形式は、初めて出会うメンバーにはよい形式であった。お互いの置かれている状況と興味の対象が理解できたようだ。今後の具体的な検討の基盤ができたと感じた。次回当たりから実際に役立つ技術交流が可能になると思った。

参考までに、ここで私自身が加わつた第2グループ「設計方法論と設計ツール」における討論の過程で印象に残つたことをまとめておく。

参加者のほとんどが大学の研究者であることがこのグループの特徴であった。なんと18名中14名が大学人であった。午前中に行われた6名の発表の中で私にとって興味深かつたテーマは、顧さんのERAモデルベースのOMS、宮本先生の設いたこれまでのソフトウェア工学研究の批判的継承、斎藤先生が説明された大規模なキャンパスネットワークの統合の構想であった。

午後、参加者全員が自己紹介を兼ねた簡単なプレゼンテーションを行い、その結果みんなが関心を持っているテーマは、以下のようなものであることが判明した：

OS (中国の人に多かった)

ソフト工学全般の動向

AI

CASE

開発環境

CAI

午後から夜遅くまで行われた技術討論では、たとえば、ベトリネットの表現と使い方といった細かな話題も出たが、大半の時間は、中国のナショナル・プロジェクトである SEP (Software Engineering Project) と、私自身も加わっている日米共同プロジェクト SDA との比較検討 (技術的および社会的の両側面から) に当てられた。

議論の内容は以下の通り：

◆プロジェクトの進行

SEP では、86年から88年を第1ステップとし、Unix ワークステーション (Sun や MicroVAX-II) の上でいろいろなツールを開発している。そして、89年から90年を第2ステップとし、32ビットの Unix WS (できれば中国製) の上に統合的なソフトウェア工学支援環境を構築しようと考えている。

一方、SDA では、モデルにもとづく環境のアーキテクチャを追求して、87年には設計のプロセス、プロダクト、および支援ツールの概念モデルについて、根本的な討論を行った。そして、88年は一転して具体的なアーキテクチャの構成要素であるプロセス・マネージャ、オブジェクト・マネージャ、アダプテーション・マネージャ等の機能分担を討論し、UNIX WS や Mac を用いたプロトタイピングを実施している。最終年度の89年には、分散型の設計支援環境のアーキテクチャについての構想の取りまとめを行なう予定である。

◆ユーザのイメージとインタフェイス

SEP では、抽象的にあらゆる階層のユーザへの対応を考えているが、SDA では、さまざまなレベルやニーズを持つユーザを想定して、環境を自由に適合 (Adaptation) させる機能を盛り込もうと意図している。

SEP では、アイコンやメニューあるいはコマンド言語の組み合わせというふつうのインタフェイスを考えているようだが、SDA では、環境が内蔵するモデル (プロセス、プロダクト、ツール) によってユーザに対して何らかのガイドを提供しようとして意図している。

◆基本的な考え方と使用する技術

SEP は、いわゆるナショナル・プロジェクトの常として、絶対に成功させ、実用的なシステムを開発することを建て前として掲げている。しかし、SDA では、研究と製品開発とを明確に区別し、自らを研究開発の最前線に位置づけて、プロトタイプを試作に止めている。

SEP は、基本的に Unix ベースの統合的環境を目指し、その構築に必要な要素ツールを網羅的に自力で開発しようとしている。一方、SDA のほうは、個々の要素技術の開発よりは、それらを統合するための概念的な枠組みの構築を主な意図とし、使えるものはすべて取り入れるという姿勢である。

◆環境の世代

開発環境の発展を、その基礎となるオブジェクト・ベースの観点から整理すると、第1世代は Unix のファイル・システム、第2世代は各種のデータベースの応用だといえよう。次の第3世代について、中国側は知識ベースの実用化を考えており、一方、日本側の参加者はハイパーメディア系のシステムを用いたヒューマンベース・アプローチといったものをイメージしている人が多いようであった。

中国の研究者の方々は、自分たちのやっている SEP プロジェクトは、第1.5ないし第2世代の環境であ

り、SDA は第2.5世代に属するという認識のようであった。

結論をいえば、設計に関する技術の目的は日中で同じであることが分かった。研究開発の内容もかなり似ている。しかし、開発のアプローチは大きく違っているようだ。今後の技術交流を介してお互いにもっと協力し合うことで意見の一致をみた。

(3) 紹興見学： シンポジウム参加者全員で紹興に遊んだことは粹な計らいであった。この小旅行で日中間の親密がかなり深まった。また紹興市と蘭亭は絶好の旅行地であった。この無用の用の働きを考慮しているあたりに中国の歴史を感じた。

(4) クロージング： 中国側はかなりの努力をした後がありありと窺えた。このシンポジウムに参加するために船と汽車を乗りついで2日かかって杭州にやってきた人もいた位である。計画、準備、資金、人員の確保、各種の手続など上海軟件中心は多忙を重ねたはずである。朱さんのクロージングのスピーチには万感が籠っていた。ついで、楊先生と岸田さんの話が続いた。「明年 再見!」、この言葉が日中ソフトウェア・シンポジウムの新しい門出であると思った。

4. 技術訪問

シンポジウム終了後、上海、武漢、北京の各地で、すでに知己となった大学の研究室等を訪問し、研究内容の見学と技術的な討論を行った。私自身が訪問した機関と主な研究テーマを以下に紹介する。

4.1 上海計算機工場（10月 8日、対応者：朱紅群氏）

1966年に設立された、人員1500名を抱えるという企業を訪問した。対応者は朱さんという技術の最高責任者であった。以下の製品を開発している。

- 中型、大型計算機：科学技術分野向け
IBM360/370 に似たもの
- 小型計算機：PDP, VAX(DEC が生産)
- マイコン：IBM/PC(AT,XT)のコンパチマシン
プリント板を購入し組み立てとテストを行う。
- WS：Apollo DN-3000 の製造（ノックダウン）

IBM のコンパチブルマシンは東海というブランド名である。現在の年間生産台数は約5000台らしい。1990年には、年間1万台を目指しているという。北京の方で製造されている長城と東海とはほとんど同じだが、漢字対応部分が少し違うらしい。IBM の機種の変更が早すぎて、それへの対応が大変だという。アメリカで新機種が出てから中国に同じものが出るまでには2年位かかるそうだ。価格は、XT が1万円で AT が2万のぐらいだそうだ。1元が35円位だからおおよその見当がつく。技術者の月給は、高い人でも200元前後らしいから、8年間のサラリーを全部つぎこんで、AT がようやく1台買えるという計算になる。

次に、計算機の利用状況を聞いた。機種では、80%がパソコンで、20%がその他である。利用分野は、企業管理、金融、軍関係の順である。Apolloのノックダウンは、累積で200台の製造実績がある。当面の課題は、計算機応用技術の開発と普及、それに人材の育成だと聞いた。上海では小学生から計算機を教育していると話してくれた。

ついで、工場を見学した。IC テスターやボードテスターが置いてあり、一応生産ラインらしきものがあったが、ラインは動いておらず、製品を作っているようすが見られない。DN-3000 を使ってプリント板の設計をフランス製の CAD を使って作成しているという説明もあった。

この会社は、計算機製造の外に販売もやっている。従業員の約半分がハード／ソフトの技術者である。技術部門の組織は以下の5課から成っている。

- 小型／中型の研究開発課
- マイコンの研究開発課
- 工業用制御の応用課
- 中型／大型／マイコンの金融応用課
- ソフトウェア研究開発課

われわれに見せてくれた工場のほかのどこかに、すごい工場があるに違いないと思った(!?)。この工場では年間5000台も作れるはずがないからだ。誰かが歩留まりはどれくらいですか、ときいたら6～7割と答えてくれた。この工場は長江計算グループの主要工場である。そういえば、上海市内で長江グループの会社を見掛けたことを思い出した。この工場に、いくつかの日本のメーカから合作の話を持ってきたこともあるそうだ。

4.2 上海交通大学 (10月8日, 対応者: 尤晋元教授)

北に清華, 南に交通あり, といわれる程の名門校らしい。計算機に関しては次の点に力点を置いている。

- 新しいコンピュータアーキテクチャの研究開発
 - 西ドイツの大学との共同プロジェクト
 - データタイプマシン
- 画像処理技術の研究開発
 - 画像処理用の専用システムで様々な映像を解析する。
 - VAX, IBM/PC, M68000 等を使用していた。

大学案内用のビデオが日本語で流れてきた時は驚いた。東京大学等の日本の大学との交流があるという。ハワイ大学の宮本先生とメリーランド大学と一緒に研究をしたという女の先生に出合った。VAX を使ってCADのプロトタイピングツールを作成していると話してくれた。アメリカではどうして Smalltalk-80 が評判が高いのかと聞かれた。少し話したが時間がなかったので、シンポジウムの私のペーパーを見て下さい、とっておいた。

この大学で面白い光景を見た。なんと10歳前後の少年が計算機を一所懸命使用しているのだ。BASIC 言語の演習かなと思って聞いてみたら、かれらはれっきとした大学生だった。いま中国では、大学入学の年齢制限がないところもある。成績さえよければ誰でも入学できる。その子は13歳だった。いわば、天才少年であった。この制度は10年位続いており20歳前に大学を卒業している人もいるらしい。

4.3 武漢大学 (10月9～10日, 対応者: 何克清教授)

この大学は長江のほとりに栄えた武漢市にある。森と湖に囲まれたとてつもなく広く美しい学校だった。宮本先生によるとアメリカでもこれほど広く綺麗なキャンパスはないという。何教授が所属する軟件研究所はソフトウェア工学では中国唯一のナショナルプロジェクトだと聞く。したがって、資金があり、多くの計算機設備を持っていた。何教授を筆頭にここの研究生はかなりの人が日本語が話せた。日立と関係があるらしい。

主として次の技術に関して議論した。

- PAD とプログラミングの連動
- オブジェクト指向技術の有用性
- プロトタイピング技術
- 設計方法論とそれのためのツールと開発環境

計算機は、VAX8600, VAXStation, IBM/PC, NEC/PC を使っていた。片山社長と落水先生、それに岸田さんが特訓講演をした。宮本先生も80分の講演をした。今の中国には、宮本先生の研究の歴史が一番役に立つと思った。なぜなら、一世代前の計算機設備が中国では流行しているからだ。私の今回のシンポジウムの論文を読んで質問してきた学生がいた。10人位の学生に囲まれて1時間以上あれこれと質問攻めにあった。内容は Smalltalk-80 に関するものだった。この経験を通じて、こうした国際交流シンポジウムの意義を実感することができた。

4.4 中国軟件技術開發中心 (10月 12日, 応接者: 鐘錫昌氏)

鐘先生のお招きに預かり中国軟件技術開發中心の人々とオブジェクト指向に関する技術討論会を行った。中国側は中国人民共和国国家科学技術委員會の新技術局長の劉氏や北京航空学院の金教授他約20名であった。自己紹介の後に、Smalltalk を中心とした技術討論と開発環境に関する議論をした。

内容は、中国の計算機設備は IBM の大型機と VAX と IBM/PC がほとんどであるから、ソフト工学の最前線を研究開発するには、もっとほかのマシン、たとえば Mac や Smalltalk WS をうまく使用するべきではないか、ということであった。私と宮本先生と佐原氏が経験に基づいた結果を説明し議論した。再利用と Smalltalk の実行速度が話題になった。

この開発中心(センタ)は、上海軟件技術開發中心と同じ性格の機関で、国家の科学技術委員會に属している。上海センタの親機関に当るらしい。

4.5 北京航空航天大学 (10月 12日, 応対者: 周伯生教授)

北京の郊外にある古い歴史が感じられる大きな木に囲まれたキャンパスを訪れた。目的は中国のナショナルプロジェクトである SEP の研究開発の一環を見学することだ。

薄暗い奥まった校舎の片隅に、AT&T/3B20 と IBM/PC を使用して開発が進められていた。主な内容を以下に記す。

- C 言語プログラムの静的、動的解析とその評価ツール
- オブジェクトのマネージメントシステム (OMS)
- グラフィックスライブラリとその利用環境
- Pascal のシンタックスエディタ

題目は結構であるが内容はいま1つであるようにみえた。全体的に欧米のペーパを参照しながら作られたツールであり、何のための環境やツールであるかが余り考えられていないように感じた。いまの中国のアプリケーションの現状では仕方のないことなのかも知れない。

4.6 北京大学 (10月 12日, 応対者: 楊芙清教授, 陳桂副学長)

今回のシンポジウムのプログラム委員長を私と一緒にやってくださった楊教授が快く我々一団を迎えてくれた。そして、副学長の陳教授も応対してくれた。このキャンパスは昔、清朝の貴族の庭園だったらしく手入れがゆき届いていた。外国人の研究者が多く見受けられた。日本を研究するための建物もあった。楊教授から北京大学における計算機関係の論文集を頂く。

陳副学長は核物理学の専門で、見るからに中国の知識人の風貌をしていた。外国人専用の学内の飯店で御馳走になった。やはり、北京の料理は外の場所の味つけとはかなり違う。日本人にはよく合った料理法だ。陳教授は来日したことが2回あり、とても刺身が好きだそう。中国の大抵の方は刺身やすしのような生身の魚が嫌いかと思っていたら、会った人のほとんどが好きだという。現実には意外なものだ、と感じた。

今まで、訪問した大学の中でこの大学が最も日米の大学に近い。

5. 感想

今回の旅行で以下のようなことを考えた。

中国は偉大な国で、社会資産は日本よりはるかに豊である。道路、建物、公園、遺跡、大学等の施設は、日本のそれらとは比較にならないほど立派である。4車線の道路に自転車道があり、更に広い歩道がある。そしてその脇には樹齢100年を超すと思われる見事な街路樹がある。こんな社会資産を日本では見られない。

中国の計算機産業は、日本の10年位前の状態に似ている。キャラクタベースのTSS全盛で、Unixの流行が始まったころにそっくりである。技術的にも大学指導型であり、すべてをアメリカに見習っている。人気のある、つまり評価の高いマシンがVAXやIBM/PCである。ビットマップやグラフィックスがまだ一般化していない。しかし、いまやダイアグラムやグラフィックス中心でないとソフト工学や最新技術が研究開発できない状況にある。

能力の高い若き技術者が熱意に燃えている。アメリカへの2～3年の留学を終えた心技ともに充実した研究者が、どこの大学にも沢山いた。かれらにいま足りないものは、よき指導と十分な計算機設備である。条件が満たされれば、きっとすばらしい仕事をするに違いない。

日本と中国は計算機産業でパートナーになれる可能性が高い官民学のあらゆるレベルで技術交流を促進しお互いの利益になる道を見つけることが可能だと感じた。日本では優秀なソフトウェア技術者が欲しい。90年代には、90万人も足りないといわれている。一方、中国では技術者とその候補生が沢山いるが、具体的な仕事と計算機とノウハウがない。人件費は、単純な貨幣の為替レートで計算すれば、約100分の1である。ソフトウェア部門では、互いに技術協力し商売が成立する可能性が非常に高いと感じた。

われわれのシンポジウムは、日本側が任意団体のソフトウェア技術者協会の主催であり、中国側は人民政府管轄下の上海軟件技術開発中心と上海対外科学技術交流中心が主催である。中国側のメンバは、ほとんどが大学の先生で当代の一流の研究者が勢揃いしている。いわば、中国側は官学のオールスターキャストであった。一方、日本のメンバは産学の連合部隊で、中々の豪華な顔ぶれであった。このように民と官が自由に技術交流ができ、お互いの研究開発に貢献しあえることには、色々な面で意味がある。多分、この産業分野では史上初のことだろう。インフォーマルではあるが通産省の強い支援もあり、今後もこの活動を継続すべきだと思う。

6. 付録：「時間のある風景」

「時間のある風景」に出会った。その風景は自分の生き方に少なからず影響を及ぼしている。

風景といえば、直ちに遠近法を連想するが東洋にはもともと遠近法概念がなく平面的な表現が主流であると思っていた。それは明治時代に西洋から輸入された技法である。それまで日本の伝統的な絵は2次元の平面であるし、本家である中国の古来からの水墨画も平面的なタッチである。東洋の物の見方と考え方は2次元の平面的なものがベースであると理解していた。

ところが、中国の自然と風景に出会ってある種の感動をえた。その感動を想起してみると、「時間のある風景」と形容するのが最も適していると思われる。

アインシュタインの物理学を学んだときから、システムの系の考え方や相対的な物の見方を身に着けたつもりでいた。だから、大抵の事象に出合ってもそのシステムの系にしたがって考えることや、系を変えて物を見つめ直すことができた。したがって、多少のことでは驚くことはない、とたかを食っていた。日本で起きるさ

まざまの事象は、その事象が発生したシステムの系とそのシステムが所有している価値尺度を考えれば、だいたいの価値基準が推し測れた。

しかし、中国ではこうはいかなかった。自分が考えられるどのような系や尺度を持ってきてもすべてスケールエラーとなり適用できなかった。中国の系は、3次元の広がり他に、時間的な広がりも持っていた。正に相対性理論である。目前の自然現象と社会現象の中で四次元空間を体験できるとは予想していなかった。

それは、武漢の晴川飯店のバルコニーから川面に浮かぶ朝陽を眺めているときだった。

薄霧に覆われたゆったりとした広い川面にオレンジ色の太陽が輝き、水面にはあの極楽浄土への一筋の黄色の途が続いていた。周りからは行き交う船の音が霧をついてしきりに聞えてくる。周りの霧の中には行き交う船の音が充満している。ポッ、ポッ、ポッ、―――、ポー、ポー、ポー、―――、ブーン、ブーン。飛び交う人の声が遠くの方から聞こえてくる。朝陽の反対側の右手には全長2 kmに及ぶ橋がある。その橋の街燈が薄闇の中で一直線に並び幾何学的な美しさで輝いている。それらの真ん中の方向にぼんやりと浮かんでいる黄鶴楼が見える。この風土の中では、空気も水も時間もゆったりとあるうねりをもってうごいている。多くの古代の詩人たちが詩と詞に歌い上げた風景をいま垣間見た。

このときである。すーっと、時間軸に沿って自由自在に目前の空間内を渡り歩けると感じた。それは例の決定的な瞬間であった。現代の主人公ロカントンが公園で樹のざわめきを見ながら感じとったあの瞬間と同じきがした。存在と時間の両方が見れる場に出合った気がする。日本の風土にはこのようなものが少ないように思う。

空間内に時間を感じるためには一種の風土が必要だ。日本の風土の中にはx軸とy軸しかないように思える。慣れ親しみ過ぎた風土では、人はその風土の一部になり盲になり本来感じるべき物も分からなくなる。異国の地にくれば、その風景からはみでているために、初めてよくそれらを感じ取れるだろう。

そういえば、10年位前にアメリカのコロラド山脈の真只中で似たような気分になったことがある。

それはコロラド山脈が織りなす大パノラマを目にして、大自然のあまりの雄大さに茫然自失となったときだ。あのときを想起してみれば、空間が停止しているように思えた。その空間はx, y, z軸の3次元空間であった。そのときは、この景色は日本のそれとは違うと感じた。日本の風景にはz軸がないと直感的に思ったものだ。

長江あたりを見ながら、老子の中に時間の概念があることを思い出した。中国には古来から時間を1つの軸として考えていた事実がある。そのことが実感として分かった、積もりになった。こうして、晴川飯店のバルコニーから眺めた風景は小生の「忘れ得ぬ風景」の1つになった。

日中ソフトウェア・シンポジウム参加報告

佐原 伸

S R A

A. Documentation and Reuse of Design 分科会報告

1. はじめに

この報告は、レポーターである筆者が、レポーターであることを忘れて討論していたため、事実の一部を主観的に伝えている確率が高い。司会は張然(復旦大学)・盛田政敏(KCS)の両氏で、発表は主に英語、討論は主に自国語で行なわれた。通訳という重労働を、張然氏が引き受けて下さった。出席者は当初11名だったが、夜から他のグループのメンバーが合流し、約15名となった。

2. 発表内容

発表は当初予定された6件以外に、3件が追加された。表1に題名を、以下に概要を列挙する。

表1. ドキュメントと設計の再利用分科会発表タイトル

発表者	所属	題名
林香	SRA	Design Documentation Support in the Large Scale Application Development
陳万軍	復旦大学	REAP, A Reusable Executable Specification Based Automatic Programming System
水谷時雄	日本システムサイエンス	Utilization of Software Parts in Application Programming System
唐稚松	中科院軟件所(Cinica)	To Represent Concurrency & Knowledge in XYZ
佐原伸	SRA	Reuse Aspect of CASE Tools
王令宏	上海交通大学	An Incremental Programming Environment Oriented C Language
盛田政敏	KCS	J-Starドキュメント・データベース
佐原伸	SRA	CASEの経験
藤野晃延	富士ゼロックス情報システム	Formal Spec., HyperMedia and CSCW (Computer-Supported Cooperative Work)

- (1) 林氏の発表は、SRAにおける大規模ドキュメントシステムの開発事例報告であり、システム開発における設計ドキュメント作成に、大きな成果をあげた例である。
- (2) 陳氏の発表は、OBJに似た代数的仕様記述言語(SEED)、知識ベースを使ったインテリジェント・マンマシン・インタフェース、プロトタイピング機能、LispまたはC言語生成機能を持ったREAPシステムについてその再利用性を述べたものである。
- (3) 水谷氏の発表は、マクロを利用した再利用可能なプログラム生成ツール、JASMACのアーキテクチャを述べたものである。
- (4) 唐氏のXYZシステムは、XYZ/Eと呼ばれる論理型言語を中心とした環境とツールの集合体で、
 - (a) 統一化されたパラダイムの実現
 - (b) 知識ベース・プログラミングの支援
 - (c) プログラムの再利用の支援
 - (d) 理論と実用性の境界の発見
 を目指している。

システムは、

- (a) 高級言語によるプログラミング¹
- (b) 仕様言語によるプログラミング
- (c) データフローによるプログラミング
- (d) 分散型並行プロセスの設計のためのグラフィック・ツール
- (e) オブジェクト指向プログラミングの管理システム
- (f) 段階的詳細化と検証
- (g) インフォーマルな要求からフォーマルスペックへの変換

などを支援する。

- (5) 佐原は、CASEツールの再利用と、曖昧性の排除のために (1)ハイパーテキスト機能の追加と(2)オブジェクト指向プロセス仕様言語の採用を提案した。
 - (6) 王氏は、C言語のインクリメンタル・コンパイラーを中心としたプログラミング環境について発表した。
 - (7) 盛田氏は、KCSで実現した、J-Starを利用したドキュメント・データベース・システムの経験について発表した。
 - (8) 佐原は、中国側参加者の多くがCASEツールの具体的イメージを持っていないため、CASEツールの概要を説明した。
 - (9) 藤野氏は、フォーマルスペックとマルチメディアあるいはハイパーテキストとが矛盾する概念ではなく、むしろこれから両者が結合していかなければならない、という主張を行なった。
- さらに、CSCWについて概要の説明を行なった。

3. 討論内容

発表および発言内容は主に次の三つのグループに分かれた。

- (1) 従来のドキュメントを解析し、データベース化するという主張をするグループ
- (2) 構造化分析用CASEツールなどを使って、構文レベルではフォーマルな方法を使おうと主張するグループ
- (3) フォーマルスペックを追求して「意味」をもフォーマルに記述しようと主張するグループ

2番目と3番目のグループは、いずれもオブジェクト指向の有効性を認めていて、自分のシステムにそれを取り込もうとしていた。また、ハイパーテキスト・ハイパーメディア・CSCWなどの考えを採用すれば、より一層効果が上がると主張するものもいた。

2番目のグループが1番目のグループを「構文が曖昧すぎる」と言って批判し、3番目のグループが2番目のグループを「意味が曖昧なので、危険すぎる」といって批判する形で議論が進んだ。結果は、唐氏と藤野氏と陳万軍氏のフォーマルスペック派に、他の2派はたちうちできず、フォーマルスペックを使ったアプローチのどれが優れているかという方向に議論が進んだ。

結論から言えば、我々の議論はすべて、唐氏の掌中にあるという感じてあった²。「あなたの意見は興味深い、しかし…何故こうしないのか…」で始まる唐氏の意見は、すべて核心を突いていて、唐氏がその分野の主要な論文はすべて読んで、自分でも実験していることが明らかだったため、他のメンバーは「I think so.」と言う以外なく、討論に至らない場面が多かった。

かろうじて、ハイパーテキスト・ハイパーメディア・CSCWに関して、唐氏はまだはっきりした見解を持つ

¹ 属性文法に従ったコンパイラ・コンパイラ(Yet another YACC)を使い、PascalからXYZ/Eへの変換ツールなどができている。

² 後日、北京で唐氏の研究所を訪問したが、唐氏のXYZシステムは、この分科会で議論されたことのすべてを考察し、整理し、実験しているという印象を持った。

ていないという立場から「興味深い」と言うにとどまり、「しかし…」が続かなかったのではとした³。

4. 全体の印象

この分科会の参加者の興味は広く分散していたため、個々の話題について突っ込んだ意見交換ができないうらみがあった。

また、唐氏が大先生であるため、中国側の若手が思い切った発言をできない面があったようだ。

B. グループ討論について

1. はじめに

Reuse aspect of CASE toolsというタイトルで、以下のような発表を行なった。

2. 発表内容

構造化分析⁴・構造化設計⁵を支援するCASE(Computer Aided Software Engineering)ツールは、ソフトウェア開発の全ライフサイクルをカバーし、あらゆるタイプのソフトウェアの開発・保守に役立つといわれている。しかし、現在のCASEツールは、

- (1) システムの、モデルとしての再利用が困難である。
- (2) プロセスベックの記述方法が明確でない。
- (3) 少しずつ機能の異なるプロセスをいくつも持たなければならないような、分散処理システムの記述には適していない。
- (4) システムを構造化分析・構造化設計のツールだけで表すことな困難である。

といった欠点を持っている。

本稿では、これらのうちの「モデルとしての再利用」に必要なアーキテクチャについて考察する。

2. 1 CASEツールの現状と問題点

現在のCASEツールの優れたものは、

- (1) データフロー・データ構造・モジュール構造ダイアグラムなどの支援
- (2) データ辞書の支援
- (3) プロジェクトデータベースの支援
- (4) ユーザによるカスタマイズが可能
- (5) ユーザーインターフェースの統一
- (6) リソースのシェアに対する配慮
- (7) 整合性チェック
- (8) 分散ネットワーク環境対応

などの機能・特徴を持っていて、大規模なシステムをネットワーク環境上で多人数で開発するための機能を、一応持っている。

しかし、一方で、

- (1) 一度作ったモデルの再利用は、データ辞書の助けはあるものの、ユーザーが自分で似たモデルを探さな

³ もっとも、これも後日北京で分かったことだが、XYZシステムは再利用に関して、ハイパーテキストを使った仕掛けを考えている最中だった。

⁴ P.T.Ward & S.J.Mellor, Structured Development for Real-Time Systems, Yourdon Press A Prentice-Hall Company (1985)

⁵ M.Page-Jones, The Practical Guide to Structured Systems Design, Yourdon Press A Prentice-Hall Company (1980)

ければならない。

(2) プロセスベックの記述方法が統一されていないため、他のユーザが作ったモデルを再利用するのが困難。

(3) 少し似ているが異なるプロセスを、うまく記述することができない。

(4) 少し似ているが異なるプロセスを、うまく再利用できない。

(5) システムを機能面から階層化されたデータフローだけで表すのは苦しい。

といった問題点があり、再利用に関する機能は十分とはいえない。

2. 2 解決のためのアイデア

上記の問題点を解決するために、下記の機能を付加することを考える。

(1) ハイパーテキスト機能

(2) オブジェクト指向プロセスベック定義言語

2. 2. 1 ハイパーテキスト機能

ハイパーテキスト⁶システムは、HyperCard⁷に見られるように、対象システムを階層化するのではなく、各オブジェクト間に自由なリンクを許し、そのオブジェクトのビューがマルチウィンドウ上の一つのウィンドウないしはアイコンとなるようなシステムの総称である。

また、絵や音声や画像といったマルチメディアを支援するハイパーテキストシステムが多い。

このハイパーテキスト機能をCASEツールに付加すると、

(1) システムを機能面から階層化されたビューだけでなく、いろいろな角度から見ることができるので、理解しやすく、再利用しやすい。

(2) 少し似ているが、異なるプロセスを参照するリンクをつけることで、重複をある程度避けることができる。

(3) システムのイメージを絵や音声あるいは画像で表せるため、直感的にシステムを把握することができ、再利用のための知識移転をはかりやすい。

という効果がある。

2. 2. 2 オブジェクト指向プロセスベック定義言語

プロセスベック定義言語として、オブジェクト指向言語を採用すると、

(1) 少し似ているが異なるシステムを、クラスの継承(インヘリタンス)機構を使ってうまく記述できる。

(2) 従来の機能面からの階層化だけでなく、クラス間のインヘリタンスの階層化によって、アプリケーションドメイン毎のモデル記述がやりやすくなり、モデルの階層化ができる。

(3) プロセスの独立性を高めることができ、プロセス単位での再利用性を高めることができる。

という効果がある。

3. 反応

唐・張両先生を除いては、CASEツールおよびハイパーテキスト・システムの具体的イメージが湧かないらしく、かなり説明をしなければならなかった。HyperCardについてもかなり説明を求められた。

唐先生は「とても面白い」とおっしゃってくれたが、後で、北京の先生の研究所を技術訪問してその訳が分かった。唐先生のところでも、同じようなアイデアで作業を進めていたのだ。もっとも、こちらのアイデアが1%しか固まっていなかったのに対し、向こうは50%くらい固まってきたとのこと。

⁶ J.Conklin, A Survey of Hypertext, MCC Technical Report Number STP-356-86 (1987)

⁷ Danny Goodman, The Complete HyperCard Handbook, Bantam Books (1987)

C. 第2回日中ソフトウェアシンポジウムの感想

1. 基調講演

楊先生のソフトウェア工学環境に関する説明は興味深かったものの、説明内容が専門家向けでなかったため、会場の暑さも手伝い、眠気を誘った。

落水先生の講演は、情報処理学会のソフトウェア工学に関する論文を分析し、日本の企業と大学の研究内容を批判したもので、先生の今までの話の中で、もっとも過激で面白かった。

野村氏の、ソフトウェア技術者に対するアンケート結果の分析が興味深かった。発表時間が少し短すぎた思いがする。

2. 分科会

分科会(Documentation and Reuse of Design)では、HyperCardについて説明を求められたが、HyperCardに関するOHPを持っていなかったため説明に苦慮した。ハイパーメディア・システムを口で説明するのはむずかしい。

中国側の発表の中では、唐教授のXYZシステムが一番独自性があり、かつバランスがとれている感じだった。もっとも発表時間が予定の2倍になり、討論時間が削られたのが残念である。

中国側の発表内容は第1日目に手渡されたが、やはり日本にいる間に概要を把握できるような準備が必要であろう。

夕方から、他のグループにいた藤野氏が参加し、分科会の趣旨とはややずれる発表をしながら、議論が盛り上がった。それまでは、発表が淡々と進み、質問もあまりないといった具合であったが、やはり、シンポジウムでは予想外の展開が起きないと面白くない。中国側参加者は、多少結果が予想できなくとも、その場で思いついたことを発言するといった、インフォーマルな形での討論が不得意なようだ。

3. その他

夜のレセプションでは、北京の周さんという若い学生が、積極的に日本人と話していた。IBMの大場氏と共同で論文を書いたことがあるそうだ。中国の若い研究者は、何とか海外に出たがっているようだ。日本のLispの普及状況に話移ったとき、たまたま相手がTao氏だったので、「日本で有名なLisp系言語の名前ですよ」と教えると、大勢集まってきてLisp人口・Prolog・Smalltalkの話になった。その中に、ハンガリー製のM-Prologを改造して、抽象データ型を扱えるようにした人がいたので少し話をした。IBM4300シリーズの主記憶5MBの上で、作業しているとのこと。自宅で、主記憶5MBのMacintosh II上のSmalltalkを使っていること、あるいはNTTの竹内氏が128MBのELISでTAOを動かしていることを話すと、とても羨ましいと言っていた。研究者が使っているシステムが貧弱なため、本来の研究よりも、環境整備に時間をとられているようである。

シンポジウム全体を通して、観光にでかけるときも、中国の若手と日本のメンバーとは違うバスという具合で、周さんのように積極的に話しかけてくる人以外、中国の若手研究者の意見を聞く機会がほとんどなかったのが残念である。次回からは、もう少し若手と交流したい。

D. 技術訪問の感想

1. 上海

上海計算機工場と上海交通大学を訪問した。

交通大学は、プログラミング言語としてPascal, Lisp, Prologに重点を置いている。CADのシステムをFPで実現しているのを見学したが、旧式のカラーグラフィック・ディスプレイを使用しているため、マルチウィンドウ・マウス・アイコンなどの機能がなく、ユーザーインタフェースの実現に苦勞している様子だった。

Micro VAXを使い、OSはVMSとのこと。

14歳の大学1年生がTurbo Pascaでプログラミング実習をしていた。小さな大学生は3人いた。中国では、成績がよければどんどん飛び級できるが、大学に入るためには、普通の受験生より成績がよくなければならないそうだ。14歳で大学に入るためには、ほとんど満点でなければならない。

計算機工場ではPC/XTを主に生産しているが、注文が少なく、皆遊んでいた。PC/XTの値段が、日本で買う場合の10倍では無理もない。他にPS/2やAppoloも生産しているが、Appoloは組み立てのみとのこと。

2. 武漢

武漢大学は、主力コンピュータがVAX-8600, Micro-VAX. VAX-8600のOSはまたしてもVMS, UNIXの方が研究に向いているのだが… MicroVAXはUltrixなのでまあUNIXだが、パークレイ版の方が研究向きなのに… 100万ドルかかった設備なので、日曜日でも使えるようになっているそうだ。しかし、学部の学生には使わせないという。

研究分野はかなり広い。ラボの基本方針は「オープン政策」で、米国や日本の大学・企業と共同で、いろいろな研究を進めている。

Smalltalkを使って、要求分析あたりの方法論・ツールを研究するグループの話聞いた。現在はゴールドバーグ女史の本を元に、VAX-8600/VMS上で、C言語を使ってインタプリタ・コンパイラ・デコンパイラを開発し、PC/XT上のSmalltalk/Vのプログラムを移植すること。要求分析ツールについては、まだ何もできていない。

ParkPlace SmalltalkやMacintoshなどの動向を話して、アプローチがおかしいのではないかと指摘したが、あまり反応はなかった。VMS上のテキストベースのブラウザを見ていて感じたことだが、研究方向を誤っていらぬところに労力を掛けすぎている。パワーはあるだけに惜しい。

他のグループでも、X-Windowのプログラムをほとんどマニュアルなしで作っていた。中国にコンピュータを輸出しているメーカーは、正しい情報を中国側に伝えていないのではないか？

3. 北京

午前中北京航空学院を訪問。午後は北京大学。翌日はCinica(中科院軟件所)が作った会社で、唐教授のグループを訪問。

航空学院ではCの支援環境・CASEツールなどを見学したが、ワークステーションを使っていないため、ユーザーインターフェイスに大半の労力を費やしているようだった。

北京大学では、学内観光のみ。

唐教授のところでは、XYZシステムのSUN上のデモを見た。SUNはイーサネットで接続されている。シンポジウムでは、ペーパーだけでイメージが湧かなかったのだが、デモを見ると、正しいアプローチで、興味深い目標に向けて進んでいるのが分かる。

フォーマルなアプローチとインフォーマルなアプローチの接点を見つけるべく、属性文法によるYet Another Yaccコンパイラ・コンパイラなどで、データフローやPascalから論理型言語XYZ/Eへの変換ツールなどを作り、「意味」を考慮した段階的設計と検証を目指している。検証されたモジュールをオブジェクトとして格納し、再利用するためのソフトウェア・データベース的なものも考えている。

万里の頂上の見学をとりやめて訪問した甲斐がある研究内容であった。

日中ソフトウェア・シンポジウム 第4セッション・レポート

中野 秀男

大阪大学

1. テーマ

設計の品質管理（チェアマン：久保 宏志，何 克清）
午前中に発表。午後から，自己紹介に続いて質疑／討論。

2. 発表者

片山 視昭（日本システム）	Application of quality function development to software development
周 隆（SCC）	An application of SQN technology
岸田 孝一（SRA）	Support environment for design process monitoring
周 暁梅（北方交通大学）	ソフトウェア信頼性成長モデル
中野秀男（大阪大学）	Evaluation of algorithm and data structure in design phase
夏 梅（武漢大学）	ソフトウェア工学のMISへの応用

3. 座長と発表者以外の参加者

金 旦貨（上海鉄路局計算中心）
野村 敏次（日本電子計算）
王 建平（上海市人民政府科学技術委員会）
張 春宝（上海大学）

4. 全体的感想

(1) 発表内容の相違

日中双方から，3名ずつのプレゼンテーションがなされたが，全体的に見て日本側の発表は，理論的なものが多かった。悪く言えば，抽象論である。中国側の発表は，事例に基づくものが多かったが，問題にしているところが，どちらかと言えばプリミティブなものが多かったように思う。これらのプレゼンテーションをベースに議論を進めることは，難しいと見ていたが，何とか討論項目はまとまった。

(2) 焦点のずれ

全体の大枠が設計工程に絞られていたものの，話はソフトウェア開発の全工程に流れた。それはそれで議論が弾み問題はないが，かんじんの設計工程における品質管理が議論されなかったことは心残りである。

(3) チョット消極的

理論的なことに関し，中国の方々が日本側の意見を聞くという態度が多々見られ，自分はこう思うという強い主張がなかったのは残念である。例外は北京北方交通大学の女性の信頼性モデルにおける，自己主張の強さである。何か，両極端を見た気がする。

(4) 有意義だった自己紹介

討論参加者全員の，現在の仕事も含めての自己紹介は，中国のソフトウェア事情を知る上で非常に有効であった。特に印象が深かったのは，鉄道局の金さんである。中国の鉄道における，コンピュータ利用の

実情が良く理解できた。

5. 金旦さんの自己紹介

上海鉄道局は中国の鉄道部に所属していて、さらに上海、杭州、南京などの6つの地区に分かれ、その下に があります。鉄道部では、VAX8350 を持ち、上海鉄路局ではVAX730, MicroVAX, LSI-11/23, PC/XT をもち、VMS 上の DECNET で、Fortran77 や COBOL や C でシステムの開発を行い、運用・管理をしている。各局や中央とのデータの交換には 1200/2400bps のモデムを用いている。保守に関心がある。

6. 発表に対する質問

Q. 野村→片山： 会社として全体的に QFC または QC をしているのか。

A. 片山： 導入したてで、モデルプロジェクトを2, 3選んでやっている。

Q. 金→岸田： モニターは管理者と開発者にどういう意味があるのか。

A. 岸田： 管理者は設計者の考え方が分かる。設計者は、設計の中で表われる問題点がトレースできる。起承転結が分かる。

Q. 片山→張： ASSMIS プロジェクトのコスト推定、結果、困難だった事、開発期間、品質、人的管理はどうだったか。

A. 張, 何： 当初、70人月の予想が50人月でできた。費用は5万元

7. 討論する内容

a. 岸田 (SRA) の発表にあったプロダクト向きパラダイムとプロセス向きパラダイムの理解

岸田さんから、プロセス向きとプロダクト向きの違いについて説明があった。詳しくは ACM/SIGSOFT の記事を読んで下さい。TQC, QFCはプロセスをプロダクトにつなぐ道具である、が結論。

b. 信頼性モデルを数学のゲームと考えては。(周曉梅)

片山 (日本システムズ) さんより、まず信頼性モデルの説明があった。それから、2人で議論。結局、

片山： どのようなモデルをたててもラフには指数関数モデル (縦横が違うが)

周： ゲームと片山： 現実の議論は最後までかみ合わず。

c. 張春室 (上海大) "OSI にもとづいた Teletex システムの開発"

張春室さんの発表のあと討論となり、その最後で可用性の判断の事が議論となった。そこで、日本と中国で、どのような場合に商品にするかが話題となった。日本では、マーケティングにより判断する。中国では、完全なものなら商品にすると回答があり、日本側から完全とは何かと質問があり、(文書、レビュー、品質《機能、性能、ヒューマンファクター》)と返事があった。

d. 討論における感想

体制の違いからの当然の観点の相違であるが、この事や歴史を含めた背景の違いを肝に命じて常に討論しないと、とんでもない思い違いをするだろう。

中国の方に質問した時、その事が中国としてのコンセンサスがあるとき、それをすべて喋るか、用意していないので答えられないのかどちらかで、個人的見解をシンポジウムの公式の席では述べられないことがほとんどであった。また逆に、個人的な席でもらされた言葉は我々は文書にしているのだろうかと思つた。

中日軟件研討会'88感想文

林 香

(株) SRA

私が参加したグループは Documentation and Reuse of Design です。座長は復旦大学の張先生と、KCSの盛田さん。とりようによっては、どうとでもとれるタイトルのついたグループでして、案の定グループ内の発表にも色々ありまして、発表順にいきますと、

- ・大規模ビジネス・アプリケーションにおけるドキュメンテーションのオンライン化の経験(私がトップ・バッターでした)

の話から

- ・フォーマルな仕様記述をベースにした自動プログラミング・システム (復旦大学の陳さん :張先生のところの院生なのですが、これはなかなか良くやっている感じでした)
- ・部品化による事務アプリケーション・プログラムの再利用 (日本システムサイエンスの水谷さん:話しなれた感じで大変わかりやすい発表でした)
- ・超メタな言語 XYZ を中核にして、Knowledge-based Programming から Programming with specification, Object oriented Programming まで何でもサポートしてしまおうというものすごい環境(中科院軟件所の唐先生: かなり先進的なものです)
- ・ハイパー・メディアを利用したソフトウェア・データベース化のアイデア(SRAの佐原さん :ハイパー・メディアは中国の方にはかなり興味を持たれました)
- ・C言語向けの Incremental Compiler を中心にした開発環境(上海工科大学の王さん: ただ一人の女性の発表者、大変熱心な発表でした)

の話まで、良く言えばバラエティに富んだ、本音をいえば一体話が噛み合うのかしらの内容的のものでありました。

私の発表内容を要約しますと、ドキュメント総量がA4版で数千ページになる程度の大規模システムの設計ドキュメント・オンライン化支援環境の構築/利用経験についてでありまして、この種のシステムにおける問題は、膨大なデータの管理と、使い勝手の良いユーザ・インターフェースの提供にあるのですが、現時点のコンピュータの能力では

ユーザ・インターフェースをとるならいわゆる WYSIWYG方式

膨大な量のデータ管理に着目するならバッチ的文書フォーマティング方式

のいずれかを選択せざるをえないわけです。私達のアプローチは、UNIX上のバッチ的文書清書ツールである troffを中心にユーザ・インターフェースの改善を行なおうとするものでして、troffの特性である、強力な文書清書能力、大量ドキュメントの管理の容易性を生かしながら、貧弱なユーザ・インターフェースを何とかしようというものです。具体的には、設計ドキュメントのフォームごとにtroffのプリプロセッサを作成し、ドキュメント入力を簡易化したわけです。また、この他にも、UNIXの既存ツールを利用して、管理面での改善も行ないました。といったことと、その成果について話をしました。

中国サイドからの質問で印象に残ったものは、確かに面白いアプローチであり、ワークステーション等のハードウェアが安くなればすべてのプロジェクトに適用できるでしょう、しかし、今のやり方はあまりにもスペシフィックではないだろうか、他のアプリケーション・ドメインのためには簡単には使えないように思える。

もっとジェネラルにすることは考えなかったのか？ というものでした。真に、適切な御意見でありまして、返答に窮したのですが、ジェネラルにしたいと思っている、しかし、そのためにはツールをメタ化しなければならないと思っており、そのうまいやり方が見つかっていないのです、と答えておきました。(苦しい!)

これは単に私個人の印象にしかすぎませんが、この質問や、発表内容にも現れているように、中国サイドの方の興味には論理指向的な傾向があるように見受けられました。たまたま今回の参加者の方が偏っていたのかも知れませんが、ハードがかなり乏しいようなので、論理の方に走る傾向があるのかもしれない。それに較べれば日本サイドは適当にバランスがとれていたように思われます。この点が私達日本サイドの、唯一の良い点だったのかも知れないとこのところ思い始めています。

ともあれ、英語、中国語、日本語いり交じっての討論が続いたわけです。幸いなことに、復旦大学の張先生が日本語が大変上手な方でして、討論に際しては日本語でも何をいつているのか良くわからないようなことまでうまく通訳していただきまして、議論の糸口さえ見つけ出せばコミュニケーションをとることは十分にできました。また唐先生は英語が大変お上手で、中国の方とも英語で議論するよう努力され、通訳の張先生の労力削減に貢献されていたと思います。

ただ残念なことは、討論の進行についてあまり深く考えずに、まずは発表と簡単な質疑応答を繰り返し、最後に討論をしたことでした。質疑応答のときに結構突っ込んだ話になっていたにもかかわらず、時間の都合で打ちきり次の発表に移っていったのですが、では討論という段になって何やら少し熱が醒めてしまい今一つ盛り上がりに欠けてしまったように思います。

そのせいか、夜の討論の後半はFXISの藤野さんの乱入を許してしまい、ソフトウェアのヴィジュアルイゼーションの話でグループ討論を乗っ取られてしまいました。それはそれでいうか、おかげでいうか、最後は結構盛り上がり、他のグループの中国の方まで参加され、結構盛り上がり終りましたので良かったのですが、次回はこう言った点も少し考慮した進行をしても良いのではないかと思います。

グループ討論を通して、印象に残ったのは、中国サイドのアメリカナイズされた派と純正?中国派の大きな違いでした。例えば、唐先生はCMUで慶応の斎藤先生と同じ頃に研究をされていたという方です。結構なお歳とお見受けしたのですが、非常に若々しく、御自分の意見をずばずば言われます。まず相手の意見やアイデアを認め、その良い点を褒めておいてから、しかしながらと反論を始めるあたりはまるでアメリカ人です。また行動も大胆で、用事があれば回りのことなど気にもせず一言挨拶されて、さっさと討論から抜け出してしまいます。これに嫌味が全くありません。さすがと言った感じです。

これに対して純中国研究派の方は東洋的遠慮もあるのでしょうか、質問は良く出るのですが、意見があまり出ず、討論にまではいたらない場合がほとんどでした。もう少し場なれして来れば、ガンガン意見の出てきそうな感じの若い研究者が数名いただけに残念でしたが、徐々に変わっていくものと思われれます。

ただ、両派に共通していた点は、熱意のすごさと、英語のうまさでした。論文はきちんと書かれていましたし、発表の準備にも相当の時間をかけた事が伺われ、ブツケ本番などと軽口をたたいていた自分が恥ずかしくなっていました。

この両派にはさまれて、純日本派も頑張ったわけですが、実践派ばかりがグループにかたまっていて、理論寄りの中国側の発表にいま一つ食い込んだ質問/意見を出せなかったのが残念でした。次回のグループわけの時に、考慮していただければ幸いです。

第2回中日シンポジウム参加記

藤野晃延

富士ゼロックス情報システム

今回が初めての中国であったが、ソフトウェア事情のことは勿論、その現在の生活習慣、および文化程度についても殆ど予備知識を持たずにでかけた。因らざるカルチャーショックを受ける態勢は整っていた訳である。

まず中日シンポジウムそのものに就いて。これは私の参加したグループのテーマがモデル化ということだったので、ある程度期待をしていたのだが、一緒になった中国側の研究者の方々は非常にbasicな、且つtheoreticalなテーマを対象としていたこともあって、余り議論とならず一寸肩透かしを喰ったような感じか。却って日本側からの片山先生の話（ソフトウェア・プロセスの記述）が記憶に残っている。その外のグループにも「出張(?)」して聴いていたが、熊谷さんのグループ（SDA vs SPA?）と復旦大学の張先生のグループの討論が中々興味深かった。あと特別講演(?)では野村さんのソフトウェア開発環境に関するサーベイが非常に良かった。これはSEAの会員アンケートをもとに分析したものだが、その変化振りを2年前の同様のものと比較して解説されたので、分かり易かった。

大学では、復旦大学、武漢大学、それに北京航空大学を技術訪問した。残念ながら余り感激するような研究内容ではなかった(?)ように思う。1つ、武漢大学のPOPE (Prototype of Prototyping Environment) の設計指針というかその技術動向と実現可能性をバランス良くmixしているのには一寸驚いた。後で訊くと何でもGEからの客員教授が1年程指導したものだとか、道理でオブジェクト指向を手慣れた感じで適用している訳だ。言葉では表現が難しいが、ここら辺の最新動向のフォロー、又それをテスト・ケースとして何処までシステムに反映させるか、これが研究部門のディレクタの腕の見せ所だろう。POPEはその意味で中々スマートにまとめ上げられていると思う。勿論、問題もあるが。

概して中国の大学の研究状況は今一つ、といった処か。先進のハードウェア/ソフトウェアが入手し難しいこともあるのだろうが、何か指導する側に「センス」が欠けているように思われる。学生や院生は非常にクレバーだし、且つソフトウェア開発能力もかなりのものと思われるので、もっと良い研究(方向付け)を与えてやればかなりの成果が期待できるのではないだろうか。一寸勿体ないなあ、というのが素直な感想か。

訪問した中ではAcademia Senica のソフトウェア研究所が、その内容、および設備(ハードウェア)についても抜きんでいた。特にProf.Tang率いるXYZプロジェクトは非常に洗練されていると感じた。XYZはテンポラル・ロジックに基づくフォーマルな言語(XYZ/E)を用意し、それをシステム中の中間表現(中間言語)とすることで、インフォーマルな言語、あるいは図で入力されたソフトウェア(プログラム)をその中間言語に変換して保持する。つまりインフォーマルな言語で書かれたプログラムをオペレーショナル・セマンティクスに基づくフォーマルな言語に変換し、プログラムの意味的な正しさを保証しようとするものと考えられる。これは同時に「実行可能なformalismを記述する実行可能なformalism」となっていて、つまりメタ・システムとして実現されている。そのために yaccより強力なcompiler-compilerが、attribute-grammarベースで実装されていた。 yaccとの大きな違いはsyntax-semanticsとも、その記述のformalismを自分自身で定義できる(つまりメタ・システム)点であろう。これらにより利用者は自分自身の「気に入った言語」をTPOに合わせて使い分けることが可能となる。この言語はデータ・フロー図であってもよいし従来の言語でも良い。所謂「conceptual programming」が可能となる枠組みが実現されている訳であり、このアプローチは非常に正解だと思ふ。こう書くと何か非常にtheoreticalな実用的で無いシステムのように思われるかもしれないので、誤解

のないように補足すると、実際にデモを見せてもらった言語トラスレータに、

DFDに基づくチャート・エディタ <--> XYZ/E

が可換となっていたし、まだimplementationの最中ではあったが、

他の超高級言語 (VHLLや4GL) <--> XYZ/E <--> 他の言語

も同様に可換可能となる。このことは一度、或る言語で実現された機能は他の言語/環境にそっくり移行できることを示している。即ちソフトウェアの再利用を支援することも意識されており、またフォーマルな言語 (XYZ/E) に変換されることでプログラムの意味的な正しさがチェックされ、所謂プログラムのテスト、デバッグという作業がなくなる (但し意味している仕様の確認の為のテストは無くならない) ことも大きなメリットだろう。

全体的に、先ずオブジェクト指向に就いて非常に関心が高いこと、併しまだそれを「抽象度の高いレベルで適用する」という処まで思い至っていないこと、それとソフトウェア開発の詳細設計プログラミング辺りの支援を主な対象としていてそのため開発プロセスそれ自体を支援するというような概念は未だ殆ど認識されていない (日本でも似たようなものか?) こと、等が特徴的かも知れない。しかし私にとっては何処の訪問も、中国の方々の熱烈で心暖まる歓迎に迎えられ、思い出深いもの許りである。あの中国の方々のホスピタリティ無くしては到底、今回の強行スケジュールを全うすることは出来なかったのではないかと、今更ながら思わずにはいられない。次回は日本にお招きする番だが、心からの誠意をもってお迎えしたい、そう願って止まない。

日中ソフトウェアシンポジウム 参加レポート

片山 禎昭

日本システム

1. シンポジウムでのグループ討論について

(1) 所属グループ :

Assesment/Q.C. of Design

(2) 発表内容 :

"Application of Quality Function Development to SW Development" という題で品質機能展開およびこれに代表される日本的品質管理思想の紹介を行った。

(3) 反応 (特に中国側) :

中国側メンバーは発表者である大学院クラスの若手と、行政・実務担当という2層構成であった。若手グループは、このようなテーマの意義を全く理解できなかったと推測される。(組織を管理する立場にならないとこのテーマの意義がわからないのは日本でも同様)

後者の実務担当グループも、全体として問題を理解したかどうか不明だが、中国軟件中心の鍾錫昌氏は、日本的 QC の思想のうち、「ソフトは技術者だけでなく、企業・組織全体で作る」というフィロソフィに対して、中国ではこれがどうもうまくいかない、という感想を述べていた。

(4) グループ討論全体についてのコメント :

上述の鍾錫昌氏が提起した「(日本では)ソフトウェアの完成を何をもって判定するか」という問題の討論が、日中のソフトギャップを象徴すると考える。

彼等は現在タテマエとして「100% テスト完了がソフトの完成」を強調しなければならない立場にあるようだ。(ドラフト段階の国家標準"軟件開発規範"にソフトの"完成"に関することがあって話題になっているらしい。)

一般に品質管理進化論の中世代前期には必ず完全検査主義が出現するが、中国のソフト品質管理はまさにこの段階に入ろうとしているような気がする。

すべての条件組み合わせで、すべてのレベルのカバレッジ 100% でテストすることが、事実上不可能ということの認識が一つの動機となってプロセス・プログラミングとか品質機能展開とかの考え方が出て来るのだが、中国が「産業としてのソフトウェア開発」について、もっともっと多くの経験を積み重ねないと話が噛み合わないように思われる。

2. シンポジウム全体の感想

(1) はじめて日中交流に参加したという立場では、非常に多くの情報が得られたし、中国のソフト屋さんとの具体的チャンネルができたという点で小生自身にとっては大変有意義であった。

(2) シンポジウムのメンバー構成と交流内容の問題点

日本側メンバー : SEA という独特のカラーを持ち特定分野のソフトウェア先進技術にハイレベルの識見・興味を持つグループが中心 (+a)

中国側メンバー : 1. 大学でソフト科学そのものを追求するのが仕事のグループ (発表論文は若手が中心)

2. ソフトウェアビジネスをこれから始めようとするグループ友好・交流とい

う大目的は大成功だったが、問題点としては、

1. 中国側が得た情報は日本のソフト技術の "A conspicuous peak of many peaks" なのだが、彼等はこのことを認識しているのだろうか？中国側の "peaks" の状況についても同様な疑問がある。
2. 日本側の発表は、日本人が聞くとそれぞれ非常に中味の濃いものであったが、バックグラウンドのちがう中国側の理解 or 興味はどの程度だったのだろうか。
3. 若手の発表は（所属グループのテーマがマネージメント寄りだった為かも知れないが）レベルに若干の問題があった。（中国で大先生たちのやっている研究のレベルはもっと高いのでは？）
4. 楊先生は別として、中国側のプレゼンテーション技術は問題。OHP シートを節約するためかも知れないが、論文自体のコピーを OHP に使うケースが多数あった。
（9月に天津で開催された、AHP 学会も同様に困ったとのこと）

3. 技術問題の感想

- (1) 中国全体の産業あるいは社会サービスは、未だ大量のソフトウェアを必要とする段階まで発展していない。このためソフトウェア産業が未成熟であり、これがまた唐 老大先生のいうように、日本と比べた場合の中国ソフトウェア技術の根本的弱点となっているように思われる。
- (2) 「立上がりの時期」に短期的採算を度外視して、大きな先行投資ができるのは社会主義計画経済の大きな長所であり、中国全体として軟件技術開発のための、このような先行投資が大学、研究所を中心に（？）行われている最中なのだろうか？各大学での軟件研究は（日本の大学でも同じであるが）、我々がメシを食っている火事場・修羅場のソフト開発とは、かなり雰囲気がちがっている。
- (3) 中国流ベレストロイカで大学にも企業活動をさせるようになってきているようで（武漢大学）、これは一つの正解かも知れない。
- (4) 上海電子計算機庁の製造ラインを見ると、社会主義計画（？）経済の弱点について考えさせられる。日本流の計算では毎日発生しているはずの「赤字」は他人事ながら気になる。同類のロスがたくさん集まって現在中国が悩んでいるインフレの一因となっていると考えればよいのか？
- (5) 同じく上海電子計算機庁で「質量（品質）第一、用戶（ユーザー）至上」という貼紙を見たが、紙背の意味や事情について、いろいろ考えさせられて興味深い。
- (6) ローティーンの大学生は一つのカルチャ・ショック。（上海交通大学）
- (7) 漢字入力の研究は”さすが！”という感じ。（上海交通大学）
- (8) いろいろケチをつけたが、ベクトルが揃って動き出せば本当に怖そうな国である。

第2回日中ソフトウェアシンポジウム

梶山 雄三

日立製作所

1. シンポジウムでのグループ討論について

グループ1（設計プロセスの分析とモデル）で、設計の上流工程である外部仕様決定段階での作業についての分析を発表した。個別の顧客対応のアプリケーション・プログラムを建設しているが、これまで実際に作業を進めるなかで、まとめてきた、サブ・プロセス及び各サブ・プロセスでの成果物の内容を中心として説明。そのため、経験に基づいて得られたものであり、理論的なものではない。

日本側の片山先生、落水先生の発表は共に設計プロセスを理論的に解明しようというものであり、観点は異なるが、同じ土俵にのれたと考える。中国側は、発表の内容も方向が遠い。この件については議論することは、できなかったのは残念。理論と実践という二つの側面から議論を進めていくことは、興味あると感じた。

発表・討論の印象としては、理論的に興味がありそうに思える、思考のプロセスのつながり、例えば、前のプロセスにどういう条件の時に戻っていくかといったことは、実践面では、大きな関心はなく、どこにでも戻っていくということになる。この様な差があると感じた。しかし、最終目的である設計作業の分析ということは同じであり、今後共、関心を持ち続けていきたい。

2. シンポジウム全体の感想

シンポジウム全体の雰囲気はグループに分かれた為良くつかめないが、全体会議等から判断する限り中国側のとり組み姿勢は非常に熱心だなと感じた。

ツールの開発、作業環境を統一的にWS上で実現しようというテーマが主のようである。内容的には、製造工程のツール群であるエディタ、PAD・ソース変換、テスター（アニメータ）機能をWSにのせようというものもあるが既に実現されているものを情報・成果が入らないため独自で遅れ馳せながらやっているものもあるのではと感じた。また、理論から入っており、実務からみた研究が少ない様にも感じた。そのため、受け入れた情報をベースにした研究になっているようだ。部品の再利用ドキュメンテーション技法といったことが話題になっている。これらは、これまでなかなかうまくいっていない。それは、これらを再利用する時に、本当に自分の考えるものと合致しているかを確かめるのが面倒だということが原因の一つ。

「成果物」を流用するのではなく、その「成果物を作り出す過程」を流用するという考えに移ってきているという話を伺った。これは、相当説得力のあるものになるのではという気がしている。

3. 技術訪問の感想

各地を訪問してのコンピュータ、ソフトウェアについての印象は、当然のことだが、まだ相当遅れているという感じ、ハードウェアについては、p1c等の小型は独自で開発する力はあるようだが、その部品については信頼性の高いものを大量に作り出すことは出来ないようだ。まだ、大型機については、研究段階では作れても、ライン・ベースで生産することはできないと思われる。このことは、技術の問題は当然あるとしても、それ以前に、中国でコンピュータのニーズがまだまだ成熟していないということが原因の様である。近代化を進め、それを追っかけてコンピュータが必要になってくる。今は近代化の途上である。

こういった状況の為、ソフトウェアについても研究レベルでは色々と推進されているがアプリケーション分野がない。ネットワークも未発達であり、大規模なオンライン・システムも数えあげる位と想像される。しかし、これからは、外国との合併会社設立も盛んになってきており、徐々にニーズが芽生えている。全て、基礎から手作りで、物事を考えているため、ハード・ソフト一体となったコンピュー利用技術の開発という面では将来画期的なものを生み出すのではないかという期待がもてる。

第2回 日中ソフトウェアシンポジウム レポート

水谷 時雄

日本システムサイエンス株式会社

1. グループ討論

(1) 自分の発表 "ソフトウェア部品化によるソフトウェア・ドキュメントの再利用"

ソフトウェアの部品化は単に再利用だけを目的としたものではない。部品化の目標の1つには、抽象化設計がある。すなわち、部品化の中に入った内容は、隠蔽され、それをを用いる人から離れてしまう、それが抽象化である。部品を用いる度合が高くなるにつれ、また、上位へ部品の構造化が進むにつれ、抽象化が高められる。

(2) 復旦大学張然研究室（陳万軍・葉雲文研究員）も "REAP: Reusable Executable Specification Based Automatic Programming System" が発表され、同じくソフトウェアの部品化による再利用がプログラムの自由化の1つとして発表された。

(3) 中国の人々への印象は、理論的枠組はしっかりしており勉強をされている。設備にはお金がかけられない事情がそうさせているかもしれない。充分、日本の人々との話のレベルは一致している。実践的な活動が少ない中で、復旦大学張然研究室では、実現化されていると思われる。

- ・ REAP (ソフト再利用による自動プログラミング)
- ・ GAUSS (汎用図形作成システム)
- ・ C-SLICING (構文エディタ・ディバッカー for C 言語)
- ・ DOC-ANALYSIS (文章メンテナンス波及リレーションシステム)
- ・ DFD
- ・ HCP

2. シンポジウムの全体的感想

(1) 交流会（パーティ・レセプション、私設交流会、バス旅行）を通じて次の印象をもった。

- ・ 研究者の方々は、ほとんど日本に来たいようです。
- ・ 大学の研究室でさえ、日本からのプロジェクト（有料）を期待しています。

(2) 街の中のとあるバーで、SUZUKI-250cc ライダの中国商工銀行のプログラマに出合った。

IBM4381 をアセンブラで
日立のM-240 L-320 を RPG で

3. 日中合作啓明軟件公司（ビーナス・ソフトウェア）の設立

(1) 日本システムサイエンス（株）他5社と中国の華東計算技術研究所との資本金総額1億円の合併会社設立にあたり、中国の全体的レベルを知りたかった。これが今回の目的の1つでもあった。情報系よりエンジニアリング系の方が技術的に進んでいるように思われた。

第2回日中ソフトウェア・シンポジウムの感想

久保 宏

富士通

シンポジウムでのグループ討論について

Assessment/Quality Control of Design のグループに属して、武漢大学の何先生と二人でチェアマンをつとめた。提出したポジションペーパーを発表するよう何先生から勧めを受けたが辞退した。発表しても議論はしてもらえなかったであろう。元々議論してもらえんと思って書いていない。中国に行くこととは無関係に書いた。ちょっと考えてみたいことがあって、ポジションペーパー提出の機会を利用して考えてみたものである。

チェアマンの努めを果たせたかどうかは分からない。どんな展開になるかは成り行きに任せることにして討論にはいった。日本人の分も含めて、提出されているペーパーを事前に読んでおくこともせずに臨んだ。どう進めるかは、六人の人の発表を聴きながら考えることにして発表にはいった。聴いている内に、たくさんの討論テーマを抽出することができたので、まあ何とかなるであろうという気分にはなってきた。

メンバーの協力を得て討論テーマの候補を列举し、中国側の人たちにその中から選択してもらうことにした。中国の人たちの研究に少しでも役立ちたいという気持ちが働いて彼等の選択に委ねた。彼等の希望は、岸田さんの発表に関係するテーマに集中してしまった。討論テーマを提起した中国人発表者が、別のグループに移ったのか討論時間にはいなくなって、選ばれたテーマを棚上げせざるをえなくなることもあった。選択されたテーマだけでは時間を使いきれず余った時間を、中国側の人のテーマの討論に提供すべく、討論テーマを追加してもらうこともした。

そんなことで、与えられた時間を使いきることはできた。その間、会話が途切れることはなかった。しかし、討論と言える状態を作り出せたとはいえない。グループの名称にふさわしく、討論の内容を、設計プロセスや設計プロダクトの品質評価や品質管理に向けようと努めたが、そこに誘導することに成功したとはいえない。

討論テーマを中国側に選択させたのは失敗であったかもしれない。そうではなく我々日本人が、中国側の発表に矢継ぎ早に質問を発して、中国側の人を無理矢理討論に引きずりだすような作戦をとった方がよかったかもしれない。

それにしても中国側の人たちの反応は奇妙である。岸田さんの発表は、プロダクト指向からプロセス指向へのパラダイムシフトを提起するものであった。古いパラダイムに矛盾を感じている人たちがこの発表に関心を寄せるのは分かる。しかし、矛盾に呻吟するほどに、古いパラダイムに苦しんだ体験があるようには思えない。彼等が、パラダイムシフトに関心を寄せるのは、奇妙である。彼等は、外国に於ける最新の研究を追いかけているだけなのかもしれない。研究が大地にしっかりと足をつけたものになっていないのかもしれない。足をつけようにも大地がないことも物語るかもしれない。このような研究態度から、優れた成果がでてくることは余り期待できない。必要は発明の母という。必要のないところに発明はない、と断ずるのは過酷に過ぎるか。

別の見方があるかもしれない。上の評論は、技術の発達についての歴史観に基づいていない。日本の現在に立脚して、日本に当てはまる物差しで中国を見ている。二つの国の産業技術は異なる発展段階にある。技術のある発展段階では、何もかもコピーすることに徹するのが正しいと、歴史を教えているのかもしれない。だとすると、その歴史の教訓に従って彼等は、正しい思考と行動を主体的に選択していることになる。

正直のところ彼等をどう見るべきかは、よく分からない。彼等とどう付き合うのが正しいかも分からない。技術史の見方だけでは十分でなく、政治体制・経済体制に眼を配る見方をしないと彼等の思考と行動は理解できないかもしれない。

シンポジウム全体の感想

中国の研究者や技術者にとって、日本との合同シンポジウムはどのような意味をもっているのか。今回のような、かなり先端的なテーマがなぜ彼等の関心と呼ぶのか。討論に即して綴った感想の前にこの疑問がある。テーマが先端的だから関心を寄せるのか。テーマに関係なく日本人と顔をつないで、日本に来る機会をつくることや、中国を脱出して日本に出る機会をうかがうことが彼等の主要な関心なのか。

技術訪問の感想

初めて中国に行った一人の日本人があれこれ悩んでいるのを嘲笑うかの如く、アメリカは中国の中に着実に地歩を固めている。日本人として日本の影の薄さを憂慮する。

オブジェクト指向を越えるもの

Midnight-Writer 大木幹雄

深夜、納品の報告書原稿を「一太郎」で網渡りで打っていると（ちかごろ、このスリルがたまらない快感になってきている）、息抜きに馬鹿なことを無性に書きたくなる。

特に最近ひどい。無能管理者の分類といった記事をずいぶん昔にSEAMAILで読んだ記憶があるが、無能社員とはいわないまでも、身勝手社員に悩まされた日は特にそうである（「自分も昔はそうだった」などという、自分も歳をとったものだと貴重な時間を浪費しなければならないので、中年は中年で勝手な見方をする。ところで、たまには「こんな社員はいらない」特集をやってほしいものである）。書いている本人がこんな状態なので、まずは眉にビッチョリ、ツバをつけてから読んでいただきたい。

さて、話は本題に入って、「オブジェクト指向ってなんだろう」と自問自答しながら、いろいろな本を読んでいると、ときには参考になるというか、イメージを膨らせてくれる本に出会う。最近、「超弦理論」なるものの解説本を読んで、自分なりにイメージを膨らませて楽しんでいる（もちろん、電車の中やトイレの中の話だが）。

電子や陽子がクォークから構成されているという話を聞かれた方は多いと思うが、それらを結びつける場の粒子が、数学的な無限少の点ではなく、弦から構成されていると考えた方が現実に合うといった理論である。弦は、閉じた弦（輪）と紐状の弦があると書いてある。多少詳しくいうと、いろいろな力（重力、電磁力、強い力、弱い力）を伝える媒体の粒子は、超弦の固有振動（すなわち、振動波）に相当するといっている。もちろん、もっと難しい概念を含んでいて、なかなか分かりづらいが、何故、深夜ボケた頭でこんな事を書いているかという、その考え方の斬新性とソフトウェアとの類似がおもしろいからである。

元来SF小説は嫌いだ、自分自身としては、「素粒子の世界の法則は人間世界、そして人間が作り出すソフトウェアの世界まで影響を与えているし、類似の構造をもっている」と、SF的な発想を信じて疑わない人間である。

多分、いくらか科学をかじった人は、極少の世界と人間世界と関連があるはずはないと思うだろうが、宇宙の法則は人間が考えたわけではないので、勝手にそう思っている。アーサー・ケストラーの「ホロン革命」に影響されたわけではないが、昔から信じていた前述の考えの根拠のようなものが「超弦理論」に見いだされるのが、なにしろおもしろい。

昔々、電子の世界では、波と粒子が同じものを裏表から見ただけで、実は同じものだと言ったとき、不思議な疑問と感動を感じた。それを再び感じさせてくれるからである（当然のことながら、オブジェクトを無限少の点に対応させて考えているから、このような感動を味わうわけであるが）。

粒子と波の2面性の関係は、電子から分子へ、遺伝子へ、人体の構成、人間社会の構成、そして人間の考え方産物・ソフトウェアへと、いたるところで類似を見せてくれる。

たとえば、人体の構成で頭脳をとれば、ニューロンとホルモンといったように、脳波を発生させる神経ループと、精神興奮剤に含まれる高分子なども、「頭脳」が機能するために必要な2面性の現れではなかろうか（念のため、これは私の勝手な思い込みです）。

ソフトウェアで粒子と波の2面性のどこに類似性があるのだと、当然いわれると思うが、「オブジェクトとメッセージです」などと、わけの分からぬこというつもりはない（ホントはそういいたいところだが）。

「データと制御」が、それだといいたい。さきの理論では、弦は輪と紐状があり、それがクォークを結び付ける力を伝える場の粒子の形状だといっているが、これを超拡大解釈すると、ソフトウェアに対応づけられる部分が出てくるように思えてくる。

たとえば、すべてのプログラムの制御構造はループとシーケンスと分岐で書けると、ある有名な先生が昔おっしゃったが、それはデータというものに作用を与えている（平たくいえば、影響を与えている）。これは、輪と紐が点粒子を結び付けているのどこか似てないだろうか（分岐も弦が相互作用して形状を変えるのに似ているが、詳しくは本を読んで下さい）。読者があまりショックを受けないように、ここで再度、今述べてることは「アリスの不思議な世界」のように、すべて白昼夢ならぬ、深夜夢であることを断わっておきます。

ところで、素粒子の世界では、粒子は他の粒子に変化したり、真空から生成されたりするが、これも、オブジェクト指向のメタクラスの考え方の一部に類似しているように感じられる。

毎度のことながら、つい、考え方が発散してしまうが、何かのメタ・オブジェクトがあって、その固有振動が特有のクラスという考え方や、オブジェクトが波としても振舞う、いい換えると、勝手なオブジェクトが調和音といったオブジェクトに分解されるといった考え方があってもいいのではなかろうか。

すべてのオブジェクトか特定の群（群論の群です）にしたがって、勝手に他のオブジェクトに変換する関係がオブジェクト間の関連設計と考えられないだろうか。群の変換の合成は、関連の多重化とは考えられないだろうか。

さらに馬鹿げたことをいえば、勝手なオブジェクトは、フーリエ級数みたいに展開することができ、その係数を決めることが、ソフトウェア設計者の仕事であるというように、理論は展開できないだろうか。

また、波のようにオブジェクト同士の干渉（あるオブジェクトを他のオブジェクトと合成すると消えてしまう）や、ある媒体（スコープやクロージャにでも相当しようか）から違う媒体に入ると屈折する（要するに性質が変化する）とか、広がりを持っている（スコープもオブジェクトに相当すること）とかがあってもいいのではないだろうか。

学生時代に情報工学を学んだ人で、ここら辺をうまく整理できる人はいないか。立派なソフトウェア研究所を持っているような会社には、優秀な人材ばかりがいらっちゃって、こんな馬鹿げたアイデアにいちいちに付き合ってはおられないだろうが、もしお暇だったら、ちょっと耳を傾けてください。

だんだん、自己陶醉といいたい放題の世界に入っているが、たまには、金稼ぎの苦悩と納期を忘れて、こんなオオザッパな考え方で頭を休めるのも悪くはない。非常に手っとり早い自分好みの方法でストレス発散をしてみました。

こんなことを書けるのは、SEAMAILだからでしょうね。学会誌だと、研究の近況とか研究メモが出ているが、変なアイデア・コーナーにでも載せて頂ければ至極の喜びと致すところです。

さあて、また現実に戻って報告書を書かなくては。では……。

（日本電子計算）

システム作りの自動化支援について

小林 俊明

ソフトウェア技術（株）

はじめに

SEAMAIL の編集長から「なんでもいいから原稿を書いてくれ」とせがまれましたので、私が日頃考えていることを書いてみました。

私は日頃なにを考えているかと言うと、いかに楽に仕事をするか…！ いかに余分な労力を使わずにプログラムを作るか…！ そんな怠慢なことを考えています。この私の夢を実現するには、どうしても高度に自動化されたラクチン開発環境と優秀なスタッフが必要です。そこで、今回はこの私の夢を実現するために必要なラクチン開発環境はどんなもので、どんなスタッフが求められているか、勝手気ままに書かせてもらいました。

1.どんなスタッフが求められているか！

知識、技能、能力という言葉の定義はなしにして、私が考えるにコンピュータ・システムを開発をするスタッフには、以下の能力が必要と思われる。

アプリケーション知識	システムの構築&保守管理能力
抽象化能力（モデル化能力）	システムの検証能力
システム・プランニング能力	プロジェクト運営管理能力
プログラミング能力	客先との交渉能力
インプリメンテーション技術	などなど……。

(1) アプリケーション知識とは？

言わずと知れた対象業務知識のこと。

(2) 抽象化能力とは？

これは私の造語ですが、だいたい次のような能力だと思って下さい。

- ・現状システムをモデル化して捕らえる。
- ・現状システム・モデルの問題点を抽出する。
- ・現状システム・モデルをモディファイして新しいシステム・モデルを創造する。

(3) システム・プランニング能力とは？

これも私の造語ですが、だいたい次のような能力です。

与えられたシステム・モデルを現在入手できるリソースで、最も効率よくかつコスト・パフォーマンスの高い形で具体化する。そんなシステム構成を設計する能力だ。これを実現するには常に最新のリソース情報を入手整理する事が必要だし、なによりも、どういうリソースをどう配置し、システムを構成すればよいかを設計する能力が欠かせない。

注：ここで言うリソースとは現在入手可能なコンピュータ資源全体のこと。

(4) プログラミング能力とは？

ここで言うプログラミング能力とは、ソフトウェアの基本設計から、プログラムのコーディングをするまでに求められるソフトウェア・エンジニアリングに関する知識と定義して置こう。

(5) インプリメンテーション技術とは？

どんなプログラムを作るにしても、システム資源のハンドリング方式がわからなければ、動くプログラムは

作れない。またシステム資源の使い方を誤ると、効率の悪いシステムになったり、デッドロックを起こしたり、システム完成後にいろいろ不都合を生じる場合がある。

注：ここで言うシステム資源とは、ターゲット環境上で使用する周辺装置や、システムサービス（メモリ管理機能やタスク管理機能など）、もっとブレークダウンすれば、機械語命令セットに至るまでのシステム諸資源を意味する。

(6) システムの構築&保守管理能力とは？

プログラム単位の世界だけで言うなら、makeとscsをどう使いこなすかだが、そんな狭い世界だけで考えるわけにもいかないだろう。なにしろシステムに幾つコンピュータが存在し、どんな関係でお互いに動作しているかなかなか難しい問題が存在する。1つの機能修正が全体にどんな影響を与えるか、そこまで考えたシステムの構築&保守管理能力が求められている。

(7) システムの検証能力とは？

早い話がデバッグ能力だ。なにかトラブルがあって手掛けたシステムが犯人扱いされた時に、いかに手掛けたシステムの正当性を証明するか。また誤りがあればいかにして誤りを検出し直すかが問題だ。

2. ラクチン開発環境にはどんな機能が必要か！

(1) アプリケーション知識の共有

コンピュータ・ソフトウェアの多くは、類似システムの開発だと言われています。類似システムの定義は明確ではありませんが、私の過去の経験からすると既存システムをリプレースする場合などは、該当すると思われれます。

例えば、今までI社のコンピュータで処理してきた業務をN社のコンピュータにリプレースする場合を想定して下さい。アプリケーション部分の業務知識はI社用のシステムを作った時に蓄積されたノウハウがかなり役に立つはずですが、しかし現実はなかなかうまく行きません。たとえ前のシステムを開発したソフトハウスに発注したとしても期待するほどの効果は上がらないでしょう。

そこで私の考えるラクチン開発環境では、この種の技術移転を支援する為に、実行環境に依存しないアプリケーションレベルの業務知識をアプリケーション知識ベースに保持し、過去の業務知識を検索&参照しながら新システムの業務モデルの定義が行える様にし、かつ設計した業務モデルの実行シュミレーションも行えるようにしたいと思っています。

きっとこれはオブジェクト指向でいうインスタンス生成みたいなものだろう。でも生成する過程で人間の介入が必要だね！

(2) 抽象化設計の支援

新規システムの開発や、アプリケーション知識ベースにインプットされていないシステムを開発する場合は、頼るべき過去の業務知識がありませんから1から勉強する必要があります。このような場合でもシステム化対象業務の分析や現状モデルの把握、新システム・モデルの定義を支援するAI的業務分析&新業務モデルの定義と実行シュミレーションも行える環境を提供したいと思います。

これがきっとアプリケーション知識ベースへのインプット作業になるのでしょう。オブジェクト指向で言うならクラスの定義かな？

(3) システム・プランニングの支援

このレベルになると実現を求められるシステムモデルが明示されているのだから、あとは現在入手が可能なコンピュータ・リソースをどう配置すれば、効率的でかつコスト・パフォーマンスの高いシステムが編成できるかを機械的に支援してあげればよい。その為にはリソースに関する知識ベースを用意し、各リソースのインターフェイス仕様をチェックしながら接続の可否判断を支援し、リソースの性能データを元に設計したシステムの性能情報も出力してくれる。おまけに各リソースの発注先情報を元にシステム価格の見積まで出してくれ

る様にしたい。

(4) プログラムの論理設計支援

ここで言うプログラムとは先にも述べた様に作成するソフトウェア・システムの全体を指す。私の考によると、実行可能プログラムを作るには2つの知識（技術）がある。1つはプログラムの論理構造を設計する能力。もう1つは設計したプログラムの論理構造を実行環境上にインプリメントする能力だ。この二つは明らかに別物だが切り放して考えることはできない。と言うのが私の考えだ。

実行環境上のOSが提供するシステム・サービスにはどんな機能があり、周辺機器をどうハンドリングすれば目的が達成できるかを知らずしてプログラムを作ることはできない。たとえばウィンドウシステムを利用したプログラムを作るとき、ウィンドウシステムがどんな機能を提供し、その機能を利用するにはどんな約束ごとを守らなければならないかを知らずしてプログラムの論理構造だけを設計する事は至難の技だ。

そこでプログラムの論理設計を支援する環境は、次の3ステップに分割して考える必要があるだろう。

① 実行環境が提供するシステム・サービスの調査支援

実行環境別にシステム・コールや標準ライブラリ、オリジナルのプログラム部品ライブラリ、流通ソフトウェアパッケージなどの技術情報をプログラミング知識ベースに格納し、利用したいサービスをブラウジングしながら調査できる環境を提供する。

② 機能のブレークダウン設計支援

実行環境が提供する機能を調査しながら、各機能モジュールをどこまで詳細にブレークダウンするか判断し、各モジュールの機能定義、インターフェイス定義、及び各機能モジュール間でのハンドリング関係の設計支援をソフトウェア工学に基づいて行う。

③ プログラムロジックの設計支援

各機能モジュールのローカルなロジックを定義するのだが、先に定義した各機能モジュール間でのハンドリング関係を損なわずに、かつハンドリング・シケンスを誤らないようなロジックの設計支援を行う。

プログラムのロジック設計ができた時点で、各言語別のプログラミング知識ベースを参照しながらソースコードの自動合成を行う。

(5) インプリメンテーション支援

プログラムの論理設計フェイズで使用したシステム資源を実際に利用できるように実行環境にインプリメントするのを支援する。

(6) システムの構築&保守管理支援

プログラムの論理設計フェイズで作成したソースプログラムと、使用するシステム資源の結合情報を元にシステムの自動生成プログラム(Makefile)の合成や、プログラムの修正に伴う変更履歴の作成、関連ドキュメント、テストデータの自動生成などを支援するものとする。

(7) テストの自動化支援

プログラムの設計情報が開発環境にインプットされるとテスト仕様書やテストデータ、テストプログラムが自動生成されるのは当たり前にした。さらに開発環境上でテストできるモジュールを自動抽出し、開発環境上での単体テスト支援も行える様にしたい。

3.おわりに

以上のような勝手気ままなことを書いてしまいましたが、私たちソフトウェア・エンジニア自身が本当に必要な開発環境はどんなもので、どんな技術を磨かなければいけないか、もう一度考え直して見る必要があるなと原稿を書いている私自身が感じてしまいました。意見のある方はどんどんSEAMAIL に投稿して下さい。編集長もきっと喜びますヨ！

処理速度への挑戦

横山 博司 (SRA)

1. はじめに

多目的モニター用カラーディスプレイとして開発したシステムの処理速度（表示時間）の改善をはかることになり、その作業に携わった一人として、その結果報告をここにまとめることにします。

簡単にシステムの紹介をしましょう。

まず、CPUとしては、インテル社の80286を使用し、CRT制御用にモトローラ社のM68000を使ったマルチCPUから成っています。ホストより表示用のデータを受信しシステム本体でもっているデータベースに従って決められたフォーマットで表示します。

簡単に、機能の紹介をしますと、登録可能画面数は、約100画面、1ウィンドウ、9画面から構成される大画面、大画面時のスクロール、拡大、データ設定（ホスト・コンピュータへのデータ送信）ができます。これらの機能のうち拡大を除いては、ソフト・ウェアで対処しています。

画面更新のとき、表示中の画面をそのまま描画してしまうとオペレータは、常に描画の流れを見ていることになってしまうため、まず、裏画面に描画し、描画終了した時点で表と裏を入れ換えます。ここで言う画面更新とは、ホスト・コンピュータよりデータを受信した時点で、ホスト・データにもとづいて画面を表示することであり、ウィンドウや大画面を使用していれば、それぞれ専用の裏画面に描画したうえで、表示し

ます。なお、ホストからのデータは、フレーム分割されることなく、1回の送信で1画面分のデータがそろっています。また、複数ホストの設定も可能で、この場合は、1画面は、複数のホストにより構成されることになります。このときも、1ホストからのデータ受信毎に画面更新します。

また、描画できる対象は、数値、文字列、棒グラフ、折れ線グラフ、トレンド・グラフ、図形パターンと移動です。

ちなみにこのシステムは、プラント制御にも用いられ、コントロール中のデータをリアルタイムで表示することもあり、処理速度が要求されます。

インテルの80286上で稼働するプログラム部分が処理速度の改善対象です。一画面の更新時間を1/2にするのが目的です。これまでに、何度か、処理速度の改善は、行なわれてきましたので、プログラムに手を入れることでの対応では不可能とのみかたより、機能を落してでも早くすることが要求されました。

2. 測定作業

とにもかくにも、どこで時間がかかっているのかを押さえないことには、話になりません。調べた結果が、表2-1 実行時間測定結果 です。測定対象画面は、1ホスト・データにて成り立っています。

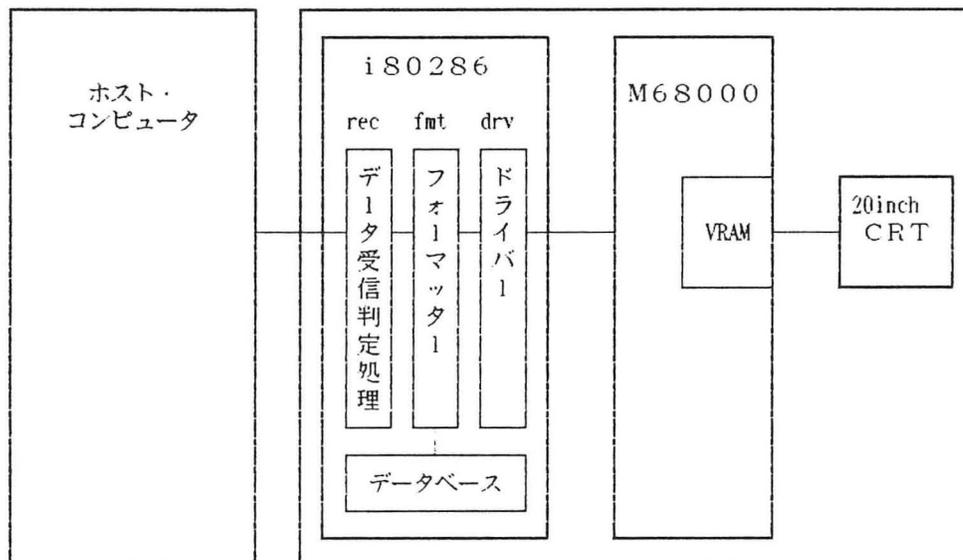


図1-1 概略構成

表2-1 実行時間測定結果

(数値表示-テキスト数: 80、文字数: 400全角)

処理	占有率%	2	4	6	8	10	12	...	100	ms
1 ホストデータ受信	-									
2 ホストデータコピー	0.33	—	1.7							
3 rec->fmtタスク切替	0.29	—	1.5							
4 マクロ判別(fmt)	0.35	—	1.8							
5 bin->asciiコード変換	2.49	-----						12.8		
6 マクロ展開	18.55	-----							95.2	
7 フォント取り込み	35.86	-----								184.0
8 マクロ展開後処理	0.27	—	1.4							
9 VRAMクリア	5.98	-----							30.7	
10 VRAMクリアタスク切替	0.18	—	0.9							
11 M68000通知	34.61	-----								177.6
12 描画終了処理	0.97	-----					5.0			
13 fmt->recタスク切替	0.10	—	0.5							
合計	99.98		513.1							

----- ; テキスト数に比例するもの
 ; 文字数に比例するもの

調べ方はプログラムにポイントを設けて、ポート出力させ、オシロスコープで出力毎の時間を測ります。なお、本測定には、ホストからのデータ受信時間は含まれていません。

測定の結果、6 マクロ展開処理、7 フォント取り込み処理、9 VRAMクリア処理、11 M68000への通知処理 だけで、全体の95.00%が占められていることが判りました。

3. 対策検討

上記の処理での時間短縮を測るためには、以下のことが考えられます。

- (1) 描画対象を減らす(処理対象を減らす)。
- (2) フォント取り込み処理をしない。
- (3) VRAMクリア処理をしない。
- (4) M68000への通知データ量を減らす。

(1)、(3)、(4) に関連していることは、現仕様では、裏画面方式を用いているため、一画面分全てを描画する必要があります。

しかし、実際には、データ全部が変更されないことが判っています。また、今回要求された画面では、数値表示や文字列表示なので、その都度VRAMをクリアしなくても、上書きすることで書き換えることができます。

しかも、描画時間も短縮されれば、オペレーターが、描画の流れをおっかけるということもないでしょう。このことを、部分描画と呼ぶことにします。

また、(2)、(4)について言えることは、あらかじめVRAMに文字を描画しておいて、VRAM上でコピーすることで対処します。そのためのVRAMのエリアは、大画面用に割り当てられたエリアを代用することとします。

このことを、コピー描画と呼ぶことにします。

以上のことから、以下の制約を設けた上で対処できることが、判りました。

- 1. ウィンドウ、大画面及びスクロール機能の削除。
- 2. 棒グラフ、折れ線グラフ、トレンド・グラフ、図形パターンと移動機能の削除。

ただし、VRAMのエリアも限界があるため、コピー描画は数値表示に限られることになりました。

4. 期待値の算出

上記の対策による期待効果を算出してみます。部分描画の処理による処理速度の改善期待値を 表4-1 に示します。ホスト・データの1/2が変化による描画対象としています。表4-2 は、コピー描画の処理による処理速度の改善期待値です。短縮時間は、合計で $-410.3 - \alpha$ ms ということとなります。

5. プログラム変更作業

ミスが入ることがないように、慎重にプログラム変

表4-1 部分描画による処理速度の改善期待値

2	ホスト・データの新旧比較時間	+25.0ms	(推定)
5	binary-asciiコード変換(1/2)	-6.4ms	
6	マクロ展開(1/2)	-47.6ms	
7	フォント取り込み(1/2)	-92.0ms	
9	VRAMクリア	-30.7ms	
10	VRAMクリアタスク切替	-0.9ms	
11	M68000通知(1/2)	-97.0ms	
		-249.6ms	

表4-2 コピー描画による処理速度の改善期待値

7	フォント取り込み => 不要	-92.0ms	((1)に加えて)
6	マクロ展開	不変+α	(コピーマクロ展開)
11	M68000通知 (97ms*(1-7w/24w))	-68.7ms	((1)に加えて)
		-160.7+α ms	

更作業を行ないました。

変更量は、システム全体の約20%にあたる約3kステップ(PL/M言語)で、バグには、ケアレスミスもなく、11件という見事なものでした。(自画自費!?)

行しないので、計測できません。そのため、改造前値も、8 マクロ展開後処理 に含めました。

(期待値との差) = (効果) - (期待値) としてあり、-値は、期待している以上に、効果があつたことを示し、+値は、その逆です。

6. 結果の測定

改造結果を測定します。

測定点や、測定対象画面、ホスト・データは同じ物を用いました。

結果は、表6-1改造後の測定結果 に示す通りです。

改造後、10 VRAMクリアタスク切替 は、実

ここでδを許容誤差と見ることができましよう。

また、マクロ展開時間(+27.6+α ms)には、表示データの変更の有無の判別処理も含まれていることから、表示データの新旧比較時間(-22.8ms)と同一視することができます。これを計算すると、+4.8+α となり、許容誤差の範中(短縮時間期待値の約1.2%)と考えることができます。

このことから、結局、M68000通知時間の +72.1ms が、期待値よりはすれた値となります。

表6-1 改造後の測定結果

(単位はすべて ms)

処理	改造後	改造前	効果	期待値	期待値との差		
1	ホストデータ受信	-	-	-	-		
2	ホストデータコピー	3.9	1.7	+ 2.2	+ 25.0	-22.8	
3	rec-fmtタスク切替	1.5	1.5	0	0	0	
4	マクロ判別(fmt)	0.9	1.8	- 0.9	0	- 0.9	δ
5	bin->asciiコード変換	4.4	12.8	- 8.4	- 6.4	- 2.0	δ
6	マクロ展開	75.2	95.2	- 20.0	- 47.6+α	+27.6+α	
7	フォント取り込み	0.0	184.0	-184.0	-184.0	0	
8	マクロ展開後処理	3.5	2.3	+ 1.2	- 0.9	+ 2.1	δ
9	VRAMクリア	0.0	30.7	- 30.7	- 30.7	0	
10	VRAMクリアタスク切替	-	-	-	-	-	
11	M68000通知	84.0	177.6	- 93.6	-165.7	+72.1	
12	描画終了処理	3.3	5.0	- 1.7	0	- 1.7	δ
13	fmt-recタスク切替	0.5	0.5	0	0	0	
合計	177.2	513.1	-335.9	-410.3+α	+74.4+α		

はずれてしまった原因は、M68000側のソフト・ウェアの負荷が増したことによるもので、まるで考えから、もれてしまっていたことによります。

このことを、除いては、ほぼ期待どおりの結果となり満足できる数値が出ました。

数値表示の場合で、ホスト・データの 1/2 が、変更されたとして、旧システムでは、513.1ms かかっていたものが、177.2ms となったわけで、実に34.5%になったわけです。

なお、全てのホスト・データが変更され再描画したとしても、約50%に短縮されるという結果が出ました。

7. おわりに

実際の作業の中にあつては、やってみなければわからないことが多く、やっつけ仕事のなりがちですが、今回のように、効果を充分に検討した上で作業できたこと、また、ほぼ期待どおりの結果がでたことは、満足できるものでした。このような機会を与えてくださった方や、一緒に作業していただいた方々には、たいへん感謝しています。

8. おまけ

1) バグの分析

コーディング以降の作業工程で、発見されたバグについて分析してみます。

表8-1 に作業工程毎のバグ件数を示します。登録されたものは11件ですが、そのうち1件は、テスト時の勘違いによるもので、もう1件は、機能仕様上、不可避の弊害であり、バグとは認められませんので、これらを差し引いた9件について掲げています。

単体テスト工程は省略しています。なぜならば、今回の作業は、システムができあがった後のメンテナンス作業であり、新たなモジュールの追加というよりも、内部の変更であることで、単体テスト用に環境を整えるのは、時間がかかりすぎて利口なやり方とは言えないからです。

2) バグの内容

バグの内容について、表8-2 に示します。表には、発見された作業工程と、そのバグが、作り込まれたのは、どの作業工程かを示しています。

さて、バグの作り込まれた工程と、発見された工程とが違うもの((1) から (8) まで) について、一つずつ検証してみます。

検証するにあたって、作り込まれた時点での発見、つまり、バグを作らないことはできたのか否かに着目してみます。

(1) については、プログラムの解析不足。

(2) については、初期時点で、画面は、オフ状態でなく、クリア状態のVRAMを表示していたわけで、実行してみないとわからない。

(3) については、数字だけでなく、スペース文字で埋めていたわけで、プログラムの解析が、不十分だった。

(4) については、(3)と同じ事と、基本設計書を熟読していれば、HEX表示が使われていることがわかった。

(5) については、CRTのマニュアルの記載が違っていたことによるので、設計の時点では、検出できなかった。

(6) については、今回追加した、比較処理の条件をしっかりとレビューしていれば、発見できた。

(7) については、設計時の話題にはあがっていたことが、反映されていなかったわけで、しっかりとレビューしていれば、発見できた。

(8) については、今回変更した、部分描画方式による変化を、しっかりとレビューしていれば、発見できた。

表8-1 作業工程毎のバグ件数

作業工程	作り込まれた件数	発見した件数
基本設計(BD)	0	-
詳細設計(ID)	7	-
プログラミング(コーディング)(CD)	2	1
単体テスト(UT)	-	-
システムテスト(ST)	0	8
合計	9	9

表8-2 作り込まれた作業工程と発見された作業工程

バグ内容	作り込まれた作業工程				発見された作業工程			
	BD	DD	CD	ST	BD	DD	CD	ST
(1) 初期画面が出ない。			○					○
(2) VRAMへの初期描画が、チラッと見える。		○						○
(3) 数値で、スペースが出ない。		○						○
(4) 数値で、HEX(A, B, C, D, F)が出ない。(※1)		○						○
(5) 数値で、1文字1ドットずつ右上がりとなる。		○						○
(6) 数値で、初期の0表示しない。		○						○
(7) 文字列、時刻が、上書きされて二重になる。		○						○
(8) 文字列、時刻が、前の表示が消えない。		○						○
(9) コンパイル・エラー。			○				○	

(※1) 'E'は、Exponent表示として登録していたので、問題とはならなかった。

次に、バグが作り込まれた工程内での検出が、可能だと判断した(1)、(3)、(4)、(6)、(7)、(8)について、バグを作り込まないのと、テストで発見するのとどちらが、時間的に、短く仕上がったかどうかを推測してみます。

(1)、(3)、(4)は、プログラムの解析不足や、設計書の読解不足によるもので、デバッグ時点で、プログラムを解析したり、設計書を読み直したわけなので、明らかに、作り込んだ工程で対処しておく方が、再テストなどのロスがなかったものと思われまます。プログラムや設計書の全てを綿密に調べることと、必要な箇所(ここでは、問題となった箇所)を重点的に調べるのとでは、後者の方が、効率的であるでしょう。つまり、プログラムや、設計書の全体にわたって概要をつかんだ上で、重点箇所を見過ごすことなく拾

い上げて、徹底的に調べあげるのが、最も効率的なやり方でしょう。

これらのバグは、製作段階で重点箇所を見過ごしてしまったことによります。

(6)、(7)、(8)は、レビュー(チェック)をしたにもかかわらず、バグが入り込んでしまったわけで、レビュー(チェック)のやり方に問題があったようです。もっと詳細について、チェック項目を洗いだし、それに基づいて行えば防げたかも知れません。しかし、その作業は、膨大なものとなりましょう。それに、その項目自体の検証が必要となり、作業が増えることでのミスがはいりこむことは、充分にありえることでしょう。このことからすると、マシンを使ったテストで検出した方が、はるかに速いと思われまます。

表8-3 作り込まれた作業工程内で対処できたと思われるものと原因

△は、できたかも知れないが、しなかった方がよかったと思われるもの

バグ内容	対処できたもの	原因	
		理解不足	レビューもれ
(1) 初期画面が出ない。	○	○	
(2) VRAMへの初期描画が、チラッと見える。	×		
(3) 数値で、スペースが出ない。	○	○	
(4) 数値で、HEX(A, B, C, D, F)が出ない。	○	○	
(5) 数値で、1文字1ドットずつ右上がりとなる。	×		
(6) 数値で、初期の0表示しない。	△		○
(7) 文字列、時刻が、上書きされて二重になる。	△		○
(8) 文字列、時刻が、前の表示が消えない。	△		○

3) まとめ

ソフトウェアの開発工程全般について言えることは、人の手が入るところには、必ず誤りが入るものなので、

- ① できるだけ、手を入れないようにする。
手を入れるとしても、機械的に、単純なアイテムに分けて一つずつ確実に。(複雑なものを、複雑なままでしない。)
- ② マシンでできるものは、できるだけマシンにさせる。
- ③ チェック(体制も含めて)は、十分に。

①については、どんな機械でも、構成されている部品の点数が少なければ少ないほど、故障が起こり難い事に似ています。

なお、プログラミング段階では、冗長すぎるプログラムは避けて、簡単明瞭で、シンプルなものとする(1モジュール数十ステップに分けて、構造的なものにするなど) ことも掲げておきます。

8. おまけのおまけ

本システムは、1986年夏に、最初のリリースが行なわれ、その後、数回の機能追加、変更が行なわれ、現在に至っています。もちろん当初から、改造計画があったわけではなく、実際に使われることでの改造が主なものです。

当然のことながら、開発にあたっては、メンテナンスの容易なものにする努力がなされました。一部のプログラムは、構造的ではなく、わかり難いものもありましたが、改造ごとに、改造対象の前後も含めて、極力美しくなるように手が加えられたこともあって、最初のリリース時点より、むしろすっきりした構造に改修された部分も沢山あります。

しかし、そのための労力、というよりも精神力は、並み大抵のものではありませんでした。担当者の方々の苦勞は、たいへんでした。

ソフトウェアの標準化や信頼性についての議論は、数々語られているようですが、現場では、まだまだ、属人的な、個人の能力にあまりにも依存しすぎているような気がしてなりません。

話のそれたついでに、例えば、自動車の新車を購入したときなど、塗装には、傷一つなく提供されることは御存知でしょうか(例外もあるでしょうけれど)。塗装は、ロボットがしますが、組み立てたり、運んだり、人の手が、介在する箇所は、いくらでもあります。一言でいえば、細心の注意が、払われているわけでしょうが、そこには、何か、そう、運用マニュアルのようなものが、きっとありそうな気がしませんか?

ソフト・ウェアの世界も、きっと例外ではないと思います。そうすれば、もっと気楽に、高品質なソフト・ウェアを生産することが、できるのではないのでしょうか。もっと、高い生産性も、実現できるのではないのでしょうか。

このことについてアイデアなどお持ちの方は、ぜひとも、きかせていただきたいものです。

なお、ここに書かれた全てのことにつきましての責任は、筆者にありますので、問い合わせなどは、筆者まで。

連絡先: 569 高槻市南平台4-16-4
横山博司

地方衛星都市におけるソフトウェア産業労働者の実態

匿名希望の地方会員

SEA MAILが、船便だろうと飛脚便だろうと、私には関係しない。しかし、会費のみで地方会員にメリットかせないのも困る。

地方会員にとって、過大評価かもしれないが、唯一の情報の入手源で有る事はまぎれもない事実であって、決して忘れてならない事であり、誰もが認める事実でもある。又、地方会員にとって、会の状況が判る情報でもある。この情報が無くなるという事は、とても怖い事である。

編集に関係されている方々の努力によってここまでやって来られたことは、既に事実として、既刊SEA MAIL数十号を見れば良い。しかし、地方会員がここでこのまま見ていれば、勝手にSEA MAILは続くと思いついてはいけない。地方会員は中央(具体的都市名は別として)で、頻繁に関催されるセミナー、分科会等に参加する事は現実問題としてかなり無理がある。(時間的、資金的等全て距離が影響する物理的な制約である)又、地方会員は都心では迷子になってしまう現実がある。(俗に言う、御上りさんでこれは私個人的な問題かも)

以上の様な現実をふまえ、今、私の現実と現状と現在と現凶(?)を全てミックスし、私が考えている事について、少々話をするにしよう。

これが、パソコン通信であれば直ぐに物議をかもしだすと、思い込んでいる私個人の見解としては、パソコン通信でよく行なわれるCHAT的な会話であっても、新聞の投書欄の様、あ〜だこ〜だ的投書(ex.最近の話題では、アグネスの子連れ論争等)もなら構わないと思う。

(もし、編集部の方で問題とあれば、前文4行をカットされても一向に構わない、この様な投稿を編集される努力は、多大なものとなることは何と無く判る様な気がしますから)

SEA MAILを出来るだけスムーズに発行させるため(?)、地方会員からの積極的な意見などをどんどん期待している(not地方会員でもいいです)。個人的見解なので決して、共感、同意などは無いと思います。その時は、反論などなんでも。

タイトルを提示したわりに本人は何を言いたいのか不明確になってしまう為、まず、本人(投稿者自信)の簡単なプロフィールを紹介する。

今から14年前の、1975年4月ソフト業界に入る。その頃は超大型機が現在のパソコンの処理レベルであって、それを多人数で使用・利用していた。元々電気系出身の者にとってハードウェアはとても興味ある物で、個人的に所有し、遊びに使いたいと思いい残業時間を有効利用(?)しハードウェアに馴れ親しんだ。

1978年頃マイクロコンピュータなる物がアメリカで作られ日本に入ってきた。これを見た時閃いた(少々オーバー)。しかし当時はまだ高価な代物。結局、入手することは出来ずに終わるが、子供がオモチャを欲しがるときのそれと、その欲望が満たされなかった時の挫折感はいつまでも心に残る物である。

そうこうしている内に、パソコンなる物が流行(ハヤ)だし、誰にでも容易に入手する

事が出来る様になり、8bitパソコンがとうとう手に入った。(その時は随分心踊るものがあった) その頃アメリカでは、UNIXがメジャーとなっていた事は当時の自分は知るよしもなかった。

それから数年間は、パソコンで学んだ(遊んだ?)。しかし16bitに主流が移りだすと、8bitは皆の心から離れ、忘れ去られた。

話が飛んでしまうが、UNIXがメジャーとなり会社でUNIXを使えばやはりパソコン的な気楽さ、便利さがあり面白い。しかし、面白かった仕事も生活環境が大きく変化しトラバユする事が必要となり、メーカー系ソフトハウスからユーザー系大企業子会社ソフトハウスに流れることとなる。

流着先は決して進んでいるとは思えない環境であり、某米国メーカー・オンリーの生産管理(製造業)なんぞと言う未知との遭遇であったために、本テーマを思いついたと言うより、本人が如実に実感した事から来ている。

随分、前置が長くなってしまったのでいよいよ本論に入りたいと思う。と、言ってもテーマのみ有るだけで個人的な見解の付いている物は少ない、このテーマで私の個人的な考え方についてぜひ会員諸氏がどの様に思い、考えているかを知りたい。そして教えてほしいのである。

現在テーマとしては、下記7種類を考えている。

- 1.仕事の都市集中化
- 2.技術者のコミュニケーションと社員の質
- 3.某社単一計算機に集中している生産管理
- 4.UNIX等パーソナル・レベルでの認識の低さ
(UNIXは生産管理等に利用出来るか)
- 5.技術力軽視・体力的労働力重視・体力的生産性重視
- 6.年功序列型
- 7.極端なクローズ世界(閉鎖的体質)

個々のテーマについて現在、私個人が考えつく事を、各テーマごとに思いつくまま書くので、それを話の糸口として行きたいと思うので宜しく願いたい。

1.仕事の都市集中化

このテーマでは自分が流れる前と流れ着いた先を比較しており俗に言うEDPの様な生産管理中心の発想が、基本で考えたもので決して他テーマとの関連はあまり考えていない。又、この業界だけかもしれないが、特に情報産業は都市集中型で発展していると思いとるところが多々見受けられるのでかなり漠然となってしまったがテーマの一つと考えて書いている。

ほとんどの企業の情報処理部門は本社に置かれている。その本社もまた一ヶ所の特定都市に集中している。現在新聞等マスメディアでは遷都論が話題となっているが、まだ具体的な動きは何もない。事実、東京と大阪に本社機構を持つ企業はかなり多いのが当

然であり、又東京・大阪に本社がある事が一流企業の一般常識になっている。ごく一部ではあるが情報処理部門(特にデータの保管場所等)に地方を採用している企業もある。

地方都市で歓迎されるのは、工場誘致を利用し地方自治体に入る収入を目的とし、若者の地方離れをくい止めるための働く場所の確保が中心である。これは、よく耳にする農業離れを防止する事を目的として地方自治体がよく利用する手段の一つと考えられる。

地方に出来るのはほとんどが工場(製造業中心)であり、工場には情報管理部門は直接いらぬ。処理結果について機械的に判断し、よりリスクの少ない手段で工場運用が出来る様な人材のみが必要であって、処理方法・処理手段等の情報処理に関係し役立つ情報の発生源と言った考えは必要としない。

地方色にもよるが、特定地方は水が良いとか等の理由で情報産業の発達している所もある。しかしこの様なケースはまれで一般的に見ると情報産業は、都市中心の考えが広く世間に常識としてなりたっている。

現在の一般サラリーマンで、都心に住宅が取得出来る人はかなり限られていると思う。都心に住宅は取得出来ない事を考え、長時間通勤を考えれば地方都市(?)で住宅は取得出来ると思う。さらに話を一歩進めて地方都市に引越し、そこで仕事を探し地域に密着する事を実行すれば仕事がないのが現実問題として浮かびあがる。仮にあったとしても都心に長時間通勤を強いられるのが関の山だと思う。

以上の様な経緯から私的判断であるが、地方都市にソフトウェアの仕事がなくても当然のような気がする。さらに、東京・大阪等の都市に集中してしまう現状により特定都市への出向及び、長時間通勤を伴う出向勤務が依然常識となっている。私の個人的なケースでは申し分ないが、地方都市に住宅取得し、やっとの思いである企業の地方工場に転職出来たが、配属された工場勤務はほんの数ヶ月で、会社の営業成績向上のため再び長時間通勤を必要とする特定都市へ出向する事となってしまった。これでは何のために地方まで来て住宅を取得したのか判らない。

2. 技術者のコミュニケーションと社員の質

SEA MAILの編集部の諸氏はかなりコミュニケーションが好きであると勝手に判断すると共に、特に夜の部の飲みにケーションには強い欲望があるとこれも勝手に判断しこのテーマを考えてもらいたいのだが?

地方都市の交通手段は現在車が中心で、免許保有者が県人口の6割をしめる。昨年(1988年)は交通事故が増加して警察は取締り強化した。このことが影響して飲みにケーションが充分出来ない。飲みにケーションは技術者間の必要な情報収集源であり、憂さ晴しの所でもあり、明日への活力となる事は一般常識である。

一般社会人での統計的數字は知らないが、飲みにケーションの大切さはかなり重要である様な気がするが、これは私だけが感じている事なのかもしれない。又、社員の質としては、仕事中心型が多くプライベート・タイムはなくてもなんら気にならない。(4、6.項等でも別途書くが)

この業界の技術の日進月歩は目を見張るばかりである。少しでも油断しようものなら明日をも知れぬ命であると、今まで思い込んでやって来た。その張り詰めた気持ちが自

己向上欲としていい部分で、仕事関係の高価な本も購入し、最初の数ページのみ読み、全てを読んだつもりで自分なりに考え、仕事に積極的に役立てていたつもりで今日までやってきた。現在の環境は決してそのような事はなくすべてが受動的である。

地方工場に勤める社員の多くは農業従事者の長男以外で、農業が嫌で工場に勤めると言うタイプが全体の大多数を占めている。社員のモラルとしては、「会社に雇って貰っている」、「働かしてもらっている」等の意識がかなりあり、仕事中心と言いながら仕事に対して責任感が殆んどない、更には仕事関係の事を積極的に勉強する気がない。だからと言って一概に無能だと決めつけるのも少々問題はあると思う。

3. 某社単一計算機に集中している生産管理

生産管理と言うと即COBOLによるカスタマイズソフトを作成する事を中心に考えている。何でも関でも、生産管理はCOBOL?で、と言った考えがかなり定着しているのが現状の様に強く感じられる。さらに、生産管理はかならずと言っていいほどカスタマイズ・ソフトを作成する、まあ場合にもよるがよほどの事がない限りソフトハウス等の出来合いのパッケージは使用される事はない。逆に出来合いのパッケージソフトの質の問題もかなりのウェイトを占めていると思われる節もあるが。

生産管理と言うとユーザーは必ず某社に見積を依頼しているところを考えると、かなりの信頼があると思えるがどんなものか個人的にはかなり疑問を持っている。国産機の見積も比較として依頼はしているが、結局某社へとなる。そんなに某社がナンバー1とも思えないのだが。

生産管理は全てカスタマイズソフトの考えであるので、プロトタイプなど時間の無駄(?)と言う考えがかなりのウェイトを占めている。又、プログラムの部品化などの考えはない。特殊多数に対する考えのツールはユーティリティーとしてあるが、UNIXの様な部品ツールの思想はない。

4. UNIX等パーソナル・レベルでの認識の低さ

(UNIXは生産管理等に利用出来るか)

UNIXは誰も知らない。知ろうとしない。必要としない。新規分野については興味がない。又、研究などについては先行投資など必要ないと考えている。

UNIX等のワークステーションは全てFA(ファクトリーオートメーション)の道具としてしか考えられない。又、考えつかない。パソコンに毛の生えた様な物的発想で、工場のライン管理位にしか使えない計算機と見ている。本当の中身を知っていれば、決してこの様な事は考えつかないはず。疑問として実際UNIXで生産管理を行っている事例が報告されていないか、どなたか知っていれば詳しく教えて欲しい。

某企業単一計算機集中により限定アプリケーションのみ使用し、3.の項でも書いているがとてもその辺の考え方に何時までたっても個人的になじめない。

5. 技術力軽視・体力的労働力重視・体力的生産性重視

製造業の地方工場の体質として、精神論的体力重視。ただ残業をすれば仕事をしていると思っている考えが強い。それでもこのごろは残業時間に対してかなり考え方が変わっ

てきたと思える。

毎日の仕事のペースでも夜の10時11時はあたりまえで、定時で帰るはさぼりと言われる様な風土がかなり強く残っている。又土曜、日曜、祝日、祭日関係なしただ働くだけカレンダーを会社と労働組合で話し生産性重視型のものが当たり前で通用している。まあ工場の操業度などと考えれば当然であると思えるが。

(祝日、祭日、日曜、土曜無視の訳の判らない週休二日)

自分の時間はなし、全て会社の時間(何年か前にあったモーレツ社員)的発想がまだ常識として残っている。

6. 年功序列型

現在どこでもまだその傾向はあると思われるが、技術力より年齢重視型の給与体系となっていてところが多いと思う。一般的に見てもその方が合理的であるかもしれないが、ソフトウェア技術者にとってはこちらかと言えば、やはり能力主義の方が有難いような感じがする。

勤めている従業員はやはり、よらば大樹の蔭的発想がありその給与体系は大企業地方工場型が如実に現れている。また人事管理についても同じ様に"人が財産"なんぞとは考えていない。替わりはいくらでもいると考えている。こんな不純(?)な考え方が基本にあるので、仕事中心で、遊びで休暇などと言っていると給与に影響がでる。

情報処理試験に関して絶対的信頼があり、神ががりの御利益があると管理者は思い込んでいる。

7. 極端なクローズ世界(閉鎖的体質)

中途採用者に対してとても排他的であり毛嫌する。まして中途採用者にスキルが有り過ぎるともっと嫌われるケースが多く見られる。

新入社員のような自家栽培的人間に対してはそれほど拒否反応はないが、やはり年齢が関係するとどうしてもスムーズに溶け込めない。この問題については個人の性格、その地方等色々な要素があると考えられるのでいちがいには言えないと思う。

結論として

かなり自序伝的な投稿となってしまったが、まずは御勘弁願いたい。

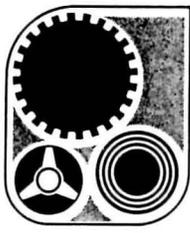
但し、トラバユして3年経った現在と、3年前の自分の考えが同じ気持ちである、その気持ちは持続してゆくと考えられる。

実際、このテーマを思い付いたのはやはり3年前だったと覚えている。

(住宅取得+親との同居がきっかけで、都会を離れ地方に移った時)

しかし、住宅取得のために田舎(?)に引っ込むより都会に残った方が幸せなのかもしれない。この業界の人にとっては・・・

そんな気持ちに揺らぎがある今日このごろです。



SOFTWARE
ENGINEERING

BUILDING A FOUNDATION FOR THE FUTURE
NICE, FRANCE MARCH 26-30, 1990

CHAIR

François Régis Valette
ONERA CERT
BP 4025
31055 Toulouse cedex, France
tel : (33) 61 55 70 90
email : valette@tls-cs.cert.fr

PROGRAM CO-CHAIRS

Peter Freeman
ICSE-12 Program
ICS Department
University of California Irvine,
CA 92717 USA
1 (714) 856-7403
email: freeman@ics.uci.edu

Marie-Claude Gaudel

ICSE-12 Program
LRI, Bat. 490
Université de Paris-Sud
91405 Orsay cedex, France
tel : 33 (1) 69 41 69 09
email: meteor@lri.lri.fr

TOOLS FAIR CHAIR

Michel Mouyssinat
Greco de Programmation-CNRS
Université de Bordeaux 1
351 Cours de la Libération
33405 Talence cedex, France
tel : (33) 56 84 60 90
email : mm@geocub.greco-prog.fr

TUTORIAL CHAIR

Bernard Lang
INRIA
Domaine de Voluceau-
Rocquencourt
BP 105
78153 Le Chesnay cedex, France
tel : (33) (1) 39 63 55 11
email : lang@inria.inria.fr

LOCAL ARRANGEMENTS CHAIR

Marie-France Kalogera
AFCET
156 Bd. Pereire
75017 Paris, France
tel : (33) (1) 47 66 24 19
Fax : (33) (1) 42 67 93 12

Sponsored By



ACM SIGSOFT



IEEE Computer Society



ASSOCIATION FRANCAISE
POUR LA CYBERNETIQUE
ECONOMIQUE ET TECHNIQUE

Other sponsorships have been requested.

In the second 20 years of its existence, it is important that the field of software engineering increasingly plan for the long term. Moreover, there is a growing necessity to evaluate the impact on Software Engineering of the major developments in computer science : (parallelism, formal methods, distributed processing, non Von Neuman architectures, automated reasoning techniques) and its most compelling applications (massive networks, life-critical systems, economy-critical systems, computationally dependent research).

The 12th ICSE will focus on efforts in different aspects of Software Engineering that are essential for building a solid foundation and for anticipating the future : Basic Research, Advanced Developments, Experiences of Technology Transfers and Analyses of Empirical Results. Contributions that describe results, not just proposals, are solicited in any content areas of Software Engineering, including but not limited to, quantitative methods, testing and verification, languages and representations, requirements analysis and specification, process analysis and modeling, project management, human factors, environments, methodologies, and applications of Artificial Intelligence to Software Engineering.

ICSE 12 will make a conscious effort to balance the program so that it will appeal to a broad segment of the field and serve as a bridge between theory and practice. In this milieu it is hoped that researchers will better understand the needs of practitioners, that practitioners will better understand current and possible future research results, and that those concerned with technology transfer can better understand both.

In each of the 4 broad categories, priority will be given to submissions that discuss topics which can be expected to be important for at least the next 5 to 10 years. In all areas, priority will be given to submissions that report results. The conference program will consist of contributed papers, invited lectures, panels, and short reports on practical experience. All parts of the program will be reviewed by the program committee for relevance and quality, making it essential that papers, panel proposals, and reports be submitted in writing by July 14, 1989.

SUBMISSIONS: Eight (8) copies in english of papers, panel proposals, or experience reports should be submitted to either (but not both) program co-chairs by July 14, 1989 at one of the addresses below :

Peter Freeman
ICSE-12 Program
ICS Department
University of California Irvine,
CA 92717 USA

Marie-Claude Gaudel
ICSE-12 Program
AFCET
156 Bd. Pereire
75017 Paris, France

Papers should be limited to 6000 words, full-page figures being counted as 300 words. Each paper must include a short abstract, a list of keywords, and the lead author's address.

Panel Proposals should include title, proposed chair, tentative panelists (including a short vita), a 2 or 3 paragraph description of the subject, format of the presentation, and rationale for the panel.

Experience Reports are intended to provide exposure of practical experience in some aspect of software engineering (e.g. using a tool or method, applying a metric, following a project management discipline, reusing Software Engineering workproducts, etc.). As with all other submissions, the primary test applied by the Committee will be the relevance of the results to future work. The contributor(s) should submit a 3-5 pages written description of the experience and a one-page outline for a 5-minutes presentation.

TOOLS FAIR : A Software Tools Fair will be held in parallel with the conference. The Tools Fair will allow conference attendees to see and learn about current software tools. Both experimental and commercial tools will be demonstrated. Those interested in exhibiting a tool, and especially those authors interested in presenting a paper describing a particular tool and demonstrating that tool, should contact the Tools Fair Chair :

Michel Mouyssinat
Greco de Programmation-CNRS, Université de Bordeaux 1
351 Cours de la Libération, 33405 Talence cedex, France

FURTHER INFORMATION: For further information and/or copy of the advance program when available, write either to ICSE12, AFCET, 156 Bd. Pereire, 75017 Paris, France, or to ICSE12, IEEE Computer Society, 1730 Massachusetts Ave., NW Washington DC, 20036 USA.

IMPORTANT DATES : Submission deadline : July 14, 1989, Acceptance notification : October 16, 1989, Final versions due: December 18, 1989.

IEEE



THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. INC.

これからの主要イベント予定 (1989.5.10 現在)

5月30日(火)	月例フォーラム「ソフトウェア工学の最新動向」(東京・機械振興会館)
6月13日(火)	シンポジウム '89 併設チュートリアル(東京・虎ノ門パストラル)
6月14日(水)～15日(木)	ソフトウェア・シンポジウム '89(東京・虎ノ門パストラル)
6月14日(水)	1989年度会員総会(東京・虎ノ門パストラル)
7月	集中セミナー in 大阪(期日、会場等の詳細は未定)
8月31日(木)～9月2日(土)	第3回教育ワークショップ「ソフトウェア設計」(熊本・阿蘇山麓)
9月 6日(水)～ 9日(土)	第7回夏のプログラミング・ワークショップ(若手の会) 「ソフトウェア開発の定量化」(岩手・盛岡)
9月20日(水)～22日(金)	秋のセミナー・ウィーク '89(東京・青年会議所会館)
10月23日(月)～25日(水)	第3回日中ソフトウェア・シンポジウム(大阪ガーデンパレス)
11月中旬	ソフトウェア工学に関する国際シンポジウム(京都リサーチパーク)
11月29日(水)～12月1日(金)	第5回環境ワークショップ(北海道・釧路)
秋 (期日未定)	第6回ソフトウェア技術交流会(開催地未定)
1月11日(木)～13日(土)	第2回テクニカル・マネジメント・ワークショップ(函館)

会員総会のお知らせ

日 時 6月14日(水) 16:30 - 17:30

場 所 農林年金パストラル(桂の間)

ソフトウェアシンポジウム第一日目終了後、情報交換パーティまでの間に開催致します。シンポジウムに参加される方は是非ご出席下さい。



ソフトウェア技術者協会

〒102 東京都千代田区隼町2-12 藤和半蔵門コープビル505
TEL.03-234-9455 FAX.03-234-9454