



SEAMAIL

Monthly Newsletter from
Software Engineers Association

Volume 1, Number 9 | September 1986

目 次

編集部から		1
情報処理技術者試験をめぐる誌上討論		2
会員の声（アンケート集計）		2
自分のための情報処理試験	佐藤 千明	4
情報処理技術者試験は無意味か	川合 胤臣	5
情報処理技術者試験の是非	加藤 周	6
誌上フォーラムのこれまで、これから		7
LISP の日本語化	杉田 研治	8
ソフトウェア・シンポジウム 87 論文募集		13
中堅技術者教育について	中園 順三	14
環境からみたドキュメントの再利用	渡辺 雄一	16
新しいパーソナル環境をめざして	熊谷 章	18
LISP 事始め	赤坂仁志・梅林信之	20
ネットワークの苦楽	栗原 正利	22
高度情報化社会と消費者問題	佐原 洋	24
アメリカ・中国雑感	村井 進	25
幹事会、分科会および支部活動		26
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		
カレンダー		30
ICSE Tools Fair 募集		31

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： 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 白井豊 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. 9 昭和61年9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

SEAMAILの9月号をお届けします。船便シンドロームからいまだに脱出できず、印刷仕上がりは9月末、皆様のお手元に届くのは10月にずれこんでまいそうです。しかし、恒例の誌上フォーラム、分科会からの技術報告、そして会員からの投稿と、内容的にはますます多彩なものになってきたと思います。

誌上フォーラム

今回とりあげた討論テーマは、毎年10月に行われる「情報処理技術者試験」です。

これは、春からの企画で、予定ではもっと多くの方に書いていただくことになっていたのですが、原稿締切まぎわになって、お仕事の関係でキャンセルがあいつぎ、結局3人だけになってしまいました（教訓：やはり早めに原稿を集めておくこと）。そのかわりアンケートのほうは、これまでにない数が集まりました。

フォーラムは読むだけのページではありません。ぜひ、読者からの反応を編集部あてにお寄せください。

幹事会、分科会および支部報告

今月の分科会および支部からの報告も、誌面の都合上、今後の予定を中心に掲載しました。

分科会、支部での成果報告は、メンバーの方からの寄稿というかたちで掲載していきたいと思っていますので、この欄では、活動結果の報告と、今後の活動予定の広報を中心にしていきます。

分科会によっては、各会員には、案内状をだしていませんので、SEANET (SEA ネットワーク)が全国ネットされるまでは、SEAMAIL 誌上で各分科会、支部の活動をお知らせします。しかし、何分にも船便ですので、興味をお持ちの方は、直接幹事または世話人のほうまでお問い合わせください。

会員からの投稿

今回は、会員からの投稿が多くありました。まだまだストックがありませんので、今回のように多くの方からの投稿を待っています。

分科会および支部からの成果報告

環境分科会、教育分科会、AI分科会、そして横浜支部からの活動成果報告を収録しました。分科会、支部共に活動が定着してきましたので、これからも、こうした成果報告を数多く掲載していくつもりです。

代表幹事からの御挨拶

代表幹事の鈴木弘さんからの、原稿を今月号こそ掲載する予定でしたが、AIビジネス関連の新会社設立に関係されていて、目の回るようにお忙しく、今回もまた間に合いませんでした。

次号にこそ、かならず書くと御本人がおっしゃっておりますので、もうすこしお待ちください。

次号予告

次号、10月号は、秋のセミナー・ウィーク報告特集 (Part I) になります。これは、春のセミナー・ウィークの報告特集を4月号でおこないましたが、セミナーに参加できなかった方ばかりでなく、参加者からも好評でしたので、引き続き特集するものです。

ただ、秋のセミナー・ウィークは、全部で21セッションありましたので、各セッションからのレポートを3回に分けて掲載する予定です。

また、成功裡に終わった盛岡ワークショップの報告も、たぶん次号にまにあうと思います。

イベントの案内

今月は、ページ数の関係で、これから暮れにかけて予定されているいくつかのイベントのご案内をのせることができませんでした。

カレンダー (P. 30) には一応入れてありますが、詳細似ついでのお知らせは、別途パンフレットをお送りするかたちにします。

情報処理技術者試験に関する誌上フォーラム

— 会員の声 —

ここに集められた会員の声は、「情報処理技術者試験に関するアンケート」として、この春SEAMAILと一緒に会員に郵送したり、フォーラムの会場で参加者に配布したりしたものの回答を集計したものです。

回答は、全部でちょうど100通集まりました。こうしたアンケートの常として、部分的に無回答があるために、合計の数字が回答者数とあわなくなることもありますが、各設問ごとに機械的に回答を集計しました。

アンケートの自由回答欄に記入されたコメントは、その中で代表的なものやユニークなものを編集部で勝手に選び、無記名のかたちで収録しました。また、回答分析とまではいきませんが、設問ごとに、編集部からのコメントを載せてあります。

設問1. 情報処理技術者の資格を持っているか

- ・ 持っている——66人
- ・ 2種——46人
- ・ 1種——43人
- ・ 特種——21人
- ・ 持っていない——34人
- ・ 受験したことがある——10人
- ・ 受験したことはない——18人

持っているというところで、複数回答がありますが、1種と2種の併取がおおいようです。つまり、2種を得てから、1種をとという傾向です。

持ってる、という回答が全回答数の約6割になっています。業界全体での数字は明確ではありませんが、この比率は、業界の水準とくらべると、かなり高いと思われます。

設問2. 資格を持って、何かメリットがあったか

- ・ あった——44人
- ・ 手当が支給された——24名
- ・ 一時金が支給された——8名
- ・ (再)就職に有利——5名
- ・ 客先との折衝で有利——5名
- ・ 自己満足——1名

- ・ ない——22名

設問3. 資格がないことで、何かハンデがあったか

- ・ あった——1名
- ・ 手当がつかない
- ・ ない——22名

設問2と3の回答をみる限りでは、資格を持っていることの「現実的」なメリットは、手当や一時金という物質的なものが大多数であり、逆にこうした金銭的なことを除くと、資格がないことによって、実作業では、あまりハンディはない、という結果になっています。このへんの実態がどうなっているのかを知るには、もっと細かなデータを集めなくてはならないでしょう。

設問4. 過去に1度でも受験して

(A) 試験のやり方についての意見

- ・ いまのままでよい——21名
- ・ 多少変えたほうがよい——27名
- ・ 1種と特種を同じ年に受験できるように
- ・ 現実的でない
- ・ 言語選択を広げる
- ・ 合格点が不明確、公表すべき
- ・ プログラミングの問題がよくない
- ・ 最新技術の問題が少なすぎる
- ・ 汎用機に重点が置かれすぎ
- ・ EDP系の問題がおおすぎる
- ・ 問題内容が古い、バッチ時代の技術がメインになっている
- ・ 微積分は必要ない
- ・ 時間が短すぎる(とくに1種・特種)
- ・ 回答形式を変える
- ・ 根本的に見直すべきだ——22名
- ・ 暗記型の答を問うのではなく、解決へのアプローチをみられる方式にする
- ・ ペーパを廃止して、端末でおこなう
- ・ 実務に則した問題にする
- ・ 言語を見直す(PASCAL, Ada, C等を追

加する)

- ・ 特種の年齢制限をやめる
- ・ 選択をもっと増やす

これらのコメントからみると、根本的に見直せという意見と、部分的でもやり方を変えたほうがよいという意見が多いようです(両方で約6割)。ここであげられている現在の試験の問題点の多くは、試験の内容がソフトウェア開発の現場に合っていないというニュアンスになっています。このことは、資格をもっていることのメリットが手当や一時金といった金銭によるものがほとんどで、実務に役立っているわけではないという回答と、一応論理的なつじつまはあっているように見えます。

(B) 受験した理由

- ・ 自発的に-----52名
- ・ 会社からいわれて-----24名
- ・ 友人などにさそわれて----1名
- ・ その他-----2名
- ・ 金のため

自発的にという人が約半分いるのは、プログラマの自己啓発に対する関心の高さを示すものだといえましょう。

設問5. これまで受験しなかった理由

- ・ そんな試験があることを知らなかった---なし
- ・ なんとなく受けそびれて-----5名
- ・ 忙しくて受験する暇がなかった-----5名
- ・ 意図的に受けなかった-----9名
- ・ 専門技術の勉強だけで十分
- ・ 試験と仕事の内容にズレがあり無意味
- ・ 関心がない
- ・ 向上心が不足している
- ・ 必要性がない
- ・ 資格そのものがナンセンス
- ・ メリットがない

さすがに試験そのものの存在を知らない人はいません。受験しないでいる人の半数が、意図的な理由をあげています。

設問6. 機会があれば今後も受けるか

- ・ 受けない-----32名

- ・ 受けたくない-----13名
- ・ どちらともいえない---31名

設問7. この試験制度そのものについて

- ・ いまのままでよい-----34名
- ・ 多少変えたほうがよい---21名
- ・ 年に数回実施する
- ・ 論文を重視する
- ・ もっと実践的にする
- ・ 採点基準と正解を公表する
- ・ 合格ラインを規定点数に変える
- ・ マイコン、パソコン分野にもウエイトを置く
- ・ もっと難しくすべき
- ・ 知識は消滅するので、資格をサポートをする講習と再試験の制度化
- ・ 分野別にする
- ・ 根本的に見直すべきだ---15名
- ・ 社会的な資格にする
- ・ 定期的に再テストをして更新制にする
- ・ 専門知識についての幅をもっと持たせる
- ・ 技術者の地位向上に役立てるような制度化
- ・ ○×、選択方式では、現実的な問題解決能力は計れない
- ・ 教養と専門知識の資格を分ける
- ・ 廃止すべきだ---8名

制度そのものの变革を望む声は、この設問での有効回答の5割をこえています。そして、その理由は、設問4の(A)と同じようなものになっています。

設問8. 他の技術者試験を知っているか

- ・ はい---46名
- ・ 技術士
- ・ システム監査
- ・ マイコン技術者
- ・ 工事担当者
- ・ 通信技術士
- ・ CISA
- ・ それを受験したことはある---7名
- ・ それを受験したことはない---33名
- ・ いいえ---32名

自分のための情報処理試験

佐藤 千明

長野県協同電算

1. 私と試験との関わり

私が最初にこの試験を知ったのは、昭和48年大学3年の時であった。当時受験するなら2種であっただろうが、その資格の価値がわからず受験しなかった。

入社後数年して、会社の方針ということで、全社的な教育・受験対策がとられた。「勉強すれば受かる。しなければ落ちる」といわれていたとうり、1回めはなんの勉強もせずに、見事に不合格となった。生まれて初めて(?)経験した不合格のくやしさをバネに、翌年猛勉強し、なんとか1種に合格できた。

その後、長野の田舎に移ったこともあり、しばらくは試験のことを忘れていたが、自己啓発の目標として、また合格一時金の小遣い稼ぎに、そして「仕事できます。資格ありません」を脱皮するために特種を受けた。時すでに32才であったが、若い頃の知識と年相応の経験が役に立って、なんとかすべり込んだ。

2. 資格と実務能力に相関関係はあるか

資格毎に、かなり主観的な感想を述べさせてもらう。

(1) 2種

これだけ受験者が多くなると、合格率が一定なら、合格者の相対レベルも下がってくるのではないか。アマチュア用であり、プロなら合格して当たり前と思う。何年やってもダメな人は、適性なしと反省して、転職したほうがいいのではないか。

しかし、そうすると技術者不足が増々深刻になり、当面の仕事を質より量でこなすしか手はなく、システムの信頼性が増々危うくなる。その意味で、教育制度の充実、ソフトウェア工学・産業の確立が望まれる。

(2) 1種

プログラマにとって必要条件ではあるが、決して十分条件ではない。つまり、この資格だけでは食えず、「資格あります。仕事できません」では話にならない。しかし、まわりの合格者をみると、一般的には有資格者は仕事ができるようだ(無資格者は仕事ができないとは決していわない)。

午後の問題は、時間をかけてじっくり考えれば解けることから、1種はある意味で頭の回転状況を調べる試験

といえる。そして、知的好奇心旺盛で、多忙の中で受験勉強を続けられる強靱な意志を持っている人が合格できるとしたら、「資格あります。由に仕事もできます」は、一般論として成り立つと思う。

(3) 特種

幅広い教養と豊富な経験が必要とされる。午前問題は幅が広い分だけ、1種よりやさしいかもしれないが、最新の技術動向が問われるので、日頃から問題意識を持った情報収集が必要とされる。午後の前半は、現実業務に即した問題が多く経験がものをいうし、論文も経験なくしてデッチ上げるのは容易ではない。逆にいえば、豊富な経験があればなんとかなるともいえ、1種はダメでも、特種は合格というケースもあると聞く。

システム全体に関する知識と経験を持っているとすれば、有資格者にはそれなりの評価をしても良いと思う。ただし、プログラミングという職業領域では、必ずしも、特種が1種より上級とはいいい切れないことも事実である。

3. 誰のための資格試験か

確かに、会社は社員に資格を取らせ、その成績によって自社の優秀性を示し、事業を有利に導こうとする。これは企業として当たり前であるし、このことを論じてみても仕方ない。会社のためにいやいや受けたとしても、合格すればなんらかの報奨があるはずだし、本人だって悪い気はすまい。最後は本人にプラスとなるのだ。

試験問題、試験勉強が実務に役立たないという意見もある。そういう人はかなり専門的、または特定業種に偏った仕事をしているのだろうが、情報処理技術者と呼ばれるからには、相応の幅広い、新しい知識は不可欠であろう。企業内教育体系の整備が遅れているため十分な研修を受けられない今、この試験をとうした自己啓発は、貴重な財産となり、将来の自分を支えることになる。

4. さいごに

中小企業診断士、技術士(共に情報部門)、CISA試験、システム監査技術者試験等、うんざりするほどのサムライ試験が待ち構えているが、上記資格はプログラマ定年後でも有効であろうから、ライフワークとして挑戦してみようと思う(恐らく受からないだろうけど)。

情報処理者技術試験は無意味か

川合 胤臣

ヒューリス

1. 私の場合

小生は10年ほど前に第1種を受験し、2回目で合格しました。顧客の担当者がやはり受験しており、彼らの情報処理に関する知識、スキルに遅れをとってはいけないというのが、受験の理由でした。ソフトウェアの開発を自らの職業としてやってきて、どれ位の能力なのかということを、チェックしてみたかったことも、理由のひとつでした。

資格を持っていて、直接何らかのメリットはありませんでしたが、客先からの評価ということでは意味がありました。とくに初対面の場合は、「1種をもっている」ということで、安心をする顧客もいました（もちろん評価しない客もたくさんいましたが）。

合格して10年もたつと、昨今の問題を見ても難しく、とても今では無理でしょう。有資格者は何年かに一度再試験を行い、レベルのダウンを防ぐ方法もあるのですが、正直のところ、仮にそのように制度化されても再試験はしないでしょう。

当社でも、若い技術者の受験率は高いのですが、SEレベルの人達はあまり関心がありません。日々の業務に追われて勉強する時間がないこともありますが、試験内容が知識偏重になってきていることも問題で、そこには受験テクニックの競争ということも生じます。詰め込みで勉強すれば合格できる、あるいは傾向を分析しておかないと合格できないということも問題です。

2. 社会的な位置付け

企業において、ソフトウェア技術者の人事考課はたいへん難しい。製品としてのソフトウェアの価値をどのように評価するかという問題と不可分ですが、現在の試験を全面的に人事考課に採用するということは考えられません。

ソフトウェア技術者に要求される能力として、基礎学力（英語、数学、文章読解、文章表現）、ソフトウェア／ハードウェアの知識、業務知識が必要ですが、現象や状況を整理し、論理的な関係を導き出す力、あるいは協調性、ねばり強さ、交渉力、体力（？）など、とても1つの試験で評価できるものではないし、ましてや絶対値

でランクづけすることも不可能です。

情報処理技術者試験の有資格者の社会的評価を、医者や弁護士、建築士と比べると、確かに低いことは事実でしょう。たとえば、建築士の場合には、設計・施工の管理について法律的に裏付けられていますが、情報処理技術者の場合にはなにもありません。

建築士の場合には、構造・計画・環境・法規（建築基準法ほか）などの学科試験と、製図の実技試験があり、学科試験をパスしても実技で失敗したときは、翌年、実技だけ受験できるという試験制度になっています。

あるとき建築士である友人にたずねたことがあります。「なぜ、建築士は法律的にその資格が裏付けられ、有資格者でないと設計できないのか」。しばらく考えてから友人は、「人の安全や都市の環境美観に関するからだだろう」との返事。「冗談じゃない。ソフトの開発だって人の安全、財産の保全に関するものはたくさんある」。結局、その場では、その違いは法律的に裏付けられている、という事実だけが原因であり、なぜ法で規定されているのかという理由は、明らかになりませんでした。

3. ソフトウェア士への登龍門

それではもし、ミスがあれば人命にかかわるソフトを開発するときに、建築士と同じようにオーサライズされた、仮称「ソフトウェア士」に依頼するでしょうか。建築基準法と同じように、「ソフトウェア基準法」なるものがあり、法でソフトウェアの開発や、品質が規定されていればそうなるでしょう。

ここまで考えると、「ソフトウェアをどう評価するか」という問題に発展していきます。ソフトを開発している現場から、もっと世間に、自分達のしている仕事を宣伝し、認知させることにより目を向けさせることが大事なのではないのでしょうか（これはまさにSEAの発足の主旨ではないか）。

現在の試験制度は、若い人達のとりあえずの目標になっているのですから、これを無意味とは考えません。「ソフトウェア士」への登龍門（一里塚）と位置づける必要があります。しかし、「ソフトウェア士」の試験科目とは、どのようなものなのでしょうか。

情報処理技術者試験の是非

加藤 周

松下電送

1. 基準としての必要性

現在、日本の社会のいろいろなところで、情報処理システムが使われており、今後ますます多くのシステムが稼働することになる。この需要に答えていくためには、情報処理技術者1人あたりの生産性を上げていくと同時に、システムをまとめあげていくためのチーム全体の生産性を上げていく必要がある。

このチームを組む場合、いつも同じメンバーで組める場合もあるが、あるプロジェクトで初めて出合うメンバーの場合も多い。チームを組んだ場合に、メンバー各員がおたがいにうまくやっていけるかどうか、そのプロジェクトが成功するか、多大な影響を与える。

メンバー各員がおたがいにうまくやっていけるには、人格的な側面と技術的な側面が、非常に重要な要素である。特に、チーム全体のまとめ役であるマネジャ、チームの推進役であるリーダー、システムを技術的に整合のとれたものとするために設計役となるリーダー、これら三者が要となって、メンバー相互が人格的にも技術的にも充分に理解し合って、プロジェクトを進める必要がある。

あるシステムが作られる時、それに参加する技術者をみると、発注者になるユーザ、それをまとめあげるシステム・ハウスの技術者、ハードウェアや基本システムを運用するために作られたアプリケーション・ソフトウェアを開発するソフトウェア・ハウスの技術者、以上4者に属する技術者の混成チームの場合もかなりある。

このような混成チームを構成する各技術者の技術水準を、各企業間をこえて設定しているのが、現在の情報処理技術者試験の制度である。したがって、混成チームを技術的に維持している最低限のメジャーが必要となり、基礎的な能力を確認する方法として現実的なものであると思う。

2. 情報処理システムの社会性

現在の情報処理技術者試験の制度では、合格するかしないかは、実際の業務上ではなんの関係を持ってはいない。たとえば、いろいろなコンピュータ雑誌から見付けてきた回路図を接ぎ合わせてハードウェアを作ったりとか、システム全体が見えないままに、各部分を請けお

たソフトウェア・ハウスで作ったソフトウェアで運用されているシステムが、情報処理の基礎知識すら持ち合わせない人々によって作られる場合もよくあることである。

しかし、こうしたことは、健全な情報社会の進み方とは考えにくい。現在すでに、情報処理システムは、社会的にパニックを起こすだけの影響力を持ったシステムとなっている。

自分が住む木造の家なら自分で設計する場合もあるが、普通は建築士の資格を持った人に依頼して設計する。そして設計したものには、責任を持つことになっている。それでは、今流行のCADを使用して設計した場合に、システムに不具合があった場合に、建築士は機械の調子がおかしかったからと、責任のがれを言い、CADの設計者も責任を感じることはないのだろうか。

危険物取り扱いには試験が必要でも、ダムを監視制御するコンピュータ・システムを作る情報処理技術者には何の資格も必要とされていない。情報処理技術は人によってかなり適性があることは周知の事であるが、その適性の確認すら行われなくて、このような社会に、重要なシステムに拘りを持つことになるだけでなく、リーダーやマネジャすらなることは、決して稀ではない。

3. どう運用すべきか

結論として、情報処理技術者試験の制度は、技術者の適性と基礎知識の確認を企業を越えて行うものとして運用していくべきであろう。

次に、情報処理システムを作り保守するチームは、この試験の合格者で編成する。もし、人数的に不足する場合は、マネジャやリーダーは最低このような技術者をあてることとする。

したがって、このように技術者が作ったシステムが社会的に問題を起こさないような人が合格するように、試験の内容を選択していく必要がある。この制度をこのように運用してはどうだろうか。

いずれ、社会的にそれほど重要でなくなるまで国家試験として、それほど重要でなくなったら民間団体に移していくようにすればよい。

誌上フォーラムのこれまで、これから

SEAMAILの誌上フォーラムは、特定の話題に関して、会員どうしがお互いに議論をたたかわせたり、意見を交換しあったりするためにあります。SEA設立のそもそもの趣旨に照らせば、この機関誌のメイン・コラムだと考えてもよいでしょう。

1. これまで取り上げたテーマ

創刊以来これまでの号では、編集ボランティアのアイデアや会員から寄せられた意見などにもとづいて、次のようなテーマを取り上げてきました。

- ★ シグマ・プロジェクト（第1号）
- ★ AIとソフト工学（第2号）
- ★ プログラマ定年説（第4号）
- ★ SDI論争（第5-6号）
- ★ ソフトウェア産業論（第7号）
- ★ 情報処理技術者試験（第9号）

それぞれのテーマに関する執筆者は、原則としてアンケートなどを通じて会員から公募することになっていますが、それではなかなか原稿が集まらないので、やむなく編集部の方で適当な人を見つけて依頼しているというのが正直な実態です。

2. 今後の予定テーマ

今後とりあげることを予定している誌上討論のテーマには、次のようなものがあります。

★ パソコン・ソフトをめぐる

権利保護、不法コピー、プロテクトにからむ技術的および社会的なことがら、価格と性能の対比、品質保証などなど、パソコン・ソフト特有の問題点を討論しようという企画です。

SEA会員のなかには、パソコン・ソフトの開発や販売に携わっている人もいれば、純粋なユーザも数多くいるでしょう。そうした相異なる立場からの意見や問題提起を、編集サイドでは期待しています。

★ 女性技術者から（へ）の提言

SEA会員のなかには占める割合はまだ5%程度ですが、最近どの職場でも、女性技術者の進出はめざましいものがあります。いかに男女雇用平等法が成立した世の中だとはいえ、こうした傾向はいろいろな問題をもたらすでしょう。すでに、2年まえ三重県合歓の郷

で行われたJISA/STCの「女性技術者ワークショップ」でも、それらの問題を予見させるような討論が行われています。

そこで、SEAMAILとしても、われわれの世界における女性技術者の問題を幅広く取り上げてみたいと考えます。とりあえず、この号に同封したアンケートにお答えください。

★ ネットワーク時代のソフトウェア開発

上滑りのVANブームも過ぎ、いよいよ日本も本格的なコンピュータ・ネットワークの時代に突入しようとしています。そうした社会環境の変化は、ソフトウェア開発の現場にどんな影響を与えるのか、われわれ自身のライフスタイルは、どんなふうが変わって行くのでしょうか。

まずは、SEAセミナー21でのネットワーク関連のセッション・レポートあたりを手掛かりに、みんなでこの問題を考えてみたいと思います。

3. 投稿および提案大募集

上記の各テーマに関する読者からの投稿をお待ちします。原稿の長さは、SEAMAIL仕上がりで1ページ（約2千字）を1単位と考えていますが、それより長くても短くてもかまいません。編集部（SEA事務局）まで、どしどしお寄せください。

また、これらの候補テーマ以外に、なにかよいアイデアがあれば、ぜひご提案ください。その場合、テーマ名だけでなく、討論すべき問題点のリストを一緒につけていただければ幸いです。あなた自身の問題提起をいきなり原稿のかたちで送っていただいてもかまいません。さらに、これまで掲載してきた誌上フォーラムの論文に対する反論や意見もお待ちしています。

前号に同封したアンケートにも書いたように、いまのところSEAの活動（分科会やセミナーなど）は、どうしても会員数の多い東京が中心になり、地方在住の会員にとっては、このSEAMAILが唯一のコミュニケーション・メディアです。したがって、フォーラムにかぎらず、地方会員からの投稿は優先的に扱うことにしたいと思います。ふるって原稿やご意見をお寄せください。

Franz LISP の漢字機能拡張

杉田 研治

協同システム開発

1. 概要

BSD UNIX上のリスプとしてよく使われている Franz LISP に漢字機能の拡張を行った。UNIX はアスキーの日本語 UNIX、稼働マシンは VAX11/780、Franz LISP のバージョンは 38.69 です。この拡張機能は、まだ確認していませんがワークステーション Sun でも使用可能なはずで、また、シフト JIS 系の OS でなくとも使用可能にすることはきわめて容易です。

これらの拡張機能を、Unix のライセンスを持っている方に実費で提供します。その手続きについては、最後をみてください。

表1に Franz LISP に関するプログラムとその機能と、さらに、今回の漢字機能を拡張したプログラム名を示します。この表1に示した、全てのプログラムで漢字機能を使用することができます。

プログラムとその実行のスキプトの例を2つ、このペーパーの最後に掲載してあります。

lisp	j!isp	インタープリタ
liszt	j!iszt	コンパイラ
lxref	j!lxref	クロスレファランス・ジェネレータ
pearl	j!pearl	データベース

表1 Franz LISP 関連プログラム

2. 漢字拡張機能の概略

シンボル名の扱い、関数、入出力と制約条件に分けて、今回実現した漢字拡張機能の概略、および実現方法を以下に説明する。

(1) シンボル名の扱い

漢字(2バイト)とアスキー(1バイト)を混在して使用することができます。

例えば、

```
-> (progn (print '漢字kanji) (terpri))
漢字kanji
nil
```

```
-> (exploden '漢字kanji)
(35519 36506 107 97 110 106 105)
-> (maknam (explode '漢字kanji))
漢字kanji
->
```

のように使うことができます。

シンボルは、プリント・ネーム、バリュー、関数とプロパティ (パッケージがある場合は、これも含む) という属性を持つ抽象概念です。従って、プリント・ネームさえうまく取り扱うことができれば、シンボル名と関数名で漢字を使用できるようになります。また、シンボルの生成と印字は、いつもリード・テーブルを介して行われるので、リード・テーブルを参照するタイミングでフックすれば、漢字を取り扱えるようにすることができます。

オブリストに登録する際に使用されるハッシュ関数は、処理上問題がないので、そのまま使用しています。漢字1文字の長さをアスキー文字の2文字分にしておくことにすると、関数や入出力の際の整合性もすべてとれます。

漢字コードは、リスプ・インタープリタ内でもファイル内でもシフト JIS コードを使用しているので、explode したシンボルが maknam でできないということはありません。

(2) 関数

後述する制約条件に示す関数以外で、漢字を使用することができます。

(3) 入出力

ここで問題となるのは tyi, tyo や exploden での漢字の表現方法だけです。表現方法は、単純に次の式で計算しています。

$(\text{unsigned})\text{上位バイト} \times 256 + (\text{unsigned})\text{下位バイト}$

漢字コードがシフト JIS で漢字1文字を長さ2で扱うことにしたので、文字列の長さ、入出力の文字長の管理も漢字機能拡張前のままで、差し支えありませんでした。

その使用例を次に示します。

```
-> (tyi)あ
33440
-> (tyo 33442)
い33442
-> #/う
33444
->
```

(4) 制約条件

実用上問題のない程度の制限しかありません。これらの制限は、ガードしようと思えば簡単にできます。

(a) 現在次の関数は漢字の使用はできませんが、近々にインプリメントする予定です。

- ・ untyi
- ・ tyipeek

(b) 次に示す関数は漢字を使用できません。 Zetalisp でもこれらの関数は使用できないのと、実用上で必要性がないと思うので、インプリメントする予定はありません。

- ・ setsyntax
- ・ getsyntax

(c) 効率が悪くなるので1バイトの仮名文字は使用できませんし、その入力のガードも行っていない。そのために、1バイト仮名文字を入力すると結果は保証されません。また、シフト JIS 漢字コードの範囲のチェックも、同じ理由で行っていません。

1バイト仮名文字を使用できるようにすることも、ガードを行うことも簡単にできますが、漢字が使用できれば、1バイト仮名文字を使用することはないし、漢字と1バイト仮名文字とを混在して使用することはないので、インプリメントもガードも必要だと思いません。

5. 変更ヶ所

変更ヶ所は大きく分けるとリーダー、プリンター、数の3つになります。変更ファイルは、いずれも Franz LISP インタープリタのカーネル部分のファイルのみです。以下のファイルが、変更をしたものです。

```
io.c      lam5.c  lam7.c
lam2.c    lam6.c  lam8.c
```

ただし、ローカルな環境に Franz LISP をインストールするためには、次のような修正が必要です。

以下のメイク・ファイルの中のパス名

```
.../usr/src/ucb/lisp/franz/vax/Makefil
.../usr/src/ucb/lisp/franz/Makefile
.../usr/src/ucb/lisp/liszt/vax/Makefile
.../usr/src/ucb/lisp/liszt/Makefile
.../usr/src/ucb/lisp/utlis/Makefile
.../usr/src/ucb/lisp/Makefile
.../usr/src/ucb/lisp/pearl/Makefile
.../usr/lib/lisp/Makefile
.../usr/lib/lispのfix.lとtoplevel.lのライブラリー・パス名とバージョン名
.../usr/src/ucb/lisp/lisztのtleve.lのバージョン名
.../usr/lib/lisp/autorun/vaxのインタープリター・パス名
```

これらの変更点を以下に簡単に説明します。

(1) リーダーとプリンター

修正で一番面倒なのがこのリーダーとプリンターですが、ファイルは io.c を修正するだけです。リーダーの修正には MSB をみて、漢字なら2バイト分データを取り込むようにします。リード・テーブルの属性に従って、エスケープ文字、マクロ・キャラクターの処理等をひとつずつ修正しました。

プリンターの修正には、漢字コードの出力のときに、ワイアード・アトムやエスケープ文字がでないようにするところが注意点です。

(2) 関数

Franz LISP の場合は、カーネル・ファンクションが sysat.c で、ファンクション・セルにバインド(マクロ関数 MK で)されるので、ここで MK されている関数についてのみ、漢字コードとの整合性をチェックすれば問題はありません。

カーネル部分の関数で基本的な機能は、全て網羅されているので、リスプで書かれた部分の修正は全く必要ありません。そして、これらの関数は rawlisp に組み込まれるので、rawlisp のみコンパイルするようにすれば、デバッグのレスポンスはかなりよくなります。修正の際にチェックした関数を以下に示します。

print	patom	pnrlen	read
ratom	readc	implode	maknam
concat	uconcat	nwrtn	flatc
alphalessp	tyi	tyipeek	untyi
tyo	termcapinit	termcapexe	int:setsyntax
int:getsyntax	aexplode	aexplodec	aexploden
ascii	readlist	gensym	cprintf
sprintf	substring	substringn	

メディアとして、1200 フィート以上の 1/2 インチ・テープを、返信用の切手を貼った返信用封筒といっしょに、下記の所へ送ってください。

郵便番号：105
東京都港区虎の門1-14-1
郵政互助会琴平ビル 3F
(株)協同システム開発
杉田研治

6. インプリメントの方法

漢字機能を拡張した Franz LISP のインプリメントの手順を以下に示します。

(1) Franz LISP をローカルなディレクトリーに再インストールする。

(2) `../usr/src/ucb/lisp/franz/*` のファイルを修正する。

Franz LISP の C 言語で書かれている部分の修正には、次のような注意が必要です。ローカル・スタック・ポインターとして、レジスターを2つグローバルに使用しているの、レジスターを使用する場合に、レジスター・セーブ・マスクを正しくなるようにしなければなりません。レジスター・セーブ・マスクは、組み込みのマクロ関数 (Savestack) で設定できます。

また、リスト・セルを作成する場合には、ガーベージ・コレクションに、そのセルがかからないようにプロテクトしなければなりません。これも組み込みのマクロ関数 (protect) で、プロテクトすることができます。めったにありませんが、ガーベージ・コレクションをオフにする必要があることもあります。

(3) `../usr/src/ucb/lisp/franz/vax` で `rawlisp` を作り、機能を確認する。

(4) いくつかの機能を確認したら、`../usr/src/ucb/lisp` で `make slow` を行う。

(5) うまく行くまで (2) から (4) を繰り返す。

(6) `pearl` をリコンパイルする。

7. 配布条件および配布方法

以上簡単に述べてきた、漢字機能を拡張した Franz LISP を、パブリック・ドメインにします。

(1) 配布条件

配布条件としては、unix のライセンスを持っていること。これは、機能拡張した Franz LISP が、BSD の unix に含まれているものだからです。

(2) 配布方法

8. 作業経緯と感想

この作業は、約1週間の検討と修正作業でできました (今年の6月の中旬)。ただし、Franz LISP のソース・コードと LISP をある程度しっかりと理解している必要があります。もちろん unix, C 言語, アセンブラに関する知識は必要です。

漢字機能を拡張しての一番の感想として、漢字が使えることによって親しみが増すことです。このことは、理解性が高まる (見てすぐに分かる) ことを意味しています。

9. 謝辞

実現にあたっては、梅林信之 (協同システム開発) 氏にコード修正のいくつかを手伝っていただきました。

10. 補足

Franz LISP の中で `curses` (もう古いパッケージですが) を使用することのできるパッケージも無償で提供できます。

実行スクリプト例-1

```
% jlisp
Franz Lisp, Opus 38.69 with Japanese
-> (ls)
[autoload csetup]
[load /staff/usr/kenji/lib/lisp/csetup.l]
/staff/usr/kenji/usr/lib/lisp/nld -N -x -A /tmp/Li2069.0 -T a0800.
jprog4.1 jprog5.1
0
-> (cat jprog5.1)
(defstruct (宇宙飛行士 :conc-name)
  国籍
```


年齢

性別

ヘルメット・サイズ

(好きな飲み物 '(ビール))

;;; (setq 熊 (make-宇宙飛行士))

(defstruct (プログラマー :conc-name)

(好きな飲み物 '(コーヒー ビール))

(好きな食べ物 '(ハンバーガー)))

0

-> (setq 熊 (make-宇宙飛行士 国籍 ソビエト))

Error: eval: Undefined function make-宇宙飛行士

<1>: `D

[Return to top level]

-> (load 'jprog5)

[load jprog5.l]

[autoload /staff/usr/kenji/usr/lib/lisp/struct]

[fasl /staff/usr/kenji/usr/lib/lisp/struct.o]

[fasl /usr/lib/lisp/machacks.o]

t

-> (setq 熊 (make-宇宙飛行士 国籍 ソビエト))

Error: Unbound Variable: ソビエト

<1>: `D

[Return to top level]

-> (setq 熊 (make-宇宙飛行士 国籍 'ソビエト))

{ソビエト nil nil nil (ビール)}

-> (宇宙飛行士-国籍 熊)

ソビエト

-> (setf (宇宙飛行士-年齢 熊) 26)

{ソビエト 26 nil nil (ビール)}

-> (宇宙飛行士-年齢 熊)

26

-> (宇宙飛行士-好きな飲み物 熊)

(ビール)

-> (setf (宇宙飛行士-好きな飲み物 熊)

(append1 (宇宙飛行士-好きな飲み物 熊) 'ウオッカ))

{ソビエト 26 nil nil (ビール ウオッカ)}

-> (宇宙飛行士-好きな飲み物 熊)

(ビール ウオッカ)

->

実行スクリプト例-2

-> (ls)

jprog4.l jprog5.l

0

-> (cat jprog4.l)

(defun みよちゃん ()

(princ 'なあに?)

(do ((x (read) (read)))

((null x) (princ 'さよなら) (terpri))

(print (eval x))

(terpri)

(princ 'なあに?)))

(defun 行く (ここ 向こう)

(cond ((equal ここ 向こう) (list 向こう 'に 'ついた))

((equal 'すごく遠く (距離 ここ 向こう))

(list '飛行機で ここ 向こう))

((equal '遠く (距離 ここ 向こう))

(電車で ここ 向こう))

((equal '近く (距離 ここ 向こう))

(list '歩く ここ 向こう))

(t 'わからない)))

(defun 電車で (ここ 向こう)

(cond ((and (equal ここ (駅 ここ))

(equal 向こう (駅 向こう)))

(list '電車 'に '乗る ここ 向こう))

((equal ここ (駅 ここ))

(list (list '電車 'に '乗る ここ (駅 向こう))

(行く (駅 向こう) 向こう)))

((equal 向こう (駅 向こう))

(list (行く ここ (駅 ここ))

(list '電車 'に '乗る (駅 ここ) 向こう)))

(t (list (行く ここ (駅 ここ))

(行く (駅 ここ) (駅 向こう))

(行く (駅 向こう) 向こう))))))

(defun 距離 (ここ 向こう)

(cond ((and (equal ここ '家)

(member 向こう '(みよちゃん 東京)))

'近く)

((and (equal ここ '横浜)

(equal 向こう 'おばあちゃん))

'近く)

```

((and (member ここ '(家 東京))
      (member 向こう '(おばあちゃん 横浜 成田)))
  '遠く)
((and (member ここ '(家 成田))
      (equal 向こう 'アメリカ))
  'すごく遠く)
(t 'わからない)))
->
nilnil
->

(defun 駅 (ここ)
  (cond ((member ここ '(家 みよちゃん 東京))
        '東京)
        ((member ここ '(おばあちゃん 横浜))
        '横浜)
        ((equal ここ '成田) '成田)
        (t '駅)))
0
-> (jliszt jprog4)
Compilation begins with Liszt vax version 8.36
source: jprog4.l, result: jprog4.o
みよちゃん
行く
電車で
距離
駅
%Note: jprog4.l: Compilation complete
%Note: jprog4.l: Time: Real: 0:8, CPU: 0:2.13, GC: 0:0.00 for 0 gcs
%Note: jprog4.l: Assembly begins
%Note: jprog4.l: Assembly completed successfully
0
-> (ls)
jprog4.l jprog4.o jprog5.l
0
-> (load 'jprog4)
[fasl jprog4.o]
t
-> (みよちゃん)
なあに? (行く '家 'おばあちゃん)
((歩く 家 東京) (電車 に 乗る 東京 横浜) (歩く 横浜 おばあちゃん))
なあに? (行く '家 'アメリカ)
(飛行機で 家 アメリカ)
なあに? (行く '東京 '横浜)
(電車 に 乗る 東京 横浜)
なあに? (行く '東京 'アメリカ)

```


論文募集

ソフトウェア・シンポジウム '87

1987年6月4-5日 ◇ 農林年金会館（東京・虎ノ門）

共催：情報サービス産業協会 (JISA)
ソフトウェア技術者協会 (SEA)

、はやいもので、ソフトウェア・シンポジウムも第7回目を迎えます。産業界の現場で、日頃忙しくソフトウェアの開発・保守・運用・管理に従事しているエンジニアたちが、それぞれの仕事のなかから得た経験や知識、アイデア、あるいは意見を交換しあうための貴重な場として、内容も年々充実してまいりました。

ここ1-2年、ネットワークの整備やワークステーションの普及、あるいは各種先進的技法／ツールの実用化など、ソフトウェア業界をとりまく状況は、急激に変わりつつあります。今回のシンポジウムでは、こうした新しい社会的・技術的環境への対応を発表や討論の主題として、従来にもまして魅力的なプログラムを組みたいと考えております。

実践的技術の経験、先進的技術の試行結果、基礎研究の成果等に関して、ヴァリエティに富む多数の論文がよせられることを期待いたします。

募集論文テーマ

特に限定はしていませんが、たとえば、次のような話題が、セッション・テーマの候補として考えられます：

- ・ソフトウェア開発におけるAI関連技術の応用
- ・現実のプロジェクト支援環境における問題点
- ・新しい開発／保守方法論の実践とその評価
- ・ユニークなアプリケーション開発の事例
- ・中堅技術者教育のケース・スタディ
- ・ドキュメンテーションの機械化
- ・ネットワークのセキュリティ
- ・プロジェクトにおける技術移転
- ・LANを用いた分散型環境の構築・利用
- ・先進的ソフトウェア・ツールの試作・実験
- ・ソフトウェア・プロダクトの流通と権利保護
- ・量的マネジメントー各種モデルの適用と評価

応募要領

応募論文の要旨を2千字-3千字程度にまとめたもの（ワープロまたはコンピュータ出力でA4判2-3ページ）を10部コピーして、

1986年11月30日までに

2人のプログラム委員長（高田佳彦または林香）のいずれか宛にお送りください。

別途編成するプログラム委員会で応募論文の審査を行い、結果を1987年1月末までに、応募者全員にお知らせします。

審査を通過した方々には、発表用の最終論文（予稿集にのせるためのカメラ・レディ原稿）を1987年3月末までに仕上げていただくことになります。

その他、パネル討論のテーマや招待講演、チュートリアルなどに関して、なにかアイデアがありましたら、実行委員長までご連絡ください。

シンポジウム・スタッフ

実行委員長
岡田正志

日本電気ソフトウェア
生産技術部
〒108 港区高輪 2-17-11
TEL 03-444-3211

ツール展示委員長
岸田孝一

ソフトウェア・リサーチ・アソシエイツ
専務取締役

プログラム委員長
高田佳彦

日本電子計算
科学技術事業部
〒103 中央区日本橋兜町 6-7
TEL 03-668-6171

林 香

ソフトウェア・リサーチ・アソシエイツ
環境開発本部
〒102 千代田区平河町 1-1-1
TEL 03-234-2611

中堅技術者教育について

中國 順三

富士通ビーエスシー

1. はじめに

本稿は、SIGENV（教育分科会）での今年の7月の月例会で発表したものに、補足・追加したものです。

最近、通産省・文部省のきもいりで新しい財団法人ができたりして、情報技術者に対する教育がクローズ・アップされてきています。とくに、中堅技術者といわれる人達への教育（再教育）は、企業の存続をも左右しかねない問題だと考えられているようです。そうした意味で、本稿が、今後それぞれの会社での中堅技術者教育を企画・実施される方々にとって、なんらかの参考になれば幸いです。

2. 中堅技術者の定義

ソフトウェア技術者の作業内容は、作業分野、プロジェクト規模、人員構成、および各社の事情によっても異なると思うが、一般的（かなり私見もあるが）に、以下の項目をクリアする能力を有することがエンジニアとして望ましいと思われる。

- 1) 最低でも2つ以上のプログラミング言語をマスターしている。
- 2) ハードウェア、基本ソフトウェア（たとえば、OS）等の知識をある程度知っており、指導のもとに開発環境（特にデバッグ環境）の設定ができる。
- 3) 指導のもとに、システム設計（外部設計）を行うことができる。
- 4) プログラム設計（内部設計）以降は、1人でもプログラムを作成することができる。
- 5) 単体／結合テストまでのデバッグ、およびテストが1人でもできる。
- 6) 最低でも、2～3人以上の部下を指導することができる。

3. 中堅技術者の立場

これも各社のやり方、プロジェクト規模等の状況にもよるが、最近の一般的な要員構成で考えると、中堅技術者の立場は最悪の場合以下のようなになる。

- 1) ソフトウェア事業のプログラマとは思えない（素人

に毛の生えた程度で、とてもプロフェッショナルとはいえない）初級技術者を抱え、

- 2) リッパとは口がさけてもいえない開発環境を使用して、
- 3) 無理難題なユーザ要求を受け入れたソフトウェアを、期限道理に作成しなければならず（時として、営業条件により受注してしまう）、
- 4) 品質においても、ある程度の保証（責任）を負わせられる。これはいわば、会社組織という中間管理職的立場である。上下の板ばさみでの苦労が多々有るが、しかし、プロジェクトが成功するには、重要な立場でもある。つまり、プロジェクト成功のキー・マンといえよう。

4. 中堅技術者の役割

「中堅技術者の定義」で述べた、定義に関する項目についてクリアするべく、自己努力をするのはあたりまえとして、それ以外に、以下の役割があると思う。各役割には、私なりのコメントをつけてみた。

- 1) 初級技術者をOJTをとうして、1人前のプログラマに育てる教育係である。
→自分の仕事を楽にするために早く自分の右腕を育てる事。
- 2) リッパとはいえない与えられた開発環境を、できる限り自分のチームの仕事に役立たせるための改善の努力をし、かつ成果をあげる。
→初級技術者でも使えるよう、バカチョン方式の採用。
- 3) 将来やらされるであろうプロジェクト・リーダーとしてのプロジェクト管理の勉強を実践で行い（これは心しておきたい）、最低でも自分の範囲内のチーム・ワークをまとめる。
→どんな難しい仕事でもチーム・ワークがよければ何とかなる。

5. 中堅技術者の育成方法

テクニカルな面については、数年来の話題となってい

る「技術移転：テクノロジー・トランスファー」について、著名な方々が色々な意見として発表されている（たとえば、W. Riddle の SEA 春のセミナーでの講演、元 IBM -現在は MCC のマネジャであるベラディーの論文）、各企業が様々なトライをしているので、それを参考にしていただくと、ここではメンタルな面について私見を述べてみる。

ある人いわく、「企業たるもの社員に対して、ロマンとマネーを与えなければならない」。この意味は、ロマン（夢ある仕事）だけでカスミを食べて生きるわけにもいかないし、かといって、「IT 商事」ではないが、マネー（給料）はそこそこ与えている（？）が、ロマンをどう与えるかが問題となる。いわゆる仕事の動機付けをどう教えるかである。

初級技術者時代は、与えられた仕事をもくもくとこなすことに努力するが、中堅になった時、自分のロマン、すなわち目標設定ができていくのが重要となる。目標設定ができていれば、現在やっている仕事の位置付け・意味付けを明確にすることができ、たとえ困難な仕事（自己能力不足も含め）であっても、前向きに対処することができ、最終的に、体得したもので、その道のエキスパートとなることができる。

そこで企業としてやるべき方法として、以下が考えられる。

- 1) 個人の目標設定に対するさまざまな情報が得られる環境を用意する。
→社内セミナーの開催/外部セミナーへの参加等
- 2) 目標に合った仕事を与える。
→個人の意見の吸い上げ
- 3) 目標にジャス・トフィットしなくても、与えられる仕事の位置付けを明確にしてあげる。
→管理者とのコミュニケーション

ようするに、どこの企業においても同じことであるが、これらのことは、特に情報処理企業にとって最も重要なことである。ソフトウェアとは人が作るものであり、いかにやる気を起こさせるか、起こしたやる気をいかに育てるかが、我々企業の使命であり、育った優秀な人材が財産となって人財となり、結果として企業の繁栄をもたらすと思われる。企業優先（単に仕事を与える）ではなく、企業と個人の「GIVE & TAKE」精神を忘れてはならない。

6. 中堅技術者育成を阻害するもの

通産省の「シグマ・システム提案動機」ではないが、ソフトウェアの需給関係からいって仕事は山ほどあり、次から次へと仕事が依頼されてくる（もっとも、儲かるかどうかは？であるが）。

しかし、ソフトウェア技術者（特に中堅以上）の絶対量が不足しており、仕事をこなす為に多忙を究めている。したがって、仕事の忙しさについて甘え、勉強がおろそかになり、新技術等に関する教育を妨げる要因となり、次の中堅技術者が育たないという悪循環となってしまう。

また、最近の若い人（これを言うともう中年だそうです）は、モータ型人間が多く、誰かがスイッチを入れて上げないと動いてくれない。与えられた仕事を、教えられた方法で仕事は行っていくものだと思っているため、従来の仕事のやり方に対して、問題意識を持たない人が増えており、本当の技術習得（自分で納得した技術）となっていない場合が見られる（こういう人が上司である場合は最悪となる）。したがって、自己啓発の面からみて教育が実につかず、中堅技術者への育成の妨げとなっている。

7. 最後に

以上、勝手な意見を述べたが、若手（数年後の期待される中堅技術者）をたくさん抱えた我々の企業において、各技術者に夢多き仕事を与え、ハッピーな仕事をしてもらおうと同時に、自他共に認める一流技術者に育成するべく、より一層の努力をしなければならないと思う。ソフトウェア技術者に栄光あれ！

環境から見たドキュメントの再利用

渡辺 雄一

電力計算センター

1. はじめに

”環境”という言葉は、最近では誰もが知っているけれども、それでいて誰も明確な説明の出来ない言葉の1つようです。それ故、一番SEAの特徴を生かせる分野でもありそうです。今までSE、プログラマ、アナリスト、プランナ等々の新しい職種を生み出してきたコンピュータ業界ですが、そのうち、エンバイアロンメンタなどという人が現れてくるかもしれない。こんなわけのわからないことを考えつつ、SIGENVの末席を汚している1人です。

第6回SIGENVの月例会での盛田さん（神戸コンピュータサービス）の発表と、それに続く自由討論が興味深かったので、自分自身の整理も兼ねてまとめてみました。会員の方々にとって、少しでも役にたてばと思います。

2. J-Starについて

まず盛田さんの発表内容の要点を列挙してみます。

- ・J-Starによるドキュメントは、従来の電話、ファックシミリ、紙、鉛筆、コピー等に代わるものとして、位置付けている。
- ・J-Starでの文書ファイル（ドキュメント）は、引出し（ドロワー）を根（ルート）として、ホルダを節および葉として、自由に木構造を構成出来る。
- ・ドロワーごとにアクセス権を設定出来る。
- ・課ごとにドロワーを与えている。
- ・現在、端末約15台をイーサネットを介して、全員（約80人）で使用している。TSSのエミュレータもあるので、J-Starの固有の処理と同様に、TSS端末としての使用がかなり多いのも事実である。事務の女性も使っているの、いつのまにか、ワープロはなくなってしまった（実は、J-Starを導入する切っ掛けとなった出来事として、事務の女性がファイル・レイアウトを作るのにワープロの出力を縮小コピーして、それを切ってピンセットを使って糊づけしているのを見た、という事があるそうです）。
- ・ワークステーションとは本来、頭に”パーソナル”の文字が付くはずである。そこで原点に立ち返って、も

っとパーソナルということを追求してみたい。

- ・その為には、パーソナルであるべきもの（個人のドロワー、ディクショナリ、ライブラリ、ツール等）は、各人が持てるような仕掛けを指向していきたい。一方ネットワーク、ファイル・サーバー等は、パブリックを追求してゆきたい。その為には、自由なコミュニケーションが確保され、パブリック・ドメインで流れるようなライブラリのような物を、どんどん増やしてゆかなければならない。
- ・（アクセス権を見直すときの）カテゴリ分類をどうするか？そしてどのような環境で利用してゆくかが、今後の課題のひとつである。

3. 討論（疑問点、問題点、主張など）

- Q：月並な質問ですが、J-Starを使ってどんな効果が得られましたか？
- A：とにかく皆楽しくやっています（笑）。なかなか定量化するのが難しくて、正確な数字は分かりませんが、大体20～30%位の効率が上がったのではないかと考えている。
- Q：アクセス権の見直しをしようとした原因はなにか？
- A：（ドキュメントの再利用について考えと言われてから）再利用について考え直してみたら、一番アクセス権の問題がそれを阻害していた。
- Q：再利用されていないのは何でわかったのか？
- A：みんながそう言っているし、また、みんなあまり他人のをを使う気もなさそうだった。再利用を全くしていなかったわけではないが、その範囲が1つの課とか、1つのプロジェクト内に限られているのが問題だ。しかしまた、プロジェクト毎に作業標準（ドキュメントの形式）が異なることも確かだ。
- Q：アクセス権を見直して、パブリックはとことんパブリックを追求したとき、それによる再利用率の上昇の見込みは？
- A：実は、パブリックの中にもものを入れようとすると、そこに鬼みtainな管理者がいて、ちゃんと申請をしないとそこに入らない。これではいけないというのが基本的に思っている。こうすると、「なぜパブリック

にいれないのか？」といって、めんどくさがっている。これはまずい。

Q: ドキュメントを「捨てる」ということについて、何かしているか？

A: システム変更をする際に、不要なものを消すこと位しかまだやっていない。まだ”いかに捨てるか？”といったところまでは至っていない。

Q: ログを取る機能はないのか？

A: 現在のバージョン5.0ではない。近々出るバージョン6.0では、もう少し使い良くなるし、最新の更新日付時刻は残るが、バージョン管理（世代管理）までは出来ない。

Q: 紙と鉛筆ではなく、機械でのドキュメントを実践していて、これは良いということはないか？

A: いろいろあるようだが、それがまだ表面までは出てこないでいる。皆溜めこんでいることは確かだが、それが表に出てこない。そろそろ蒸し干しをしないとあかんのですが。

Q: (盛田氏の言う) ディレクトリとは何か？

A: どこに何があるのか？ということを示した「辞書」のイメージ。これが現在ない。その為、「あのプロジェクトのドロッワーを捜せば、多分こんな物はあるだろう」といった具合で使っている。そのようなわけで、課とか部とかという属人的単位でドロッワーを構成したことがまずかった原因の一つと考えている。

紙の世界だと、バラバラめくるということが出来るが、機械だとそれが出来ない。また現実の世界だと（神様みたいな）事務の女の子がいて、「あの書類どこだっけ？」と言うと、「それは………」と言って捜してくれる。J-Starのような性格の処理には、そのような機能（それはひょっとしたら知識ベース）が必要だ。紙の世界を機械にただ乗せ換えただけで、まだ機械のメリットを引き出す「仕掛け」が十分でない。

Q: 使い勝手そのものは？

A: (J-Starのことを) 馬鹿だと言う人もいるが、私は重宝している。

Q: 以前のSIGENV例会で、J-Starはソフトがオープンコネクションでなくて使い辛いという意見があったが。

A: そこらへんの話になると、J-Starそのものの問題というよりも、XEROXのポリシの問題ではないか。唯このままでよいわけはなくて、いずれは—

ずづつでもオープンになって行くのではないか（またそうしてゆかなければならない）。

Q: J-Starに不満はないのか。

A: 不満は一杯ある。不満はあるけれども、ほかに良いものが見当たらない。

ドキュメントという言葉の持つ意味を、もうちょっと整理してみる必要があるのではないか？

4. 私の感想

ドキュメントの再利用環境ということで見直してみたら、実はファイルに対するアクセス権をいかに設定するかという、運用形態が問題であったようです。ならば、ドキュメントに限らず、再利用を推進するための環境作りは、新たな機能をただ求めることだけではなくて、もっと地道に運用形態の在り方を整理検討してゆくことが重要と思われます。

それとは別に、ドキュメントの在り方の論議は、極端な見方をするならば、仕様化技術への葛藤のあらわれとも受け止められます。いまだ確固たるプログラム作成技術があるわけでなくうに、近年のハードウェアの急激な進歩や、コスト・ダウンにより開発運用環境の変化が生じてきており、もろもろのドキュメントの見直しが、必要になってきていることは確かです。そのためには、現状の環境でどのようなドキュメントが必要かを、もっと議論していかなければならないと思います。

ドキュメントは、設計方法論・手法に依存することが多いのですが、多少の無理があっても、いくつかのカテゴリに分類整理してから、もう一度その環境も含めた議論をする必要が感じられました。

5. おわりに

SEAの春のセミナー・ウィークでも、ドキュメントの機械化ということで、J-Starをケース・スタディにしたセッションがあり（講師はもちろん盛田さん）、そのレポートが、SEAMAILの4月号に掲載されているので、導入の経緯などは、そちらを読んでいただきたい。

最後に、盛田さんのいわれた環境に対する言葉をひとつ引用して、本稿の結びとします。

「いままでは、端末がホスト・マシンにぶら下がっていたけれども、これからはワークステーションにホストがぶら下がるようなイメージで、自由に（端末から）ホストを選択する環境を指向しなければならないのではないだろうか」

新しいパーソナル開発環境の構築を目指して

熊谷 章

パナファコム

1. はじめに

この報告は、6月のYDOC例会での熊谷さんの発表をもとにして、横浜支部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。

ソフトウェア開発環境、それもオフィス・ワークの遂行にも違和感なく適応するようなものが望まれるが、現在の開発環境下で、そのような全く新しいモノを、試行的にでも構築しようとすると多くの問題がある。

例えば、知的なマン・マシン・インタフェースであるとか、日常の定性的な情報をデータ・ベース化して、フレキシブルに使うためのDB管理システムといったようなものを、ある程度評価できるようなプロトタイプとして作成したいと思っても、ヤレ、百人月だの千人月だの、途方もない時間と手間と費用がかかり、製品を作るならいざ知らず、実現可能性の評価や、そのコンセプトの有効性を評価したいといったシステムの有効性を確かめる上での、いわば前哨戦の段階で、既にオテアゲといった状況になることが少なくない。

実行スピードには眼をつむるとしても、もう少し手軽に、素早く、そして自由度が高く、ソフトウェアの開発が出来ないものだろうか。

この解決のために、我々パナファコムでは、パーソナル・オフィス・エンバイロメント(POE)と呼ぶ、強力な開発環境として「LIPS」を開発し、統合化された環境の構築を目指しての試行錯誤を行っている。

以下では「LIPS」について、その概略を述べる。

2. LIPSの由来と開発のアプローチ

(1) 最初の契機

従来とは変わったシステムとして、POEの開発を目指した。POE開発のためには、UNIXとCでは、あまりに不十分('84ころ)であった。

→新しいプログラミング環境が必要

(2) LIPS

AIのエキスパート・ツールや第5世代プロジェクトからヒントを得て、マルチプル・プログラミング環境を考えた。

$LIPs := (Lisp + Ingres + Prolog) * n$

この時の's'は複数形を意味したものであったので、小文字になっていた

(3) LIPSシステム開発における Smalltalk の威力にシビれる('84末)。

LIPs → LIPS (Sは Smalltalk の S)

3. LIPSのアーキテクチャ

自由度を高く、かつ、拡張が容易なようにコンポーネント形式とし、余りガチガチに結合させてはいない。

Lisp, Ingres, そして Prolog の結合に関して、有機的な結合の度合は、かなり密接したものになっているが、Smalltalk については、ユーザ・インタフェースを担当していること等から、他の3つよりさら、ルーズな結合状態にしてあり、フレキシビリティが高いものになっている。

4. LIPSの狙い

1) ハードウェアに非依存なマルチプログラミング環境の作成

→ベースとして UNIX を採用する

2) 新しいメタファ(パラダイム?)の導入による新機能の実現

◆述語論理ベースの推論、

宣言的プログラミング→ Prolog

◆ダイナミックなデータ表現形式としてのリスト構造

AI共通言語→ Lisp

◆オブジェクト指向→ Smalltalk

これらのメタファにより以下の点を改革する。

◆知的な対応が出来ないマシン

→誰でも使えるマシン(知的インターフェース)

◆単なるデータの集合(通常のDB)

→構造化された、意味のあるデータ(知識)

◆開発の不可能さ(whatの追求が主眼なのに、howに手間取る

→開発可能に(実行スピードを除けば、whatの試作が容易)

5. LIPSの特徴

(1) マルチ・プログラミング環境

親言語を Prolog とし、Prolog から Ingres, Lisp, Smalltalk の諸機能を使用できるようにする。

Lisp との結合: 2つのリストを結合するプログラム例

《in Prolog》

```
append ([],X,X).
```

```
append ([A:X],Y,[A:Z]) :- append (X,Y,Z).
```

《in Lisp》

```
(append lambda ( X Y )
```

```
(if (null X) Y
```

```
(cons (car X) (append (cdr X) Y ))))
```

《Prolog から Lisp の呼び出し》

```
append (X,Y,Z) := (lambda (X Y) (append X
Y)).
```

↓ Prolog に変換

```
append (X,Y,Z) :- lisp( append (X,Y,Z)),!.
```

必ずバック・トレース禁止の!が付加される

(2) Lisp との結合

Lisp で用いられるプロパティ・リストを Prolog からアクセスする機能

《アクセス》

```
prop (Name, Attribute, Value).
```

《変数への読み込み》

```
prop (pen, color, V).
```

pen の属性 color の値を V に読み込む

《変数への書き込み》

```
prop (pen, color, BLACK).
```

color の値を BLACK にする

(3) Ingres との結合

Ingres の機能のうち、C 言語から呼び出し可能な、EQUEL プリアプロセッサ機能を使って実現している。

《Ingres 起動、DB のリレーション名獲得》

```
ingres (Database, relation)
```

《リレーションを構成する属性》

```
relation (Relation, Attribute)
```

《リレーションのタプルを事実として読み込む》

```
retrieve (Relation)
```

《リレーションの創成》

```
create (Relation, Attribute)
```

《リレーションを壊す》

```
destroy (Relation)
```

《Ingres の終了》

quit_ingres (Database)

(4) プログラム開発環境

《ユーザ・インタフェース》

◆LIPS のプロンプト「*」に対して、「事実」と「ルール」の入力が行える。修正はその場で直ちにエディタを呼び出し、修正できる。

・バッファの表示

・バッファにファイルを入力

・バッファをファイルに出力

・実行、再実行コマンド (Prolog の

consult, reconsult 機能)

・Shell の呼び出し

◆デバッグ環境

・範囲トレース

・指定述語トレース

・ブレークパッケージ (1 行実行, why, how 型の説明)

why 型: より幹に近いゴールトリーの表示

how 型: 葉に近い、既に達成されたゴールトリーの表示

◆syntax と記述能力の向上

1 例として、あるインタプリタを書いたとき、

Prolog = 数ステップ

Lisp = 数十ステップ

C/ Pascal = 数千ステップ

6. LIPS のインプリメンテーション

Smalltalk-80 の inter-process communication 機能を介して、ブラウザの 1 プロセスとして動く。

7. 今後の課題

今後の課題としては、以下の 3 点が挙げられる。

(1) Prolog → Lisp へのトランスレータ

高速化、実用化のためには必要

(2) Smalltalk → C へのトランスレータ

C 言語開発への Smalltalk 環境の流用

(3) スタンダード・スペックへのアプローチ

意外に大切である。

現行 Franz → Common Lisp

DEC-10 Prolog → Quintus Prolog

先日のSEAセミナー21でも、熊谷さんを講師とするLIPSのセッションがあった。そこでどんな討論が行われたのか、興味深い。

LISP 事始め

-実践的AI言語講習会に参加して-

赤坂仁志・梅林信之

協同システム開発

1. はじめに

現在私たちは、JSDのFASETプロジェクトのなかで、AI向き言語といわれているLISPを開発主要言語とした仕事をしようとしている。それは、状態遷移（言語・図形表現）を用いて、設計の自動化を行おうというものである。計画は5ヶ年であり、今年は2年目で、一部プロトタイプの開発にはいる。

梅林的これまでの環境は、VAX11/780でUNIXをOSとするハードウェアで、C言語を中心にいくつかのプログラムを開発してきただけで、実務経験はほとんどない。そして、赤坂のほうは、実務経験はCOBOLを中心に大型計算機上で2年あり、協同システムにきて、UNIX環境での経験が半年である。つまり、2人ともLISPでの実務経験は、今年の5月になってからである。

2. LISP環境

われわれが普段使っているUNIX (4.2BSD) では、Franz Lispがほとんど自由につかえる。自由という意味は、LISPを使用すると資源（メモリ、ディスク）を大量に使ってしまい、UNIXはTSS環境であるために、「TSS、みんなで使うと遅くなる」をさらにひどい状態にしてしまうのだが、われわれのところでは、端末数こそ多いが、現在のところ開発にあまり使われていないために、重いジョブをいくら動かしても他人に迷惑をかけないからである。さらに、2台のWS (Symbolics-3670, Tektronix-404) では、いわゆるAI環境でLISPが使用できる。加えて、UNIXの日本語端末として使用しているPC-9800には、パソコン用のLISPがいくつか導入されている。

こうした環境が恵まれているとは思っていたが、今回の講習会に参加した方々のLISP環境を、講習会でのアンケートの結果として、事務局から提供されて、自分たちがあまりにも恵まれ過ぎていることがわかって、かえってプレッシャを感じた。

3. 言語を覚えるには

言語を覚えるには、その言語でプログラムを作るのが一番とはよくいわれている。事実、われわれも、これまで知らなかったC言語は、人から教わるより、本を中心に目の前の環境でマスターできた。しかし、LISPは本を読んでも（読んだのは、講習会で副読本として使用されたのと、その他2冊である）、Cほどにはプログラムがつかれない。

これもよくいわれていることだが、新しい言語であるLISP, Smalltalk, Prolog等（これら言語が、新しいという語弊があるが、現場での実績という意味でとしておきたい）を習得し、使いこなすのは、これまでの既存の言語（COBOL, Fortran等）に害されたひとたちより、最初から新しい言語を使っている人や頭のやわらかい若いエンジニアのほうがはやい、ということである。その理由のひとつに、言語のspecを覚えるだけでなく、そこで培ってきたプログラミング・テクニックも覚える必要があり、そのテクニックはソースを読むか、口こみでしか伝わらなかった、というLISPの歴史も関係している。

われわれ2人は、24才と25才ということでは十分わかいし（エンジニアとしてです）、既存言語に毒されるほどの経験もないのに、ちょっと複雑な問題にあたると、自力での解決が時間がかかってしまっていた。これからプロトタイプの開発に入るのに、これでは困ると思っていたときに、「実践的」というこれまでにないキャッチフレーズにあたり、自費で参加した。

4. なにが実践的だったか

第1回目の講義は、比較的やさしかった。これまでに知っていることのおさらいが多かったからである。それでも、本に書かれている内容よりは、はるかに判りやすかった。それは、4回の講義全体にいえることであったが、とうりいっぺんの表面的な説明でなく、処理系内部の説明がほとんどの場でされていたからである。

たとえば、データ構造の説明は、内部でどうもたれているか。シンボルは4組（名前、関数、プロパティ、ヴァリュー）によって構成されている、等々である。LI

SPの命である関数の説明も同様であった。これは、どの本にも説明されていないことである。

2回目の講義の最初に、前回の講義の復習としての練習問題がだされた。ここで、いままで自分では判ったと思っていたことが、そうではないと思い知らされた。この練習問題の解答も、処理系内部の説明がなされ、単に問題ができた、できないでなく、自分が納得できたように思えた。

5. そして宿題

そして、3回目の講義の冒頭で、次回までの宿題が出された。提出すれば、講師が添削をして送ってくれるという。ここまで面倒をみられると、受講する側としても熱心にならざるをえない。といっても、出来のほうがいまいちで申し訳ないが、2人の素直な感想としては、これから真剣にLISPに取り組むための方向を示していただきたきとする。この先、LISPを自分のものにするかは、これからの取組かただとは思いますが、できれば、実際のマシンを講義に使用した、いわば、応用篇のアドバンスト・コースを催っていただけたらと思う。

6. テキストについて

さらに応用篇を希望したのは、実は3、4回目の講義が突如として(!?)むずかしくなったからである。前2回に較べて、内容が増えたせいであるが、この感じは、われわれだけでなかったようである(毎回前講義でのアンケート結果を提供してくれた)。ひとつの言語をそれなりにでもマスターするには、4回*2時間では無理ということだろうか(講師も、もっと話たいことがあり、2時間*14回が理想とか)。

講義のテキストが、講義の前に毎回配られたが、できれば事前に手元にあれば、予習ということができたように思えた。しかし、その内容の豊富さからいえば、このテキストを作成した講師の苦労がわかる。SEAでこのテキストに手をいれて(説明文と実例を増やす)、出版したらどうだろう。これまでにない、実践的な教科書ができと思うのだが。

7. 最後に

とにかく、「実践的に」有意義な講習会であった。細かい文句(たとえば、会場、人数、時間)をいえばきりがないが、コスト・パフォーマンスを考えなくても、自費で参加したかいがあった。

それと、講師がLISPをマスターしたのは、仕事の合間をみての独力と聞いて、これはやらねば、という気

になっただけでも収穫であった。■

実践的 A I 言語講習会の報告

1. 目的

7月8日を第1回として、週1回の連続4回のLISPの講習会を開催しました。

この講習会の目的は、最近になって、ソフトウェア開発にAI技術の適用が期待されてきており、こうした期待を実現するためのハード/ソフトの環境が、ようやく手近になってきたものの、AI言語(LISP等)といわれる言語を習得するのは、独習では、なかなかむずかしいようである。そこで、これらAI言語を、実際のソフトウェア開発の現場で毎日使っているエンジニアを講師にして、実践的に勉強しようというものでした。

テーマ自体のタイムリー性か、価格の安さか、定員15名のところを、3倍以上の申込があり、急挙会場を広い所に変更しましたが、それでも20名以上の方をおことわりする盛況でした。

2. 講師

講師には、個人的興味でUNIX上のFranz LISPの日本語化をやってしまったという杉田研治さんをお願いしました。

3. アンケートから

4回の講義の度に、受講者のかたからアンケートの協力をいただきました。その中から、興味深い解答を載せます。

(1)これまでのLISPの経験について

LISPを使って仕事をしている(いた)—— 3人

LISPは独学で覚えた

プログラムを使った————— 4人

本のみ————— 8人

今回が初めて————— 12人

(2)現在の環境でLISPが使用できますか

できる————— 16人

できない————— 10人

4. 今後の予定

今後の予定としては、今回と同様な形式で、Prolog, Smalltalkを検討しています。

また、今回の参加者の希望が多かった、LISPのアドバンスト・コースの検討と、今回参加出来なかった方のために、同じコースの再開(11月頃)も検討しています。■

ネットワークの苦楽

栗原 正利

ソフトウェア・リサーチ・アソシエイツ

1. はじめに

日本のアカデミック・ネットワークもようやく起動にのり、大学・研究所のUNIXユーザを中心とした電子郵便とニュースのシステムJUNETが運用されているが、ここでは、私自身が80年にUNIXに出会ってから、今日のネットワーク環境に至るまでの経緯を振り返り、その変遷と運用について述べてみる。

2. UNIX間の通信

(1) 単純接続

80年夏、自分の会社に最初のVAXが導入された時点では、何しろマシンが1台しかないで、当時話題をさらったイーサネットのようなLANは、我々にとってはまだ無縁のものであり、かたや広域ネットワークといっても、日本では法律の手厚い保護により自由な接続はままならない状況であった。

82年に入り2台目のVAXとマイクロ・プロセッサを使ったUNIXマシンが何台か導入され、これらのマシン間でのファイル転送機能が必要になった。もともとUNIXには、端末回線を利用したリモート・ログインのためのコマンドcuと、ファイル転送およびコマンドのリモート実行のためのシステムUUCPが用意されていた。当然我々もこれらのツールを使って社内のネットワークの第1歩を踏み出した。

まず手始めにcuによるリモート・ログインを利用し、ファイルの転送はこのコマンドの持っているファイル転送機能によって実現した。cuコマンドは接続操作も容易であり、なかなか便利なコマンドであるが、このファイル転送はテキスト・ファイルに限られ、また信頼性ならびに端末回線の占有使用の問題があるため、VAXのような複数人で利用するシステムでの本格的利用には適していない。

(2) ファイル転送

そこでファイル転送のためにUUCPのセットアップを行い、cuとUUCPを目的に応じて使い分けようにした。

ところが当時使っていたUUCPは多くの問題を含んでおり、はっきり言って優しく扱ってあげないとすぐに

死んでしまう、わがままなシステムであった。例えば1Mバイトのファイルを転送すると、受信側のUUCPは転送が終了した時点で転送完了のACKを送信側に返す前に必ずコアダンプする。その結果このファイルの転送は完了していないことになり、UUCPが再起動されると、再度この1Mバイトのファイルを転送しなおすという、お粗末な無限ループに陥ってしまい、他の要求はいつまでたっても処理されないことになる。

このため毎朝出社と同時に、UUCPが正常に動作しているか否かをチェックする日々がしばらく続いた。

(3) 電子郵便

UNIXにはmailという電子郵便用のコマンドが用意されている。このコマンドは1つのマシン内におけるローカルな電子郵便機能にとどまらず、UUCPのリモート実行の機能を利用してマシン間の電子郵便に使用することもできる。

この時点で使用できた宛先の指定方式は、いわゆるUUCPスタイルの方式であり、利用者自身がどのようなマシンを経由して電子郵便を配達するかを、指定しなければならない。この方式はシステムとしてはシンプルで良いが、たくさんのマシンのリンク関係を全て人間が覚えているのは非現実的であり、何十台何百台とマシンが接続されたネットワーク環境には適した方法ではない。

3. 公衆回線

(1) 電話回線からのアクセス

当社のVAXには、初めから自動受信のACUモデムが接続されており、直結回線が足りなくて、社内から電話回線を使って、同じビルにあるVAXを使ったりしていた。80年のことであるから当然300ボーの低速モデムであり、かつそれが数十万円もしたのである。

UNIXでは、電話の着信信号を検知してモデムを受信可能な状態にする方式ではなく、最初から受信可能な状態にしておき、ACUが電話を受けモデムがキャリア信号を検知した後、UNIXに対して回線が使用可能になったことを伝える方式を取っている。この方式自身に問題はないのであるが、当時のACUモデムはタイムアウトの概念がなく、誰かが間違って電話をしてビーとい

う音で間違いに気付く電話を切る、あるいは電話からアクセスしようとして電話したが途中で気が変わって電話を切ってしまった場合でも、受信側のモデムは無限にキャリアを待ち続けるため、それ以降誰も使っていないのにお話中となってしまう。

(2) 遠隔地通信

UUCPは、電話回線を使った広域ネットワーク用に作成されたソフトウェアであるため、遠隔地であってもお互いの了解と自動発信のできるACUモデムさえ手に入れば、容易に電子郵便およびファイル転送を実現することができる。

80年に購入した300ボーの自動受信ACUモデム以下の値段で、2400ボーの自動発信可能なACUモデムが手に入るようになった。この種の新しいACUモデムでは前述のような問題は発生せず、またこれらのACUモデムを使った公衆回線による遠隔地通信を活用し始めたころにはUNIX自身も改良され、UUCPの問題もほとんど解決されていて、毎朝の日課は自然消滅していった。

4. LAN

我社へのイーサネットの導入はつい2年前のことである。現在では、端末回線をTCP/IPプロトコルで利用できるソフトウェアの導入により、東京本社/別館/大阪支社の5台のVAXを、どこからでも会話型でアクセスできるようになった。

しかし便利な反面、イーサネットに接続されているマシンの数が増えると、なかなかその運用は面倒なものになる。そもそもイーサネットは、マルチ・アクセスのシステムであるため、イーサネットに接続されている任意の2つのマシンはそれぞれ直接通信することができる。つまり、イーサネットに新たに1台のマシンを接続するには、その1台の設定を行うだけでなく、そのイーサネットに接続されている全てのマシンの設定を変える必要があることを意味する。

単に新しいマシンの存在を設定するだけであれば、たいした作業量ではないが、ネットワーク全体にわたって利用者の管理を行う場合、あるいはネットワーク・ファイル・システムなどによって資源を共有するような場合には、その設定はかなり面倒なことになる。今後の方向として、ワークステーションのような個人向けのシステムがLANにつながることを考えると、今までの手操作ではとても対応できそうにないように思える。

5. 広域ネットワーク

JUNETは、大学のUNIXユーザを中心とした電子郵便とニュースのシステムである。JUNETではUUCP方式の宛名とは異なり、ユーザ名とサブドメイン名を指定する方式をとっている。

この方式は、転送経路をシステムが自動的に選択するため、UUCP方式に比べ参加するサイトが増え、転送経路が複雑になった場合に強みを発揮する。しかしそれは逆に、新しいサイトをJUNETに加えるためには、そのサイトが接続されるサイトではその情報をシステムに教えなければならないことを意味する。この種の設定は、ネットワークに限らず新しいものを接続する場合には、どんな場合でも必要であるが、前述のように、イーサネットのようなLANに、既に多くのマシンが接続されている場合だと、なかなか面倒なことになる。

JUNETは、参加各サイトの協力・合意によって運用されているネットワークである。このためJUNETに加入する組織ではpostmasterと呼ばれるネットワーク管理者を置き、宛先不明の郵便の処理やシステム設定変更作業などをど行うことになっている。

JUNETの電子郵便の機能は、非常に有効なものである。最近では漢字の使用できるUNIXも多くなってきているため、郵便と同じように利用することができる。

一方ニュースは、USENETと連動したシステムであるため、あるトピックに関してネットワーク上で議論することができる。また公開ソフトウェアの配布、種々の情報流通も行われるため、UNIXユーザとしては欠かすことのできないシステムである。恐ろしいことに、1日に1Mバイトもの情報が来る。

6. 情報ネットワーク

JUNETのような組織を越えたネットワークは、とにかく内にこもりがちな我々を、外部と結んでくれる1つの重要な生命線である。しかしそこには、1つのルール、つまり「GIVE&TAKE」の精神があり、これを心がけないとネットワークを有効に活用することはできない。このことはSEAにもまったく同様にあてはまり、SEAは我々会員の「聞くだけの参加」ではなく「ものを言う積極的な参加」によってお互いの利益を生む1つの情報ネットワークだと考えられる。

SEA会員の情報ネットワークへの積極的参加と、1日でも早く、SEANETが全国に張られることを願っている。

高度情報化社会と消費者問題

佐原 洋

東海大学

1. 消費者への影響

高度情報化社会の到来が消費者にどのような影響を与えるか、ということが現在問題になっている。とくにいわゆる「消費者問題」の分野での影響が、筆者の参加している東京都消費者対策審議会で検討されているが、この問題に関して、以下簡単に私見を述べたい。

まず第一に、当面その対策が重要視されているのは、信用供与を受けようとしたときに、本人が誤ってブラックリストに載せられていたため、信用供与をうけられなかった、という消費者被害の問題である。また各種の個人情報専門機関にあつめられていることは、プライバシーの侵害ではないか、ということが論じられている。しかし、実はブラックリストの問題は、必ずしも情報化の中心的推進者であるコンピュータ・システムの問題ではなく、データのインプットミスの問題であり、かならずしも高度情報化社会における本質的な問題とはいえない。むしろ後者の問題のほうが、その被害は顕在化していないが、将来大きな問題となる可能性があろう。すでに結婚相手を探すためにもコンピュータのお世話になる若者が多くなっている時代である。

2. 消費者問題

高度情報化社会における消費者問題は、社会の文化的、社会体制的な変化と、それが消費者の自主性をそこなわせる方向に影響を与えることから生ずる消費者問題である。その端緒は、昨年発生した豊田商事など、さまざまな詐欺的商法にあらわれていると考えることもできる。

詐欺的商法の歴史は、洋の東西を問わずかなり古いが、ここ数年爆発的な増加を示すに至ったのは、一方では所得の増大、とくに高齢者の貯蓄額の増大ということもあり、他方では家族関係その他の人間関係の希薄化と、これに代わって、怪しげな財テク情報の横行、そしてこれに乗じて企業と消費者の情報格差を巧みに利用した、悪しき意味での情報化時代の反映ではなからうか。

しかし、こうした情報格差を悪用した詐欺的商法による消費者被害ということは、被害者は少なくないにしても、基本的にはもっと大きな問題がある。それは人間や社会、そして文化までもが、コンピュータ使用に便利

ように枠をはめられる、ということである。

3. マークシートの恐怖

たとえば、国、公立の大学入試の共通一次試験は、マークシート・カードによっておこなわれている。マークシートを使うということは、実は受験生の頭脳を、そしてややおおげさに表現すれば、精神までマークシート式に替えてしまう、ということではあるまいか。

筆者自身の経験によると、大学生のなかで、マークシートの成績のよい者は文章が書けないし、文章の書ける学生はマークシートが得意ではない、という傾向が年とともに進行している。これはまさに恐怖に近い現実である。文章が書けないということは、自分で意見を持てない、自分の思想を持てない、自分の青春を持てない、ということである。こうした傾向が続くならば、学生だけでなく、すべての国民、すべての消費者がマークシートのしかものごとを理解できなくなるかも知れない。

4. コンピュータへの過信

コンピュータ過信にともなう製品やサービスの墮落は、直接にはコンピュータミスではないが、コンピュータにより設計、製造、運営、保守、点検されているので危険はないはず、という過信があったのではないか。

三洋電機ファンヒーターによる酸欠事故、日立の冷蔵庫、洗濯機の発火事故なども基本的には、コンピュータで設計、製造、点検しているものなら安全という油断があったのだらう。特に問題になりそうなことは、以前はメーカーがかなりの長時間、社内での使用実験をして商品の欠陥発見に努力していたものが、最近はコンピュータ・シミュレーションによって、これに代えているのではないかと、推測されていることである。

コスト面では合理化がすすむが、消費者被害という点で、このましくない結果となったのではないか。また各種の情報機器が、やがて家庭でも利用されるようになることは当然であろうが、消費者にとっての消費コストの増加、また機器の操作のむずかしさ、という点でも大きな問題がある。ひとくちでいえば、高度情報化社会とは消費者にとって、「便利ではあるが、暮らしにくくなる」ということではなからうか。

アメリカ・中国雑感

村井 進

日本コンピューター・システム

1. はじめに

最近、仕事でアメリカと中国を訪問する機会を得た。仕事とはいえ、見知らぬ国々を旅するのはそれだけで楽しいものである。ましてや、ほとんど両極端に位置するこの2カ国を、短期間のうちに見ることができたのは幸せであった。その時のとりとめのない印象を記してみる。

2. ロサンゼルス

ロサンゼルスは空はスモッグに覆われていた。片側5車線で網の目のように張られた高速道路を、渋滞さすほど走っている車の排気ガスのせいである。飛行機で近づくにつれ、地上の景色が霞んでゆくのがよくわかる。赤茶けた砂漠の上を飛び越えてくると、いかにも人の活動する地に着いたという印象があった。

ロサンゼルスで、人が歩いているのは中心街だけである。少しはずれると車以外は見あたらない。もちろん、夕暮れ時の裏通りなど、歩いている人はまずいない。また、街角にはコーラの自動販売機など全く置いてない。お金の入った機械など、街角において置けばたちまち破壊されてしまうのであろう。

町中を走っている車は、バラエティーに富んでいる。超高級車からほとんど屑鉄まで走っている。ひどいのになると、屋根もボンネットも無くなっていて、エンジンの回転がそのまま見えていた。しかしどうあれ、1人1台以上、車を持っていることに変わりはない。したがって、バスなどの公共交通機関はあまり発達していない。旅行者といえども、車がないと非常に不便である。ただし、ダウンタウンのまわりだけは、Dushというマイクロバスが走っている。安くて便数も多く便利だったが、大柄なアメリカ人が小さなバスに乗って走り回る姿が、とても愉快であった。

朝の出動時刻になると、ロス空を何台ものヘリコプターが飛び交って行く。私は、てっきり交通情報を流しているのだと思った。でも違っていた。タクシーの運転手によると、「ビッグ・ビジネスマンは、ヘリコプターでビルの屋根の上に通勤するんだ」とのことである。

アメリカとは、機械化によつて効率を追求する社会である。人手は出来るかぎり省かれている。たとえば、ホ

テルを出るときは、コンピューター制御のロックを掛けることになる。

2. 北京

北京空港は、どこか中国的な香りのする近代的なビルである。しかし、雑踏は少ない。楊子江、黄河の流れる深い緑の大地に、そのまま降り立ったような印象がある。空は思ったほど澄み渡ってはいない。たぶん黄砂のせいであろう。

北京のタクシーは非常に安い。30分走っても、日本円に直すと2000円位にしかない（しかしそれは、運転手の給料の約半月分に当たっている）。数人で中華料理を食べに出るときなど、いきおいタクシーを利用することになる。行きに20分、食事中1時間半待ってもらって帰ってくる。中華料理が無茶苦茶おいしいのも重なって、まるで殿様になったような気分が味わえる。どういうわけかタクシーは、最新型の日本車が中心であった。

北京の通勤手段は、かの有名な自転車である。通勤時になると、全く同じ型の自転車が、まさに大群を成して走っている。そしてその走り方とくると、交通規則などあってなきがごとくである。さらに、それをかきわけけるように2両連結の大型バスが走っている。よくもまあ交通事故が起きないものだと感心してしまった。後で聞いた話では、最近交通事故は増えているという。しかし、事故の恐ろしさを、皆実感していないようである。

夜の中国の街角には、人が溢れている。そこらじゅうの路地に椅子を持ち出して、井戸端会議に花を咲かせている。なんだか、すっととけ込んでゆけそうな安心感がある。表通りには、これは間違いなく昭和20年代から30年代にかけての銀座である。街灯のしたに柳が揺れ、その下をアベック達が行き交っている。

中国は、ある意味では、非常に非効率的な社会である。ブルドーザー一台で半日で終わる仕事を、何人もの人が集まって一日がかりで行っている。我々が泊まった宿舎には、各階毎に係の人がいた。用事は全てその人に頼むし、また出かけるときは、鍵を預けて行くのである。

幹事会報告

昭和61年度第2回：7月24日（木）

昭和61年度第3回：8月27日（水）

会 場：いずれも機械振興会館

議 題：

1. 会員更新

ボランティア会員の未更新は、現在約30パーセントある。更新意志のないかの判定は、もう1度手紙を送るが、それでも更新されない場合は、会員抹消とする。また、今後の更新手続きは、3回の督促状の発送を基本とする。

2. 会員増加活動

8月中旬時点で、個人正会員数が500名をこえたが、より有意義な活動を行うために、「1人が1人以上」をスローガンにした会員増加運動を行う。そのために、会員にはレターを発送して運動を行ってもらおう。

また、事業所単位で加入の賛助会員に関しては、幹事がそれぞれの運動を展開することとし、そのための資料を作成し、各幹事に発送する。

3. ソフトウェア・ショー

ソフトウェア流通促進センターのご好意によって、ソフトウェア・ショーの会場に、当協会のブースが設置される。そこで配布する資料として、当協会の案内を盛り込んだ資料を作成する。

また、ショー期間中に、ソフトウェア相談員を当協会の中から依頼があり、了承された。メンバーの選定は、次回の幹事会で決定する。

4. 秋のセミナー・ウィーク

今年度1番大きなイベントとして企画されている、秋のセミナー・ウィークへの中間申込状況が報告され、予想を越えた申込（延べ550人）があった。詳細の報告は、次回の誌上でセミナー委員長から行われる。

5. 情報化月間記念フォーラム

SEAとして情報化月間に参加し、特別フォーラムを開催することがセミナー委員会から提案され、承認された。案内パンフレットは、すでに全会員のお手元に届いているはず。

（記：野辺）

A I 分科会 (SIGAI)

1. 第4回月例会

A I 分科会 (SIGAI) の第4回月例会を、9月11日（木）に、機械振興会館の会議室で開催しました。

第4回は、東洋情報システムの山本稔さんから、パソコン上のエキスパート・システム構築ツール「(Super) Brains」の、その特徴、長所、欠点を話いただきました。

2. 次回以後の月例会予定

第5回月例会

日時：昭和61年10月9日（木）19:00-21:00

場所：機械振興会館 B3階 9会議室

テーマ：「オブジェクト指向入門」

Smalltalkを中心に

坂下秀 (ASTEC)

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：未定

A I 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)の野辺良一までどうぞ。

管 理 分 科 会 (SIGMAN)

第5回の月例会を、9月19日（金）19:00～21:00に、SEAMAIL編集部と共同で、東高円寺のSEA事務所でおこなった。

管理分科会への連絡、問い合わせは、沖ソフトウェア(03-454-2111-ex-3134) 芝原雄二までどうぞ。

教 育 分 科 会 (SIGEDU)

8月は夏休暇でおやすみした教育分科会の会合を、9月30日（火）に機械振興会館で下記の要項で開催します。

日時：9月30日（火）19：00～21：00

場所：機械振興会館6F-S1

テーマ：未定

報告者：杉田義明（SRA）

なお、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」というテーマで半日のフォーラムを青年会議所（東京）で開催します。

「新人教育」、「中堅技術者教育」というテーマでの、第3、4回の討論の報告は、SEAMAIL誌上に行われていますが、技術者教育に関心のあるかたは、杉田（SRA-03-234-2611）、または中園（BSC-03-501-4159）迄。

再利用分科会(SIGREUSE)

8月27日（水）に第6回討論会を、山本邦雄（横川北辰電機）さんから、自ら作成し、実際に運用している部品化システムについて、設計方法論、技術面および運用面からの解説を行っていただいた。

今後の予定

今後の討論会、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、当日に会場においてください。なお、SIGREUSEへのお問い合わせは、世話人までご連絡下さい。

第7回討論会（第4水）

日時：9月24日 19：00～21：00

場所：機械振興会館 B3-9

発表者：村井 進（日本コンピューター・システム）

テーマ：コロラド大学認知科学研究所の「オブジェクト・オリエンティド方式による画面生成支援システム」の紹介

第8回研究会（第2水）

日時：10月8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8回討論会（第4水）

日時：10月22日 19：00～21：00

場所：機械振興会館 B3-9

テーマ：未定

第9回研究会（第2水）

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）

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

連絡先：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3回の会合を、9月2日（火）にソフトウェア流通促進センター会議室で行った。

テーマは、「プログラムの同一性の実験」ということで、「迷路通り抜けプログラム」を Fortran で会員各自が作り（あるは誰かに作ってもらい）、同じ仕様書からどの程度違う（同じ）プログラムができるのか、ということの実験結果を自由討議した。

この実験の目的は、同じアルゴリズムを解くプログラムにおける同一性は判別できるか、1部を手直したプログラムの判断基準はあるか、を究明しようという遠大なものです。

討議に先立ち、佐野先生から、最近米国で、「プログラムのアルゴリズムに著作権がある」という判例がだされたが、プログラムがどんなに違っていても、アルゴリズムに著作権があるのは納得がいかないので、この実験で確認したい、という問題提起があり、討議を開始しました。

会員5人のプログラムと、佐野先生が某大学のプログラミング未経験の学生4名のもの、全部で9本のプログラムが集まり、まずは学生のものからの分析を行いました。

た。

討議は、分散と集中を繰り返しながら活発に行われましたが、時間切れで結論までに至りませんでした。

そんなわけで、次回第4回は、

日時：10月6日（月）18：30～

場所：ソフトウェア流通促進センター会議室

テーマ：「再びプログラム同一性の考察」

の要領で行います。

この討論の結果ができれば、SEAMAIL誌上でまとめたいと思っています。

SIGSLP への問い合わせ等は、SRAの能登(03-234-2611)にどうぞ。

環境分科会(SIGENV)

1. 第8回月例会

第8回月例会として、8月20日（水）19：00～21：00に、機械振興会館で、「ソフトウェア開発環境としてのメインフレームのTSS」をテーマにして、メインフレームのかたに紹介していただき、その後で討論を行いました。そのまとめは、次号のSEAMAILで報告します。

環境分科会の月例会は、毎月第3水曜日に、機械振興会館で行っていますので、ソフトウェア開発環境に興味のあるかたは、特に連絡はいりませんのでどうぞご出席ください。

ネットワーク分科会(SIGNET)

前号でもお知らせしたように、SIGNETを実質的に活動させたいと思います。そのために、本号に同封したアンケートには是非回答して、事務所（東高円寺）にお送りください。

関西支部

SEA関西AI分科会開催のご案内

お待たせしました、関西でもようやく分科会が充足することになりました。最も希望の多かったAI分科会の第1回会合を、下記の要項で開催しますので、是非ご参加ください。

日時：昭和61年10月1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：ゲストをまじえて、AI開発環境についての意見交換、および今後の分科会の運営方法についての討議。

ゲスト：野辺良一（協同システム開発）

参加費：会員-500円

一般-1,000円

SEA関西第6回研究会

今回の研究会は、大阪大学の鳥居先生が主宰されている「高信頼性ソフトウェア研究会」と合同で、下記要項で開催します。

日時：昭和61年9月24日（18:00 - 20:00）

昭和61年9月24日（9:00 - 15:30）

場所：KDDハローイン

プログラム：

9月24日

懇親会-東京から大勢参加します

9月25日

・データ構造にもとづくソフトウェア設計の一方

白井義美（日本電子計算）

・制御用ソフトウェア開発環境の現状と問題点

岡村博（三菱コントロールソフトウェア）

・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）

・検査部門でのデータ活用法

藤原裕文（富士通）

・あるソフトウェア・パッケージのバグ・データの

時系列分析

石原千秋 (SRA)

・プログラム作成能力の評価尺度とデータ収集ツール

松本健一 (大阪大学)

SEA関西10月度月例会

日時: 昭和61年10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、盛田政敏 (078-391-8291-神戸コンピューターサービス)、臼井美義 (06-488-6021-日本電子計算) にどうぞ。

横浜支部 (Y-DOC)

1. 第5回 YDOC 会合

「納涼討論会・会食会」となった8月の YDOC は、佐原さん (NCC) の Mac と UNIX にたいする思い入れ (?) の程が伝わったのか、質疑応答が相次ぎ、盛況でした。Mac と UNIX (VAX) の良いところをうまく連携させて、使いやすい環境を実現しており、その使い心地は、次のように表せるとのこと。

UNIX + Mac > (MVS + VOS3 + OS IV) * *
(N5200 + 5550)

この発表の内容は、オフィス・ワークとしての「開発環境」、あるいは「ヒューマン・インタフェース」や「使いやすさ」等に問題意識を持っている方には、おおいに参考になると思われますので、YDOC で整理し、SEAMAIL に載せたいと思います。

2. YDOC とは

最近、会員が加速度的に増加しているとのことで、あらためて YDOC の説明をしてみたいと思います。

YDOC は、神奈川県に在住しているか、あるいは勤務地が神奈川県にあるソフト/ハード等の技術者を対象とし、その親睦と交流を図りながら、来るべき情報化社会

の中核を担う技術者の、相互の啓発の場となることを意図した、技術者のための集いです。

月1回の月例会を催しており、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ですので、気軽に参加して下さい。以下、今後の月例会の開催予定を示します。

3. 今後の予定

第6回 YDOC 開催予定

日時: 9月26日 (金) 18:30~

場所: 神奈川県政総合センター (横浜駅西口、三越裏)

9階 902会議室 (Tel. 045-312-1121)

費用: 1名 ¥1,000 (会場費、資料代等)

内容: 「AI: エキスパート・システムの構造」

伊野 誠 (日本ユニバック)

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) までご連絡下さい。

第7回 YDOC 開催予定

日時: 10月31日 (金) 18:30~ 場所: : 神奈

川県政総合センター

テーマ: 「失われたパラダイムを求めて」 松原 友夫 (日立SK)

SIGCAI 結成へ

大木幹雄さん (日本電子計算) ほかの方々から、CAI をテーマとする分科会を発足させたいという提案がありました。同封のアンケートをお読みになって、その趣旨に賛同する方は、ぜひ積極的に応募して下さい。

S E A これからの活動予定

- 1986 -

◆9月(September)

24日(水)

再利用分科会第8回討論会(機械振興会館)

30日(火)

教育分科会第5回会合(機械振興会館)

◆10月(October)

1日~2日

S E A 情報化月間記念フォーラム

「ソフトウェア技術と産業の未来を考える」

(東京・東条会館)

6日(月)

法的保護分科会第3回研究会

(ソフトウェア流通促進センター)

8日(水)

再利用分科会第8回研究会(機械振興会館)

9日(木)

A I 分科会第5回月例会(機械振興会館)

15日(水)

環境分科会第9回月例会(機械振興会館)

22日(水)

再利用分科会第8回討論会(機械振興会館)

22日~24日

UNIXワークショップ'86 in 横浜(ホリデー・

イン 横浜)

28日(火)

教育フォーラム「新入社員教育」(青年会議所)

◆11月(November)

6日(木)

7日(金) A I 分科会第6回月例会(機械振興会館)

)

盛岡ワークショップ報告会(青年会議所会館)

12日(水)

再利用分科会第9回研究会(機械振興会館)

22日(水)

再利用分科会第9回討論会(機械振興会館)

◆12月(December)

6日(木)

A I 分科会第6回月例会(機械振興会館)

- 1987 -

◆3月(March)

18~20日

春のセミナー・ウィーク(機械振興会館)

S E A 会員状況

昭和61年9月13日現在の入会会員状況は以下の通りです。

正会員--535名(先月比102名増)

賛助会員--14社(先月比3社増)

<地域分布>

	勤務	居住
岩手 =	2	2
新潟 =	5	5
栃木 =	2	2
茨城 =	2	4
埼玉 =	8	49
千葉 =	9	42
東京 =	367	213
神奈川 =	32	108
長野 =	6	6
富山 =	1	1
静岡 =	3	3
岐阜 =	1	3
愛知 =	8	6
滋賀 =	1	2
京都 =	4	8
大阪 =	48	38
奈良 =	0	2
兵庫 =	15	19
福岡 =	3	3
熊本 =	5	5
長崎 =	1	1
鹿児島 =	3	3

<男女分布>

男 = 502

女 = 33

<年齢分布>

20-24 = 21

25-29 = 112

30-34 = 148

35-39 = 146

40-44 = 56

45-49 = 30

50-54 = 7

55-59 = 6

60以上 = 5

会員の事務管理は、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。

TOOLS FAIR



March 30-April 2, 1987
Monterey, CA, USA

9TH International Conference on SOFTWARE ENGINEERING:

Formalizing the Software Process

Sponsored By:



IEEE COMPUTER SOCIETY
Technical Committee on Software Engineering



IEEE Institute of Electrical and
Electronics Engineers, Inc.



ACM SIGSOFT



Agence de L'Informatique
Etablissement Public National



afcet

The Fellowship of Engineering,
United Kingdom



Information Processing of Israel
Israel Section of IEEE

Singapore Computer Society

Singapore National Computer Board



Software Engineering
Association of Japan

A

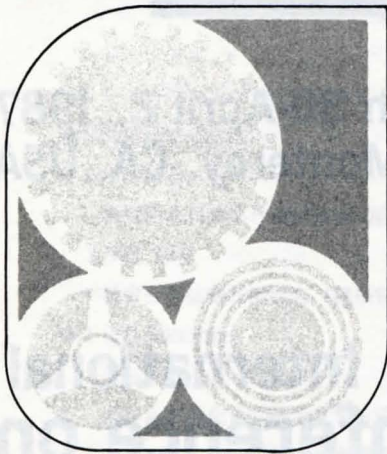
Tools Fair is being held in conjunction with the International Conference on Software Engineering. The purpose of the Tools Fair is to provide conference attendees an opportunity to view a variety of software development tools that are becoming available, both from the research community and from industry.

The Tools Fair will have two components:

- a) Exhibitions—An exhibition area will be available with booths set up for the purpose of demonstrating individual tools. Those wishing to exhibit a tool in the exhibition area should complete the application form on page 3. A \$1,200 fee will be charged for booth rental in the exhibition area. Demonstrators will be responsible for arrangement of the interior of their respective booths and must constrain their demonstration to that booth.
- b) Tools Presentation Track—A large room equipped with a video projection system will be available and tool presentations will be scheduled as a track, parallel to the paper sessions. Proposals for presentations in this tools track will be submitted to peer review by the Tools Fair program committee. Those wishing to provide presentations in the tools track should follow the instructions on page 2.

TOOLS FAIR

March 30-April 2, 1987
Monterey, CA, USA



9TH International Conference on **SOFTWARE ENGINEERING:** Tools Presentation Track

In conjunction with the 9th ICSE Tools Fair, there will be a special track of tool presentations, held in parallel with the other conference sessions. This track of presentations will be devoted to the description and demonstration of particularly interesting or innovative tools. The objective of this track is to make it possible for the developers of such tools to present their tools more thoroughly and in more depth than can be done at a booth in the Tools Fair. Those who are selected to make presentations in this track are welcome, and encouraged, to also display their tools at a Tools Fair booth.

The Tools Presentation Track is intended to complement the other tracks of ICSE 9 by offering descriptions and demonstrations of software development tool capabilities that currently exist in production or prototype form. Like the other tracks of ICSE 9, however, it will be strictly technical in nature. Presentations that are oriented toward marketing, as opposed to the technical aspects of a tool's capabilities, will not be accepted for the Tools Presentation Track. Presentation proposals will be carefully reviewed by the Tools Fair committee to ensure appropriate levels of technical content.

Presentation Format

Presentations in this track should generally be tutorial demonstrations of a software development tool and its capabilities. Each presentation will be approximately 20 minutes in length. Presentations will take place in a lecture hall equipped with projection devices capable of displaying video output on a large screen. This should facilitate live demonstrations of a tool's capabilities as an integral part of each presentation. The demonstration should illustrate the most novel, most important features of the tool, and should be geared toward a technically sophisticated audience. Questions from the audience will be encouraged, and thus the presenter must be prepared to provide substantive technical information in answering such questions. For this reason, it is crucial that the presenter have a deep technical understanding of the tool being presented; marketing representatives will almost certainly be inappropriate as presenters.

Selection Criteria

Proposals for presentations in the Tools Presentation Track will be evaluated by the Tools Fair committee based on both the quality of the tool to be described and the quality of the proposed presentation. Tool quality will be judged on the novelty of the tool's capabilities, the potential value of those capabilities to software developers, and how well the tool delivers its capabilities and the tool's robustness. Quality of the proposed presentation will be judged on the depth of technical content, freedom from marketing "hype," and appropriateness of the demonstration for illustrating interesting aspects of the tool.

Proposal Format

Each proposal for a Tools Presentation Track presentation should begin with a one page summary, clearly indicating the essential contribution made by the tool, how the tool is novel, and how the proposed presentation will demonstrate this. The summary should be followed by a description to the proposed demonstration, sufficiently detailed to enable the Tools Fair committee to assess the quality of both the tool and the presentation. A detailed outline of the presentation is a minimal description. A (possibly partial) transcript of the presentation and a sample of the intended demonstration is better. A videotaped example of the proposed presentation and demonstration may be best of all. Finally, every proposal must identify and provide a short biography of each person who will be taking part in the presentation.

Deadline for the proposal is November 1, 1986. Proposers will be notified of selection by January 15, 1987.

9TH ICSE TOOLS FAIR EXHIBITION APPLICATION

ORGANIZATION _____
NAME OF TOOL _____
PERSON RESPONSIBLE _____
ADDRESS _____
CITY _____
STATE _____ ZIP _____
COUNTRY _____
PHONE _____
TELEX _____

NOTE: Although exhibitors will be responsible for providing all required equipment and communications, the committee may be able to arrange for resource sharing if several exhibitors have similar needs.

HOST COMPUTER _____
OPERATING SYSTEM _____
PERIPHERALS _____
COMMUNICATIONS/PHONES _____
POWER REQUIREMENTS _____

I agree to submit payment of \$1,200 U.S. by January 30, 1987.

Checks should be made payable to IEEE Computer Society.

SIGNATURE _____

Attach a 100 word technical description of the tool.

Additional information concerning logistics will be provided upon receipt of application

Deadline for application is January 30, 1987.

Send application to either of the Tools Fair co-chair:

Larry E. Druffel
Rational
1501 Salado Drive
Mountain View, CA 94043

Prof. Jack Wileden
Department of Computer & Information Science
University of Massachusetts
Amherst, MA 01003



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