



SEAMAIL

Monthly Newsletter from
Software Engineers Association

Volume 1, Number 10 | October 1986

目 次

編集部から		1
代表幹事からのメッセージ	鈴木弘	2
セミナー 21 レポート (その 1)		3
総括 (アンケート結果を含んで)	杉田義明	3
ソフトウェア開発現場とネットワーク	寺本雅則 / 芝原雄二	4
プロジェクト管理のエコロジー	松原友夫 / 平山伸一	6
環境におけるプロジェクト管理	深瀬弘恭 / 大槻亮人	8
システム監査と EDP 部門	吉村成弘 / 佐原伸	12
盛岡ワークショップの報告	松尾正敏	15
ソフトウェア開発環境としてのメインフレーム TSS	田中慎一郎	18
Mac と Unix の世界	佐原 伸	22
ハンドヘルド・ワークステーションの夢	佐久間まゆみ	25
中国旅行記-連載その 1-	野村敏次	28
幹事会, 分科会および支部活動		32
幹事会 A I 分科会 管理分科会 教育分科会 再利用分科会		
法的保護分科会 環境分科会 横浜支部 関西支部		
会員の声		34
カレンダー		37

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・メーカ、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つて行われ、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： 鈴木弘

常任幹事： 岸田孝一 長井剛一郎 盛田政敏 吉村鉄太郎

幹事： 稲田博 臼井義美 岡本吉晴 落水浩一郎 皆藤慎一 木村高志 久保宏志 斎藤信男 三枝守正
杉田義明 辻淳二 鳥居宏次 中園順三 針谷明 松本崇純 松原友夫 水谷時雄 三浦信之

会計監事： 近藤秀朗 吉村成弘

常任委員長： 岸田孝一（会誌編集） 盛田政敏（企画総務） 吉村鉄太郎（技術研究） 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)：岡本吉晴 久保宏志 引地信之 松尾正敏 水谷時雄

管理分科会(SIGMAN)：岸田孝一 塩野富教 芝原雄二 鈴木信裕

教育分科会(SIGEDU)：大浦洋一 杉田義明

再利用分科会(SIGREUSE)：青島茂 阿倍正平 村井進

AI分科会(SIGAI)：安倍昭敬 坂下秀 臼井豊 高田佳彦 広川昭八 野辺良一 藤野晃延 横山憲一

ネットワーク分科会(SIGNET)：鈴木弘

法的保護分科会(SIGSPL)：能登末之

支部世話人 関西支部：臼井義美 盛田政敏

横浜支部：熊谷章 林香 藤野晃延 松下和隆

SEAMAIL編集グループ：大槻亮一 大西亮一 岸田孝一 佐原伸 沢田寿実 芝原雄二 関崎邦夫 田中慎一郎
長井修治 野辺良一 藤野晃延 山内徹 渡辺雄一

SEAMAIL Vol. 1, No. 10 昭和61年10月1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）

〒166 東京都杉並区高円寺南1-5-4 高円寺サンハイツ404

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

定価 500円

編集部から

お待たせしました。船便の10月号です。なんとか今月中（もとい、先月中）に発行すべく努力したのですが、ほんのちょっぴり(?)遅れてしまいました。ゴメンナサイ。

しかし、11月および12月号は同時編集を開始しましたので、どちらも、発行予定月の半ばには、お手元にとどくと思います。

苦しい財政のうちから返信用封筒を入れるという投資をしたおかげで、9月のアンケートに対しては、200通に近い回答がありました。ありがとうございました。結果は、整理がつきしだい、次号以降に順次のせていきます。

それでも、まだ約400人の方がNoResponseです。いまからでもかまいませんので、ぜひ御回答ください。

セミナー・ウィーク特集

9月3～6日に、機械振興会館（東京・芝公園）で開催されたSEAセミナー21の報告を、今月から掲載します。ただし、セッション数が21もあるので、3～4回に分けて紹介して行きたいと思います。今月はまず、A1, F1, F3, G1の4セッションと、杉田セミナー委員長の総括報告です。

盛岡ワークショップ報告

第4回目の夏の若手ワークショップが、9月10～13日に、岩手県盛岡市で開催されました。テーマは、「プロジェクト・プランニングのケース・スタディ」。近々報告書が発行されますが、ここでは、プログラム委員長の松尾さんに、全体のサマリーを書いていただくことにしました。

なお、このワークショップの討論結果を材料にして、プロジェクト・プランニングのフォーラムが11月7日（金）に開催されることは、すでに御案内の通りです。

会員からの投稿

ばちばち、会員からの投稿が集まりつつあります。今月とはくに、この夏北京の国際会議に出席された野村さんから、大作の紀行文が寄せられました。全3回という、SEAMAILはじまってからの連載物になりました。

会員アンケートの回答のなかに、「機関誌にこんなページがほしい」といった要望がいくつかありましたので、編集部からお答えします：

SEAMAILは、手作りの雑誌です。もし、あなたがそう望まれるのであれば、自分でまず、お望みのコラムをスタートさせるきっかけとなる原稿を書いてください。

われわれ編集部の役目は、ただ単純に、みなさんから寄せられた原稿をCompileするだけだということをお忘れなく。

分科会および支部からの成果報告

環境分科会、横浜支部、そして関西支部から、それぞれのミーティングでの発表や討論をまとめたレポートをいただきました。しかも、神戸の佐久間さんの原稿は、自筆のイラスト入りです。

分科会、支部共に活動がだんだん定着してきました。これからも、こうしたレポートを逐次掲載していきたいと思いますので、よろしく。

幹事会、分科会および支部報告

前号でお知らせしたように、一般の原稿を優先するために、分科会・支部関係のニュースは、開催予定が中心になっています（それが船便じゃ役に立たないというお怒りは、まあ、抑えてオサエテ！）。

会員からのメッセージ

第2号に、「会員の声」というコラムがありました。これは、設立時のアンケートに寄せられた声をまとめて掲載したものです。

その後、何人かの会員の方から、まとまった原稿でなく、たとえば、呼びかけや問い合わせのように、簡単なメッセージをのせる欄がほしい、という意見をお聞きしました。

そこで今月号から、定期的にこのコラムを設けしました。探し物、仕事の愚痴、ちょっといい話等々、どうぞ編集部（東高円寺事務所）までお気軽にお寄せください。

代表幹事からのメッセージ

鈴木 弘

構造計画研究所

春の総会で私が代表幹事に推薦されてから、半年もの時が過ぎてしまいました。もっと早くみなさんに御挨拶のことばを書く予定だったのですが、例になく（！？）仕事に追われ、今日までのびのびになりました。あらためて、ここで私の所信と、会員各位へのお願いを述べさせていただきます。

1. SEAへの期待と必要性

ソフトウェア産業はこれまで、毎年20%の伸びで成長してきており、今後も全体としては、その伸びが続くだろうといわれています。しかし、たとえば、計算センタなどは売上が頭打ちになってきていますし、その一方で、ネットワーク、AI、ワークステーションなどの新しいビジネス・チャンスが開けつつあります。

これらの現象は、ソフトウェア産業の質的・構造的変化を物語っているわけですが、われわれ個々の技術者も、こうした産業構造の変革に、これからは、いやでも対応していかなければなりません。

私自身がこの世界に足を踏み入れてから、かなりたちましたが、そのあいだ常に、質・量両面における人材の不足と、開発工程の改善の必要性を感じ続けてきました。この問題は、ただ単純に人を増やしただけでは解決しないことは、皆さんよくご存じの通りです。

現在、ソフトウェア開発の仕事には、日本中で40万人以上が従事しているといわれています。これらの技術者たちは、しかし、社会構造上は、ソフトウェア・ハウス、計算センタ、メーカ、ユーザ、大学・研究所など、さまざまなセクターにばらばらに位置付けられ、それぞれの現場での問題点を解消することに追われて、共通の技術的・社会的課題を認識する余裕を持ちえていません。そうしたことが、ひところマスコミで流された悪いイメージとあいまって、ソフトウェア技術者に対する一般の認識を薄めている原因になっています。

SEAは、既存の社会組織を横断するプロフェッショナル・ソサィエティとして、

- ・技術者としての社会的責任への自覚の浸透
- ・技術者であることへの社会的認知
- ・技術向上のための相互の交流と情報交換

を促進することをめざします。

2. これからの展望とお願い

SEAは、あくまでもソフトウェア技術者が個人として参加し、相互の交流を通じて、それぞれの人的・技術的向上を図るための場です。ある種の知的職業に従事する専門家の団体として類似のものに、たとえば医師会や建築士会、弁護士会などがありますが、これらはSEAとちがって、すでにその存在がきちんと社会的に認知されています。

われわれの活動の難しさは、おそらくそのへんにあるでしょう。すでに、昨年の設立にさいして、一部から労働組合または政治団体に発展するのではないかと、といった見当ちがちな誤解があったりしたのは、そのためです。こうした誤解をとりて、SEAをより発展させるためには、会員1人1人が強い自覚を持って、地道に活動を展開して行く以外にないでしょう。

9月末現在で、会員は600人をこえ、特定のテーマをもった研究分科会も7つが活発に活動をしており、その成果もいくつか出はじめました。また、こうした活動は、ともすると東京が中心になりがちですが、関西と横浜で2つの支部が独自の活動を行っていますし、いくつかの地域で新しい支部設立の動きもあります。

そうした成果をPRするために機関誌の定期的な発行が必須であり、さらには、セミナー等のイベントの開催による一般への広報が重要だと考えていましたが、これまでのところ、多くの方々の努力のおかげで、順調にそうした土台作りも進んでいるようです。

代表幹事としての私の第1の目標は、組織としてのSEAの安定化です。昨年12月の正式設立以来半年で会員数が600人に伸びたということは、かなりの成果だと思いますが、組織および財政基盤の確立という点から考えると、まだまだ十分とはいえません。すでに幹事会からのお手紙でも申し上げてありますが、さしあたり会員1000人の確保をめざして、会員各位には、それぞれ「1人が1人を！」を合い言葉に組織拡大のため御努力をお願いしたいと思います。

SEA 秋のセミナー・ウィーク

総括報告

杉田 義明

セミナー・ワークショップ委員長

9月3～6日に東京の機械振興会館でおこなわれた秋のセミナー・ウィーク（SEAセミナー21）は、わが協会としては、春のセミナー・ウィークに続く2回目の大型公開イベントでした。

しかも、前回は、日本ユニックス・ユーザ会（jus）との共催でしたが、今回は単独開催ということで、SEAならではの特色をだそうと、パンフレット等でも御案内したように、プレゼンテーションと自由討論を半々にした、インフォーマルなミニ・ワークショップという新しいスタイルに挑戦することを、企画しました。

この新しい試みは、講師や事務局が不慣れだったせいも、セッションによっては、ややうまく行かなかったところもあったようです。また、受講者の方にも、すこし戸惑いもあったように見受けられました。

しかし、受講者アンケートの結果からみると、この新しい試みは、全体としては、好感をもってむかえられたようです。代表的な声をまとめると：「自由討論の時間が短く感じられた。同じようなことを、いろいろな所や人が考え、悩んでいることがわかった。再度同じテーマで、もっと突っ込んでディスカッションがしたい。そのためには、自分ももっと勉強しておきたい」といったところです。

編集ボランティアの方々の御協力を得て、全セッションの記録（要約）を、今月から順次SEAMAILに掲載して行きます。読者サイドからの御意見・反論・コメント・感想等が活発に寄せられれば、ひきつづいて、誌上でのミニ・ワークショップが可能ではないかと考えています。そして、その内容を反映したかたちで、来年3月に予定されている春のセミナー・ウィークの企画を立てられたらと思っています。

幸い今月号から、「会員の声」欄が復活していますので、短いもので結構ですから、セミナーに参加できなかった方々も、ぜひいろいろな声をお寄せください。

次の表に、各セッションの参加者数（招待者を除く）と、受講者からの評価をまとめてみました。評価の数字

は、受講者アンケートの中の「内容は期待通りだったか？」、「仕事に役立つか？」という2つの問いに対する回答を+2から-2までの5段階で採点し、集計したものです。セッション F2 のアンケート票が、なぜかどこかにまぎれて、見つかりませんでしたが、どのセッションもプラスの評価になっています。

セッション番号	参加者数	評 価
A1	16	+17
A2	15	+12
A3	19	+27
B1	18	+24
B2	11	+8
B3	23	+35
C1	32	+12
C2	42	+52
C3	35	+16
D1	20	+22
D2	32	+22
D3	27	+6
E1	25	+6
E2	46	+13
E3	34	+31
F1	34	+12
F2	51	*
F3	20	+12
G1	21	+12
G2	9	+24
G3	10	+22

今回のセミナーでは、教育分科会のメンバーをはじめとして、多くの会員の方々に舞台裏でお手つだいいただきました。この誌面を借りて、心からお礼を申し上げます。

SEA 秋のセミナー・ウィーク

セッション A1

ソフトウェア開発現場とネットワーク

講師： 寺本 雅則

報告者： 芝原 雄二

1. はじめに

このセッションでは、ソフトウェア開発現場とネットワークとの関わり合いをテーマに、具体的な話題としては、シグマ・プロジェクトおよびC&Cサテライト・オフィスを取りあげて、講演がおこなわれた。

シグマ・プロジェクトは、すでに御承知の通り

- ・ソフトウェアの信頼性・生産性の向上
- ・ソフトウェアの重複開発防止
- ・ソフトウェア開発設備の充実／ノウハウの蓄積／技術の向上
- ・技術者教育の効率化

等をねらった国家プロジェクトである。

一方、C&Cサテライト・オフィスは、講師が勤務する日本電気の社内で試みられている新しい職場環境をめざした実験であり、

- ・ソフトウェア技術者の質的・量的確保
- ・女性をはじめとする潜在労働力の活用
- ・広域分散型雇用の促進—地方の時代—
- ・ソフトウェア開発形態とその技術の確立

等を目的としている。

講演のなかでは、また、上記の話題に関連して、UNIX環境で提供されるネットワーク機能、プロトコル、マルチマイクロ・プロセッサ、分散ファイル・システム、分散OS等についてもお話があった。

このレポートは、これらの講演およびそのあとに行われた自由討論のなかから、とくに報告者自身にとって印象に残ったことがらをピックアップし、私見をまじえながらまとめたものである。

2. 開発現場でのネットワークの必要性

ソフトウェアの開発現場で最近考えられているのは、ネットワークの導入によって、デバック環境を改善しようということである。ターゲット・マシンと開発マシンが同じという状況は、しだいに例外的なものになりつつある。とくに、マイクロソフトウェアの世界では、ほと

んど稀といってよいのではないだろうか。

こうした環境下では、開発マシンで作られたプログラムを、ネットワークを介してターゲット・マシンに送ることによって、最終確認テストが容易になる。

しかし、ネットワークのもたらす真の効果はそんなところにあるわけではない。

ソフトウェア開発という仕事は、ただ単に、マシンを使ってプログラムをコーディングし、デバッグすることだけで成り立っているのではない。グループ内の進捗会議、ユーザや他部門との打ち合わせ、文献調査その他の技術情報の収集といった、さまざまな作業を含んでいる。これらの作業を円滑に遂行して行くためには、人間と人間とのあいだの多様なコミュニケーションが必要である。そこにネットワークを活用することによって、仕事の量を軽減し、さらには、入手する情報の量をふやすことが可能になるはずだ。

報告者自身も、最近、仕事の関係で、大量の文献を調査する必要にせまられたが、全部自分の足を使って捜さなくてはならず、そうした技術情報の検索システムや、技術者どうしのコミュニケーションがネットワーク化されることの重要性を痛感させられた。

世の中一般を見回してみても、ネットワークの必要性は、最近ますます高まりつつある。たとえば、通勤難、大都市部の高騰するオフィス、労働力の地方への分散、女性をはじめとする潜在労働力の活用といった社会環境の変化によってもたらされる諸問題を解決するためにも、ネットワークは大きな力となる。

コンピュータ・ネットワークは、もともと、各種ハードウェア資源の共同利用を目的として生まれたものであるが、最近では、パソコンの普及も伴って、電子メール等による情報の交換にウェイトが移行して来ているように見受けられる。

また、この他にも、新しいメディアとして、音声会議、静止画会議、動画会議等の会議システム、高速デジタル

回線、パケット回線を利用したネットワーク・システム等がある。これらを統合することにより、時間や場所を意識しないソフトウェア開発環境の提供が可能になる。

3. ネットワークのソフトウェア開発への利用

電子メール、電子掲示板、電子ニュースレター、ファイル転送、遠隔ログイン等のネットワーク・サービスを利用した開発の効率化は、そろそろあちこちの会社で試行され始めているが、そうした環境で仕事ができるのは、いまのところまだまだ恵まれた一部の人間でしかない。今後、シグマ・プロジェクトの成果が、直接・間接に世の中に普及することによって、ネットワーク環境の利用者層が大きく広がることを期待したいものである。

そうなれば、技術情報の問い合わせ・紹介、公開ソフトウェアの配布、磁気テープやディスク上の情報の交換、遠隔地からのコンピュータの利用といったことがいっそう容易になり、よりよい環境でソフトウェアの開発ができるようになる。

たとえば、いまのように机上に参照用の部厚いファイルを常備したり、長いリストを広げたりといったことは、必要でなくなるだろう。在宅勤務も（物理的には）可能になるし、また、プロジェクト・チームをいくつかの場所（地方都市）に分散した新しいマネジメント体制を考えることもできる。田舎の別荘で、好きな酒を飲みながら、ときどきはプログラムを組みたいという第1世代の老プログラマたちの夢も実現できるかも知れない。

4. LAN導入のインパクト

ここまでは、ネットワーク一般について議論を展開してきたが、近い将来において、われわれ自身の開発現場にどのような改善が可能だろうか、という観点から考えてみると、LANの導入による環境の分散化が、もっとも速効性が高いのではなかろうか。

ソフトウェア開発とは、チーム・メンバーのそれぞれが、お互いに協力しあいながら、ソース・コードを含む各種の情報を作りだし、蓄積し、検索する連続的なプロセスである。

これまでの開発環境では、ふつう、情報は集中的に管理されている。たとえばTSSの場合、マシン室のディスク群に、このことは、開発成果物の管理という観点からすべての情報が一括してはいっている。すれば便利だが、個々の情報を作りだし、利用する技術者の立場からは不便点が多い。

高性能のパーソナル・ワークステーションが普及し、

それらがLANで接続され、何らかの仕掛けによるファイルの共有化が可能になれば、情報を「物理的」には分散し、「論理的」には統合した理想的な環境が実現できる。このことは、たとえばUNIXの世界では、すでに夢ではなく、現実の形になりつつある。

このようなLAN環境の導入にさいして一番問題になるのは、運用管理である。分散化された情報を、どう管理すればもっとも効果があがるのかを考えていかなければならない。そのためには、ハードウェアの知識だけでなく、コミュニケーションの知識が、運用管理者に求められてくる。また、応用ソフトがLAN普及のキーポイントになるであろう。

LANの構築にさいして、プログラミングやデバッグの支援を中心にするのか、打ち合わせ等のコミュニケーション支援を重点に考えるのかによって、マシンの配置、ソフトウェア・サービスの提供等の形態は、かなり異なってくる。C&Cサテライトでは、そのようなことを考えたオフィス空間の実現が試みられている。

このことに関連して参加者からいくつかの質問があり、未来のソフトウェア・オフィスのあるべき姿について、討議がおこなわれた。

ネットワークの構築以外にも、照明の位置、机の高さ、作業空間の広さ、個人を仕切るしきりの高さ等々、多くの人間工学的な要因を考慮する必要があり、C&Cサテライトは、そうしたことがらも含めた実験を意図しているということであった。

5. おわりに

かつて、ソフトウェア開発の3種の神器は、机、紙、そして鉛筆であった。今やそうした「紺屋の白袴」的状况をどうやって脱却するかが大きな課題と考えられるようになってきた。このセッションに参加して、われわれが1日の1/3以上を過ごしている環境を、より人間性を重視したものに変えて行くことの重要性を、あらためて強く感じた。

ハードウェアのコストやネットワークの使用料は、飛躍的に安くなってきている。反面、人件費の高騰は、人材不足もからめて、ますますソフトウェア産業発展の障害になりかねない。解決のひとつの方向として、あくまでも人間性尊重を第一義に考えたネットワークの構築を意図すべきであろう。これは、ともすれば人間を消耗品扱いにしているという業界の悪いイメージを打破することにも繋がると思う。

SEA 秋のセミナー・ウィーク

セッション F1

プロジェクト管理のエコロジー

講師： 松原 友夫

報告者： 平山 伸一

1. なぜプロジェクトは失敗するのか

トム・デマルコが、その著書「ソフトウェア・プロジェクトの管理」のなかで

- ・ 15% のプロジェクトが全く役に立たない
- ・ 100-200% のコスト超過はあたりまえ

と述べているように、ソフトウェア・プロジェクトは、一般に、失敗して当然のように考えられている。しかし、それには、いくつか特有の原因がある。

まず、ソフトウェアそのものの性格として、人間の目には直接見えないということがあげられる。このことは、トラブルが起こった場合に、修正ループがどうしても大きくなってしまい、せっかく助っ人をつぎこんでも、その効果が少ない、といった問題を発生させ、それがコスト超過の原因にもつながってゆく。また、F. ブルックスのいう「人・月の神話」という落とし穴に管理者を陥れる。

さらには、技術移転がうまくいかないという結果や、プログラマたちのあいだに自己流の職人芸が横行してしまうという悪循環をもたらす。気のきいた工夫や技術上のアイデアを世間に発表し、ある種の名譽を得る機会に乏しいという状況も、プログラマが自分たちの仕事をお座なりにすませてしまう原因となっている。

次に、客観的評価手法の乏しさがあげられよう。ソフトウェアの大きさ、工数、コスト、生産性、どれをとってみても、その評価指標はあいまいであり、品質評価のポイントさえ、人によってまちまちである。まるで、綿アメをゴムの物差しではかるようなものでしかない。

次には、ソフトウェアが人間的要素を多分に含んでおり、それがまた、きわめて重要な役割を占めていることがあげられる。システムに、人間的要素が含まれることが多ければ多いほど、そのシステムに関する予想は、むずかしくなる。

たとえば、そのよい例として、勝てると信じて深入りしてしまった第2次世界大戦やベトナム戦争があげられ

よう。経済学の分野では、オイル・ショック以降の石油価格の変動がそうだし、技術的な領域での例を考えれば、最近のジャーナリズムをにぎあわせたチェルノブイリ原発事故がある。ソフトウェア・プロジェクトの場合にも、同様の例は枚挙にいとまがない。

2. プロジェクトを混乱に陥れる構造

プロジェクトが大規模になり、内容が見えにくくなるにつれて、一般に、管理は強化される傾向にある。また、一旦トラブルが起こると、その反省としても管理の強化がもたらされる。しかし、それによって、ほんとうにトラブルは解消するのだろうか。

計画から大幅にずれてしまったプロジェクトに対しては、かならずその原因の調査がおこなわれるが、「何とかしたい、何とかなるはずだ」というバイアスがかかった管理者の主観的評価にもとづいて、状況判断がなされ、善後策をとるための意志決定がくだされる。しかし、多くの場合、そうした主観的判断は、誤りであることが多い。

G. ワインバーグは、こうした判断ミスの背景として、管理者の側に、現状の技術的評価に対する無力感や、本当のことはわからないという不安感が存在する、と述べている。

その結果、かれらにとって唯一安心できる対策として管理の強化がおこなわれるのだが、それは、往々にしてカタストロフィを生んでしまう。最近の例としては、マルコス政治の末期における混乱があげられよう。

3. 自由と統制

全体としての自由を保つためには、デモクラシーの原則と少数者保護のルールが必要である。こうしたルールの積極的意味として、チームワークについて触れる。

サッカーの名選手ベレについて、仲間のアレイヤが、「ベレという男は、いつでも、どこでも、どんなときでも、周りにいるわれわれに対して、最善をつくそうとする男です。そんな男に対して、どうしてひどいボールを

蹴ることができますか」と語っている。

この「ひどいボールは蹴れない」というルールは、民主的に暗黙のうちに決定されており、全体として、よい結果を生みだしている。つまり、チーム員全員がプロジェクトの意味と目標を知り、全体の行動原理と自らの役割に照らして、自律的に目標に向かって行動することが、プロジェクトがうまくいくための、ひとつの理想形であろう。

プロジェクトの意味と目標をよく知るためには、共通の場で、共通に認識できる情報が必要である。しかし、一般に、人間の思考や使う言葉の意味は、共通化されていない。人間の思考パラダイムは、使う言葉や職業（手段や環境）と密接に関係しており、また、心理学という刷り込み現象の例をだすまでもなく、初めてのものに影響を受ける。つまり、各自の視点がそれぞれ異なるわけであるが、そのどの視点も部分的には正しく、これが多くの論争や行き違いを生むもとになっている。

こうしたことから、共通の場をつくるために、技法や概念の標準化が必要になってくる。ただし、そのさい、決して誤解してはならないのは、標準化の真の目的が、より広い場の下での自由を確保することだ、ということである。

4. ホロンのプロジェクト管理をめざして

ホロン (Holon) とは、全体を意味するギリシャ語の単語ホロス (Holos) に、粒子とか部分を示す接尾語オン (on) をつけたものであり、「全体子」と訳され、それ自体「個」であると同時に、「全体」との調和を保つということの意味している。

最近のワークステーションの高性能化は、それらを自律性を持った「個」として位置付けることを可能とし、一方、ネットワーク技術の進歩は、これらの「個」が「全体」に協調することの可能性をもたらした。つまり、技術的な側面だけを考えれば、LANを活用した分散型の開発環境は、すでに十分ホロンの環境であるといえよう。したがって、そこにホロニック・システムの思想を組み込めば、個々の開発者が自律的に行動しつつ、理想のプロジェクト環境を、段階的に作りあげて行くことができるだろう。

以下に、開発環境をホロニックにするための条件を列挙しておこう：

- ・いつでも好きなときに使える、LANで結ばれたワークステーション。

- ・ネットワークで結ばれた場が十分に広く、その上に蓄積された情報が多い。
 - ・人やプロジェクトの変化（ゆらぎ）に対応できる柔軟なツールとシステム構成。
 - ・自らの振舞いと、その全体の中での位置付けが一目でわかるシステム。
 - ・その振舞いデータが仕事の副産物として把握できるシステム。
 - ・上方に開いていて、下方に選択的に閉じているシステム（監査システムにしないため）。
 - ・自分のワークステーション上で開発作業をやりながら、プロジェクト全体のステータスを知ることができるシステム。
 - ・人やシステムの振舞いデータを、すぐにシステム改善にフィードバックできるシステム。
 - ・開発作業であれ、プロジェクト管理であれ、たいいていのやりたいことがワークステーション上でできるシステム、つまりツールの選択の幅が広いシステム。
- こうした諸条件を満足させるには、環境を構築するさいのちょっとした気配りが大切であり、また、それさえあれば比較的簡単に実現できるのではないだろうか。しかも、ソフトウェア開発では、生産物と管理情報を同じシステム上で扱えるのだから。

5. おわりに

以下に、まとめて代えて、セッション参加者からいただいたアンケートの声をいくつか載せておく：

- ・情報処理業界で使われている用語のあいまいさを十分に認識できた。
- ・もうすこし簡潔に、重点を絞って話してほしい。
- ・時間が足りなかったが、今後のプロジェクト管理を行っていくうえで確信がもてた。
- ・プロジェクトのメンバーとのコミュニケーションを考え直す必要があることがわかった。
- ・プレゼンテーションに時間をとりすぎて、自由討論の余裕がなかったのは残念だった。内容的にも、3時間で自由討論は無理であろう。
- ・話の範囲が広すぎて、ポイントがつかめなかった。ホロンの話をもっと集中的に話たほうがよかったと思う。ホロンの概念自体は、非常に興味深かった。
- ・「概念」の説明としてはよかった。そのなかから、プロジェクト管理へのヒントを得ることができたが、できれば、いくつかの具体例で説明してほしい。

SEA 秋のセミナー・ウィーク

セッション F3

環境におけるプロジェクト管理

講師： 深瀬 弘恭

報告者： 大槻 亮人

1. はじめに

このセッションの前半は、講師自身の豊富な経験にもとづいて、環境整備の重要性、プロトタイピングの有効性、ネットワークの重大さ、開発環境の分散化などを内容とする講義が行われた。その要旨は、すでにSEAMAIL 7月号に紹介されているので、ここでは、後半の質疑および討論の内容を報告する。

2. 開発マシンとターゲットマシン

K氏： 開発マシンとターゲットマシンの関係について、パソコンの場合と汎用機の場合に分けて、お聞きしたいのですが。

深瀬： 私の会社はパソコンの仕事をしていますので、具体的に話すために、パソコンの場合をとりあげます。その前に、このなかで開発マシンとターゲットマシンを分けて開発している人はいますか？

全員： (しばし全員沈黙)

深瀬： われわれのところでは、ホストとワークステーションの両方にクロスコンパイラがのせてあり、スタンドアロンでもクロス環境でも、同じように仕事ができるようになっていきます。

具体的にいえば、MS-DOS側にもCのコンパイラやシンボリックデバックがのっています。ホスト側では、Cコンパイラ、リンカ、デバックを使い、最後の実行形式のモジュールまで作ってしまうことが可能です。もちろん、リロケータブルなモジュールをばらばらに作り、それをダウンロードして、ワークステーションで仕事することも可能です。デバック自体は、ターゲットマシンで行い、世代管理はホストマシンでやるようにプロジェクトを組んでいます。ファイルが分散して、世代管理ができなくなることを避けるために、ホスト側を倉庫番に使っています。

個別のデバックは、ターゲットマシン（パソコン）でやるのが可能です。メインフレームの場合は、ふつう、こうしたことは不可能でしょう。開発現場にターゲット

の環境がない場合が多いのです。回線がない、ディスクの構成が違う、データベースの中身が違うなど、いろいろな障害があるわけです。

A氏： われわれの会社は、一番旧式の方法ですが、じかにホストのメインフレームに端末をつないで仕事をしています。最近、親会社から独立したばかりなので、いまは客先（親会社）の環境下で作業をしていますが、2年以内にUNIX環境におきかえることを考えているのですが。

深瀬： 私がもし、ホスト向けの大きな仕事（COBOLでのアプリケーション開発）を請け負ったとしたら、COBOLでの開発環境（シンタックスチェッカ、シュミレータ）をホスト側にまず作るでしょう。そして、ワークステーション側には、DB/DCの基本的な部分までエミュレートする環境を作ってしまうのです。

現実には、ANSIのCOBOLはUNIX上にもパソコン上にもあります。メインフレームのCOBOLは方言が多いので、いくつか機能を追加しなければならないでしょうが、一番ネックになるのは、DB/DCの取り扱いや、画面設計をすることだと思います。

しかし、小さいプロジェクトでは、ある程度単体テストがすんだ時点で、ターゲットマシンに持込んで統合テストをした方が、早いだろうと思います。われわれの場合には、そのような仕事が多いので、さきほど述べたような環境を、除々に作りあげてきたわけです。

UNIX上には、パターン・マッチングなどのストリング処理に非常に強いツールが、数多くあります。また、yaccというコンパイラ・コンパイラがあるので、そうした環境が簡単に作れるという利点があります。実行形式のオブジェクトを出す必要がなく、シンタックス・チェックのためのプリプロセッサを作るだけなら、かなり手短かにできます。

この点についてもっと踏み込んでいうと、メインフレームの場合、その中のある特定のアプリケーションにき

わめて依存した形でソフトを開発しなければならないというのが、悲しい現実だと思います。

これからも、各社各様のワークステーションが出てくるでしょう。たとえば、A社の汎用機のソフトを作るには、A社のΣ-O Sが乗ったA社のワークステーションを買わないと仕事ができない、といった可能性が十分にありえます。

それは、開発者側にとっては、非常な負担になります。結局、A社の仕事を受けているかぎり、いつまでたってもA社の環境から逃れられない。つまり、せっかくの設備投資が、ある特定機種向けの開発環境に閉じこめられてしまう結果に終わるという危険性があるのです。

3. 環境とマネジメント

Y氏： 現在の開発現場の人間には、環境整備に対する自発的な気構えが足りないような気がするのですが。

深瀬： 環境を整備していく時に、最初にぶつかる障害は金の問題で、次は、全社的にやらなければダメだからしばらく見送ろうといった「いいわけ」です。

金の話は、利益の出ている部署は簡単ですが、出ていない部署ではむずかしい。これはあたりまえです。「環境整備をすれば利益が出る」といっても、「今利益が出てないのに金をかけても利益が出るわけがない」と拒否されてしまうでしょう。

わが社の場合は、社内の資料の検索、交通費、旅費の精算、会議室の予約などを、3つのビルにネットワークを張ってやっています。このための設備投資は、会社の間接経費のなかに入れてあります。ですから、開発現場がその経費を持たずに、自分達が使う分だけの経費を考えればよくなっています。

Y氏： 私は、ユーザの社内でアプリケーション開発のプロジェクトを担当しています。プロジェクトの要員は5～6名いますが、自分たちの努力で開発環境を改善しようとは考えていません。与えられた環境の中で仕事をして行くことに慣れてしまっており、管理されていなければ安心して仕事ができないようです。

深瀬： VAX11/780 (1MIPS) のTSSでは、たとえばGandalfのような(重い)環境は、UNIXを1人で占有することになるので、使えません。しかし、数MIPSの性能を持つパーソナル・ワークステーションを導入すれば、そんなことは問題でなくなります。しかも、最近では、そのほうがはるかにやすい。また、汎用メインフレームのマシンも、リプレースの中

古機ならタダみたいな値段で買える。いずれにせよ、ハードは非常に安く手に入るようになってきたので、自分たちが必要とする環境が簡単に作れます。

ソフトハウスの皆さんは、ハードを安く買うことにあまり興味をお持ちでないようですが、役に立つ開発支援ツールは、一般にCPU資源をよく食いますから、ワークステーションなどを活用して、そのことで他人に迷惑をかけないようにする。そういう環境作りを皆さんも考えたらいかがでしょうか。

また、若い人が開発環境に関して余り興味を持たないということも事実です。それは世の中にある他の開発環境を知らないのが原因で、自分がいま置かれている環境について、よしあしの判断がつかないからでしょう。

そういった意味からも、開発環境を整備にさいしてリーダーシップをとるのは、プロジェクト・マネージャの責任だと思います。ふつう、マネージャは、いろいろな雑用に追われていて、コーディングや設計を自分自身の手でやる時間なんかとれないでしょう。では、何をすればよいかといえば、部下のエンジニアたちがうまく仕事のできる環境をセットアップすることに力を入れるべきだと思います。

O氏： プロジェクト運営上のさまざまな管理業務は、どこまで機械化できるでしょうか。

深瀬： 皆さんは、どこまでやられていますか。

O氏： ほとんどやっていません。本人に報告させるのですが、あまり信用できません。たとえば、設計フェーズがこれだけすんだといっても、実はコーディングを行っていたりするケースが多い。むしろ、むりやり計ってしまう仕掛けを作った方がよいのではないのでしょうか。

深瀬： 私の会社でも、原価計算のために、開発の各フェーズごとの工数を調べていますが、残念ながらペーパーワークです。月に1回会社の会計監査のために作っているだけで、実際のプロジェクト運営には役立ちません。

ただし、プロジェクトがどのくらい進んでいるかは、週報が電子メールで飛び込んでくるので、すぐにわかります。ツールのどこにどんなバグがあるか、それを避けて通るにはどうすればよいか等の情報も、即座に電子掲示板で利用者にブロードキャストしています。紙を使うと回覧に時間がかかるので、コンピュータ・ネットワークを利用しているわけです。

プログラム自体の管理は、かなり機械化されています。そのためのツールは、原則として既存のものを使ってい

ますが、しかし、できあいのツールでは、なかなか現実の仕事の管理は思うようになりません。したがって、「ソフトウェア開発ツールは自分の手でメンテナンスすべきだ」というのが私の考え方です。

UNIXには、プログラムの世代管理のためのツールとして、SCCS、RCS、SPMSの3種類が用意されています。われわれはその3つを使いくらべてみて、RCSを採用することにしましたが、ちょっと大きめのDBソフトウェアなどは支えきれない。ですから自分たちで手直しして使っています。ある意味ではツールを進化させてやるわけです。ある程度素性のよさそうなツールは、このようにして使いこなさなければならないと思います。自分たちのしごとのやり方をツールに合わせるのではなく、ツールのほうを自分たちで使いやすいように進化させるのです。

O氏： ツールを修正する専任のスタッフがいるのでしょうか。

深瀬： おりません。何か問題が起きた時には、ツールを使っている人を集めて討論させます。そして、みんなで分担して手を入れます。専任を置くと、現場の声が聞こえなくなるので、よくありません。勝手な思い込みでなおされて、せっかく素性のよかったツールが悪くなってしまうたりする。

時には、ツールに手を入れるために、プロジェクトの工期が延びてしまうこともあります。それでもよいと思います。次にまた同じような仕事をする時に、役に立ちますから。

4. プロトタイピングについて

M氏： プロトタイピングの長所短所は何か。開発モデルはウォータフォールのままでよいのでしょうか。

深瀬： 一般にプロトタイピングの欠点は、ドキュメントがないことだといわれています。それに対して、ウォータフォール型の開発では、自動的に多量のドキュメントが残るようになっている。

しかし、実際には、結局意味のあるドキュメントは残りません。基本設計書を見てメンテナンスをすることはありえないし、では、詳細設計書はどうかといえば、これも役に立たない。結局、ソースをみてメンテナンスする羽目におちいる。

ウォータフォールは、大規模プロジェクト（長期間・多人数で、その間に人が入れかわること）を前提として考えています。人がかわっても、ドキュメントを生かそ

うという発想です。しかし、現実には人がかわったら、その人が何を考えていたのかは、結局わからなくなってしまいます。

この意味では、プロトタイピングとウォータフォールとのちがいは、紙一重ではないかと思います。

プロトタイピングのよい点は、1人のエンジニアのアイデアにもとづいて、それを具体化するまでにいろいろ実験をくりかえしていくので、内容が一貫していることです。つまり実験というかたちで、論理設計を行っているのです。

また、変更がきた時でも作り直すのが苦になりません。いちいち基本設計書に戻って修正し、詳細設計書も直し、両者の整合性をチェックしてからコーディングをするのとくらべて、大幅に工数を減らせます。しかし大規模なプロジェクトはそれを許してくれません。

プロトタイピングの欠点は、標準的な物の作り方が存在しないことでしょう。つまり複雑で大規模な物は作りにくい。しかし、われわれは、それよりもメリットの方を重視して、プロトタイピングを採用しています。

M氏： ドキュメントもないプロトタイプが商品になり得るのでしょうか。

深瀬： UNIX上のソフトには、ふつう、仕様書などありません。アイデアを書いた簡単なメモとソース・プログラムだけです。それで、十分に商品として通用しています。こうした考え方のお陰で、UNIXの世界では、使いやすいソフトがどんどん出てくるのだと思います。

ただし、ユーザがそうした考え方を受け入れてくれるかどうかは、また別の問題です。特に、旧来の開発手法しか知らないお客の場合は難しいでしょう。まずは、社内の開発環境作りに使うのがよいと思います。

われわれは4.2BSDでソフトを作っています。しかし、客がシステムVがいいといえば、システムV用のソフトも作ります。工作機械つまり開発環境と商品とは、まったく別のものだと考えています。

5. 環境への投資が重要である

T氏： 進捗の把握を、時間（期間）の両面から定量的に行いたいのですが、何かよい方法はないでしょうか。

深瀬： いくつか考えられますが、私の経験では、意味のある数値が得られたことはありません。たとえば、「あのプロジェクトはどうか」と聞くと、「順調です」と答えが返ってくる。しかし、最後に行って「実は....」と

なってしまう。これは、初めに見えなかった部分に、ただだけの工数がかかるかわからなかったためです。

特に、パソコン関係の仕事では、ハードもソフトも見えない部分がほとんどなので、工程管理は最初から放棄しています。汎用機のソフトはロジックの大半が見えているので、ある程度のマイルストーンは設定できるでしょう。

T氏： おっしゃる通りで、途中で聞くのと最後の方で聞くのと話が違ふことが大半で、どうしてよいのか困っているのですが、何か知恵はないものでしょうか。

深瀬： 見通しのないプロジェクトの工程管理などできないとお答えするしかありません。しかし、工程の管理はできなくても、工数を減らすための環境の整備は可能です。初めに見えなかった部分で遅れても、環境の力で素早くリカバリーできる。そのための投資が重要だと考えています。

たとえば、ワークステーションを導入するとか、ネットワークを引くことなどです。見通しのきくアプリケーション開発の場合にも、工数を短縮し、開発コストを下げる上で、環境投資は役立ちます。

T氏： 最後に各社のワークステーションの特徴や選び方についてお聞きしたいのですが。

深瀬： われわれがワークステーションを選ぶ時の基準は、価格、ネットワーク、OSの3つです。

一番先に考えるのは、もちろん価格です。技術の進歩や需要の増加に応じて、価格は変動します。昨日までのベストセラー商品が、価格・性能の両面で他社の新製品に追い落とされ、明日からは全然売れなくなるといった状況は、ワークステーション・ビジネスの世界では日常茶飯事です。そのあたりの見極めが大切です。

アメリカでは、SUNみたいに、日本的感覚からすれば高価なワークステーションが、何万台も出荷されています。エンジニア1人の年俸より安ければ、1台入れてもよいという計算のようです。つまり、あちらの経営者は、それなりのワークステーションを導入すれば、生産性は自動的に向上するものと楽観的に考えている。

日本では、なかなかそうはいきません。ワークステーション1台あたり生産性が何%上がるのか、見掛けだけはきちんとした積み上げ資料を用意しなければならない。まあ、2-3年もすれば、ワークステーションの値段も、いまのパソコン位にさがり、そうすれば爆発的に普及するでしょう。

ネットワークもまた、きわめて重要なファクターです。どんなにハードが高性能で、よいソフトがそろっていても、特殊なネットワークの中に閉じているワークステーションは、外の世界と接続するときに、そのためのゲートウェイを作らなければならないので、かえって高くなります。その点、SUNのように、オープン・システム・アーキテクチャを採用している製品は、ユーザの立場からすれば安心だという結論になります。

特定のハードウェアのためにチューニングされたOSはよくありません。パフォーマンスは多少落ちて、自分でいろいろ手を入れられるOSであることが絶対条件です。現時点では、UNIXしかないというのが、結論になります。

T氏： 国産で安いワークステーションがいくつか出はじめましたが、どうお考えですか。

深瀬： 今日はメーカーの方もいらっしゃるのですが、ホンネがいいにくいのですが（笑い）、一般論として、国産ワークステーションはあまりおすすめできません。

アメリカのワークステーション・メーカーは、新しいコンセプトの設計に対して、すぐれた人材を大量に投入しています。開発投資もかなりのものです。その結果マーケットで多くの実績を持っていますが、これだけ台数が出れば、ソフトもハードも洗練されるのはあたりまえのことです。

しかも、ハードもネットワークもオープンなアーキテクチャになっているものが多く、いろいろな付属装置をユーザが自由に選んで、サード・パーティーから購入できます。ディスク・コントローラでも何でも、全部サード・パーティーからマーケットに流れているので、それだけ信頼性も高いのです。

それに比べて、国産のワークステーション（特に汎用メインフレームの製品）は、自社の閉じた世界の中で作られていることが多く、何か問題が起きた時に救いようがありません。その意味で、あまりおすすめできないのです。

6. おわりに

後半の討議も盛り上がりの内に、残念ながら時間が来てしまいました。当セッションの講議・討論を通し、自分の周囲のソフトウェア開発環境を真剣に見直し、改善を進めて行くことの重要さを、あらためて再認識させられました。

SEA 秋のセミナー・ウィーク

セッション G1

システム監査と EDP 部門

講 師： 吉村 成弘

報告者： 佐原 伸

1. システム監査とは

1. 1 定義

システム監査とは何かといえば：

「システム監査は、監査対象から独立したシステム監査人が、情報システムを総合的に点検・評価し、関係者に助言、勧告するものであり、セキュリティ対策の実行性の担保及びシステムの有効活用を図る上で効果的手段と考えられる。

システム監査は、システムの信頼性、安全性、効率性を高め、よって情報化社会の健全化に資することを目的とする」（通産省「システム監査基準」より）

といったところが一応の定義ですが、今日は、効率性の話は省略し、コンピュータ犯罪に焦点を当てて、お話ししてみたいと思います。

システムの安全性の観点からみると、コンピュータ犯罪の報道では、ほとんどの場合肝心なことが抜けています。つまり、どうやったかがわからないのです。今日は、このあたりのことを紹介しながら、システム監査について、皆さんと一緒に、考えていきたいと思っています。

ところで、通産省はシステム監査の制度化を考えていますが、本来、システム監査は、誰かに押し付けられて行うものではなく、企業が自ら行うべきものであろうと考えます。本日の話は、おもに米国での事例が中心になりますが、米国では、EDPAA（EDP Auditor Association）という協会があり、8大監査法人はそれぞれ独自の監査パッケージを使用して、監査を行っています。

1. 2 必要性

システム監査の必要性としては：

- ・ 会計監査の前提条件として
- ・ コンピュータ犯罪の防止
- ・ コンピュータの有効利用
- ・ プライバシーの保護

などが考えられます。

システム監査を正しく行っていないと、監査法人に責任が生じる可能性があります。米国では訴訟になる可能性があり、実際、エクィティ・ファンディング事件（後述）では、このような訴訟が起きています。

このため米国におけるシステム監査は、「監査をやった証拠を作るのが目的になっている」という傾向が見られます。エクィティ・ファンディング事件では21億ドルの粉飾があり、97,000通の保険証券の3分の2が架空契約であったにもかかわらず、監査法人は無罪になりました。3枚中2枚の証券が偽造だった訳で、まともな監査をやっていたら発見できたはずなのですが？

2. わが国の実情

2. 1 わが国における問題の取り上げ方

以下に示すような機関が、「システム監査」を問題として取り上げています：

- ・ 通産省
- ・ （財）日本情報処理開発協会
- ・ 日本公認会計士協会
- ・ ユーザー団体

2. 2 実施状況

わが国における「システム監査」の実施状況については、雑誌「データ・マネジメント」の昭和60年7月号に、警察庁が調査した結果が報告されている。それを次ページの表に示します。

実施しているというのが、全体の16パーセント弱というところに、今後とも多くの問題をはらんでいるように思えます。

実施している	183社	15.9%
社外専門家による	(82)	(7.1%)
社内専門家による	(101)	(8.8%)
実施していない	827社	71.9%
その他	101社	8.8%
無解答	40社	3.5%

3. 米国での歴史

3.1 エクィティ・ファンディング事件

この事件は、EDP部門に分からないように、保険計数部で架空証券作成プログラムを作り、97,000通の保険証券の3分の2以上が、架空契約だったという衝撃的な事件だった。21億ドルの粉飾事件として有名です。

架空証券はコンピュータで作成され、コードで識別されていました。会社は倒産し、先程お話しましたように、監査法人が告訴されましたが、無罪となりました。

3.2 始まり

1950年代にその萌芽が見られ、1960年代に入ってその認識が高まり、1970年代にほぼその考え方や手法が確立しました。だいぶ古いものですが、以下のような参考文献があります：

- ・ P. Waterhouse, Audit encounters Electronic Data Processing, 1956.
- ・ W. S. Boutell, Auditing with the Computer, 1964
- ・ 監査業務とEDP (日本経営出版会)

4. コンピュータが企業に与えたインパクト

4.1 手作業時代の事務処理

会計の世界では、物と帳簿を分けるのが原則でしたが、コンピュータ時代で、この原則が問題になってきました。伝票と帳簿のこれまでの役割は、ふつう、次のようになっています：

伝票： 組織を横断して情報を伝達する。

帳簿： 伝票から転記、分類・集計を行う。

4.2 コンピュータによる事務処理

ところが、コンピュータ時代になって、省力化のために短絡化現象が発生し、組織を横断して伝票が流れるあいだに幾重ものチェック（内部統制）がなされていた手

作業時代に比べて、かえって危険性が大きくなりました。

4.3 内部統制に与えたインパクト

(a) 内部統制の定義（狭義）

狭義の内部統制は、以下のように定義できると思います：

「財産を保全し、会計記録の正確性と信頼性を確保するための組織計画、手続き及び記録から構成されている」

(b) 会計業務に与えた変化

会計システムがプログラム化され、会計業務が現業システムに埋没してしまった。このため、会計部門の牽制機能が低下しています。

(c) 対象部門（インプット部門）の問題点

事務能力の低い末端からインプットされ、手作業段階でのチェックの節が消滅したために、問題がでてきました。手作業時代はチェックの節毎に、人間が意味的なチェックを行うことが可能だった訳ですが、コンピュータではこのようなチェックがかなり困難です。このあたりで、AIといえますかエキスパートシステムが、システム監査にとっても、重要な意味あいを持って来るだろうと思います。

(d) EDP部門の問題点

以下に示すような問題点があります。

- ・ データ処理が電算室にすべて集中する
- ・ 能率を第一とする精神風土がある
- ・ プログラムの不備に乗ぜられると非常に悪い
- ・ オペレーション段階での介入の可能性がある
- ・ ドキュメンテーションが整備されていないケースが多い
- ・ 修正、改ざん、架空取引の挿入などが証拠を残さずにできる

最近のパソコンのシステムには、このような全体の整合性を保った修正が容易にできるものが多く、脱税などに流用される危険が大きいようです。

(e) システム全体のコントロール

システム全体のコントロールをしているのはコンピュータであって、人間は部分的にシステムに関与しているだけになっています。さらに、人間が部分的にしか関与できないため、知識が目に見えなくなって、全体を見渡せる人間の養成にも支障をきたしています。

4.4 コンピュータ犯罪

(a) ウェルズファーゴ銀行事件

特別な技術・知識を持っている訳ではない2人の人間（オペレータと詐欺師）が、銀行の本支店間で金をクライ回しし、3,000万ドルを横領した事件です。つまり、自分達の口座に金を振り込み、検査の前に支店から振り込まれたような操作をします。この振り込みがばれそうになると、今度は本店から支店に振り込む訳です。

これは本支店取引に関するシステムのチェックが甘かったことに起因しています。チェックは10日毎にしかありませんでした。このため犯人達が一度成功すると、何回でも同じ手口で横領できたわけです。

(b) ハーロウの犯罪

農産物の販売を行っている会社の、システムと経理の両方の責任者を兼ねた人間が、システムを改造して会計士も感心する程の帳簿をコンピュータで作っていました。会計士はそのシステムの販売を勧めたほどだったそうです。しかし実際には、会計士が来る前日に偽造帳簿をコンピュータで打ち出していた訳ですが、こんなことは、従来の手作業のシステムでは考えられなかったことです。

(c) 日本の銀行の例

女性行員が歯医者に行くといって出て行ったきり、戻らなかった。昼のチェックで1億8千万円の誤差がでたが、銀行は打ち間違えと思って彼女の帰りをまったのです。が、実際には、4つの支店に振り込んだ金を横領して海外に逃亡していた訳です。

この事件で巧妙なのは、1億8千万という数字が、例えば2億3千万円の3と1を打ち間違えて23億1千万円と打ってしまったと推理できてしまう点です。専門家ほど引掛かる手口です。各支店からの引き出しも、全額おろすと疑われるため、一部を残していました。この事件は銀行の第3次オンラインで、現金勘定のリアルタイムのチェックができれば防げるようになるかもしれません。

(d) 内部統制を厳格に守ってもまずい例

これまでの例は、内部統制を厳格に行っていれば、何とか防ぐことができた事件なのですが、内部統制を逆手にとった犯罪も起こりました。

倉庫に出荷指示票がまわされ、A社に納入します。その分在庫が減るのでB社に注文をだし、B社から品物と請求書が届きます。しかし、A社への請求はいつまでも発行されないように、システムが改ざんされています。

ここでA社とB社は、システムを改ざんした人間のダミーとなっています。この場合、手作業時代の分業によって、犯罪を防ごうとする考えが通用しなくなります。つまり、今の場合、各担当者は規則通り仕事をしている訳です。しかし、不正は棚卸しのときまで発覚しません。

(e) ジェリー・シュナイダー事件

パシフィック電気が捨てたシステム仕様書を高校生が拾い、全部揃えました。分からないところは、新聞記者を名乗って取材したのです。彼は、システムの全体を理解し、システムにアクセスし、自分がほしい品物を出荷するよう指示したのです。

この事件は、会計上はほとんど完全犯罪です。しかし、注文を受ける媒体が電話であり、受けた人間が判断すれば、あまりにも変な注文であること（注文の単位が少なく、品物の組み合わせがおかしい）が分かったはずですが、この点で、エキスパートシステムの活躍する余地がある訳です。

この事件は、高校生たちの仲間割れで発覚しましたが、当事者のジェリー・シュナイダーは、このことを本に書いてベストセラーとなりました。

(f) 刑罰

コンピュータ犯罪は罪が軽く、企業側もあり告訴したがりません。また、犯罪の立証も困難な訳です。たとえば、脱税をプログラムのミスにするとすれば、その立証の困難さはおわかり頂けるでしょう。

5. 参考文献

これまでお話ししてきたことの詳細については、以下にあげる文献を参照してください。

- ・EDPシステムの内部統制 日本公認会計士協会編 第一法規出版（株）
- ・システム監査の基礎知識 鳥居壮行著 日本生産性本部
- ・システム監査ガイドライン 監査人協会東京支部編 日経マグロウヒル社

第4回夏のプログラミング・ワークショップ報告

松尾 正敏

ソフトウェア・リサーチ・アソシエーツ

1. はじまり

「夏のプログラミング・ワークショップ」—正式名称(?) : 若手の会 in 盛岡—も4回目を迎えました。

その昔、ソフト協技術委員会時代に、ワークショップというかたちの技術交流活動を始めたばかりのころ、「これから何年か先にはこの業界の中心になるであろう20代の技術者を対象としたワークショップを開いては?」と話していたのが現実になり、めぐりめぐって、自分自身がプログラム委員長をつとめることとなりました。

このワークショップのねらいは、鎖国会社(社会ではない)の弊害を若いうちから取り除こうという、中年・熟年技術者の気配りからでています。もちろん、議論の中身も大切ですが、日本のソフトウェア・コミュニティに電子ネットワークが真に定着するまでに、もう少し時間をかけて、若手技術者の輪を広げておく必要があるのでは、と私自身も考えています。

さて、今年の若手の会は、夏も終わりの9月10日—13日の4日間に、例年通り盛岡で開催されることとなり、企画運営スタッフも自然に(?)次のようにきまりました:

実行委員長

堀江 進(日本電気ソフトウェア)

プログラム委員長

松尾 正敏(SRA)

プログラム委員

臼井 義美(日本電子計算)

熊谷 章(パナファコム)

野村 行憲(岩手電子計算センター)

深瀬 弘恭(アスキー)

森田 政敏(神戸コンピュータサービス)

事務局

辻 真理子(日本コンピューター・システム)

中島 千代子(ソフトウェア技術者協会)

今年の討論テーマは「プロジェクト・プランニングのケース・スタディ」でしたが、「テクニカル・マネジメントを中心に」というサブタイトルをわざわざつけたのは、これまでプロジェクト管理を主題にしたワークショップ

が、ともすればグチのこぼしあいに終始してしまったことへの反省がこめられています。

招待講演に関しても、これまでと趣を変えて、この業界以外の人を選ぼうということで、演出家、建築家、弁護士など、いろいろな候補者が浮かびました。結局、盛岡のローカル色を全面に打ち出そうということになり、地元の野村さんをお願いして、郷土の歴史に詳しい照井さんと建築家の久慈さんをお招きすることになりました。

プログラム委員が頭を悩ます以前に、不思議なことに、参加募集はすでに始まっており、何人かの人からはポジション・ペーパーが送られてくるという状態で、募集締め切りの日が迫ることとなりました。

2. 楽しみと悩み

参加募集の締め切りは7月15日でしたが、ほとんどのポジション・ペーパーは、締め切り前の1週間に殺到するというありさまで、この業界における「納期」の概念を如実に反映していました。納期遅れもいくつかありましたが、全体としての比率は少なく、一昔前に比べて、仕事を納期どりに何とかこなしてしまう習慣が定着してきたのだな、と感心した次第です。

結局のところ、申込は27社から55人にもなり、定員は25人なので、さあどうしようということになりました。もし、計画がち密に立てられていたら、30人の方々に「残念でした」と伝えねばならなかったのですが、そこは、ソフト協の時代からの伝統で、「どうにかなるさ」と、まるでいつものプログラム開発の場合と同様に気楽に計画変更してしまい、最終的には32人を選抜しました。これにスタッフの8人が加わり、公式参加者は40名と決定したのが8月の下旬頃だったと思います。

ちなみに歴史(JISA/STCレポート)をひもとくと:

第1回 . . . 47名

第2回 . . . 50名

第3回 . . . 61名

もの人間が集まっており、ワークショップの「適性規模」をはるかに越えていたのですが、今回は最適ラインに若干近づくことができたようです。

参加希望者が増加し、かつ、ポジション・ペーパーの内容が充実することは、望ましい現象ではありますが、反面、人数の制限によって参加できない方が生ずるのは残念なことです。それでも、現在29才以下の方でしたら、来年もまたチャンスがありますので、ぜひ申し込んでいただきたいと思います。

3. いよいよ始まり、そして

ホテル東日本での初日のセッションは、オリエンテーションと堀江実行委員長の基調講演で幕を明けました。納期を守ることに忠実な参加者の皆さんは、時間通りに集合した安堵感からか、はじめてこの種の会合に出席する緊張感からか、新幹線の旅の疲れのせい、はたまた、昼食後の心地よさのせい、会場には異様な静けさが漂うひとときでありました。別の見方をすれば、プロジェクト管理データの収集と分析で有名な堀江さんの話を一言も漏らさずに吸収しようとする使命感にも似た食欲の現われであったのかもしれませんが。

次は、いわゆるプレイング・マネージャである5人のグループ・リーダー（つまりプログラム委員）が、実際に「プロジェクト・プランニング」をどのように行っているか、私生活、いや、会社での仕事ぶりについてそれぞれの実態を明らかにするパネル討論となりました。これは、グループ討論の基本方針を明らかにする目的で行なったもので、参加者がどのリーダーと残りの2日半の運命を共にするかを決意する手がかりにさせていただきました。

まあ、5人が5人とも、さすがに年季の入っていて、もっと時間があつたらと思うほどに密度の濃い意見が数多く飛び交い、思わず感心させられました。何かの機会に、また同じ組み合わせでパネル討論を行なうと面白いのではないかと感じつつ、時計の針を恨めしく見ながら討論の幕を閉じました。

初日の終わりは、ワークショップ恒例の自己紹介を延々と2時間半ほど続け、途中から話のつまみに飲みものと食べものを加えて、参加者相互の出会いの緊張を多少なりともほぐすことができたようです。やはり、日本人の証明とでも言える謙虚さ（人見知りの癖）は、そここに現われてはいましたが、新人類よりも少しばかり年をとった20代の人々は、自己表現がなかなか上手なようで、ユニークな自己紹介も多かったのです。

こうして最初のプログラムも時間の経過とともに消化され、盛岡の夜の静寂さが戻ってきました。

4. 集合と離散

2日目と3日目は5つのグループに分れ、

・実際に経験したプロジェクトを持ち寄り、それらをもとに各グループで討論の対象とするプロジェクトを定義する。

・このプロジェクトを始めるに当たって、どのようなプランにするか

を検討し、プランを作ることを目標にしました。

企画・運営サイドの人間にとって、ワークショップの楽しみの一つは、どのようにグループが形成されて行くかを観察することです。

今回は、グループ討論の場所がかなり広い会議室だったので、そこに5つの島を作り、あらかじめそれぞれの島の所有者（グループ・リーダー）を決めておいて、席取りゲーム方式で分けることにしました。すると不思議なことに、バランスよく5つのグループができあがり、なんの調整もいらないという事態に、何か不吉な予感を感じたものです。こうして、グループ・リーダーとその仲間たちも決まり、各自持ち寄ったプロジェクトの説明から、グループ討論が始まりました。

2日目の午前中はどんなプロジェクトを想定するかについて討論し、午後は、各グループからその結果を報告し、他のグループの人々がいろいろ意見を出すかたちで討論を進めました。何といてもハイライトは、「地上観測ロケット打ち上げ用システム」プロジェクトであり、「本当に打ち上げができるのかな？」とか「打ち上げの後始末」プロジェクトの提案などがあり、会場に和やかな雰囲気を出すように努力したグループ・リーダーの方々の心配りに、感動を覚えたわけです。

夕方のプログラムは、楽しみにしていた照井氏の講演で、「みちのく夜話（日本人の心のふるさとを語る）」と題して、遠く縄文・弥生時代まで遡り、江戸時代そして現代と、岩手県が「みちのく」である所以について、楽しく語っていただきました。

講演の後に起こったできごとは、ただただ「！」マークを何個か並べなければ、表現しようのないことでした。時間の使い方は各グループの意志に任せていたのですが、2つのグループが簡単な打ち合わせの後、夜の町へ食事に去ってしまったにもかかわらず、残りの3つのグループは、食事の時間も惜しんで討論を続けていたのです。部屋の半分には人がいて、もう半分にはテーブルと椅子しかないという光景が、今でもまぶたに浮かんできます。

さて、「この原因は？」と考えてみると、答えは一つ、なんとグループ・リーダーの血液型の違いだったのです。その2つの血液型が何であるかは想像にお任せすることにして、2日目も無事夜が訪れました。

3日目の朝、この日も天気はよく、みちのくの太陽は期間中ずっと、プログラマたちを歓迎してくれました。

しかし、とうとう事件が起きました。あるグループのリーダーである熊谷さんが、「夜8時には戻ります」との置き手紙を残し、急用のため早朝の新幹線で東京に帰ってしまったのです。突然のことで、残されたグループの若者たちには多少の戸惑いはあったようですが、上司をあてにしないのがこの業界の常識なのでしょう、このグループのテーマとしていた「火消し」を見事に実践していました（深夜に、最終の新幹線で、熊谷さんは戻ってきました。新幹線はたしかに便利ですが、災いのもとでもあるようです）。

この日の夕方には、盛岡にある多くのビルの設計を行っている、建築家の久慈氏に、建築の世界での才能の引き出し方や、ソフトの世界との違いなどについてお話しをいただきました。

プログラムの上では、グループ討論の時間をたっぷり取ったつもりでしたが、実際にやってみると議論はつきないもので、グループによっては議論が不完全なままに終わったところもあったようです。しかし、初めて顔を合わせた者同士にしては、限られた時間内でかなり突っ込んだ意見の交換ができたものだと感心させられました。盛岡という地の利を活かした討論方法も、それぞれのグループごとに、いろいろ（静かな会議室で夜遅く迄とか、自然の中で発想の転換をはかったりとか）工夫していたようです。

こうして、2日間にわたる討論も終わり、各グループは最終日に向けてまとめを行なったのち、夜遅くまでフォローアップのミーティングを、市内のあちこちで続けていました。

5. 最後の集まり

第4日午前中の最後のセッションでは、これまでの討論のまとめを、各グループから発表してもらいました。それぞれのグループ内での討論は、かなり盛り上がったはずであるにもかかわらず、この全体討論ではあまり質問や意見が出ず、どちらかといえば静かに終わったのが、奇異な感じもしました。当初は、パネル討論の形式をとるつもりだったのを、これまでどうりの発表形式に

したことで、かえって盛り上がりを欠いてしまったようです。

それでも各グループの発表は、趣向を凝らしたものや、長すぎる（発表時間が）ものや、少々議論不足気味のものなど、結構バラエティに富んでいて、聞いている方は楽しめました。内容の詳細は、別途発行されるワークショップ・レポートを読んでいただくことにして、この報告もこのあたりで終わらせていただきます。やはり、慣れない原稿書きで、後半は疲れのせいかわくだけになってしまいました。

6. 蛇足（鬼を笑わせる話）

今回のワークショップで、もっとも大変だったのはプログラム委員の方々と、準備のために高価な頭脳と肉体を無償提供していただき、ありがとうございました。この場を借りて御礼申し上げます。

いまから考えると、企画がかなり成り行き任せのところもあり、事前のミーティング回数が少なかったような気がします。次回は、少なくとも6ヶ月前ぐらいにプログラム委員会の編成を終わって、時間に追われることなく準備を行なうことをおすすめします。

また、一番問題になるのがテーマの設定ですが、今回は、テーマが先にあって、それからスタッフが決まりました。「プロジェクト管理」はテーマとしては誰もが興味を示し、かつ、若手の人々にとっても身近な問題ではあります。しかし、今回のテーマが果たして目的を得たものであったかどうかは、残念ながら、未だに判断が付きません。来年は、この討論テーマの設定の問題についても、プログラム委員会で十分論議をしていただきたいと思います。

ワークショップというイベントは、ボランティア活動を前提とするSEAにとって、ある意味ではもっとも重要な活動であることは間違いありません。今後、若手の会に限らず、たとえばそれぞれのSIGが運営するワークショップなど、バラエティに富む企画が次々に出てくることを期待します。

とはいえ、SEAの活動はまだ始まったばかりです。ボランティアの絶対数が不足していることは否定できません。今回のワークショップの参加者の中からも何人かのアクティブなボランティアが生まれ、それらの人々の企画運営による若手だけの「若手の会」が近く実施されることを祈りつつ、筆を置きます。

ソフトウェア開発環境としてのメインフレームTSS

田中 慎一郎

協同システム開発

1. はじめに

SIGENVの8月例会では、「最近のメインフレームTSSの動向」というテーマで、日立製作所、日本電気、富士通の各社の方にお話していただき、参加者との討論の場を持ちました。

今回はその模様を、私が受け止めた範囲で、簡単にご紹介しようと思います。当日の発言者のなかで、そういう風には言わなかったとか、発言の主旨がちょっとちがうとかいう異論のある方、また、当日出席されなかった方で、討論内容に関して賛否いずれかの意見がある方は、どしどし誌上にコメントまたは御意見をお寄せください。

2. 各社からの環境の紹介

2.1 日立製作所

日立製作所からは、新対話システムの核である対話型環境基盤製品：

MODE1: Multi-media Open-ended Dialogue
Environment 1

および総合形テキスト・エディタ：

ASPEN: Advanced Editor System for
Programming Environment

を中心に説明があった。

MODE1はこれまで2層であったプログラム群（システムレイヤ、アプリケーションレイヤ）の中間に位置するミドルレイヤである。それは、いわば対話の部品化をねらったもので、複数のアプリケーション・プログラムを連結させるための機能（アプリケーション・プログラム間連結機能）と、対話に使用するパネルを制御する機能（パネル制御機能）の2つを軸として、各種メディア（グラフィックス、ビジネスグラフィックス、画像、表データ）を処理するための共通機能を、ひとつにまとめたものといえる。

エンドユーザは3種類のパネル（メニューパネル、データパネル、ガイドパネル）を通じてコンピュータ内の処理と対話が可能であり、また、データの論理化、抽象化がすすめられているため、単純化された形でデータを取り扱えるようになっている。具体的には、基本的な制御をつかさどるための

MODE/BS(MODE/Base)

グラフィックスを処理するための

KGRAF(Kernel Graphics Function)

BGRAF(Business Graphicsd Feature)

対話環境の基本要素を対話形式で定義するための

MODE/PNL(MODE/Panel Service)

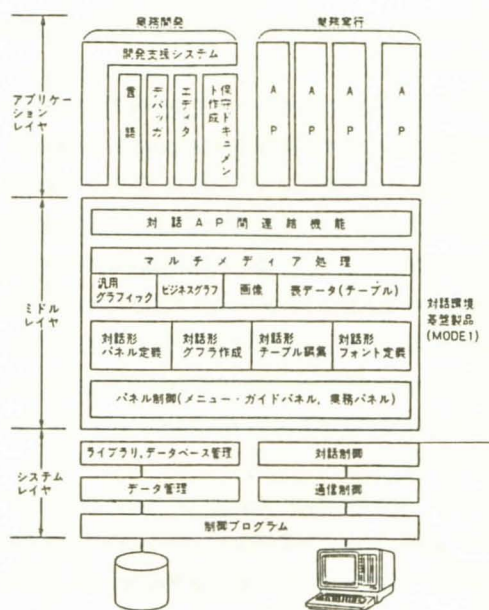
MODE/TBL(MODE/Table Service)

MODE?GRF(MODE/Graphic Service)

MODE/FNT(MODE/Font Service)

がある。

新対話システムの構成



ASPEN は、プログラム開発作業を総合的に支援することをめざしたエディタである。

特徴としては、

- ・画面コマンドと行コマンドの両方が使える
- ・同時に複数個のプログラムを表示・編集できる（8ファイルまで）
- ・正規表現が使える

- ・ UNDO がある
- ・ 画面分割、ページマップによるスクロール、プログラム内の特定行の固定表示ができる
- ・ 編集を保留にしたままで、コンパイルやコピーなどができる
- ・ コンパイラと連携できる（自動的にエラーメッセージ領域がとられたり、エラー行の色が変わったりする）
- ・ TSS コマンドが自由に活用できる

などがあげられる。

その他の使い勝手として、

- ・ 画面はツリー構造をしているが、慣れれば直接飛んでいくこともできる。
- ・ ヘルプ機能が充実している。
- ・ 出力リストはスプールにためておいて編集することもできる

なども付け加えられた。

2.2 日本電気

日本電気からは、比較的一般的な話として説明が行われた。

まず TSS への要求としては、

- ・ エンドユーザ指向
- ・ 利用／操作の容易性
- ・ 開発／保守支援ツールの整備
- ・ 豊富なプログラミング言語
- ・ データベース利用環境

などがあるが、最初の2つにあらわされているように、プロ向きから一般向きへの傾向が顕著である。

TSS EXECUTIVE の機能分割サブシステム構成は、右下の図に示すようになっている。

コマンド体系は、

- ・ 利用者カタログ／ファイル保守
- ・ コマンドプロシージャ
- ・ デファード処理
- ・ 対話型リモートバッチ処理
- ・ デバッグ／トレース
- ・ テキストエディタ編集／支援
- ・ プログラミング言語
- ・ データベース操作／検索

となっており、特にデバッグ／トレースはエンドユーザによる開発を意識している。

テスト・デバッグサポートとしては、

- ・ TSD/T
- ・ RBUG
- ・ ATAC
- ・ DSP

があり、ソースおよびオブジェクトのデバッグが行えるが、マルチメディア対応を意識している。また現在問題となっているのは、開発過程の情報をどう持つか、ドキュメント等にどのように反映させるかという点である。

テキスト／データ支援としては、

- ・ 一括入出力
- ・ 行エディタ
- ・ テキストエディタ
- ・ 日本語エディタ
- ・ Runoff
- ・ Beautifier

があり、ドキュメンテーションを意識したものとなっているが、アंकと日本語の混在はいまのところできない。

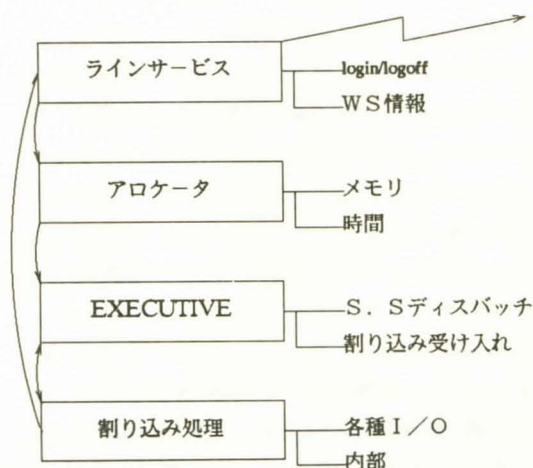
言語に関しては、JCL なしで実行できる環境をめざしている。

またデータベースとしては、

- ・ INQ
- ・ IDS
- ・ ADBS
- ・ RIQS

があるが、ACOS 6 と 4 とで一元化されている。

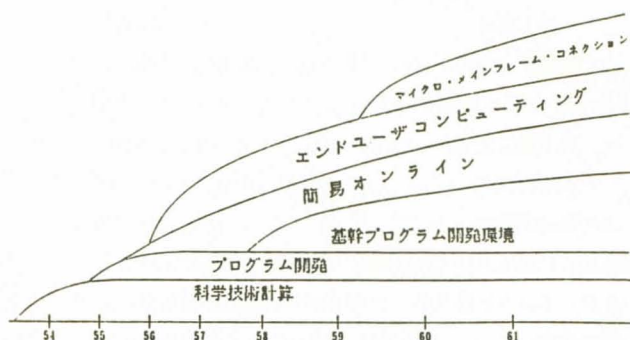
TSS EXECUTIVE
機能分割・サブシステム構成



2.3 富士通

富士通からは、TSS 利用形態の推移（一般的な説明）のあと、オンライン・データベース・システム（AIM）と TSS との関係について説明が行われた。TSS 利用形態の推移を、下図のようにしている。

TSS 利用形態の推移



富士通では TSS の簡易性を評価し、AIM (Advanced Information Manager) で動くシステムを TSS でも使えるようにするとともに、APS (Advanced Printer Service) と組み合わせて統合化している（とかく一緒に集めると、統合といったがのですが、との発言には共感をおぼえた）。

APS はセンタプリンタとリモートプリンタを同じイメージで使えるようにしたものである。

テスト環境としては、ITS (Interactive Testing Support Facility) があり、本番環境と分離された開発環境を提供している。すなわち、開発時には簡便な TSS を利用し、ITS でテストを行い、必要であれば、本番稼働時に AIM 環境に移行するわけである。

以上は、主に基幹システムを構築するための仕掛けであるが、エンド・ユーザ・コンピューティングのためには INTERACT がある。

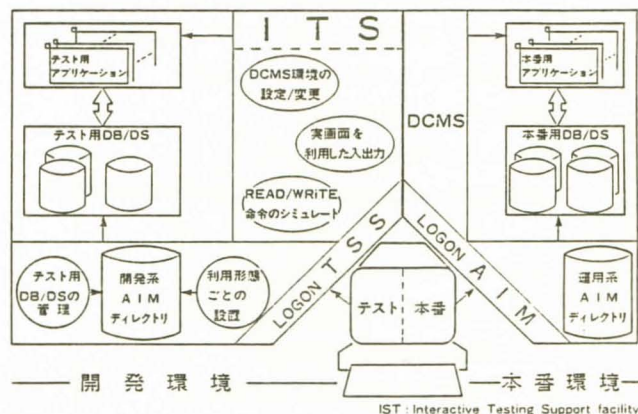
INTERACT は、事務作業の中の帳票作成処理、グラフ作成処理、簡単な計算処理から、企画/計画部門の専門的な統計・予測処理まで、エンドユーザが自ら利用できるツールを数多く備えている。メニュー・ハンドラを利用して画面を作り、コマンド・プロシジャで制御を

行くと、簡単にプロトタイプを作れることもできる。

現在は、マイクロ・メインフレーム・コネクション (LINKDUST) として会話型データ交換を行えるようなホスト〜パソコン連携システムへ力を入れている。すなわち、従来のホストシステムでは、データ処理からメニュー/グラフ出力まで、すべての機能をホストで行っていたが、会話転送を利用してマン・マシン・インタフェース部分をパソコンで実現することにより、ホストを、大量のデータ管理や高速処理などの、本来のデータ処理に専念させようというわけである。

本番環境と分離された開発環境

—開発系 AIM ディレクトリと ITS—



3. 質疑および討論

後半は3社の方への質疑を軸とした討論となったが、皆さん、個人の立場で発言していただいたので、以下では社名・人名は省略する。

A: 最近感じるのは、ソフトウェア開発環境が、どんどん「一般向け」あるいは「素人向け」の方向に進んでいるのではないかと、ということです。われわれソフトウェア・ハウスで働くプロフェッショナルにとって、メインフレームの世界はどうなっていくのだろうか。たとえば、あいかわらず徹夜を続けているプログラマにとって、何かいいことがあるのでしょうか。

X: 別にメインフレームに固執する必要はないんじゃないかな。それよりも、メインフレームとワークステーションの役割分担をすすめて、マン・マシン・インタフ

エースの向上をはかるべきでしょう。

Y: われわれプログラマにとっては、ワークステーションでも、メインフレームと同様のことができればすむのですから、実帳票・実画面イメージが得られるような環境が、ワークステーション上に作られることになるでしょう。

そして、メインフレームの TSS はその他の使い道、たとえば科学技術計算や、データベースその他の資源の共有、バカチョン式のユーザ・インタフェイスという利点を生かす方向に向かって行くでしょう。

Z: 単純作業で行けるものは、半自動化して工数を削減するのも一つの方向ではないでしょうか。エンドユーザにできることは、自分自身でやってもらうようにするだけでも、われわれの仕事はかなり楽になります。こうしたことは、ツールを充実させることで可能になるでしょう。ワークステーション化という方向も、もちろんあります。

A: 現状をみると、あまりにも手工業的なソフト開発のやりかたが多い。ダメな人には、いかにいいツールを与えてもダメだ、という経験的事実があるので、逆に、できる人の生産性をより高める道を考えたらどうでしょうか。たとえば、拡張性の高い、モディファイしやすいツールなどが考えられますが。

X: TSS が使いこなされるようになってきたのは、せいぜいこの5年ぐらいであって、登場からはだいぶ遅れがある。したがって、いまあるツールなども、一般に使いこなされるようになるには、だいぶ時間がかかるのではないのでしょうか。

拡張性のあるツールには興味はありますが、そのようになっていくのは大変だと思います。たとえば、中身をユーザに見せるには、それなりの手間がかかるし、モジュール化すると性能が落ちるなどの問題もあるでしょう。さらに、自己拡張可能なツールの製品責任はどこにあるのか、メンテナンスはどこがするのか、といった問題も生じてきます。

どちらかといえば、初心者用と熟練者用とで、ツールが分化していくということになるでしょう。

A: 拡張性のあるものだと、だれかが改良すれば、みんながその恩恵を受け、それに刺激を受けて新しい改良が行われるというかたちで、技術進歩の波及効果が大きいと思うのですが、どうでしょうか。

Z: ツールの機能評価をさせると、みんな意見がちが

って、まとまりがつかないことのほうが多い。それほどいいことかどうかは、疑問です。

A: みんなでたたきあって、ツールをよくしていこうというメカニズムが、まだ社会的に確立されていないんでしょね。

X: そういうやり方のほうがいいものができるでしょうね。でもいまのところは.....

B: そのうちそうなるんですかね。

X: きつとなるでしょうね(笑)。

A: ものを作るとき、それはその人の経験を反映したかたちで作られる。というより、そういうかたちでしか作られない。だから、それを別のところに持って行ったときには、使う人の経験に合わせて、当然モディファイされるべきだと思うのですが。

Y: 生産性を上げるという意味では、ツールの活用のほかに、再利用をいかにうまくやるかも大切です。プログラムのうちで、本当にシビアにロジックを考えて新しく作らなければいけないのは、せいぜい2〜3割ですから、ノウハウの発散を防ぐ工夫が必要でしょう。現状では、パソコン、オフコン、メインフレームとそれぞれにちがいますから。

それから、何をどういっても、COBOLの使われている率がまだまだ高いのですから、もう一度COBOLアプリケーションをどうやって支援するかということに、たちかえる必要があるでしょうね。

B: とにかくも、メーカーとユーザのはざまにいる人たちに、光が当たるように祈ってます。

X: やっぱり分化でしょう。

A: 理想はあっても実体はない。ツールのみならず、人間の問題もあると思います。

4. おわりにかえて

私はソフトウェア・ハウスで育った関係で、メーカーの人たちとの接触が、これまでほとんどなかった。また、ここ数年はUNIXを使っているの、今回話題としてとりあげられた各システムは、まったく知らない世界で、以前に比べるとずいぶんよくなったという印象をうけた。

しかしそれでも、ソフトハウスのエンジニアたちが、使いにくいメーカーのツールで苦勞しているという話は、いまだに数多く聞く。それを解決する方法のひとつとして、今回のような、ツールを作って供給する側と、それを使わざるをえない側との討論が、もっと頻繁におこなわれる必要があるのではないかと思う。

MAC と UNIX の世界

佐原 伸

野村コンピュータシステム

この報告は、8月の横浜支部例会での佐原さんの発表をもとにして、世話人の藤野晃延がまとめたものです。したがって、文責はすべて藤野にあります。

1. はじめに

私が日常使っているソフトウェア開発環境は、残念ながら最新の高価な EWS や、あるいは Smalltalk-80 のような先進的ソフトでもありません。

VAX 上の UNIX (BSD) と、片手でも持てるような可愛らしいパソコン Macintosh です。そして、私自身、現在のこの環境、つまり UNIX と Mac の組み合わせがとても気に入っています。というより、現状では、これより強力で使い勝手のよいものはちょっとみあたらないのではないかと、とさえ思っています。

そこで、典型的な私の1日の仕事を紹介し、その中で UNIX と Mac がどんな役割を果たしているかを示し、オフィス・ワークとしてのソフトウェア開発に特徴的なポイントについて、考察してみます。

2. UNIX と Mac と私

オフィスでの仕事では、やはり文書処理に費やされる時間が最も多く、それはたとえば、仕様書を作成したり、スケジュールを立てたり、技術資料をまとめたり、あるいはプログラムを作成したりと、内容はいろいろですが、広義の文書処理とみて差し支えないでしょう。

そこでの、私の仕事は、だいたい次のようになります。

- (1) 目の前の端末から、漢字 UNIX 4.2BSD (by アスキー) にログインし、
- (2) jvi (vi の日本語版, by アスキー) で文章の入力を行い、
- (3) VJE-Σ (by VACS) で漢字文章変換をする。

(以上 on VAX)

ドキュメントが文章だけでは、説得力に欠けるので、説明の図や写真が欲しくなります。その場合には、

- (4) MacDraw (by Apple) で図形作成・編集を行い、
- (5) MacVision (by Koara Technologies + ビル・アトキンソン) でイメージ入力し、

- (6) MacPaint (by Apple) で絵とイメージを編集する。
(以上 on Mac)

次に、作成した Mac 上の図やイメージを、UNIX 上の文章ファイルと一緒にし、最終的なドキュメントとしての体裁を整えなければなりません。そのために、

- (7) MacTerminal (by Apple) で Mac を VAX/UNIX に接続し、
- (8) macget で Mac のファイルを VAX/UNIX にファイル転送する。
- (9) 転送されたファイルを macpic (by ariza@sra) で pic コードに変換し、
- (10) 日本語 pic, tbl, ditroff (by ikadai@sra) で清書する。

ここで、LBP に実際に出力する前に、最終出力イメージを端末上で確認できると大変便利です。そのようなツールができないか、SRA の松尾さんに質問したら、とっくに作ってあるということでした。したがって、

- (11) pcview (by matsuo@sra) で最終出力イメージを確認し、意図したものと異なる部分があれば、再び (10) に戻る。OK なら LBP 出力して完成。

そのほかに、作成した文書の管理、ネットワーク経由の転送、といったことも手軽に行えます。たとえば、

- (12) RCS で作成文書のバージョン管理を行い、
- (13) junet (管理人 /dev/jus) で、ネットワーク上の他のユーザに転送し、意見をうかがい、
- (14) m でニュースを読み、最近のトピックについての海外情報を端末にいながらにして入手することができ、
- (15) MH で電子郵便を読み、先ほど転送した資料の内容についての、他のユーザからの批評や意見が自分宛てに届いていれば、それを参考にして改訂版の作成が可能となる。

(以上 on UNIX)

さて、以上で私の典型的な1日の仕事が終了しました。もう、気付かれたことと思いますが、UNIX と Mac の組み合わせにより、単に計算機、あるいはワープロとしてのみ使うのではなく、製図機、複写機、写真機、電話、ファ

クシミリ、スクラップ・ブック等としても活用できるわけです。

UNIX と Mac を組み合わせが、なぜこんなに強力なのか、それを説明する前に、UNIX と Mac の特徴を簡単に比較してみます。

3. 使い易さを追求したユーザ・インタフェース

Mac の特徴は、何といてもそのユーザ・インタフェースで、その設計思想は、親しみ易さと一貫性を融合させ、コンピュータを可能な限り使い易くすることにあり、その目的の追求に、300 人月もの労力を費やしたともいわれています。

私は自宅にも Mac を持っていますが、残念ながら、ほとんど使用することができません。私が使う前に、子どもたちが占有してしまいます。当然、子供ですからマニュアルなど見るわけありません。しかし、ディスクのセットだけしてやれば、あとは勝手に楽しんでます。このことから、Mac の使い易さの程がおわかりいただけるでしょう。

実際、ユーザ・インタフェースに関してみれば、いわゆる Lisp マシン等の高性能ワーク・ステーションをも含んだ無差別級の比較でも、Mac に軍配が上がるだろうとまでいわれています。

一方、UNIX の方はどうかといえば、プロフェッショナル向きであるだけに、きわめて寡黙で、決して親しみやすいとはいえません。

次ページの表 1 に、Mac と UNIX のユーザ・インタフェースを (Mac のユーザ・インタフェースの概念を中心にして)、比較してみました。

Mac の User Interface 第 2 版 (by Espinosa) という手引書には、表 1 に示したようなことが、形式的、実用的にまとめられ、使いやすさと一貫性のあるユーザ・インタフェースの実現のための指針が説かれています。

たとえば「モード」を形式的にあらわすと、「対話型コンピュータでは、1 つのモードは、ユーザ・インタフェースの状態を指し、その状態はある期間継続するが、いかなる特定の目的とも関連することなく、オペレータの入力に対して解釈をくだす以外には、何等の役割も持たない」ということになります。

何のこともやらさずバリわからず、それにしがったユーザ・インタフェースの実現など、私には到底無理な話ですが、しかし、このような哲学的な努力が Mac の使いやすさを支えていることは確かです。

4. 機能の組み合わせの妙

ユーザ・インタフェースでは、Mac に一歩遅れをとった感のある UNIX ですが、機能の豊富さ、機能を組み合わせての新しい機能の実現容易性等で、Mac をはるかに引きはなしています。複雑な機能を、既存のツールの組み合わせで実現しようという UNIX の基本思想は、コマンド・インタプリタ、入出力ダイレクション、パイプやフィルタといったエレガントなメカニズムによって実現され、ツールの豊富さとあいまって、きわめて魅力的なものとなっています。

さらに、ほとんどのツールがパブリック・ドメインであることも特徴で、さきに紹介した UNIX のツール類も同様です。機能面からの UNIX と Mac の比較を、表 2 に示します。

これらのことから、データの量がある程度まとまっている、複雑な処理を行いたい、同時にいくつかの処理をかたづけたい、といったときなど、UNIX は頼りになる相棒です。

5. 結論

以上述べてきたことをまとめると、次のようなことがいえます。

- (1) すべての処理を終了すると、ディスクが OS によって自動的に排出されるなど、Mac はだれでも安心して使えるよう工夫されている。逆に人間が安心して、まったく不注意なミスを招く場合がある (複写時のディスクの入れ換えミス等)。
- (2) 簡単な仕事、視覚的な仕事は Mac にやらせるのがよい。
- (3) 複雑な仕事、プログラマ向きの仕事は UNIX にやらせるのがよい。
- (4) Mac も UNIX も、熱狂的ユーザが多く、ソフトの質がかなり高い。
- (5) 最終的な結論として、UNIX と Mac は、それぞれ両方の極の雄であり、その最大の長所が、他方の欠点をちょうど補完するような形になっている。

このため、UNIX と Mac を組み合わせると、相乗効果が非常に大きくなる。その効果のほどは、

$$\text{UNIX} + \text{Mac} > (\text{MVS} + \text{VOS 3} + \text{OSIV})$$

$$** (\text{N 5200} + \text{5550})$$

と表現しても過言ではないと思います。

Mac	UNIX	意味
Toolbox	shell	ユーザ界面
Inside Mac	なし	マニュアル
モードレス	モードあり	モード（状況）の変化
グラフィック	テキスト	表示形式
アイコン	文字列	操作の対象
マウス	キーボード	位置の設定
ポインタ	カーソル	現在位置
クリック	不定	指定（実行の指示）
ダブルクリック	不定	ショートカット（省略）
プログラマブル・キー	ただのキー	ユーザの分類・識別
タイプ・アヘッド	タイプアヘッド	先行タイプ入力
オーディオ・フィードバック	不定	確認、注意の喚起
ディスク・トップ	コマンド	ユーザ・モデルの世界
ウィンドウ	ディスプレイ	表示形態の単位
リソース・ファイル	不定	環境の柔軟性・再定義化
スクロール・バー	不定	視野の変更、スクロール
タイトル・バー	なし	ウィンドウの位置変更
サイズ・ボックス	なし	ウィンドウのサイズ変更
ディスク・アクセサリ	コマンド	文房具の機能
フォント	一部有り	表現（文字の大きさ）
タイプ・スタイル	一部有り	表現（文字の形態）
ボタン	サブコマンド	下位機能のON/OFF指定
チェック・ボックス	オプション	オプションの選択指定
ダイアル	オプション	アナログ式オプション
ドラッグング	複数指定	対象の範囲（複数）指定
ダイアログ・ボックス	オプション	対話的な実行指示・指定
警告ボックス	エラーメッセージ	警告状態の通知

表1 - ユーザ・インタフェースの比較

UNIX	Mac	備考
shell	なし	コマンド言語、強力で汎用的
マルチ・プロセス	シングル・プロセス	バックグラウンド機能が可能
ツールキット	ツール単体	ツール間インタフェース
マルチ言語	言語なし、オプション	自由度が高い

表2 - 機能の比較

ハンドヘルド・ワークステーションの夢

佐久間 まゆみ

神戸コンピューターサービス

1. はじめに

本稿は、9月25日に大阪のKDDハロー・インで、「高信頼性ソフトウェア研究会」と合同開催された、SEA関西支部の9月度研究会で発表したものをベースに、補・加筆したものです。

ソフトウェアを開発しているわたしの環境は、あまりよいとはいえません。会社にはマシンはあるのですが、ほとんど計算サービス業務につかわれているので、客先のマシンを借りてソフトウェアの開発をしています。ですから、ユーザがかかわると、その度に各地にいくようになります。

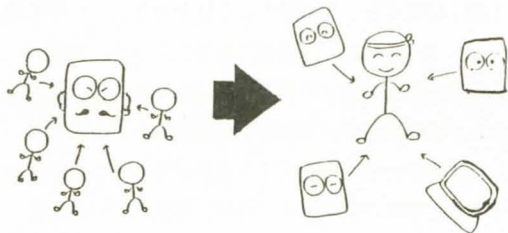
今日お話ししようと思っている内容は、最近わたしが所属している部で、「開発環境」をテーマとした部内ミニ・ワークショップが行われまして、そこで話したことがもとになっていますが、「ソフトウェア技術者にとって気持ちのいい環境」ということを、あくまでも技術者の立場から、マネージメントや経営サイドの観点をまったく無視して考えてみたいと思います。

2. 現状と問題点

2.1 マシンについて

まず一番大きな問題であるマシンについてですが、先程もいいましたように、自社にあるマシンは、計算業務に使われているので、わたしたちソフト開発部隊は、ユーザのマシンを使わせてもらいに出向いていきます。ともかく、マシンがないとソフトウェアが開発できないのですから、そうならざるをえないのですが、これをなんとか解決したい。

つまり、わたしたち人間がマシンを中心に動くのではなく、人間の動きにあわせて、マシンが使う人の周りに集まってきてくれるような環境にしたい、ということになります。



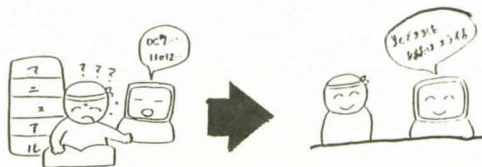
2.2 ソフトウェアの問題

次に、ソフトウェアの環境についてです。わたしは、大型のホスト・マシンを使うことが多いので、そのときの問題ということになりますが、何かコマンドを叩いて、それにエラーがあったとします。すると、なにかメッセージがでてくればまだいいほうで、ほとんどの場合、ただわけのわからないコードがでてくるだけです。

そこで、コードの内容を調べようと思ってマニュアルをみるわけですが、そのコードの所をみますと、他のマニュアルの名前がでてきて、そっちをみるようにとあります。そこで別のマニュアルをめくると、また別のマニュアルの名前がでてくる。こうして3冊目のマニュアルにとりかかった結果が、サポート・センターに電話をするように、書いてある。「いったいなんなんだ!」と、思わずマニュアルを投げ出したくなるほどの「親切さ」です。

必要な情報は、10冊以上あるマニュアルの中にあるらしい（ないときも多々あるようです）のですが、本当に必要な情報を手にいれるるまでには、大変な思いをしなくてはならないわけです。

ひとつの解決法としては、コードにはかならずメッセージをつけるようにし、そのメッセージも簡単にいいから、わかりやすい日本語にするくらいの辞書機能がのぞましい。つまり、使う側のことをもっともっと考えて欲しいということです。



2.3 設計時の問題

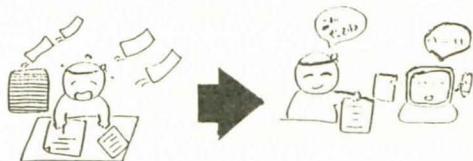
その次は、設計段階での話です。世の中には仕様書というものがありまして、この仕様書にたいして、変更という名の手直しが頻繁におこなわれます。多くの場合と

いうか、ほとんどが手書きの仕様書でして、その変更によって影響のあるところを捜しだして、整合性をそこなわないようにと気を使いながら（手書きですから、あくまでも気を使うだけです）、またまた手書きで直すのですが、それが終わらないうちに、また次の変更がくるという状況がふつうになっています。

ソフトウェア技術者というのは、ソフトウェア＝プログラムをつくるのが仕事かと思っていたのですが、まるで一昔前の植字工みたいなもので、修正された原稿をもとに、1字1字、ピンで文字を探している気がします。

特に、プロジェクトの終わり近くの1ヶ月位などは、仕様書などのドキュメントの書き直しを含めた整理に忙殺されて、まるで情けなくなってしまうこともあります。

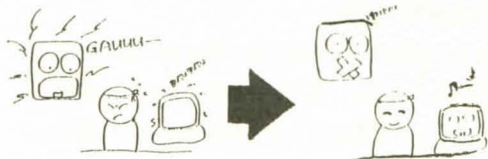
ここで、わたしのいいたいことは、ソフトウェア開発にとって本当に重要でないこうしたつまらない作業に、多くの時間をさかなければならないのは悲しい。もうそんなことを止めたい、ということです。人間が自らやって効果があがることに専念し、そうでないどうでもいいことは、機械にまかせるようにしたいのです。



2. 4 周囲の環境

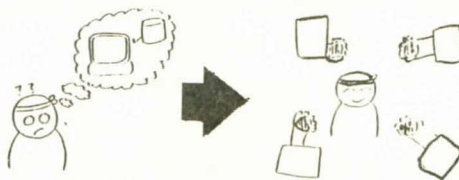
最後に、周囲の環境の問題点をあげてみたいと思います。

まずは雑音です。この問題は、マンマシン・インタフェースということで、エルゴノミクスの研究テーマのひとつにもなっているようですが、今のコンピュータはかなりの音をだしてしまっていて、特にホストマシンはすごい音がします。これが集中を要する仕事には邪魔になるわけです。わたしは特殊なヒトで、多少くらい雑音がするほうが能率が上がるのですが、普通は邪魔な存在です。そこで、個人に合わせた作業環境が作れたらと思うのです。静かなほうがいい人には静寂を、そしてわたしのような人間には、好みにあわせて、たとえばBGMが流れ



るような作業環境があればいいと思います。

もうひとつ、周囲の環境に関してですが、ソフトウェアを開発するときには、ユーティリティとかツールが有効なことはわかっている、それがどこにあって、どう使っているのかわからないことがよくあります。周囲の人たちに聞いてはみるのですが、閉じられた世界では、自分たちがこれまで使ったものしかわからない。どうも、もっといいものがあるはずで、よそにはたくさんあるらしい、と思えるのですが、手に入ってこない。せめて情報が入手しやすいように、もっとオープンで、よその会社の人とも情報交換ができる情報ネットワークをつくるべきだと思います。



2. 5 まとめてみると

これらをまとめてみると、以下のようになります。

- ・人間がマシンの所へ行くのではなく、マシンが人のところへやってくる。
- ・利用者の立場にたったソフトウェアが必要である。
- ・本来人手をかけるべきところでないところは、ツールをつくるなりして機械にまかせる。
- ・個人によって作業がしやすい作業環境の設定が行えるようにする。
- ・情報が入手しやすい情報ネットワークをつくって、広く使えるようにする。

3. 夢のワークステーション

これまで述べてきたことは、どちらかというと一般的なことでして、これから、もうすこしわたし個人に関係のふかいお話をします。どちらかというと夢物語になるのですが、これから話すようなワークステーションが使えれば、わたし自身がもっと仕事をやる気になると思います。

まず第1にですが、手で持ちはこびができるワークステーションです。わたしは部屋にとじこもって仕事をするような暗いタイプのプログラマではありませんから、今日のように天気のよい日には、公園なんかで仕事したい。別に端末に向かわなくても仕事ができることがあるのですが、端末に向かっていると思

って安心する人がどこかにいるので、とにかくどこにいても端末に向かえるようにすればいい。そんな意味から、とにかく持ち運びができることが条件になります。人によっては、雨の日でも外でやりたというので、防水の必要もあるでしょう。

マウスはなかなか便利ですが、いまの方式では、長時間使用していると肩がこってしまうのでなにか工夫が必要です。わたしの同僚は、足で操作するというアイデアをだしたのですが、女の子の場合そうもいかないことがあります。そこで、視線によってポイントを決めることができないかと思ひます。つまり、ファッションブルなアイ・センサー・サングラスのようなものがあれば、昼間の太陽の下でも、役立つと思ひます。

次に、キーボードの変わりに、音声による入力を基本にしたい。もちろん、コンピュータからの応答も音声です。ただ、キーボードはオプションでつける必要があります。たくさん人のなかで仕事をしているとき、皆がめいめいに、大声でしゃべったりしたらうるさいので、そんなときにはキーボードで、ということになります。

メール機能は、情報の収集等にかかせませんが、ハンドヘルド・ワークステーションは、コードがないのが大前提になっています。このコードレスというのは、是非実現して欲しいのです。単に持ち運びに便利ということだけでなく、オフィス内でも、美観をそこねますし、つまづくものをわざわざ広げているようなもので、とにかく邪魔以外のなにものでもありません。そこで、世の中にはいくつもの人工衛星が打ち上げられているので、そのひとつをつかって、必要な情報はすべて電波で流すことにしたらどうでしょう。でも、そのために大きなアンテナを持ったりするのはごめんです。

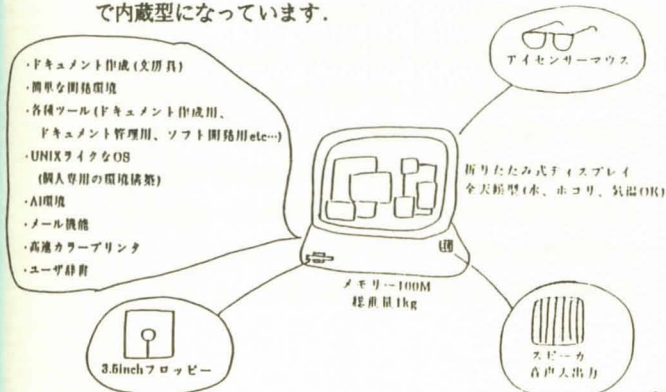
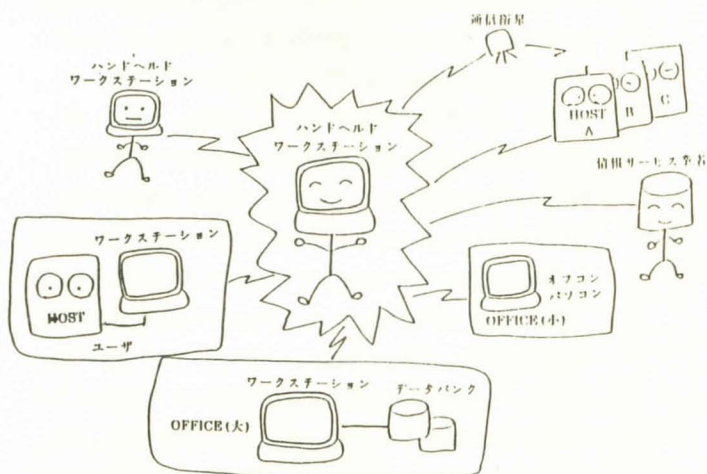
その他、ドキュメント作成・管理のためのツールとか、ソフトウェア開発にとってツールは、すべて用意されているのは当然です。OSは、個人の環境が設定できれば、UNIXでもなんでも構いません。プリンターはカラーで内蔵型になっています。

4. おわりにかえて

夢のようなことをあげたようですが、本音をいえば、技術者として働ける環境が欲しいということです。

わたしのところでも、J-Starを導入して、環境を少しでもよくしようとしているのですが、いかんせん機械の台数より使いたい人のほうが多くて、根本的には改善されていないというのが現状です。夢のようなといいましたが、現在の技術をもってすれば、決して不可能ではないわけですし、必要なのは、わたしたちのように、日々ソフトウェアを開発している個人の手に、こうしたワークステーションがわたることです。そして、このワークステーションが、機械中心で作られるのではなく、あくまでも使用する人間のことを考えた、つまり、人間中心主義で作ってもらいたい、ということです。

世の中では、ソフトウェア技術者は最先端技術を応用しているといわれているのに、その実は、これまでに話したように悲惨なことが多いように思われます。ソフトウェア開発で必要と思われる環境が用意されておらず、その足りないところを、わたしたちの努力なり、苦勞なりで埋めているという、無駄な作業をなくして、わたしたちのもっている能力をフルに駆使できる開発環境が、一日も早く実現されることを望みたいと思ひます。



中国旅行記

連載 - その1 -

野村 敏次

日本電子計算

1. はじめに

この夏、8月18日から8月20日の3日間にわたり、北京(Beijing)で、ソフトウェア工学に関する国際ワークショップが開催された。たまたま、SRAの岸田孝一さんの御紹介で、NCSの村井進さんとともに、その会議に招待され、プレゼンテーションする機会が与えられたので、ついでに西安(Xian)、上海(Shanghai)にも立ち寄り、中国におけるソフトウェア開発の現状について見聞してきた。

会議の内容そのものは、プロシーディングが立派な本になっているので、それをお読みいただければよいと思う。この旅行記では、私が見て感じた中国について、いくつか書いてみたいと思う。単なる読み物として楽しんでいただければ幸いである。

2. 中国到着

8月16日(土曜日)の午前10時過ぎに成田を出て、北京国際空港についたのは午後2時近くであった。空港特有の華やかさはほとんどなく、ひっそりとして暗い感じが漂っていた。入国審査にしても、税関にしても、係官はほとんど口をきかず、機械的に書類に目を通し判を押してくれるだけである。所々についている薄暗い蛍光灯が、そうした感じを強くしていた。荷物の出てくるまでの間に、日本円を中国元に交換したが、ここでも全くの無言。差し出された用紙に、パスポート番号、氏名、金額を記入して現金と一緒に差し出すと、しばらくして、中国元と交換証が無造作に窓口から投げ出されてきた。まあ、考えてみれば、中国語でいろいろいわれても、こちらは何も解りはしないのだから、無言でことが運ぶなら、それが一番楽でよい。

トンネルを抜けた感じで到着ロビーに出ると、出迎えの人々で混雑していたが、われわれの出迎えの人は見当たらない。B4版位の紙に、名前を書いたものをかざしている人が何人かいるが、その中に岸田もなければ、野村、村井もない。互いに顔は知らないし、こちらは迎えの人の名前すら知らない。人混みの中を名前を書いた紙

を覗いて捜すこと約10分。居ました、Dr. Kishida, Dr. Nomura, Dr. Murai, 藤野副総裁(あとで日本電気の藤野さんのことだとわかった)という名前を書いた紙を持った人たちが!

迎えに来られた人の中心は、北京航空学院計算機系の副主任で副教授の金茂忠(Jin Maozhong)さんであった。日本語を上手に話し、われわれが北京滞在中に主として面倒を見てくださった。今回のワークショップは、国家科学技術委員会の下部機構である中国ソフトウェア技術開発センタの主催だが、北京航空学院も事務局としてこれを手伝っていて、金さんは、5年ほど前に福井大学に留学した経験もあり、われわれとの窓口になられたようである。

3. 宿舎へ

われわれをのせたマイクロバスは、地平線までも続くかと思われる真っ直ぐな道を、かなりのスピードで走っている。道の両側には鈴懸けや白樺に似た木々が整然と植えられ、木のトンネルを潜っていくような錯覚におちいる。両側の景色は森や林や畑が交互に現れ、人家はほとんど見当たらない。30分以上走って、ようやく広い道にでたが、先程とは全く異なった、緑のない建物だけの光景となった。

片側四車線の道を車がクラクションを鳴らして、かなりのスピードで走っている。その合間をぬって、いたる所で人と自転車が道路を横切っている。交通規則があるのかないのかはわからなかったが、自転車や歩行者が信号をほとんど無視するのは、肝を冷やすし、驚かされたことのひとつである。

土ぼこりのためであろうが、車外の景色は霞がかかったようにぼんやりとし、太陽は出ているが影が薄い中を、さらに10数分走って、われわれがようやく宿舎に着いたのは、4時近くになっていた。外国に来たという印象はさほど強く感じなかったし、広大であるという感じもさして受けなかった。

4. 宿舎

われわれの宿舎は、中国国際経済技術交流中心（China Center of Economic and Technical Exchange）の宿泊施設で、6階建ての近代的なビルであった。20畳程の応接セットのある部屋と、10畳程の寝室、バス／トイレが付いていて、アメリカや日本のホテルと較べても、むしろ広々とした感じのする良い部屋であった。備品としては、6人分の応接セット、机、冷蔵庫、テレビ付きで、クーラーも程よくきき、想像していたイメージとはかなりかけ離れた立派なものであった。因みに、冷蔵庫とクーラーは日立、電話は日電、テレビはソニーであった。

ただ、閉口したのは、部屋が何の臭いかわからないが、臭うのである。これは、上海でも同じであったが、窓を開けておいても、外出から帰り部屋に入ると、その臭いが強烈に鼻についた。

部屋には鍵がかかるようにはなっていたが、各階に寝ずの番の人がいて、ドアを開けると何でもすぐ用が頼めると共に、不審な人をチェックしているので、治安の面では極めて安心できた。

5. 史跡

北京では、月曜日から3日間会議が予定されているので、空いているのは着いた翌日の日曜日だけであったが、その日に中国ソフトウェア技術開発センタの招待で、万里の長城と明の十三陵に案内していただけることになった。アメリカ人、デンマーク人、中国人、シンガポール人、日本人等国際色豊かに15人程の招待者を乗せたマイクロバスは、やや薄曇りの中を万里の長城に向けて、朝の8時に宿舎を出発した。

（1）万里の長城

現在、一般人が見られる万里の長城は、北京の西北約80Kmにある八達嶺にある。行くまでの道幅は広く、信号もほとんどないので、車はかなりのスピードで走っている。所々で道路の拡張工事が行われていて、土ぼこりがすごい。工事現場には、大勢の人々と、トラクタとローラーが各1台、それに数台のトラックがいたが、人による作業が圧倒的に多く見受けられた。また、休憩をとっている人の多いのも印象的であった。

長城に近づくにつれ、道幅は細くなり、車も次第に込みだし、ついには前に進まなくなってしまった。車は観光バスが多かったが、渋滞して止まっているそれらの車に向かって、手にゆで卵を持った人々が、大声で売り付けようとしている光景が見られた。何故か昔の記憶が呼

びさまされたような、一抹のわびしさを感じた。これなどは年のせいかもしれない。

八達嶺に着いた時には、既に11時40分になっていたが、目の前に見る歴史的な建造物には、いい難い感動を覚えた。2500年前の周朝末期に、北方の異民族の侵入を防ぐため構築され始め、秦の始皇帝のときには、30万の軍兵と数百万の農民を徴発して、大量のレンガを用いて現在の長城の原型を造ったとされている。

ところが、その感動も、余りにも多い人の波に瞬時にして消え去り、人とぶつかりながら、真夏の太陽が照りつける中を、かなり起伏の激しい長城の上を汗を拭きながら歩くことになった。観光客の多くは中国の人で、時たま日本人やアメリカ人がいるだけであった。お年寄りの姿も多く、北京近郊における最大の観光地であることを、改めて認識させられた。

120m毎に立つ烽火台に立って見渡せば、長城が山々の尾根を延々と続く雄大さに、しばし暑さも忘れて見惚れていたが、その建設の困難さは、想像を絶するものがあつたと思われる。6000Kmに及ぶというとてもないものを造った皇帝を愚かであると批判することもできようが、広大な中国全土を城壁で囲もうとした壮大な発想は、せせこましく生きるわれわれ現代人には、及ぶ術もない。強大な権力者と虐げられた人民の合作ともいべき、この地上最大の建造物が、多くの人々に限らないロマンを末長く与えてくれることを祈りつつ、長城を後にした。

（2）明の十三陵

万里の長城から20Kmばかり北京の方に戻った所に昌平の街があって、ここから東に10Km程行った所に明の十三陵がある。これは、明朝第三代目の皇帝である永楽帝以下13人の皇帝とその后たちの陵墓である。この内、最も古くて大きなものは、永楽帝の陵である「長陵」であるが、未だ発掘されていない。

われわれが見たのは、二番目に大きい「定陵」であり、ここには、第14代の万暦帝と2人の后が眠っている。十三陵の入り口の大宮門からは、両側に10m間隔位に、功臣像／文臣像／武將像、馬／象／ラクダ／カイチ等々の石像（それぞれ、かなりの大きさである）が並んでいた。ここもかなりの人出ではあったが、万里の長城ほどではなく、落ち着いて見学することができた。日本の神社・仏閣と同じような感じで、周りにはさまざまな土産物店が軒を並べている。その様は、日本の観光地の風情

とたいして変わる所はなかった。

陵墓をみるために、薄暗い石段を地下5階位まで降りると、そこは地下宮殿という言葉に相応しい大理石造りで、ひんやりした冷気が漂っていた。広さもかなりのもので、棺や玉座、それに壺等のさまざまな埋葬品が陳列されていて、墓の中というイメージは全く感じられない。史実によれば、万暦帝は10歳で即位し、22歳で自分の墓を作り始め、6年の歳月と国家財政の2年分の銀を費やして造ったのが、この定陵であるというから、立派であるのは当たり前なのかもしれない。

時間は既に午後1時を大分回っている。そこで昼食ということになり、外人用の接待場所で、渴いた喉を北京ビールで潤しながら、美味しい北京料理を食べることができた。中華料理はひとつの歴史だという。広大な史跡をみたあとで、歴史を食べるという、かつてない経験を満喫することができた。

6. 国際会議

18日(月)8時半より、国際経済技術交流センタの3階ホールで、

International Workshop on

Software Engineering Environment

すなわち国際軟件工程環境討論会が開催された。ワークショップとなっているが、実質はシンポジウム形式であり、初日はシングル・セッションであったが、2日目以降は、2つのパラレル・セッションで、発表および討議が行われた。

会場に入るとわれわれ招待者には、前の席2列が用意されていて、そこだけはお茶の用意がされていた。会場は200人近く入れる広いもので、ほとんどが中国の人々であったが、期間中満席に近かった。会場の一番前には、演台とOHPの他に、一段と高くなったところがあり、20人程度座れる席が会場と向かい合わせて用意されていた。会場の一番前の高い所には、赤い横断幕が掲げられ、「国際軟件工程環境討論会」という白い布を切って作った文字が張り付けられていた。

(1) オープニング・セレモニー

ちょうど8時半、一段高くなって会場に向かい合っている席に、10数人の男女が着席する。国家科学技術委員会並びに党の偉い方々らしいが、紹介もないので誰だか判らない。会議に慣れていないせいか、司会等で若干のモタつきはあったが、やがて、その雄壇の中の一人が中国語で挨拶を始めた。英語の逐次通訳はついたが、喋

る人も通訳する人も、それぞれ勝手に原稿を読んでいるのがおもしろかった。テレビ局も来ていて、雄壇の人々や会場の人々を撮影していたが、カメラで写真を撮る人は誰もいなかった。残念ながら、われわれも会議の様子を写した写真は一枚も持っていない。

2人の挨拶が済み、セレモニーが終わると同時に、雄壇の方々はサッと席を立たれて消えたが、その席は会議中の3日間ずっとそのままであった。

(2) 基調講演

講演者は二人で、最初はカリフォルニア大学サンフランシスコ校の Wasserman 教授であり、次はヨーロッパ共同体の ESPRIT プロジェクトの Hunke 氏であった。

Wasserman 教授は、

From Programming Environment to

Software Engineering Environment

というタイトルで、中／大規模のソフトウェア・システムの開発のためには、TSSターミナルとエディタ／コンパイラ／プリティプリンタ／リンカ／ローダ等々といった基本的なツールセットだけではなく、要求分析／仕様記述／設計／テスト、さらには履歴管理やプロジェクト管理等の技術的支援が重要であることを説き、開発者は、単なるプログラミング環境よりも、それ以外の作業を自動的に支援してくれるソフトウェア開発環境を欲するようになるとし、そして、自らが大学で研究していた USE システムを発展させた新しいソフトウェア開発環境 IDE の概念を紹介された。

Hunke 氏は、

Beyond the Life-Cycle View

- The Industrial Context

というタイトルで、ライフサイクル・モデルをプログラミングを中心に発展させていく考え方を説明したが、印象としてよくまとまっていて、モデルとしては非常に興味あるものであった。

(3) 各セッション

各セッションは、5-6の論文発表で構成されており、時間は3時間半づつに区切られていた。一緒に行った人の中では、岸田さんが初日の午後の4番目、村井さんが2日目の午前の2番目、そして私が最後の日の午前の2番目となっており、やはり、最後の日というのは、心配事が先にあるだけ気の重いものであった。

40篇の発表論文は、外国の論文が15篇で、残りは中国からの論文であった。特に2年ほど前から始められ

ている

ISEE (Integrated Software Engineering Environment)

という UNIX をベースとしたナショナル・プロジェクトに関する論文発表が多かった。日本でも、これと類似のプロジェクトが進行しているので、その概要について簡単にふれておく。

このプロジェクトは、中国ソフトウェア技術開発センターが中心になって、10の大学や研究所を巻き込んだ共同の国家プロジェクトである。1984年8月からスタートしており、2段階で開発を行う予定で、第1段階では、個々にツールをインプリメントし、第2段階では、それらの統合化を行う。現在、いくつかのツールは第1段階を終了しているところである。

最初の1年半の間に、4回のレビュー・ミーティングを実施し、多くの海外の専門家も招いて意見を聞いている。マンパワーとしては、既に約200人年を費やしている。このプロジェクトの目的は、ソフトウェア開発における全ライフサイクルを支援する統合化されたソフトウェア工学環境を確立することで、21世紀の情報化社会に向けてのアプローチである。現在、10のツールを作っており、それらは以下の通りである。

- ・ DSE: Design Structure Editor System
- ・ CSED: C-Based Syntax-Directed Editor
- ・ STE: Software Testing Environment
- ・ SIB: Software Information Base
- ・ UI: User Interface Tool - 2プロジェクト
- ・ C-CROSS: Cross Compiler for C
- ・ DMSS: Decision Making Supporting System
- ・ CMS: Configuration Management System
- ・ EASYCODE: A 4th Generation Language
- ・ USE: A Universal Structured Editor

来年には、計画の第2段階に入り、最終的に統合化されたものは、AT&Tの3B-5か、DECのVAX 11/785の上にインプリメントされる予定になっている。

7. 食事について

旅の楽しみのひとつは食べ物である。中国に来たからには中華料理を食わねばならぬ。メニューは、肉、野菜、魚、麺等に分かれていて、漢字を見れば、おおかたの見当はつけられる。酒の種類も、ビールか茅台酒か紹興酒だから、銘柄にこだわらなければこれも適当に頼める。ただ、残念なのは、持ってこられた料理の名前とか材料

が判らないことだ。

このように書くと、食事はいとも簡単にとれそうであるが、実際はなかなか思うようにはいかない。ホテルの食堂を除いて、有名な料理店に行くには予約がいる。しかし、これらの店には、日本語や英語のわかる人はほとんどいない。従って、ロビーに英語で予約の電話をお願いすることになる。

予約なしで行ったら、店先で交渉して入らねばならないが、ほとんどは言葉が通じない。そこで筆談となるわけであるが、日本にない漢字も多いし、意味も若干異なっており、思うようにはすまないのが普通である。

会議も終わった20日の夜、7人で食事に行くことになった。場所は、宿舎から車で20分程のところで、西四南大街にある「同和居飯荘」という山東料理の有名店である。この店を選んだのは、ここだけにあるという有名な「三不粘」というものを食べるためであった。2台呼んだにもかかわらず、タクシーは1台しか来ないので、まず、3人が最初に行き交渉し、4人は後から来ることに決まった。

店の場所はすぐに判ったが、入るまでがたいへんな苦労だった。英語も日本語も通じないので、さっそく筆談に入ったのだが、「今、われわれは3人おり、後から4人来て、総勢7名となる」ということを、相手に判ってもらうのに10分以上要した。店の方も4人できて、店の入り口でやりあっているの、通行人が立ち止まってみてきた。どうやらわれわれが使う日本の漢字が、中国の人に理解してもらえなかったことに原因があるらしい。「後から4名来る」というのは、「后から四人来る」としなければならず、「合計、あるいは総計7名は、「共七人」とすべきであったことが後で判った。

次は金額である。これは割合早くまとまった。相手が「40元」と書いてきたので、「各人？」と問うと、「共280元」、「可、不可」ときたので、可に○をつけて商談成立だ。この間15分位だったが、人目にも気になるし、かなり疲れた。まあ、一度店に入ってしまった、個室であるし、気楽に飲み食いでき、すばらしいディナータイムをすごすことが出来た。因みに、「三不粘」とは、豆の粉と砂糖と卵黄でできているデザートで、確かに箸にも歯にも、もちろん入れ歯にも全くつかない。食べた後の入れ物は、洗ったように奇麗であるという評判は、まさにその通りであった。味も、少し甘味のある、おいしい食べ物であった。 [全3回]

幹事会報告

昭和61年度第4回：9月25日（水）

会 場：機械振興会館

議 題：

1. 会員拡大活動

9月22日現在での会員数が報告され、年度目標の1,000人を達成するためには、月100人の増加が必要である。そのために、前回の幹事会で作成することが決められていた新入会員募集のパンフレットを、ソフトウェア・ショウの会場で配布する。また、同じものを会員に発送し、会員拡大に協力してもらうことが、あらためて確認された。

また、正会員の所属している会社数は、延べ300社をこえていることも報告された。

2. ソフトウェア・ショウ

ソフトウェア・ショウ期間中に、当協会のブースで広報活動することと、流促センタ・ブースでソフトウェア相談コーナーを担当するボランティアの割り当てが、当日出席した幹事のなかから決定された。

3. 秋のセミナー・ウィーク

秋のセミナー・ウィークの参加者数、および予定収支が、セミナー委員長から報告された。参加者数の詳細は、今月号のセミナー総括報告を参照されたい。

4. NISAとの共催イベント

10月31日に、長野県長野市で行なわれる情報化月間記念フォーラムに関して、NISA（長野県情報処理協会）から共催の依頼があった。招待講演者3人がいずれもSEA会員であること、長野県にSEAの活動を広報し、支部結成をめざすことにも役立つということで、全員一致で了承された。

5. 幹事の増員

5月の総会で、幹事会に一任されていた幹事の増員について、いま幹事が不在の支部や研究会の世話人に、とりあえず幹事就任をお願いすることが決定された。

（記：野辺）

A I 分科会 (SIGAI)

1. 第5回月例会

10月9日（木）に、機械振興会館の会議室で開催。「オブジェクト指向入門：Smalltalk を中心にして」というテーマで、アステックの坂下さんにお話をいただきました。その後に参加者から、Smalltalk でアプリケーション

を開発している苦勞談がいろいろ出て、来年3月の例会では、そうしたケースの事例報告をしてもらうことに決まりました。

2. 次回月例会予定

第6回月例会

日時：昭和61年11月13日（木）19:00-21:00

場所：機械振興会館 B3階 9会議室

テーマ：プログラム自動生成システムとしての
DRACO

赤坂 仁志（協同システム開発）

AI 分科会の月例会は、毎月第2木曜日におこなわれます。連絡および問い合わせは、協同システム開発（03-503-4981）の野辺まで。

管理分科会 (SIGMAN)

10月の月例会は、準備の都合上休会としました。

次回会合予定

日時：11月14日（金）19:00～21:00

場所：東高円寺SEA事務所

テーマ：討論形式「プロジェクト・リーダーの資質とは？」
問い合わせは、沖ソフトウェア（03-454-2111 ex-3134）芝原まで。

教育分科会 (SIGEDU)

教育分科会では、第6回の会合を兼ねて、10月28日（火）に、「新入社員教育を考える」というテーマで、半日のフォーラムを青年会議所（東京）で開催しました（参加者53名）。

11月の月例会は、11月19日（水）PM6時半から、日本電子専門学校のアオシマ・メディア・センタを訪問し、最新の教育設備の運用状況を見学して、意見交換を行います。参加費は1000円。

参加希望者は、杉田（SRA: 03-234-2611）、または中園（BSC: 03-501-4159）まで。

再利用分科会 (SIGREUSE)

今後の予定

これからの討論会、研究会の予定を以下に記します。ソフトウェアの再利用に興味のあるかたは、ご出席ください。出席のための連絡はとくにありませんので、当日に会場においでください。なお、SIGREUSE へのお問い合わせは、世話人のいずれかまでご連絡下さい。

第9回研究会(第2水曜日)

日時: 11月12日 19:00-21:00

場所: 機械振興会館 B3-9

テーマ: 各自のテーマに基づく意見交換

第9回討論会(第4水曜日)

日時: 11月26日 19:00-21:00

場所: 機械振興会館 B3-9

テーマ: 未定

世話人: 青島(SRA: 234-2611)

阿部(NES: 444-3211)

大西(JSD: 503-4981)

村井(NCS: 433-8171)

法的保護分科会(SIGSLP)

第4回の会合は、予定通り10月6日(月)に、「プログラムの同一性の実験」その2、ということで開催された。次回は、次の要項で、同テーマに関する第3回目の討論を行う。

日時: 11月7日(金) 18:30~

場所: ソフトウェア流通促進センター会議室

テーマ: 「プログラム同一性の考察」Part 3

問い合わせ等は、能登(SRA: 03-234-2611)まで。

環境分科会(SIGENV)

第11回月例会

日時: 11月19日(水) 19:00~21:00

場所: 機械振興会館,

テーマ: 未定

問合せ先: 引地信之(SRA: 03-234-2611)

関西支部

SEA関西AI分科会第2回会合

日時: 昭和61年11月12日(水)

18:30~20:30

場所: 日本電子計算(株)大阪支店 会議室

大阪市西区江戸堀1-9-1

肥後橋センタービル 12階

テーマ: 「AIの基礎について」-中野秀男(阪大)

「SUN/3におけるAI環境」

-江口隆己(伊藤忠)

参加費: 会員-500円, 一般-1000円

SEA関西11月例会

日時: 昭和61年11月14日(18:30-20:30)

場所: SRA関西支社会議室

テーマ: 未定

研究会・月例会・分科会を含めて、関西支部へのお問い合わせは、下記のいずれかまで:

盛田政敏(KCS: 078-391-8291)

臼井美義(JIP: 06-488-6021)

横浜支部(Y-DOC)

YDOCは、月に1回の月例会を毎月最終の金曜日、あるいは最終の祝祭日の前日に行っています。自由な懇談会といった趣きですので、気軽に御参加下さい。

毎回、集会の後には中華街に繰り出し、中華料理と紹興酒による宴を持っており、そこでの多方面に亙る多彩な話題の交換には、皆さんの好評を頂いています。興味ある方は是非どうぞ。

第7回YDOC開催予定

・日時: 10/31(金) 18:30~

・場所: 労働福祉センターB2会議室

横浜市関内、スタジアム並び

・費用: 1名¥1,000(会場費, 資料代等)

・内容: 失われたパラダイムを求めて

松原友夫(日立SK)

・連絡先: 参加は自由です。問い合わせ等があるかたは、藤野(045-622-2163)まで。

その他の分科会

先日会員に配布したアンケートをもとにして、近々のうちに次の3つの分科会:

SIGCAD

SIGCAI

SIGNET

がスタートします。御期待ください。

なお、既存のSIGへの参加希望者のリスト(郵送用ラベル)は、それぞれの世話人の手にお渡ししました。

会員からのメッセージ

この欄は、会員相互の誌上伝言板です。

愚痴（仕事上、プライベート？）、探し物、最近考えていること等々、自由な意見交換をする場です。

とりあえず、今回は、9月号に同封したアンケートで寄せられたメッセージのなかから、いくつかをランダムに選んで掲載します。

SEANET が運用を開始するまでのあいだ、電子掲示板の代わりにどんどん御利用ください。

田口 明義

10年程前に、ジェネレータ方式によるプログラム・ソース作成システムを作成したことがある。その方法が、いまでも通用するかどうかに関心がある。プログラム・ソース作成については、環境の変化に伴わない開発手順に違いが出てきているとは思いが。

それに加えて、現在、要求仕様書、プログラム設計仕様書の自動作成にも興味を持っている。

安西 宏

SEAから受け取る郵便物の宛名が、変な所で改行している。他の人もそうでしょう。紺屋の白袴の現代版みたいで、変に嬉しいです。いつもソフトじゃいじめられてるもんだから。

後藤 浩

私の住む家を探しています。会社（丸の内線、国会議事堂前）から1km以内。ちょっとぜいたくかな？

ところで、UNIXワークショップの案内に出ていた「ハッカーたちのBOF」のBOFとは、Birds of a Feather（類は友を呼ぶ）の意味だそうです。知らなかったもので、電話で確認しました。

日比野 修三

名古屋に住んでいるため、なかなか活動に参加できません。

コンピュータは万能ではないし、あってほしくもありません。そろそろコンピュータの進歩にも、ドクター・ストップをかけたいと考えています。第2の原子爆弾に

したくはありません。

岡村 博

「最近の若い者は、仕事より遊び」。どんなに納期がせまっても遊びにでかけてしまう。なんでワシが土・日に出勤せんとあかんねん。当方、若干30歳。

山崎 桂

最近のユーザは、どうして短期決戦ばかり望むのでしょうか。全体で何十万ステップにものぼるシステムを、たった半年でカット・オーバーさせようなんて、狂気の沙汰としか思えない。でも、これが出来てしまうのだから、この業界って、一体どうなっているのでしょうか。すべてこれ、人海戦術の成せる業。いったい品質はどこへいったんだろう。

北条 正顕

bit誌10月号のコラム「あれふ・ぜろ」に、SEAMAILのことが紹介されていました。ついに、わがニュースレターもメジャーなってきましたネ。

小林 伸二

客先の人事移動で、プロジェクトが揺れ動き、迷惑を被ることがたびたびある。少なくとも自分が関与している間は、人事移動なしであってほしい。たとえそれが昇進であっても。

佐藤 雄一

何でもAI、AIというけれど、なにをしたらAIと名前がつくのでしょうか。何でもいいから、何かこじつけたらAIと名前がつくみたいに思えます。それだったら、別にAIといってもいいわなくても、同じように思うですが。

AIという言葉は、エスニックと同じで、単に流行なのでしょうか。

佐久間 まゆみ

来年の目標は、「今年こそ地味に生きる！」に決めました。

小林 文子

最近、会社で限りなく落ち込んでいる自分を、どうすることもできません。会社においても、SEAの会合でも、カレーライスのお皿にちょこんと座っている「フクジンヅケ」に自分の姿をみるのです。ああ、「フクジンヅケ」よ奮起せよ。

河村 一樹

情報処理技術者の適性とは何なのだろう。世間には、プログラマ適性検査などがあり、多くの企業で利用しているようだが、はたして個人の適性を正しく反映するのだろうか。プログラマという特殊な職業と、属人性の強い仕事内容という面からみても、どうも疑問をもつ。多くの学生たちの姿を見つめている立場にあって、考えずにはいられない。これからの研究テーマにしてみたいと思う今日このごろである。

柳田 俊一

「アルファベット3文字の術語・略語をこれ以上増やさない運動」を、みんなではじめませんか。漢字3文字は大いにけっこう。中華人民共和国に見習おう。コンピュータは電脳、プログラムは程序、デバックは除錯、などなど。

細野 広水

某セミナーで、ハード・メーカーの総務部長さんがおっしゃっていた：「ライフサイクルが極端に短いパソコンやワープロを買うとき、明日にはもっと良い機種がでるのでは、と悩まずにすむうまい買いかたは、死の前日に買うパソコンだけだ」と。

中山 照章

いずれは、SIGNETでこの欄のような情報を集めましょう。面白いメッセージは誌面にも発表することにして。

池沢 忠男

年間の文献をまとめてくれるところはありませんか。英語の論文までなかなか手がまわりません。

浦野 昇千

この3年間、毎日毎日 Mac のことばかり考えていまし

た。その Mac がもうすぐやってきます！

咲間 繁

苦勞して、やっとツールを完成させたにもかかわらず、ちっとも普及しません。ツールは、どのようにすれば普及するのでしょうか？

野々下 孝治

実験データ（時系列データ、スペクトルデータ等の数値データ）のデータベースの勉強をしたいと思っているのですが、数値データベースの勉強をするためのよい参考書をご存知ありませんか？

日下 正敏

HELLO! ソフトウェア・ショウの会場で入会。

噂のSEAに入会できて、ラッキーでした。ネットワーク・アプリケーション、マイコン系開発、CAD/C G系開発と転職し、アーキテクチャとビジネスのLINKを目指して、新環境下でスタートの身。よろしくおねがいします。

松浦 智之

"Inside Mac" という文献を探しています。

伊藤 恵一

シグマの会議に出たときのこと。「この問題については、各自の宿題にします」といわれ、その「宿題」という響きが懐かしかった。仕事上での納期というのは遅れがちだげど、宿題というと絶対にやらなくてはならないという感じがする。でも、学校にいていたころはどうだったっけ。

諏訪 茂男

7月に Mac Plus を購入しましたが、かなりのカルチャー・ショックを受けました。ほとんど MAc のスイッチを入れない日はないくらいです。

Inside Mac も Pascal も持っているのですが、不思議となかなかプログラムを作る気にならないのです。将来、ソフトを作る人と、使う人の分業化が進行するのではな

ど思ったりしています。

もちろん Mac の使い心地は、J-star より上！

小沢 周平

システムが完成したとき、満足感と後悔をいつも感じる。100%満足のゆかないシステムへの後悔は、自分自身が進歩していることの証か、自己満足か。

松沢 英明

LISPコンパイラ作成法の文献を御存知の方は、御教示下さい。

大木 幹雄

ソフトウェア産業というと、人材育成会社からレンタルビデオ屋さんまで幅広い、ソフトウェア・サービスなどというところにさらに広くなり、かなりイカガワしい（私の好きな夜の）商売を含む印象を、世の人に与える。だから、コンピュータ・ソフトウェアと呼ぶべきなどというつもりは毛頭ない。システムのな思考で仕事ができるのであれば、コンピュータなどなくても、自分はソフト屋さんだと考える余裕を持ちたいものです。

武田 喜美子

女性会員のネットワークを作りませんか！

笠原 俊哉

休日の過ごし方について、SEA会員の皆さんは、どのように過ごされていますか？

野村 行憲

SEANET 構築を切望する。そうすれば、地方会員の私でもSIGに参加できよう。だから SEANET は、第2種パケットが利用可能（第1種パケット）が前提で、マルチ・プロトコルのサポートを希望する。当然、私の MAc で利用できなければ意味がない。

田中 康和

現在、ソフトウェアの品質や生産性の向上を何とかしなければと考えているのですが、これといって手だてがなく、暗中模索の状態です。やはり、家内工業的手法から抜け出せないのでしょうか。

沼田 千賀子

友人が在宅勤務を探しています。子供が2人いるため、常勤は厳しいとのこと。でも、世の中、在宅でできる仕

事はないのか。厳しい現状のようです。彼女のような優秀な技術者を放っておくこの業界、損をしていませんか？

鈴木 裕信

コンピュータは時間と空間をゼロに近づけてくれるはずなのに、なぜか私は、朝の満員電車に乗らなければならない。シグマのようなシステムは作れても、日本社会の土壌には根つかない。といったら、いいすぎか？

松村 好高

専門誌となると、ついつい技術的専門分野のトピックスが多くみられるが、大規模（とくに事務処理分野）の開発では、プロジェクトの運営・管理が大切である。つい今年の夏も火消しに狩りだされたが、皆どう思っているのだろう。本当に、現実では情けなくなることが多いようにおもえるが。

伊藤 正次

（株）タンデムで売られている PC-9801 用の Smalltalk-80 を購入し、使ってみての一言：これほどひどいものとは思わなかった、すぐにおちる。

近藤 直文

UNIX 端末に、パソコンを使用しており、カナ漢字変換をパソコンでローカルに行っています。しかし、パソコンのカナ漢字変換ソフト（VJE-α、一太郎の ATOK）では、vi を使う時に、アスキー文字コマンドと漢字の切り換えが繁雑で困っています。NCRやDECの変換ソフトのように、常時アスキー文字入力で、変換キーを押したときだけカナ漢字変換をするソフトで、PC-9801で動くものがあれば教えてください。

編集部から

今後この欄への投稿は、お気軽に（たとえば葉書などで）、どんどんお寄せください。また、投書者とコンタクトを取りたい場合は、事務局までお問い合わせください。

S E A これからの活動予定

◆10月 (October)

28日 (火)

教育フォーラム「新入社員教育」(青年会議所)

31日 (金)

NISA・SEA情報化月間記念フォーラム

「これからのソフトウェア環境を考える」

(長野県長野市)

第7回YDOC(神奈川県労働福祉センター)

◆11月 (November)

7日 (金)

プロジェクト・アラニング・フォーラム

(青年会議所)

法的保護分科会第4回研究会

(ソフトウェア流通促進センター)

12日 (水)

再利用分科会第9回研究会(機械振興会館)

関西支部AI分科会研究会(JIP関西支店)

13日 (木)

AI分科会第6回月例会(機械振興会館)

14日 (金)

管理分科会月例会(高円寺SEA事務所)関西支部

月例会(SRA関西支社)

19日 (水)

環境分科会月例会(機械振興会館)教育分科会月例会

(日本電子専門学校)

26日 (水)

再利用分科会第9回討論会(機械振興会館)

31日 (金)

第8回YDOC(神奈川県労働福祉センター)

◆12月 (December)

6日 (木)

AI分科会第6回月例会(機械振興会館)

19日 (金)

SEA設立1周年記念パーティ

または

海外視察結果の報告フォーラム

+ 忘年パーティ

(いずれにせよ会場は未定)

S E A 会 員 状 況

昭和61年10月13日現在の入会会員状況は以下の通りです。

正会員—598名(先月比63名増)

賛助会員—18社(先月比4社増)

<地域分布>

<男女分布>

勤務 居住 男 = 563

女 = 35

北海道 = 1 1

岩手 = 2 2

新潟 = 5 5

栃木 = 3 3

茨城 = 3 6

埼玉 = 8 51

千葉 = 10 42

東京 = 412 244

神奈川 = 37 119

長野 = 6 6

富山 = 1 1

静岡 = 4 4

岐阜 = 1 3

愛知 = 8 6

滋賀 = 1 2

石川 = 1 2

京都 = 5 10

大阪 = 52 41

奈良 = 0 2

兵庫 = 18 23

福岡 = 3 3

熊本 = 5 5

長崎 = 1 1

鹿児島 = 3 3

<年齢分布>

20-24 = 30

25-29 = 131

30-34 = 162

35-39 = 160

40-44 = 60

45-49 = 30

50-54 = 9

55-59 = 6

60以上 = 5

会員の名簿管理は、パソコンを使用しておこなっています。住所、会社部署等の変更があったときは、なるべく早く事務局までお知らせください。



ソフトウェア技術者協会

〒166 東京都杉並区高円寺南1-5-4 高円寺サンハイツ404

TEL.03-312-3256 FAX.03-318-3909