



SEAMAIL

Monthly Newsletter from
Software Engineers Association

Volume 1, Number 8 | August 1986

目次

編集部から	1
SEA SEMINAR 21 特集	2
ネットワークとその応用	2
寺本雅則 長井剛一郎 村井純	
ソフトウェア開発環境	6
岸田孝一 熊谷章 盛田政敏	
AIとソフトウェア工学	10
斎藤信男 横井俊夫 玉井哲雄	
ソフトウェアの再利用	14
大木幹雄 水谷時雄 村井進	
ソフトウェア設計技法	18
伊野誠 白井義美 落水浩一郎	
プロジェクト・マネジメント	22
松原友夫 堀江進 深瀬弘恭	
一般的トピックス	26
吉村成弘 佐野稔 辻淳二	
分科会および支部活動	30
AI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	36

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 白井義美 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：白井義美 盛田政敏

横浜支部：熊谷章 林香 藤野晃延 松下和隆

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 V o 1 . 1 , N o . 8 昭和61年8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

セミナー21特集

今月号は、SEA SEMINAR 21の総特集号としました。9月3日～6日に予定されているこのイベントは、春のセミナー・ウィークに続く、SEAとしての2回目の大行事です。

SEAでなくてはできない企画をということで、各セッションを、プレゼンテーションと自由討論を半々の形にし、インフォーマルなミニ・ワークショップ・スタイルというこれまでにない試みにチャレンジすることになりました。

こうした新しい試みに対して、講師陣の意気込みは強く、そのことはここに集められた各講師のポジション・ペーパーからもうかがえるでしょう。

当初、このSEAMAILは、8月の上旬に皆さんの手元におとどけし、セミナー参加応募の参考にしていただく予定でしたが、編集長が仕事(?)で2週間ほど日本をるすにしたり、編集ボランティアたちも忙しかったりして、また船便になってしまいました。

そこで、方針を変更して、各講師のポジション・ペーパーを1ページに編集しなおし、いわば「拡大誌上フォーラム」のかたちで、こらからの会員のみなさんの意見交換、情報交換のための材料を提供しようということになりました。実際にセミナーに参加できない方々の(特に地方在住の方々)の積極的な誌上討論参加を期待します。各講師のポジション・ペーパーに対する意見・質問・コメントなどをどしどし編集部(東高円寺事務所)までお寄せ下さい。

分科会・支部から

現在、分科会は7つが発足しており、ネットワーク分科会を除いて、月例会を中心に活発活動しています。また、支部も関西と横浜の2つが、分科会以上に活動していますが、今月の分科会および支部からの報告は、誌面の都合上、今後の予定を中心に掲載しました。

夏は、夏期休暇等もあって、若干参加者の数が減るなどがあるようですが、活動は従来以上に、活発に行われています。こうした分科会および支部からの成果報告は、次号から再び、順次掲載していきます。

Call for Papers

今月号では、CALL for Paperを2種類

掲載しました。

ひとつは、来年3月30日～4月2日にアメリカのモンタレーで、第9回ISCEに併設して行われるTOOLS FAIRのものです。SEAも協賛団体になっていきますので、みなさんからの積極的な応募をお願いします。

もうひとつは、来年6月4～5日に東京で行われる、「ソフトウェア・シンポジウム'87」の論文募集です。次回から情報サービス産業協会(JISA)とSEAとの共催になります。こちらのほうにも、積極的に論文を応募ください。

代表幹事からの御挨拶

前号でお約束した、代表幹事の鈴木弘さんからの、その後挨拶を今月号こそ掲載する予定でしたが、今回も間に合いませんでした。次号にこそ掲載したいと思います。ご容赦ください。

次号、9月号は、これまでと同様の編集になる予定です。誌上フォーラムでは、「情報処理技術者試験」を取り上げます。その他には、SIGからのレポートや会員からの投稿を掲載します。

また、前号でもお知らせしたように、秋のセミナー・ウィークの各セッションに参加して、その講演および討論をまとめる役割の臨時編集ボランティアを追加で募集しています。セミナーへの参加は無料になります。希望者は、どうぞSEA事務局までご連絡ください。

秋のセミナー・ウィーク
セッション・グループ A

ネットワークとその応用

これまで、コンピュータ・ネットワークに関する技術開発やシステム構築がアメリカを中心に進められてきたのは、広大な地理的国土とその上に散らばった分散型の社会構造からくるニーズのせいであることは、いうまでもありません。一方、すべてが東京を中心とする文化的・社会的構造を持つ日本の場合には、そうしたニーズがどちらかといえば低く、ネットワーク化が遅れていたといえましょう。

しかし、先頃の電気通信法の改正によって、この状況は一変したようです。世界のどの国とくらべてもひげをとらない日本のハード・ソフト技術の現状からすれば、今後あらゆる面でのネットワーク化が急速にすすむものと予想されます。そのことは、われわれソフトウェア技術者の日常や産業の形態に対しても、かなり大きな影響をもたらすでしょう。

このセッション・グループでは、ソフトウェア開発へのネットワークの応用をメイン・テーマとして、さまざまな角度からその問題点や可能性を探ることを意図しています。

セッション A1

ここでは、「ソフトウェア開発現場とネットワーク」というテーマで、寺本雅則（日本電気）さんから、御自身の経験を土台としたケース・スタディー形式で、ソフトウェア開発の現場にネットワークを導入し利用するさいのいろいろな問題点とその解決方法を探っていただくことにしています。

LANやワークステーションの組み合わせによる環境の分散化、開発マシンとターゲット・マシンとの結合や機能分担、在宅勤務やリモート・オフィスの具体化にともなう管理上の問題点など、討論の材料にはことかかないでしょう。

講師の寺本さんは、日本電気マイクロコンピュータ・ソフトウェア開発本部長代理、かねてから、ソフトウェア生産データの収集や分析に強い関心をお持ちであり、その意味で今回のセミナーでは、ネットワークが開発現場にもたらす影響について、単なるお話だけではなく、

いくつかの定量的データを用いた問題提起が期待されます。

セッション A2

来るべきネットワーク社会で、中心的な役割を果たすのは、光ファイバ、衛星通信、CATV、ビデオテキストといった、新しいコミュニケーション媒体を取り入れた、マルチ・メディア・ネットワークでしょう。しかし、これらの話題は、これまでマスコミ等で華やかに取り上げられている割には、その実態や技術の動向は、意外に知られていません。

このセッションでは、もともとTV送信技術のコンピュータ化に関係したことがきっかけでこの業界に入り、その後ご自分でCATVビジネスなども手がけられて、ネットワークの技術動向に造詣の深いシステム工学社長の長井剛一郎さんを講師にお迎えしました。セッション・テーマは、「マルチ・メディア・ネットワークの現状と今後の展望」。全員での討議を通して、今後ソフトウェア技術者や業界が、VANビジネスに何を期待し、どう取り組むべきかを考えてみたいと思います。

セッション A3

日本における代表的なネットワークといえば、JUNETを挙げることに異論はないでしょう。「UNIXネットワーク環境」をテーマとするこのセッションの講師村井純（東京工業大学）さんは、2年前に小規模な実験からスタートしたJUNETの構築や運用に関して、中心的な役割を担って来られました。

また、慶応大学や東京工業大学における教育研究のためのソフトウェア開発環境のネットワーク化も積極的に手がけられています。こうした具体的な経験にもとづいて、ネットワークの技術的側面からみた、ソフトウェア開発環境の分散化のあるべき姿と問題点を討論しようというのが、このセッションのねらいです。

セッション A1

ソフトウェア開発現場とネットワーク

寺本 雅則

1. 開発現場でのネットワークの必要性

ソフトウェアを開発する者にとって、通信とはどのような意味を持つものなのでしょうか。

わたくしたちは単にターミナルの前にすわって、単一のマシンのソフトウェアをデバックしているだけでしょうか。実はそうではなく、仕事を拡大し、掘り下げていくためには、日常まわりの人間と、予想以上に多くのコミュニケーションを行っていることが必要なのです。

グループの進捗会議から、得意先との折衝、特定の技術情報の収集から、技術的ディスカッションまで、その形態は多種多様にわたっています。

2. 新しいメディアの開発への利用

このような、双方向あるいは単方向のコミュニケーションは、急速に電子化されていく（あるいはもうすでにされている）傾向にあります。

通信の形態は、そのメディア、ネットワーク・トポロジー、情報の交換の仕方等において、ますます多様化していくでしょう。電子メールや電子掲示板の普及はそのあらわれですが、将来はどう発展するのでしょうか。

また、ソフトウェアを開発したり、そのソフトウェアを走らせるために用いられるコンピュータとコミュニケーションとは、密接に関連しており、わたくしたちの会社でC&Cと呼んでいるような発展形態になるものと考えられます。今日、コンピュータの利用技術は、社会におけるさまざまな問題の解決や仕事の能率を向上させる上で、重要な役割を占めているわけですが、ネットワークを交えた利用形態のみが、個人・組織・地域等の壁をこえた複合的な問題の解決を可能にするものだからです。

このような変化は、ソフトウェアを開発しているわたくしたちの現場にも、いやおうなく浸透していくでしょう。現実には、そうした動きが、わたくし自身の周りでも、すでにいくつか起こりつつあります。たとえば、勤務地を会社の事情でなく個人の事情に合わせた在宅勤務の誕生やサテライト・オフィスの出現などは、このような傾向のあらわれだと、みることができでしょう。

3. ネットワークのインパクト

ソフトウェアを開発するための技術を対象とする学問分野（すなわちソフトウェア工学）は、いまだ完成されたものとはいえない状況にありますから、わたくしたちが現場で直面する日々の問題は、自分たち自身の知恵で解決していかなければならない、というのが現実です。このような状況下で、LAN（ローカル・エリア・ネットワーク）やWAN（ワイド・エリア・ネットワーク）等の現場への導入によるインパクトが、そろそろ色々な問題となってあらわれてきつつあります。

4. コンピューティング・パワーの分散化

たとえば、ソフトウェア開発においては、ホスト・コンピュータの負荷が開発効率を左右することが多いのですが、最近では、LANによるコンピューティング・パワーの分散化が進んでいます。

その場合、ホスト・コンピュータの負荷均一化に効果のある分散環境とは、どんな形なのでしょうか。また、そのとき、どのような現象が利用者サイドに生じるのでしょうか。さらに、分散ファイル・システムを導入すると、環境の利用形態はどんなふうに変化し、それに対して現実はどう対処したらよいのでしょうか等々、開発環境の利用者や運用者のサイドからみた問題点が、いくつも考えられます。

今回のセミナーでは、以上のようにソフトウェアを開発する現場と、ネットワークとのかかわりあいから生じてくる問題点を、いくつかの観点から、わたくし自身の経験を交えて考察し、将来のネットワーク環境のあるべき姿を、参加者の方々と一緒に考えてみたいとおもっています。現実にネットワーク環境を利用されている方々、またはこれから環境のネットワーク化を考えておられる方々に、それぞれの具体的な問題意識にもとづいて、活発な討論をいただきたいと存じます。

(日本電気)

セッション A2

マルチ・メディア・ネットワークの現状と今後の展望

長井 剛一郎

1. 新しい文化の始まり

衛星通信、CATV、ビデオテックス、パソコン通信、これらはニュー・テクノロジーの開化と共に微妙に入り組み、まったく新しい文化を作ろうとしている。

昭和58年2月、我が国最初の実用通信衛星「さくら2号a」(CS-2)は、予定時刻通りの正確さをもって打ち上げられ、衛星時代へ日本も本格的に参入することになった。

かくて郵政省の衛星利用のパイロット計画が産業界の注目を集めるようになったが、この計画の実験内容は次の通りである。

- (1) コンピュータ・ネットワーク
- (2) 新聞紙面伝送
- (3) CATV (有線テレビ) の番組分配

これらは、昭和52年に打ち上げられた実験通信衛星で得られた実験結果をもとに、実用通信衛星(CS-2)でさらに高度な利用実験をし、これを63年に打ち上げ予定の次期実用通信衛星(CS-3)での民間本格利用促進につなげようというものである。

上であげた実験内容のうち、(1)と(2)は、すでに昭和59年3月から実験が開始されている。その主力となるコンピュータ・ネットワークの実験は、地球局を保有する10社が中核となり、9グループ19社が当初参加しており、昭和59年度中にさらに18社が新規に参加して、13グループ37社で、目下実験が進められている。

また、アカデミー・コンピュータ・ネットワークの実験として、米国のカーネギー・メロン大学と東京工業大学との間で実験が行われ、国際間のネットワーク実験も幅広いものになりつつある。

2. ニュー・メディア・ブーム

次に、昭和50年代後半に起こった、「ニュー・メディア」ブームがある。このとき、御三家として脚光をあびたのが、「都市型CATV」、「キャプテン(文字図形情報ネットワーク)」に代表される「ビデオテックス

(双方向文字図形情報システム)」と、「文字多重放送」の3つである。そしてこれらは現実のものとして動き始めたことは、マスコミで大きくとりあげられたこともあって、御存じの方も多であろう。

郵政省の「CATVの振興策」、建設省のインテリジェント・シティ構想の中における「アーバンCATV構想」等はその例である。

また、ビデオテックスは、NTTの「キャプテン」の伸び悩みはあるけれども、プライベート・キャプテンは地方の各地で、着実に根を下ろし始めている。

NTT、富士通、松下、読売新聞の協同出資で設立された「テレメディア・サービス」は、群馬、静岡、長野、福島各県、神戸、京都市等の各地で事業を展開中であり、さらに積極的な姿勢がみられるなどは、この1例である。

しかも、これらのDB作成においてもコンピュータ・ネットワークは欠かせないものになりつつある。

3. VANサービスの本格化

昭和60年4月1日の、電気通信事業法の施行に伴い、従来種々の規制を受けていた通信回線が自由化された。

これによって、第1種電気通信業者による通信網の整備、新設、さらに第2種電気通信事業者による新しいサービスの提供が計画、実施されてきており、いわゆるVANサービスが本格的に進展してきている。

このような状況の中で、不特定多数を対象としたネットワーク・サービスの有望なもの1つとして、パソコン通信サービスが注目されてきている。そして、ここで使用されているパソコンは、当然ながらビデオテックスの端末にもなり得る。これらが相互に結合されたとき、一大ネットワーク社会が出現するものと思われる。

今回のセミナーでは、こうした現状をふまえて、今後のVANビジネスにどう取り組むべきかを、討議していきたいと思う。

(システム工学)

セッション A3

UNIXネットワーク環境

村井 純

1. . 環境技術の課題

ソフトウェア技術者にとって理想的な計算機環境は、知識と発想を表現する知的作業から、実際のソフトウェアを取り扱う技術的作業までを、きわめて効率よく支援できるものでなければならない。

既存の技術を単純に集積したような環境は、利用者を混乱させ効率を下げる。現在のほとんどのメインフレーム上の環境はそうである。新しい技術を取り入れる場合も、単純にそれを既存の環境に追加しただけでは、全体のアーキテクチャ的なバランスを破壊してしまうこともある。

我々が直面している課題のうち、技術的に重要なのは、日本語の処理と、ネットワーク技術の取り扱いである。ソフトウェア技術者の要求に耐えるようなかたちでこれらを導入するためには、環境全体の論理的構造に正しく整合したかたちにしてやらなければならない。

環境におけるネットワーク技術を考えるにあたっては、

- ・ 現在利用している既存の基礎ネットワーク技術の整理
- ・ 分散化という視点から見た開発環境の構造
- ・ これから構築すべき分散環境に関するアイデア

の3点が重要である。

2. セミナーでの討論テーマ

今回のセミナーでの具体的な議論は、一般的な分散環境の構築が最も活発に行われているUNIXを中心とし、次の各項目を議論したい。

(1) BSDのネットワーク・プロトコル

4.2BSDは、OSの構造そのものを、ネットワーク技術を前提として設計したUNIXであり、したがって、ソフトウェア専門家の要求に応える環境を提供することができる。また、階層的なプロトコル構造を汎用的に導入するための機構が組み込まれており、DARPAのTCP/IPが実装されている。ここでは、OSの構造としてネットワーク技術をどのように取り入れているのかを中心に、4.2BSDカーネルの構造を分析する。

(2) UUCPネットワーク

UUCPは、シリアル回線を用いたネットワーク・プロトコルで、現在約4000ものシステムがそれによって結合され、巨大なネットワーク・コミュニティを構成している。ここでは、広域ネットワークの最も基本的なプロトコルとしてのUUCPの構造に加えて、USENETなど、UUCPプロトコルを用いた実際のネットワークの構成に関する議論を行う。

(3) 既存のアカデミック・ネットワーク

ソフトウェアの研究開発者にとっての環境は、LANによる分散環境だけでなく、技術者間の情報交換を基調として、ネットワーク・コミュニティも重要な役割を果たす。これらの両方を同時に正しく利用する環境を構築するために、ARPA、CS、JUNETなどの現状や問題点、OSに関連した技術的方式などを考察する。

(4) 分散環境構築の素材

ユーザがネットワークで結合された個々のシステムの存在を意識せずにすむような「分散環境」は、一般的に、設計がむずかしい。これは、個々のシステムの管理体制、OSの機能、ハードウェアの機能などの差異によって、構築可能な環境が異なるからである。ここでは、これらの問題をどのように解決すべきかという点に注目しつつ、NSFなどの要素技術について、今後の動向を展望する。

(5) 分散化の視点から見た環境の構造

現在、我々が必要としている個々の技術の整理と、個人環境からそれらをアクセスするための理想的な構造を基盤として、これまでに分析したネットワーク関連技術をどのように組み合わせ、取り入れて構築していくべきかを、今後実用化が進むことが予想される新しい技術も考慮しつつ、将来の分散環境の構造に関する議論を行う。

(東京工業大学)

秋のセミナー・ウィーク
セッション・グループ B

ソフトウェア開発環境

さまざまなツールを統合化した環境の概念が、ソフトウェア・エンジニアリングの中心的な技術課題としてとりあげられてから、すでに数年の時間が経過しました。

その間に、環境構築技術の発展のために、さまざまな実験的プロジェクトが世界各国で同時並行的に進められ、また産業界でも、現実の開発生産性や品質の向上を目的とする環境の改善が、それらの研究成果を踏まえた形で、おこなわれてきました。

一方、ハードウェア技術の革新は、数年前にはまだ夢であった超高性能パーソナル・ワークステーションを低価格で商品化しようところまで進み、それらをLAN等で接続することにより、ソフトウェア・エンジニアにとって理想的な環境を実現するための技術的・経済的条件は、ほぼ整いつつあるように感じられます。

こうした背景を踏まえて、このセッション・グループでは、環境をめぐる3つの話題をとりあげて、討論します。

セッション B1

ここでは、「環境構築と技術移転」というタイトルで、岸田孝一さん(ソフトウェア・リサーチ・アソシエイツ)にお話をいただきます。つまり、実際に環境の構築・改善やツールの導入を試みようとする場合に、最大のネックになるだろうと予想されるその人間的・社会的側面について考えてみようというわけです。

講師の岸田さんは御存じの方も多いと思いますが、日本におけるUNIXブームの仕掛け人のひとりであり、自社内における環境構築や利用の実績はもとより、JSDプロジェクトやSIGMA計画のプランニング等を通じて、環境をめぐる技術移転のさまざまな側面について、豊かな経験をお持ちです。

セッション B2

AIとソフトウェア工学とを融合させる動きが最近注目を集めていますが、そのひとつの具体的な形は、新しいAI的パラダイムを支援するプログラミング環境を具体化することです。

熊谷章さん(パナファコム)のセッション「マルチプログラミング環境LIPS」では、講師自らが開発した複数のAIプログラミング言語(Lisp, Prolog, Smalltalk)とデータベース(ingres)を1台のパーソナル・ワークステーション上で同時並列的に支援する環境を実例としてとりあげ、未来における新しい開発環境のイメージについて、討論をおこないます。

講師の熊谷さんは、長岡の環境ワークショップやその報告を兼ねたエイプリル・フォーラムでのユニークな発言からもわかるように、人間性中心主義・不易流行・マイノリティ重視を旗印に、プログラミング・カルチャーは自ら土地を耕さなくては生まれてこない、と主張するB型人間です。

セッション B3

次世代における理想の環境について議論することもたしかに楽しいが、いま自分たちが使っている開発環境の中にどんな問題があり、とりあえずそれをどう解決するかをまず考えようというのが、盛田政敏さん(神戸コンピュータサービス)の主張です。セッションのテーマは、「大型ビジネス・アプリケーションの開発環境」講師は、いうまでもなくSEA関西支部の代表世話人、会社では部下たちからけむたがられるくらいに新しもの好きのソフトウェア・マネージャです。

第3次オンラインを例にあげるまでもなく、ビジネス・アプリケーション分野での開発需要は、ますます増加しています。にもかかわらず、その開発支援環境については、改善の努力があまり真剣におこなわれていないようです。COBOLとソフトウェア工学との相性(?)の悪さは、かねてから指摘されていますが、それをそのままにしておいては、環境論議はいつまでたっても机上の空論に終わってしまうでしょう。

セッション B1

環境構築と技術移転

岸田孝一

1. はじめに

「環境」ということばを、この世界で最初に耳にしたのは、たしか1979年に A. Wasserman を招いて Ada についてのセミナーを開いたときだったと記憶している。ちょうど DoD が Stoneman の要求仕様をまとめていた時期で、Unix とどこがちがうのかといった生かじりの会話をかわしたおぼえがある。

翌年の夏に、Unix が自分の会社にはいつてきた（というより、わたしがむりやり導入したといったほうが正確な表現だろう）。その準備のために、いろいろな人の話をきいたなかで、あるアメリカ人がしてくれた忠告が、いまだに耳にのこっている。

「Unix はたしかによい道具だが、それはひとつの土台にすぎない。ほとくの会社はたかだかプログラマ数人のちいさなソフト・ハウスだけれど、Unix のうえに自分たちの環境をつくるのに半年ちかくかかった。きみのところは、はるかに大世帯だし、これまでの経験もたりないから、かなりの時間を必要とするだろう。そして、残念ながら、その時間をどうやって短縮するかはきみ自身の問題だから、ぼくはあまり力になってあげられない。」

そのときは、ただ、そんなものかとおもって、ききながしていた。実感をもって真剣にかんがえはじめたのは、Unix を自分でつかいだして、しばらくたってからのことである。

2. 環境づくりのむずかしさ

ソフトウェア・システムとしての環境は、さまざまなツールを統合したものと定義される。そこで問題は、システムを構成すべき個々の要素のえらびかたやそれらの統合のしかたが、技術の進歩にともなって、たえずかわってゆくことだ。

このことは、1981年から5年間、JSDのSME Fプロジェクトに関係し、Arcturus, Gandalf, Tool-pack など、アメリカのR&Dコミュニティでの環境プロジェクトと同時並行的に情報交換をしてゆくなかで、きわめてつよく実感された。

いくらか手前みそになるが、SME Fがナショナル・プロジェクトにしてはめずらしく（！）成功をおさめた理由は、そうした世界の最先端の技術動向と直接に接触をもち、なおかつプロジェクトに参加した各社の技術者のみなさんが、積極的に技術的・人間的交流をおこなって、自分の視野をひろめるための努力をおこたらなかったからだとおもう。

別のいいかたをすれば、環境の構築や導入・利用には、つねに技術移転の問題がつきまとい、そのことの解決には、たんに技術的な知識や見通しだけでなく、環境開発者自身の人間的な思想がどこまでそのアーキテクチャに表現されているかが、キーポイントになってくるのである。たとえば、Unix がプログラマたちに好まれるのは、Ken Thompson や Bill Joy の個性が直接肌感じられるからであり、Macintosh の魅力は、まさに Alan Kay の革命的発想のなせるわざであろう。

その意味で、さる2月の長岡ワークショップで熊谷章さんがいわれた「教養のない人間に環境をつくる資格はない」は、名言だとおもう。

3. セミナーでの発表と討論

今回のセミナーでは、こうした環境構築における人間的側面でのいろいろな問題を、参加者の方々といっしょにかんがえてみたい。

わたしのほうから提供できる材料は、過去6年間の Unix を土台とする社内環境の構築に関するデータ、JSDプロジェクトでの経験、海外のワークショップなどで仕入れた情報、環境プロジェクトとしてしばらくは実質的にテイク・オフできそうもないSIGMAへの批判的意見などである。

具体的な討論のテーマやすすめかたは、参加者の方々の要望にしたがって、その場でフレキシブルにかんがえたい。

(ソフトウェア・リサーチ・アソシエイツ)

セッション B2

マルチ・プログラミング環境 LIPS

熊谷 章

1. コンピュータと言語

コンピュータは、人間の思考のためのメタファーを提供する。さまざまな知的活動にコンピュータを用いることにより、われわれは、物事を表現するための豊かで新しい言語を獲得できた。それらの言語に含まれている概念を比喩的・類推的に用いることによって、人間は自らの思考（ときには思考に関する思考）をより強力に表現しうる。

コンピュータを利用したモデルは、また、正確でもある。何らかの理論をコンピュータ上に実現すれば、きわめて細心な研究者でさえ見落とすような、概念的な誤りを発見することができる。思考と実験のサイクルを何回か繰り返すうちに、頭では思いつかなかったような障害にであうこともありうる。

コンピュータを使った問題解決は、その問題が何を要求しているかを明確にする。一度そのプログラム（モデル）を実行してみれば、問題がどれだけの情報処理を要求しているかが、はっきりする。

2. 複合的プログラミング環境

プログラミング言語はメタファーの表現媒体として、きわめて重要な意味を持つ。ある問題が解けるか否かは、それを解くのに用いられるメタファーいかんによって決まるといってよい。世の中にいろいろなコンピュータ言語が存在する背景には、こうした理由がある。それぞれの言語は、固有の表現力や概念、あるいは機能を持っているのである。

コンピュータで解決すべき問題が複雑になるにしたがって、言語の持つこのメタファーはますます重要になるものと考えられる。重要というよりは、問題の解法を実現できるか否かの決め手になるであろう。

そうした意味で、理想的なプログラミング環境は何かといえば、それはマルチ・メタファーを可能にするような環境である。1つのプロセス（コンピュータ内における実行単位）内で、複数のメタファーを同時に実行できることが、複数の側面を持つ現実の複雑な問題を解決する上で、実用的であり、意味がある。

そうした環境をマルチ・プログラミング環境と呼ぶ。

3. LIPSの狙い

われわれは、エキスパート・システムの開発ツール、人工知能システム、第5世代プロジェクトなどに強い関心を抱き、そうした問題領域をターゲットとする新しいマルチ・プログラミング環境として、LIPSと呼ばれるシステムを開発した。

LIPSとは、Lisp, Ingres, Prolog および Smalltalk の頭文字をあらわす。

すなわち、現在AI関連のアプリケーションで用いられている3種類の言語が持つそれぞれの特徴（記号処理、動的記憶管理、宣言的知識表現、推論機能、オブジェクト指向、ビジュアル・プログラミング等）と、リレーショナル・データベース・モデルの概念とを、ユーザがすべて同時並行的に利用できるような環境を意図している。

また、LIPSはUNIX上に具体化されているので、ハードウェアから独立したプログラミング環境であり、その他の言語（C, Fortran, アセンブラ等）とのリンクも可能である。

LIPSプログラミング環境下では、実際1つのプログラムのなかで、ユーザは任意の言語を自由に組み合わせ使用することができる。たとえば、マン・マシン・インタフェース部分に“S”，決定されたアルゴリズムの実行に“L”，宣言的知識と推論に“P”，大量のデータの保存と検索に“I”，スピードの要求される部分のプログラム部分にはCやアセンブラを使うといった具合である。

4. セミナーでの討論

将来のプログラミング環境においては、こうした複合的な言語をいかに支援するかが、きわめて重要な課題になるものと予想される。

今回のセミナーでは、LIPSの開発および試用にさいしての技術的な問題点を具体的に提示し、おなじような問題意識をお持ちの参加者の方々との討論を通じて、環境の未来について考えてみたい。

(バナファコム)

セッション B3

大型ビジネス・アプリケーションの開発環境

—構築のための具体的方策をさぐる—

盛田 政敏

1. 呼びかけ

環境というと、UNIXハッカーの専売特許のような感があるが、ハッカーたちの環境が、現実的なビジネス・アプリケーション開発環境を支援して成果をあげている、という話は、あまりきかない。

ビジネス・アプリケーションの開発環境はどうなっているのか？ 技術者たちはメインフレームの提供する環境に満足しているのだろうか？ あるいはシグマが解決してくれるだろうと期待して待っているのだろうか（今のところ、期待は薄いようだ）？ ビジネス・アプリケーションを開発しているプログラマ、またそのマネージャたちは、これからの自分たちの環境改善についてどう考えているのだろうか？

ひとくちに環境といっても、いろいろなものがある。制御系とビジネス系では、かなり異なった様相を呈している。また、同じビジネス系でも、大型メインフレームといわゆるオフコンを同じに考えるには、少し無理な点もあるようである。さらには、それぞれの人の立場によって、環境の概念が多様な捉え方をされているのが現状だと思う。高尚な話でなくてよい。環境構築や環境改善、環境転換のための具体的方策や知恵を探ってみたい。

このセッションで、とくに「大型ビジネス・アプリケーション」とことわったのは、次の理由からである。

まず、ビジネス・アプリケーションの開発が、世の中の需要としては最も多いと思われるのに、なぜか、こういうタイトルのついた開発環境がとりあげられない（一度やってみたらどうか）。さらに、限られた時間の中でこの討論であるので、ある程度的を絞って、議論が発散しないようにしたい（ある程度共通の土俵で）。

2. 討論の内容と進め方

大型ビジネス・アプリケーションの開発環境について、環境構築のための具体的方策（考え方、アイデア、事例

）を探ってみたい。

ソフトハウスとしての我々の部門におけるアプローチを1つの事例として紹介し、できれば参加者の中からも2、3人の方に事例報告をいただき、これらを材料にしながら、全員参加の討論を通じて、環境構築のための具体的方策を考えてみたい。

3. 参加してほしい方

たとえば、

- 開発環境（マシン）があるが・・・
 - ・こんな風にやっているよという人
 - ・効果のほどに疑問のある人
 - ・プログラマの支援しかしていないと不満な人
 - ・いまいち使い勝手がよくないと気に入らない人
 - ・多くの人が使っていて思うように使えないと憤慨している人
 - ・コストがかかりすぎて、と財布の紐が心配な人
- 開発環境（マシン）を持ちたいが・・・
 - ・持たなくても、結構うまくやっていると人
 - ・シグマ（UNIX）の動向が気になって、と日和見の人
 - ・うちの会社では、とあきらめている人
 - ・よくわからないが、どうすればいいんだという人
 - ・高くつきそうで、と思案している人
- 開発環境（マシン）がなくて
 - ・夜間、休日にユーザ先マシンを使いながらアホらしい、と思っている人
 - ・部下のために何とかやろうと思っているマネージャの方
 - ・身軽でせいせいしているという人
- 俺も一言しゃべりたいという人
- 俺の所はこんなに快適な環境だよと自慢したい人
(神戸コンピューターサービス)

秋のセミナー・ウィーク
セッション・グループ C

AI とソフトウェア工学

AIブームはまだ続いているようです。それに対して期待ばかりが上滑りして足が地についでいないと批判する人たちもいれば、他方では、最近のソフトウェア工学の行き詰りを指摘し、そうした状況を打破するために、AIや知識工学との結びつきが必要だと指摘する人もいます。本誌第2号でも、AIとソフトウェア工学の接点に関する誌上討論を特集しましたが、そこでもやはり2通りの意見がだされています。

いずれにせよ、ソフトウェア技術は、ひとところに留まってはできないでしょう。事実、オブジェクト指向やルール型プログラミング等の新しい概念や技法の導入によってプログラミング方法論が多様化したのは、AIの研究成果のおかげだといえます。

これまでAIセミナーというと、ただの言語講習会だったりすることが多いのですが、今回のセミナーでは、ミニ・ワークショップという形式を活用して、ソフトウェア開発にAIをどう生かすべきか、あるいは生かせるかどうかをメイン・テーマに、地に足のついた討論が展開されるものという期待があります。

セッション C1

スタンフォード大学、カーネギーメロン大学に留学し、日本におけるソフトウェア工学（OSや開発環境）の代表的研究者のひとりである斎藤信男先生（慶応大学）には、「知的プログラミングを目指して」というテーマで、ソフトウェア工学とAIの融合によって、ソフトウェア開発の改善がいかんして可能かを展望していただきます。

そのあとで、受講者の積極的な参加をえて、具体的なAI/知識工学技術の現場への導入の方策を探ることを考えています。それは、ひとつの統合的な環境を構築することから、あるツールの一部にAI風のお化粧をほどこすことまで、いろいろなアプローチがありうるでしょう。

セッション C2

現在のAIブームをまきおこしたきっかけは、御存じ

第5世代コンピュータでした。ICOT（新世代コンピュータ技術開発機構）において、プロジェクトの推進に従事してこられた横井俊夫さんには、ソフトウェア工学と第5世代および人工知能とのかかわりを中心に、考察・討議していただきます。テーマは、そのものずばり「AI、第5世代、ソフトウェア工学」。

AI手法の適用事例としての知的プログラミング、普遍的な問題解決過程としてのプログラミング、AI向き言語と環境等々、とりあげる話題はいろいろあります。おそらく、情熱的で理想家肌の横井さんの熱弁に受講者がまきこまれて、かなり「飛んだ」議論がくりひろげられるのではないのでしょうか？

セッション C3

AIを実現するのもソフトウェアであるという観点に立って、春のセミナー・ウィークでは「エキスパート・システム」の講演をいただいた玉井哲雄さん（三菱総合研究所）には、今回は「ソフトウェア技術という視点でAIをみる」というテーマをとりあげていただくことになりました。

玉井さんは、これまでプログラムの静的解析、数学的検証、自動合成などソフトウェア工学とAIとの境界領域におけるツールの試作・開発を数多く手がけてこられ、現在は人工知能開発室長として、エキスパート・システムの開発に取り組んでおられます。そうした経験を踏まえた内容の濃い討論になることでしょう。

セッション C1

知的プログラミングを目指して

斎藤 信男

1. AI技術の影響

ソフトウェアの生産性を向上させることは、ソフトウェア工学の重要でかつ本質的なテーマである。そのためには、過去にも多くの技術の開発と普及が行われてきた。最近では、ソフトウェア開発環境が生産性向上のためのキーとなる技術として期待されている。

AI技術は、あらゆる計算機技術を利用し、またあらゆる計算機技術に影響を与えるものとして、最近目ざましく発達してきた。一面から見れば、従来では無視されていたか、あるいは不可能であったことが、応用ソフトウェアの中で簡単に実施できる。たとえば、自然言語風のインタフェースを応用ソフトウェアに組み込むことは、AI技法を使用することにより可能となる。

一方、AI技術を効率よく実施するための計算機の追求は、新しい計算機アーキテクチャを生みだし、また新しいプログラミング・パラダイムを生みだしている。それ自体、計算機技術の大きな発展の源となる。その意味で、AI技術は、計算機科学のひとつの大きな分野として認知されている。

2. ソフトウェア工学とAI技術との関連

ソフトウェア工学とAI技術との関連を考えると、次のような問題が考えられる。

- (1) AI用ソフトウェアを如何に効率よく開発するか
- (2) ソフトウェアの開発のために、AI技術が如何に役に立つか

前者は、ソフトウェア工学からAI分野への寄与であり、後者はAIからソフトウェア工学への寄与である。これからのAI分野の拡大は、AI用ソフトウェアの開発が急激に増えることが予測されるので、ソフトウェア・エンジニアリングあるいは業界にとって(1)は重要な問題である。

一方、ソフトウェア工学における現状をブレークスルーする技術として、AI技法が一番期待できる。これは、場合によっては、従来のソフトウェア生産の体制やプロセスを大幅に変更させる力を持っている。ここでは、主

として(2)の問題を具体的に考えてみたい。

ソフトウェアの開発においては、色々のソフトウェア・ツールが開発されてきた。最近では、それらを集大成してソフトウェア開発環境として実現することが行われている。個々のツールのなかには、かなりの工夫がなされているものも少なくない。プログラミング・エディタには、たとえば構文向きエディタのように、何らかの知的判断をエディタで行っているものもある。このような知的判断をくらすのは、いわゆる知識ベースが得意とするところである。このような知的判断をくらすと、大きな効果を現すソフトウェア・ツールがたくさんあるであろう。プログラムの動的性能を分析するツールがあるが、たとえばどの部分を変更すればよいのかを提示してくれるツールに拡張することができる。文章道プログラミングで、色々のアルゴリズムの検索や比較が即座にできれば、より優れたプログラムが記述できる。

3. AI技術との接点

これらのAI用ツールを実現していくためには、ソフトウェア工学的なツールが既にたくさん実現してあるような環境、たとえばUNIXの環境に、AI環境も実現することが必要である。AI専用マシンも効率的には悪くはないが、プロトタイプの開発には使えなくても、運用マシンとしてはあまり適していないだろう。むしろ、専用チップとして汎用マシン内に組み込まれるほうが、能率がよく、使いやすい。

ソフトウェア工学からすれば、多くのAIツールを細かいレベルの作業に実際に利用して、その効果を確認する必要がある。それらが定着したところで、改めてソフトウェアの開発プロセスの見直しをおこない、本格的な自動化へと進ことができよう。ソフトウェア自動生産こそソフトウェア工学の最大の目標であり、それが実現して初めて近代産業として確立されたといえるであろう。AI技術との接点は、そのような展望を持って議論されなければならない。

(慶応義塾大学)

セッション C2

A I ・ 第5世代コンピュータ ・ ソフトウェア工学

横井 俊夫

1. A I とソフトウェアのかかわり

A I とプログラム、プログラミング、ソフトウェア問題とのかかわりは、非常に深く、多様である。まず、大きく3つのかかわりあいを見ることができる。

第1は、プログラミング、すなわちプログラムを作るという人間の典型的な1つの知能活動として、A I の対象テーマとのかかわりあいである。自動プログラミングについては、プログラムの自動合成を目指すという言葉尻をとらえ、その取組の小規模さと成果の未熟さをあざけられる場面もあったが、仕組みを分析し、明確なものにしようという態度には、真しなものが感じられた。自動プログラミングの“自動”とは、自動車の“自動”と同程度という認識が、行き渡っていなかったからの誤解であろう。いずれにしろ、このあたりの感覚は、A I に対する冷静さを保つためにも、それぞれが磨きをかけねばならぬものである。

第2は、プログラミングを普遍的な知的活動の枠組みとして、A I の土台と見なすかかわりあいである。人間が学習をし、知能を拡大していくことは、頭の中に次々と新しいプログラムを合成していく過程だとみなせる、という心理学からの指摘がある。知能の本質と思われる学習という機能も、とかく散文的な語り合いに終わっていたが、プログラムの合成という明確な枠組みが与えられると、その細部の仕組みにわたっての議論が可能となる。そして、そこでの成果が、プログラミングの問題そのものへの解決の糸口に還元される。

第3は、A I 向きプログラム言語、プログラミング環境、マシンとして、A I 研究・開発のツールとしてのかかわりあいである。A I 特有の記号処理を適切に効率良く表現する機能を取り込んだ言語が開発された。この取り込みは、既存言語の延長に行われたのではなく、ラムダ計算や述語計算という整地された土台の上で行われ、奥に明解な計算機構として組み立てられた。また、A I システムは、特徴として記号処理を多用するが、他の処理形態が不用というわけではない。むしろ、数値処理をはじめごくごく一般のデータ処理をも効率良くこなし、し

かも計算機構の上に記号処理と一様に統合することとなる。A I 向き言語は、真の汎用言語である。記号処理機能は、自己参照機能の裏付けとなり、インタプリタをはじめデバッキング・ツールの高機能化を容易にし、他には見られない良質なプログラミング環境が提供される。A I 向き言語の機能の高さは、核となる計算機構の高度化に支えられているが、それは処理系に相応の負担をかけることになる。この負担は、専用マシン等、ハードウェア技術を活用すれば容易に克服され、余分の利得を得ることも可能である。処理系の多少の重さにこだわり、手にする高いソフトウェアの生産性を軽視するようでは、ソフトウェア工学の基本的視点が揺らぐことになろう。

2. 補足的コメント

以上の3つのかかわりあいを彩るものとして、さらにいくつかのコメントをつけ加えておこう。

まずは、ソフトウェア工学とエキスパート・システムとのからみに関してである。ソフトウェア工学上の問題も有効な適用対象として選ばれているが、これらを成功させるためには、知識の体系化が可能となるよう分野を適切なものに特定化すること、そして、分野を一般化するならば、プログラム変換や定理証明等の形式的アプローチの堅固な土台をまず礎くことである。

次は、ソフトウェア工学と自然言語（日本語）処理とのからみに関してである。枠組みや手法をどうするかはともかくとしても、ソフトウェアにかかわる作業は、あるレベルでは、ドキュメント処理に関する作業と見なしうるという考え方がある。ソフトウェア開発・保守支援システムは、何種類かのドキュメント処理を支援するシステムとして取り組むのが有力なアプローチである。

最後は、ソフトウェア工学と第5世代コンピュータとのからみに関してである。第5世代コンピュータ・プロジェクトは、あらゆる技術を動員してのソフトウェア生産性の飛躍的向上の達成を目的とするものである。色々の単純化はあるが、このあたりが最も誤差が少ないのではなかろうか。

(新世代コンピュータ開発機構)

セッション C3

ソフトウェア技術という視点でAIをみる

玉井 哲雄

1. どんな人が参加するだろうか

このSEAセミナー21を、参加者との意見交換や討論を重視したミニ・ワークショップとするという企画は、誠に面白い。自分が講師という役割さえ割り当てられなければ、大いに賛同するところだが、自分の受持をどう運営すればよいかを考えると、やや途方にくれる。

まず、表題のようなテーマをあげたとして、参加者はどんな気持ちでこのセミナーに参加するだろうか。

(1) 「日頃、ソフトウェア開発に携わっているが、難しい問題がない。AIは、この問題に何かの解決法を与える武器になるかもしれない」と期待する人。

(2) 「AIがブームになっている。会社でもビジネスになるのではないかという観測から、色々な動きがある。どのくらい商売になりそうなものか、ちょっとのぞいてみよう」という人。

(3) 「AIと計算機科学は、どっちがどっちを含むのかはよく分からないが、とにかく密接な関係だ。ソフトウェアをやる人間にとっては、その意味でAIについて知見を持つことは必要だろうが、参加者は(講師も含め)AIをどう捉えているのだろうか」という人。

(4) 「AIブームは、所詮ニュー・メディア・ブームなどと同じように、上すべりで終わるのではないか。皆、どう思っているか、聞いてみよう」という人。

(5) 「AIソフトウェアは、小規模システムとしては気のきいたものができるかもしれないが、現実の大規模で複雑な問題を扱うには、従来からの方法で作ったソフトウェアでないと無理なのではないか」と考える人。

2. どんなセミナーに

このような色々な意見の人が参加することを、歓迎したい。もちろん、それらの人の抱くさまざまな疑問に、すべて納得のいく解答を与えることはできないだろう。なるべく私の現在の考え方を明確に示した上で、討論を進めるほかはない。

とくに次のような問題を、とりあげたい。

(1) AIと対象としてのソフトウェア開発の問題点
多くの既存の工学分野や産業分野が、AIや知識工学

に期待し、とり込もうとしている。皮肉な見方をすれば、このところ学問的にも産業的にも沈滞気味の分野ほど、その動きが顕著である。ソフトウェア工学がAIに関心を寄せるのも、ソフトウェア工学自身がやや行き詰り気味だという事情があることは否めない。しかし、AIの適用対象として、ソフトウェア開発や保守作業が、どのような特徴と問題点を持ち、AI適用に本当に向くのかを、まず明らかにする必要がある。

(2) ソフトウェアとしてのAI

AIを実現するのも、ソフトウェアである。AIソフトウェアは、ソフトウェア工学的見地からみて、どのような特徴があるだろうか。たとえば、エキスパート・システムの開発は、典型的なプロトタイプ手法によって行われている。また、AIからはルール指向型やオブジェクト指向型など、新しいプログラミング・パラダイムが道具としてソフトウェア工学に提供された、とみることができるともいえる。知識表現のさまざまな方法は、システム開発の要求分析・定義の場面で使えるかもしれない。このように、AIソフトウェアは、ソフトウェア工学に材料を与える宝庫である。

(3) ソフトウェア技術者としてAIにどう取り組むべきか

ソフトウェア技術者にとって、AIの何が当面役に立つか。将来的には何ができそうか。その成果を取り込むには、どんな準備をしたらよいか。この辺の見きわめが重要であろう。とくにこのような問題は、一意的な解がないであろうから、余計に突りある討論に期待したい。

3. その他

当セミナーで、名著や名プログラムを読む形式がいいかとも考えた。しかし、時間やテーマを考え、全面的にそういうスタイルにするのは今回は見合わせた。ただ、部分的に、たとえばAIの考え方についてHofstadterのエッセイを読むとか、AIプログラミングについて、Susmanの本の一部を紹介するとか、考えてみたい。

(三菱総合研究所)

秋のセミナー・ウィーク
セッション・グループ D

ソフトウェアの再利用

ソフトウェア工学の究極の目的が、ソフトウェア生産性の向上であることは、いまさら論を待たないでしょう。そしていま、この課題を解決する最後の決め手として、熱い期待を込められて語られているのが、AI技術の応用であり、ソフトウェアの再利用です。

1983年の9月にアメリカのロード・アイランド州で Workshop on Reusability in programming という会合が大企業 I T T の主催で開かれました。このワークショップでは、再利用の分類に関する研究から、技法・方法論、さらには実際のツールの発表と討論が行われ、ソフトウェアの再利用という問題が、一躍クローズ・アップされる引き金になりました。

こうした背景もあって、部品合成をはじめとして、ソフトウェア開発の効率、信頼性をあげる再利用ツールがいくつも研究されています。しかし一方で、ソフトウェア開発現場での再利用の実現の実態は、いくつかの調査によっても、まだまだの感があります。

SEAが、今回のセミナーでは、再利用の実践的な側面に焦点を絞り、講師自らが現場でおこなっている再利用のツール作りやそれらの応用を中心に討論を進めたいと考えています。討議の内容が、I T T のワークショップを越えるものになることが期待されます。

セッション D1

「部品による自動プログラミングは可能か」をテーマにする大木幹雄さん（日本電子計算）は、目下、ディラックの向こうを張って「部品合成場の理論」を研究中とか。

本誌創刊号のプログラマ・インタビューからもわかるように、部品化や再利用の実践に早くから手をつけて来られました。また最近、Smalltalkを活用した部品合成にも取り組んでおられ、テーマとあわせると研究開発的な要素の多いお話になりそうですが、JISA/STC 再利用部会長としてのさまざまな調査分析の経験も踏まえて、再利用に関する話題を豊富にカバーした討論になることでしょう。

セッション D2

再利用の効果という点から考えれば、知識の再利用と設計情報の再利用に多くのことが望めそうですが、水谷時雄さん（日本システムサイエンス）を講師として迎えた「ソフトウェア再利用を中心とした設計技法」のセッションは、そうしたテーマをとりあげます。

講師の水谷さんは、これまで、データ中心設計によるプログラミング・ツール「JASPOL」の開発、さらに、ソフトウェア部品合成プログラミング・ツール「JASMAC」の開発にたずさわっておられ、現場での再利用の実践に強い情熱を持っておられます。ともすると理論だけに終わりがちなテーマに対して、実際のなアプローチが期待されます。

セッション D3

村井進さん（日本コンピューター・システム）は、SEAの分科会の中で一番活発に活動している、再利用分科会（SIGREUSE）の代表世話人です。その活動を通して、また自社においても、実際に使用できる再利用のツールの研究と、それをより効果的に使用できる環境の研究および実現構築を行っておられます。

「プログラミング言語は、もはや技術者の範囲を飛び出して、万民のものとなりつつあるという感じがします。しかし、本当に上手にプログラムを書ける人はあまりいないようです。やっぱり、他人が作った良いものは、どんどん再利用するのが得策だと思います」と、主張する村井さんは、「実用的な再利用ツールおよびその運用体制」と題されたこのセッションで、再利用ツールを本当に効果的に使用するための、運用体制および開発環境の具体的なイメージを討論したいと考えておられるようです。

セッション D1

部品による自動プログラミングは可能か

大木 幹雄

1. 夢は現実に

具体化したい機能を入力すると、それに必要な部品が自動的に選択・組み合わせられ、プログラムが合成されるツールは果たして夢であろうか。最近のAIマシンや、エキスパート・システム構築用ツール開発の進展をみると、このような夢は一步一步、実現に向かいつつあるように感じる。

例えば、知識処理システムで扱うルールは、知識の集約の意味で部品と対応付けすることができる。その結果、ルールの推論機構を部品間の関連をたどる部分に流用し、ある程度自動的にプログラムの合成を行うようなツールも、そう困難なくできるようになっている。

このようなAI技術を、いかにソフトウェア部品の合成に応用し、自動化を図って行くかが、セミナーを通じて共に考え、討論したいメイン・テーマである。

2. 部品合成と推論エンジン

AIというと、推論エンジン・モデルがまず存在し、それが何に應用できるかを考えるようなイメージがあるが、実際にはAIの推論エンジンといっても、診断エキスパート・システムや言語理解システム用のほかに、適用対象によって数多くのものが考えられる。

部品の合成に必要な仕組みを、エンド・ユーザの使い易さから追求すると、逆に推論エンジンに新たな機構を取り込まなければならない部分が、かなり出てくるのではないかと考える。

特に最近、ソフトウェア開発に関するプロセス情報の蓄積と検索がソフトウェア再利用において、重要であるとの認識が高まってきている状況を考えると、開発プロセスに関する知識を自動的に蓄積し、それを部品合成のプロセスで再利用するような独自の機構が強く要求されているように感じる。

このような点から、セミナーでは部品合成とAIの推論エンジンとの関連を主にお話する予定である。このとき、部品合成に、なぜAIの技術を導入しなければならないのかの必然性を、過去の経験や現実的な問題点と照らしあわせて紹介してみたい。また、このような仕組み

が、どのような経験的な事実に沿って必要とされるに至ったかについての背景も紹介してみたい。

ただ、AIの推論エンジンの細部については専門家でないため、経験に立脚して効果があると思われる部分しかご紹介できないが、部品合成から見たAIの有効部分はかなり理解しているつもりである。具体的な事例は事欠かないので、管理者にも、ツール屋さんにとっても、参考になるのではないかと考えている。

3. 再利用ツールは普及が大切

現在私は、会社においてAI技術を応用した各種のツール開発の責任者の立場にいる。どちらかというツール屋稼業が性にあっており、部品再利用支援ツールについては、マクロ部分の展開をベースにしたCOBOLプリ・プロセッサ-PRECObOLをはじめとして、いくつかのツール開発を今まで担当してきている。

最近、Smalltalk-80やInterlis p-Dと悪縁苦闘しており、関数やクラスを部品としていかに知的に検索し、結合するかの方法に頭を悩ましている。個人的な興味についてさらに言わせていただければ、プログラマに仕様書を渡し、コミュニケーションを取りながらプログラムを開発していく過程と、仕様書を理解しプログラムを合成するツールの機能とを、認知モデルによってどう対応付けるかに興味をもっている。

とはいうものの、再利用などという泥臭い問題は、どうのこうのいうよりも効果を見せることが第一で、実際にはこのようなAI技術よりも、むしろ普及に対する施策のほうが重要ではないかと考えている。普及に関しての苦労は数多く経験しており、開発方法論、設計手法と絡め、時間があればこれらの点にも触れてみたい。

4. 参加される方への希望

以上のことを考慮にいれて、セミナーに多数参加して頂けると幸いであるが、できれば部品再利用ツールの使用経験をお持ちの方で、部品合成の方法に関して積極的な意見をお持ちの方が、多数参加して頂けることを期待している。

(日本電子計算)

セッション D2

ソフトウェア再利用による ソフトウェア開発の自動化へのアプローチ

水谷 時雄

1. ソフトウェア自動設計へのアプローチ方法

ソフトウェアの自動設計に向けて、どのようなアプローチをすべきかは、以下のような方法が考えられる。

(1) ソフトウェア技術の学習・熟練

ソフトウェア技術の知識と適用業務の知識とが要求されるシステム・エンジニアより、ソフトウェア技術の学習をより少なくさせたい

(2) 技術的情報の隠蔽

電子部品のように、その内の論理は隠し、その外の用途機能のみ知らしめることが技術移転であろう。

(3) ソフトウェア部品化

サブルーチン/モジュールと呼ばれる処理手順の部品化と同様に、データあるいはデータ構造情報の部品化が必要となる。さらに、物理的データ構造から離れ、処理手続きを内包した論理的・抽象的データも部品化となるであろう。

(4) 情報資源管理

ソフトウェア部品化がすすむと、一層、情報資源管理が必要となろう。

(5) ブレークダウン設計からアセンブル設計への移行

ボトムアップによる部品が蓄積され、より上位の論理的・抽象的部品化がすすむにしたがい、ソフトウェア設計のブレーク・ダウンの深度が浅くなり、ソフトウェア設計の大部分が、部品の合成（アセンブル）となるであろう。

建築界がそうであるように、製造工法により、設計法が定まるのである。ユニット工法の世界では、設計図とは要求定義とほとんど変わらない。

2. データ構造の部品化もできる部品化法とは

従来のモジュール技法では処理手順の部品化しかできない。部品化には以下の技術的要件をみたすべきであろう。

(1) インタフェイスの不整合チェックを可能な限り川上（コンパイル時点）で行う。

(2) 部品の汎用性をより高めるため、部品にインテリジェンスを持たせる。部品が連絡データの形式を規定すべきでない。

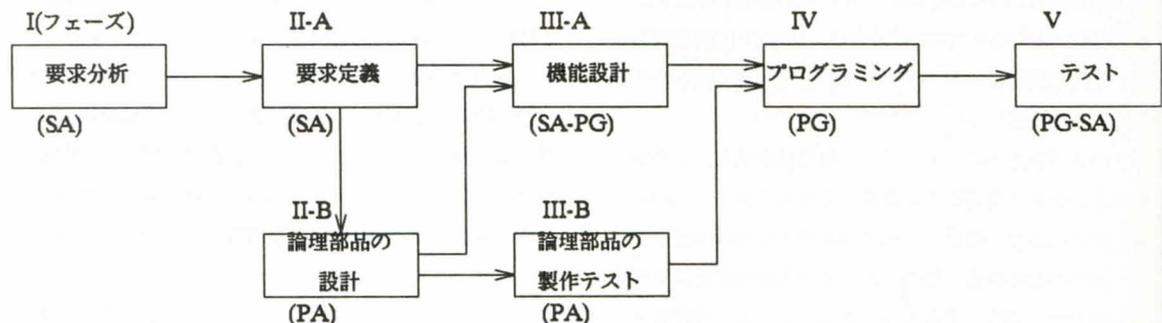
(3) 単なる処理部門だけでなく、データ構造情報も部品化でき、さらに、処理をともなった下層のデータまでも部品化できることが望まれる。

これらの要件をみたす部品化法の1つとして、ソースメコンポーネンツ合成法がある。

3. 抽象化設計によるプロトタイピング開発

プロトタイピングは単に工期を短縮することにとどまらない。未分析の情報を残しつつ、設計からプログラム開発をし、実行でき、並行して詳細分析の結果で設計追加・改良を加える。すなわち、より抽象的なレベルで設計をとどめてもお実行可能なプログラムとなる（下図を参照）。

(日本システム・サイエンス)



セッション D3

実用的な再利用ツールおよびその運用体制

村井 進

1. はじめに

本セッションでは、JISA再利用部会およびSEA/SIGREUSEの調査研究を通じて整理されたソフトウェア再利用ツールの現状と、それらを効果的に使用するための運用体制および開発環境について、具体的な例をあげながら検討をおこなう。

2. 再利用の要求とその問題点

開発経験をノウハウとしてのみ使用するのではなく、システム自体を再利用して、新規開発工数を削減したいという要求は、類似システムの開発を何度も経験している技術者にとっては、共通の要求であろう。しかしながら、既存のソフトウェアの再利用はそんな簡単ではない。大部分が共通なシステムでも、ほんの些細な点の相違によって、再利用の際に、大きな変更工数を必要とすることもある。

さらに、ソフトウェア再利用には、どの範囲で再利用を行うかという問題がある。個人レベルでの再利用には、大きな効果は望めない。逆に、広い範囲で再利用すると、そこには他人の作成したソフトウェアを理解するという、非常に大きな問題点が発生してくる。

これらの要求および問題点克服が、「ソフトウェア再利用技術の開発」の動機となるものである。

3. 再利用ツールとその環境

結論から先にいえば、再利用性を考慮しないで作成したソフトウェアの再利用は非常に困難であり、また、再利用を支援する仕掛けのない環境でのソフトウェアの再利用も非常に困難である。

従って、ソフトウェア再利用を行う上で重要なのは、再利用性の高いソフトウェアを作成するためのツールおよび環境と、実際に再利用を行うためのツールおよび環境である。ここでいう環境とは、開発体制とツール運用体制を包含したものである。

本セッションでは、話を進めるガイドラインとして、ソフトウェア再利用技法に関して以下の分類を行う。

4. ソフトウェア再利用技術の分類

ソフトウェア再利用技術には、大まかに言って、アプ

リケーション分野での開発対象の整理・定式化と、ソフトウェア記述言語の高度化の2つの視点が存在する。

(1) AP分野での開発対象の整理・定式化

アプリケーション分野での開発対象の整理・定式化とは、端的にいえばプログラムのパターン化とモジュール化である。いかなるプログラムも、単なる処理の流れとしてでなく、ある一定の機能や論理の表現としてとらえることができる。

パターン化とは、これらの機能や論理をいかにして汎用的な処理パターンに帰着させるかという問題であり、モジュール化とは、いかにして部分的に完成された処理を組み込み可能な形で提供するかという問題である。

パターン化のツール(標準化手法)としては、PARADIM, EAGLE, STEPSなどがある。モジュール化のツール(サブルーチン群、関数群)としては、画像処理サブルーチン群やC言語のライブラリ群等がある。

パターン化は事務計算分野に適しており、モジュール化はシステム制御分野や基本ソフト分野に適している。

(2) ソフトウェア記述言語の高度化

ソフトウェア再利用という観点からの高度な言語とは、ソフトウェアを部品として記述し、それを合成する機能をもった言語をさしている。これらの言語には、大きく分けて2つの傾向がある。

1つは、部品記述および部品組み合わせ記述を、言語仕様として完全に取り込んだ言語であり、Ada, Adapt,あるいはオブジェクト・オリエンテッド言語などがある。

もう1つは、既存の高級言語(たとえばCOBOL)に部品記述機能および部品合成機能を付加したものであり、SEA/Iの部品記述および合成機能、JASMAC, PRECOBOLなどがあげられる。これらはいずれも言語であると同時に、シンセサイザあるいはジェネレータと呼ばれるツールでもある。

(日本コンピューター・システム)

秋のセミナー・ウィーク
セッション・グループ E

ソフトウェア設計技法

1968年のNATOの会議を始点とすれば、ソフトウェア工学という分野が誕生してから今年で18年になります。ひとつの学問が成熟するには、18年という年数は、あまりにも短いでしょうが、しかし、これまでに多くの成果が達成されています。

このセッション・グループでとりあげるのは、現在、機械化がもっとも遅れており、ソフトウェア開発の現場で一番問題点が多いと指摘されている「ソフトウェア設計技法」です。過去に多くの方法論/技法/支援するツールが、提案されていますが、しかし、一方で、それらが現場のソフトウェア開発にはちっとも役立っていないという指摘もあります。原因はいろいろ考えられるでしょうが、ひとつのヒントとして、今年の3月に行われた、春のセミナー・ウィークでの、W. Riddle氏の講演「ソフトウェアにおける技術移転」が参考になるでしょう（本誌第4号）。

ともあれ、わたしたちのまわりには、ソフトウェアの設計という仕事を改善するための方法論や技法そしてそれらの支援ツールがかなりの種類存在することは、間違いのない事実です。それらをどうやってソフトウェア開発現場に導入して、成果をあげていくかが、大きな問題でしょう。

セッション E1

これまで26年間、基本ソフトウェアの開発・保守、日本語処理、自然言語処理、知識工学等の研究開発などの仕事に従事してこられ、現在では、ソフトウェア開発支援の技法/ツールの開発提供を担当しておられる、伊野誠さん（日本ユニバック）には、「ソフトウェア設計技法JSDと支援ツール」というテーマで、現場で必要とする設計技法とそれを支援するツールについて、JSD（ジャクソン・システム開発法）をひとつのケースとして、お話いただきます。

ただ単に、JSDという設計方法論の紹介でなく、この技法を支援するツールの実際を示しながら、一般的な技法とツールの関係、適用上の問題点などが討論のテ

マになるでしょう。

セッション E2

毎号本誌にユニークな関西支部便りを書いて下さっている白井義美さん（日本電子計算）には、「データ構造に基づくソフトウェア設計の一方法」というテーマをとりあげ、要求定義から詳細設計にいたるまでの設計の各工程ばかりでなく、保守の工程にまで使用できる設計書は、どう作成したらいいのか、そのために必要な記述事項はなにかを、現場での経験をもとに論じていただく。

特に講師自らがすすめておられる、図式表現を取り入れたデータ構造を中心としたソフトウェア設計（いわゆるアコーディオン・フロー）の実際が示されるでしょう。そしてその後、参加者とその利点や短所を討議することによって、現場にとって、よりよいシステム設計の方法が具体的に探りだされるはずで

セッション E3

ソフトウェアの設計にも、新しい波は押し寄せてきています。いや、むしろ新しい動きを、積極的に取り入れていこうという態度が必要とされています。オブジェクト指向パラダイム、関数型や論理型のプログラミング等の、新しい方法論や技法のAI技術の採用なども、その1例といえましょう。落水浩一郎先生（静岡大学）には、こうした新しい世の中の動向を、「設計方法論の最前線」というテーマで、展望していただくことになっています。

そして、これら新しい技術動向の見極めをとうして、それらを、どう現場に適用していくかを考えることが、このセッションでの討議のスタートになります。つまり、単なる新しい技法の紹介に留まらない、積極的な討議が3s期待れます。

セッション E1

ソフトウェア設計技法 JSD と支援ツール

伊野 誠

1. はじめに

ソフトウェア設計技法なるものは、世の中に存在するのか、しないのか、というところから問題提起をしたい、と思います。世の中には設計技法がいろいろあるという立場でものをいうとすれば、昭和59年4月に情報処理学会主催で行われたシンポジウム「プログラム設計技法の実用化と発展」以来、複合設計、ワーニエ・メソッド、ジャクソン・システム開発法など多くの技法が紹介されています。したがって、世の中には有名な技法がたくさんあるというわけです。

一方、何もないという立場では現場で役に立ちそうなものはない、その証拠として上記のような技法はほとんど現場では使われていない、という主張になるでしょう。

2. 開発現場ではどうなのか

それでは、現在、ソフトウェアやシステムを開発している現場において実態はどうなのか、という議論に進みましょう。

今年3月に情報処理振興事業協会技術センターから「ソフトウェア開発に関する実態調査」なる報告書がだされています。この調査は、コンピュータのメーカーやユーザおよびソフトウェア・ハウスなど多様な業種の、多様なソフトウェアの種類、例えば事務系、技術系、制御系、基本ソフトウェアなどを開発している現場に対し、その経験・知識の豊富な専門家にインタビューを行い、その結果をまとめたものです。システム設計技法利用の現状は次の通りです。

- (1) 要求定義フェーズと比べると、設計フェーズでは技法の使用例がかなり見られる。
- (2) しかしながら、ひとつの技法を全面的に採用している例はほとんどない。大部分は、設計手順などを定めた標準工程の中に各技法の精神や方法の一部をとり入れているというものである。
- (3) データベース設計については、特定の技法の導入例は非常に少ない。

また、いろいろな技法についての問題点として次のことが上げられています。

- ・ 上流と下流の不整合
- ・ 乏しい表現力
- ・ 設計フェーズの一部のみ支援
- ・ 設計技法ごとのモデル作成
- ・ モデルおよび設計結果の検証可能性が低い

以上の実態は、我々のまわりでの開発現場を見て素直に、そんな感じですねといえるものだと思います。さて、ではこのセミナーにおいて私自身、何をお話し、何を議論したいと思っているかを述べることにします。

3. JSDを1例にとって

まず、ソフトウェア設計技法の具体例としてジャクソン・システム開発法 (JSD-Jackson System Development) についてご紹介します。このJSDは、比較的新しい技法であり、リアルタイム処理や制御系アプリケーションなど、動きのあるシステムの設計に向いているということで、日本ユニバックから英国ジャクソン社のインストラクター・コースに2人出しています。そのようなこともあって、JSDとはどんな技法であるのかその概要をご紹介したいと思います。

さて、技法の具体的な例をご紹介した後の議論として期待していることは次の通りです。

- (1) システム設計の技法としてJSDは使えそうか、あるいは使うのはむずかしいか
- (2) 使えるとして、このような技法は役にたつのか、たたないのか
- (3) 支援ツールは、具体的な技法があってはじめて役にたつツールが考えられると思うが、技法とツールの関係はどうなのか
- (4) JSDの支援ツールとしてどんなものがあり、どんなものが考えられるか
- (5) 一般的にいて、システム設計フェーズでの技法とツールについて現場として何が必要なのか

以上いろいろと言いましたが、議論したいことが山ほどあり、ぜひ皆さんのご参加をお待ちしています。

(日本ユニバック)

セッション E2

データ構造に基づくソフトウェア設計の一方法

白井 義美

1. はじめに

一般のアプリケーション・システムの開発では、エンド・ユーザの要求をいかに正確に効率よくシステム設計に反映するかが大きな問題である。これに対して、要求定義、基本設計、詳細設計の各工程においていかなる方法を採用すれば効果的であろうか。また、どのような設計書の様式を利用すれば的確にシステムの内容を表現できるのか。さらに、保守に役立つ設計書とはどのようなものでなくてはならないのか。

ここでは独断と偏見に満ちたシステム設計技法の一例を取り上げ、システム設計を進める上で、考慮しなければならない各種の事項のうち、次のような点について、実務経験に基づいた意見交換を行いたい。

2. 要求定義書とシステム設計書

要求仕様書とシステム設計書は、それぞれ異なった目的で作成されるが、その両者の間には溝があるといわれており、これを解決する一方法として、システム設計に携わるすべての人々が、共通に理解できるような設計書があれば非常に便利である。また、要求定義から基本設計、詳細設計に至るすべてのシステム開発の工程に、同じ様式の設計書が一貫して利用できれば、なお便利でると考える。

この様に、ユーザ、設計者、プログラマの間に誤解を生じさせないようにするためには、形式的な記述が必要であり、わかりやすい設計書であるためには、自然言語的な要素が不可欠であろう。これらの要因を満足させるにはどのような方法が考えられるであろうか。

3. データ中心型記述と制御中心記述

システム設計にはデータの流を中心に記述するものと、制御の流れを中心に記述する方法がある。システムの中には、データの流に基づいて書かれたものが理解しやすい場合もあるが、制御の流れでシステムの内容を的確に表現できる場合もある。もし、その双方を同時に表記できれば、もっと理解しやすくなるであろうか。

4. データ構造とプログラム構造

事務計算系のシステムでは、データ構造をベースとし

た設計法が便利であるといわれており、ここで取り上げた設計方法も、データ構造に基づいてプログラム構造を決定しようとしている。

この場合、データ構造やプログラム構造をどのように表記すれば理解しやすいか、データ構造とプログラム構造の関係をどのように表現すればいいのか、また、どのようにしてデータ構造からプログラム構造を導けばいいのか等について論じたい。

5. システム設計における図式表現

理想的なシステム設計書とは、データ構造、データの流、データの内容、プログラム構造、制御の流れ、処理の内容等の要件がすべて同一帳票上に記述されており、なおかつ理解しやすいものであろう。そのためには、図式による表示を効果的にシステム設計書に取り込むことによって、視覚的な面から設計書の内容を理解できるようにすべきである。

それでは、いかなる様式のシステム設計書に、いかなる形の図式を記述すれば、これらの要件を無理なく記述でき、しかも理解しやすくなることが可能であろうか。

6. モジュール化と最適化

システムのモジュール化は、プログラムの複雑さを軽減し、多くのプログラムの同時作業を可能にする点で、非常に効果的である。モジュール化の方法は数多くあると思われるが、どのような方法がどんな場合に有効なのかを考えてみたい。また、システム設計中に発生する各種の最適化手法について、その必要性や功罪について論議したい。

7. 保守のための設計書

開発段階で作成されたシステム設計書が、保守に有効に利用されていないことが、しばしば見受けられる。それらの原因の多くは、システム設計書自体が、プログラムの変更に応じて、適時アップデートされていないことが挙げられる。なぜ設計書が見捨てられるのか。どうすれば保守に役立ち、またアップデートするに足る設計書になるかを考えたい。

(日本電子計算)

セッション E3

設計方法論の最前線

落水 浩一郎

1. 80年代の新しい動向

80年代の新しい動向を、冷静に把握したいと思いません。また、設計やプログラミングに関する本音について、意見の交換をさせていただきたいと思いません。

そこで、討論の叩き台として、以下の2つの話題をまず私から提示させていただきます。

(1) 最新の研究・技術開発の動向の把握

大学院の講義を利用して、ソフトウェア設計の紹介記事を読みました (IEEE Tran. on SE, Vol. SE-12, No.2, Feb. 1986)。そのうち、以下の論文についてご紹介します。

(a) Conventional Approach (定義、設計、実現)

ここでの要件は、以下のものである。

- ・マンマシン・インタフェースのシュミレーション
- ・問題解決とソフトウェア構造の対応の良さ
- ・並行処理システムの解析

(b) Operatinal Approach (実行可能仕様+自動変換)

ここでの要件は、以下のものである。

- ・関数仕様の効果とプロトタイピング環境
- ・データフロー図式
- ・ベトリネット
- ・Entity-Actionモデル

これら論文のポイントをご紹介する形式をとらせて頂きます。70年代の各種方法論と異なり、実システムの開発に適用した経験はありませんが、その当時感じたいくつかの壁や改良点に関して答えてくれる実感のある論文を選びました。

(2) 種々のプログラミング・パラダイムの特徴と適用分野

操作的アプローチには魅力を感じています。新しい研究課題として、種々のプログラミング・パラダイムにも

魅力を感じています。プログラミングに対する自己の能力が拡大される実感がして。

それ以上にエキスパート・システムは私を引き付けます。ソフトウェア工学のどこに適用すべきか、できるのかという疑問を抱きつつ。

2. 新しいプログラミング・パラダイム

一方、それらも含めて、どの方法論も設計やプログラミングの際に完全ではないことにはがゆさも覚えます。

たとえば、学生諸君にプログラミングを指導するとき、卒研で共にシステムを設計する時、複合設計やジャクソン法やパール・ネックレスや抽象データ型を十分に説明し、演習をしたうえでとりかかります。しかし、いつも途中からは自分の世界です。または、よりよいIDEAを出した学生諸君に世界をとられてしまいます。分析をした結果をまとめて、そしてつなぐ世界です。最後にそのとき採用している方法論の記法に表現を変換して終わります。

次の年に新しい学生諸君にシステムを説明するときにはいつも「ところでそれは良いとして、結局はどう考えたかというところから始まります。実際の設計過程と、種々の方法論には微妙なずれがあります。このような問題について種々のプログラミング・パラダイムの特徴と適用分野を、比較検討することを手掛かりにして、皆様と討論させていただきたいと存じます。

以下の項目を討論の出発点として準備します。

- ・手続き型言語に基づくプログラミングの問題点
- ・オブジェクト指向パラダイムの適用分野
- ・手続き型プログラミングと関数型プログラミング
- ・論理型プログラミングのとらえかた
- ・手続き型プログラミングとルールベース・エキスパート・システム

(静岡大学)

秋のセミナー・ウィーク
セッション・グループ F

プロジェクト・マネジメント

いま、プロジェクト・マネジメントは、新しい局面をむかえようとしています。

技術的な面からすると、プログラム開発の自動化（機械化）に向けての展開、パーソナル・ワークステーションの大量導入、開発環境の分散化、AIなどの新しい技術の発展とその応用等々の技術的変革が、もうすでに、いろいろなところで実現されてきています。

社会動向および人的な局面からすると、第3次オンラインをはじめとして、ソフトウェアの膨大ともいえる需要の増加、さらに生産されるソフトウェア品質向上の要求、そして、それらを支えるべきエンジニアの量の不足と質の低下といった問題が山積しています。

これらにどう対応していくかが、これからのソフトウェア・マネージャに求められていると思われま

セッション F1

SIA時代からJISAに至る6年間、業界における技術標準化の中心的役割を担って来られた松原友夫さん（日立ソフトウェアエンジニアリング）には、「プロジェクト管理のエコロジー」というテーマで、これまでのソフトウェア・プロジェクト管理の問題点を、できるだけ洗い出し、なにが原因でそうした問題がはっせいするのか、どうそれを解決したらよいかを考察していただきます。

そして、そうした考察をふまえて、講師が提言するホロンのプロジェクト管理の意味するところを、参加者とともに討議することによって、成功するプロジェクト管理の秘けつを探ります。

セッション F2

9月に盛岡で開かれる若手のワークショップの実行委員長でもある堀江進さん（日本電気ソフトウェア）は、これまでに自分が担当された「すべて」のプロジェクトの中でさまざまな管理データを収集し、その分析結果を次の作業、次の工程、そして、次のプロジェクトで利用するということを実践して来られました。もちろん、数字だけでは何も解決しません。堀江さんの現在の関心は、

人間的側面を考慮したテクニカル・マネジメントというあたりにあります。

「人間的側面からみたプロジェクト管理」というテーマのこのセッションでは、そうした豊富な経験に基づいて、マネジメントに必要な応用技術を、人間的側面を中心にとらえるという立場からの議論が展開されます。明日から役立つプロジェクト・マネジメントが探られることが期待できます。

セッション F3

大学では物理学科のあまり勉強をしない学生であったため、なんとなく情報処理業界に入ってきた深瀬弘泰さん（アスキー）は、大型技術計算のアプリケーション分野の仕事からはじめて、メインフレームの事務処理（バンキング）、OSまわりのSEを経て、小さなコンピュータの面白さにひかれて、ミニ/マイクロの世界へ移籍し、現在は、UNIX関連の技術部隊と、社内OAシステムのアプリケーション部隊のマネージャという2足のわらじをはいておられます。

「環境におけるプロジェクト管理」をテーマとするこのセッションでは、特に技術革新がもたらすマネジメント・スタイルの変化についてお話いただき、そして、先端の開発環境の構築・運営に携わってきている講師の経験をとらえて、新しいプロジェクト管理のありかたについて考えます。

セッション F1

プロジェクト管理のエコロジー

松原 友夫

1. マネージメントは成功の前提か

(1) 問題と管理強化の危険な悪循環

- ・チャレンジャー事故原因報告書
- ・非行と校則強化の悪循環
- ・プロジェクトのトラブルと管理手段の増加

(2) 自由と統制のバランス

- ・両極端は放任とオーウェリアン社会
 - 言論の自由が保証されていない国は意外に多い
 - 自由の脆弱性
 - 自由を守るには強い意志と節度のある統制が必要
- ・プロジェクト管理における統制過剰
 - 統制過剰を生む背景
 - 多いトラブルとその膨大な被害
 - 客観指標の不足
 - プロジェクト規模の増大
 - 自発性の不足, 統制受容型人間の増加
- ・過剰統制の盲点
 - 自発性の衰弱, 一方通行コミュニケーション, 視野狭小による硬直的な判断
 - 「ほんね」と「たてまえ」の乗離
 - 統制意図に迎合したレスポンス

(3) 自由と統制は表裏一体

- ・チームワークの極意: 川喜田式探検チーム
 - 定まった役割分担の下で, 普段から上下にとらわれない自由な議論ができていたチームほど, 緊急事態への対処が迅速で効果的である。
- ・あなたの組織の自由度自己診断

2. 思考のパラダイムの頑強さ

(1) 手段や環境との密接な異存関係

- ・日本人は日本語で考える
- ・プロは分野ごとにパラダイムを持つ
- ・プロのソフトウェア技術者は何で考えるだろうか

(2) 初体験シンドローム

- ・はじめに作られた思考パラダイムは強く, その後の思考を支配する
 - 雑がふかしたときの刷込み現象

- COBOLシンドローム, 巨大OSシンドローム

(3) 大きな視点の慣性

- ・「相手の立場にたって考える」ことは次元の異なる視点への任意の変換が必要, 言うべくして行い難し
- ・自己への拘りや立場の非可逆性が視点変換の障害
- ・見えにくいソフトウェアでは一層難しい

(4) 共通のパラダイムを作るには

- ・意図的誘導型: スローガン, 目標管理
- ・自発性依存型: 自律ルールによるダイナミックな自主判断

3. 共通の場を作るための標準化

(1) 標準化は節度ある自由を確保する

- ・標準化の目的はより広い場での自由を確保すること
 - 力に任せた勝手気ままなルール作りは多くの人々の自由を奪うもの
- ・局所に閉じた標準化の優先度は低い
 - ソフトウェアでは広域の標準化は手つかず

(2) 用語定義, 評価法, 測定法の標準化は議論の客観化を促す

- ・測定法 (例えばプログラム規模) の測り方が組織毎に異なる産業はソフトウェアのみ, 生産性や品質の客観的比較はほとんど不可能, 独善を助長する
- ・あなたの組織では, 生産性や品質をどうやって測っていますか, 他社の数値と比べたことがありますか

4. ホロンのプロジェクト管理の提案

(1) 全体と個の調和

- ・ホロン: 全体子, ぜんたいとの位置づけを知り, その上で個として自由に振舞う要素物質はどこまでも独立した個の要素に分解できるという還元主義のアンチテーゼ

(2) まとめ

- 電子コミュニケーションや開発環境をうまく使えば, 全体との位置づけを知りつつ自律的で自由度の大きいプロジェクト遂行が可能である

(日立ソフトウェア・エンジニアリング)

セッション F2

人間的側面からみたプロジェクト管理

堀江 進

1. プロとアマの違い

ソフトウェア技術者は、実に多くの知識を学ばなければならぬ。プログラミング言語、ジョブ制御言語、TSSのコマンド、データベース、オンライン・システム、種々のツール、担当する専門分野の知識等。

新入社員は、これらの知識の一部を教えられ、「さあOJTだ!」のかけ声とともに実務につく。このような新人は、きっと(a)のようなコーディングをするだろう。

(a)CIRCLE=6.283185*R・・・確かに動くけど

(b)CIRCLE=2*3.1415926*R・・・もう一歩

(c)PARAMETER(PAI=3.1415926)

CIRCLE=2*PAI*R・・・よくわかる

新人はプログラミング言語を知っていても、それを使って優れたものを作る術を知らない。まして、どのようなものが優れているのかも知らない。そのため、プログラムのなかで、どうしてマジック・ナンバーを避けるべきかを知らない。言語仕様などの基礎技術に対して、このような技術は応用技術といえるだろう。このような応用技術の有無は、開発期間、性能、品質で雲泥の差を生む。これがプロとアマの違いではないだろうか。

ソフトウェア開発のほんの一部のコーディングでさえ、このような状態である。まして、プロジェクト管理でこのような技術の有無は、プロジェクトの正否にかかわってくる。プロジェクト管理の中心であるリーダーが、何を理解しておくべきかを、いくつか挙げてみよう。

2. プロジェクトのメンバーは?

新しいプロジェクトを考えてみよう。特別な場合でない限り、次のようなメンバーが集まる。適任と思える人、あまり適任とは考えられない人、そして、未熟な人。このようになるのは、現在進行中のプロジェクトから、メンバーを都合しなければならぬからである。その結果、プロジェクトから抜けられそうな人、抜けてもいい人、そして新入社員が集まる。つまり、多くのプロジェクトは、かなり悲劇的な状態からはじまる。

3. プロジェクト・リーダーは?

リーダーは一般に、次の3通りのいずれかで決まる。

- ・自力型・・・いわゆる餓鬼大将
- ・選出型・・・メンバーの互選による
- ・任命型・・・上位の機関、人からの指名

企業では自力、選出型でなく、任命型でリーダーがきまる。ということは、幼い頃から今まで、一度もリーダーの経験がないひとが、ある日突然、リーダーになるのである。このプロジェクトもかなり悲劇になる可能性を十分に秘めている。

4. モニタリング

経験の浅いリーダーがすべきことは、プログラムをいくつかに切り、それぞれの担当者を決めることではない。それぞれの工程、それぞれの作業が、終わったときはもう遅い、ということにならないように、途中で内容をチェックする(モニタリング)が必要である。

5. プログラムの見積り

プログラム規模の見積りに関してのモニタリングの例を示す。プログラム機能が決まった段階(機能設計)と、詳細な処理が決まった段階(詳細設計)で見積もってみよう。この結果と最終規模を比較すると、次のような傾向がでてくるだろう。

最終規模>機能設計後の規模>詳細設計後の規模

詳細設計後のほうが、機能設計後より見積が悪くなる。詳細設計後は処理が具体化されて、コード・イメージがわかっているため、より良い見積りになると一般に思われている。しかし、データを取ってみると、そうならないことが多い。

もしあなたの懐具合が寂しいときは、初めての飲み屋よりは、一度飲んだことがあるところに入るだろう。見積りでも、このこととおなじ現象なのだ。見えないものは恐ろしい、見えるものは安心だというわけだ。

「マジック・ナンバーは、どうして避けるべきか」ということと同じような、マネジメントに必要な応用技術を、人間的側面を中心に、定量データ、モニタリングについて考えてみようではありませんか。

(日本電気ソフトウェア)

セッション F3

環境におけるプロジェクト管理

—黄色い部屋はいかに改装されたか—

深瀬 弘恭

1. 由緒正しい密室犯罪

皆さんが本格推理小説のファンならば、ガストン・ルルーが1907年に発表した「黄色い部屋の謎」をご存じだろう。密室犯罪のものとしては、プロットの確かさからいって、たぶん最高水準である。ポオの「モルグ街の殺人」は煙突が出入口になっていたし、ドイルの「まだらの紐」は毒蛇がぬけられる小さな穴があった。

ソフトウェアを作るとは、要求定義を明確にし、機能仕様の設計をし、その方法論を選択し、ユーザ・インタフェースを作り、最終的には、ブラック・ボックスを作ることにはかならないように思える。ユーザからすれば、オペレーティング・システムが何であれ、そのアルゴリズムが何であれ、自分の望む結果が得られればよいわけである。

それでは作る側の問題は何か、それは密室犯罪の犯罪者の立場に似ている。いかにプロットを組み立てるか、そしてフェアな方法（ソフトウェア・コンセプトのあるべき姿、これが大切である）でそれを実現するか、ということであろう。しかもその実行可能性（応答時間、ユーザ・インタフェース）を押し測る、あるいは前もって予行演習をしてみることであろう。そこで問題となるのは、密室のあり様（環境）である。もちろん、ボワロの「灰色の脳細胞」（いい意味でのハッカー）も必要には違いないが。

「黄色い部屋の謎」は現代推理小説からすれば、古色蒼然とした仕掛けではある（80年も前に書かれたのだから無理もない）が、ではソフトウェアの開発環境はどうであろう。開発環境としての「黄色い部屋」は現在、いかに改装されたのだろうか？

2. 誰も常にあらゆることに気がつくわけにはいかぬ

現在のソフトウェア・サイズは、幾何級数的に巨大化してきていることは、良し悪しは別として、周知の事実である。1例をあげれば、銀行のオンライン・システム

がよい例である。60年代の1次の1行当たりの開発費は60億円であったが、70年代の第2次では200億円となり、80年代の3次にいたっては700億円とも800億円ともいわれている。

「黄色い部屋の謎」は、「誰も常にあらゆることに気がつくわけにはゆかぬ」という事実の上に成り立っている。現代の複雑で巨大化した「黄色い部屋」ではなおのこと、このことが言える。これは犯罪者にとってはよいように思えるが、完全犯罪を目指す者にとっては落とし穴になりかねない。なぜなら、自分自身も気がつかないぬけ穴があるかもしれないのだから、そこをボアロのような探偵に衝かれて、ガラガラとプロットが崩れてしまうかもしれない。それではどうするか？

3. 部屋の改装

「黄色い部屋」を現代的犯罪の舞台として改装することを考えよう。その前に、我々にはダ・ビンチはすぐにはみつきりそうもないので、それぞれの分野のプロフェッショナルである「黄金の7人」を捜しておく。「ゲリラ線教程」の教えるところに従って、組織化された同時多発ゲリラを可能とする改装をほどこす。いいかえれば、プロトタイピング手法を多用できる分散型環境に移行する。ウォータ・フォール型の「部屋」のづくりでは、犯罪者の思い込みを左右されやすく、実行時の急な状況変化に対応不能であろう。

私企業では、この改装費に関して、費用対効果の比率は厳しく監視されるわけであるから（公的機関のプロジェクトも本来そうであるはずだが）、分散環境の構築については、そうでない場合と比べて、総合的に秀でていくというシナリオを用意しておかなければならない。これはソフトウェアの側面からみると、ツールの整備、データベースの活用であり、ハードウェア面からみると、急速な技術革新をいち早く取り込むことのできるネットワーク構成、マシン構成についてのシナリオである。

(アスキー)

秋のセミナー・ウィーク
セッション・グループ G

一般的トピックス

セッション・グループA-Gでとりあげたテーマは、ソフトウェア開発に関する諸問題をいわば技術的に解決しようというものです。しかし、それらが単に技術面だけではすまないことは、皆さんよくご存じの通りです。

このセッション・グループでは、ソフトウェア開発の現場を取り巻く一般的な事柄で、特に重要と思われるものをとりあげました。

セッション G1

日能公認会計士共同事務所の代表公認会計士でおられる吉村成弘さんには、SEAの会計監事もお願いしていますが、今回のセミナーでは、「システム監査とEDP部門」というテーマで講演と討議をしていただきます。

講演では、システム監査とは何かという定義（意外に判っていないことが多い）、そしてその必要性和世の中の事情が説明され、とくに、コンピュータ犯罪とシステム監査に対するシステムのアプローチについて考察がさます。また、システムの信頼性・安全性に対する意識の高揚と統一的および具体的な対策についても考察と討議が行われます。

これまで、システム監査のセミナーはたんなる講演形式だけのものが多かったようで、今回は参加者とともに、具体的な事例をもとに参加者全員で討論をしてみたいと考えています。

セッション G2

今回のセミナーの講師の中で、経歴面からみた一番の変わり種は、「ソフトウェアの法的保護」というテーマで講演と討議をしていただく、佐野稔先生（コンピュータ法務研究所・弁護士）でしょう。

その経歴を簡単にみても、昭和43年東京工業大学卒業（物理学科）。同年4月東京大学大学院（素粒子論専攻）に進み、途中で軌道を外れ、47年司法試験合格。修習を経て50年弁護士登録。その後、再び軌道を外れて、59年4月、東工大に戻ってコンピュータのH/W、S/W（特にOS）を学び、コンピュータ関連法務のみの専門弁護士をめざして研究中。

この簡単な経歴からみても、「ソフトウェアの法的保護」問題の講師としては、まさに最 でしょう。

セッション G3

ソフトウェア技術者のなかには、ひじょうにユニークな個性を持った人が、他の世界よりか多くみられるといわれています。そして、こうした人々を含めた情報処理技術者は、社会でより重要な役割を担っています。

「SE道（どう）について語り合おう」というテーマで、システム・コンサルタントとして活躍中の辻淳二さん（辻システム計画事務所）と共に、この情報処理技術者として、日々の仕事の中で、何を考え、何をすべか等が討議されます。

マネージャだけでなく、プログラムを作っているエンジニアの人たちにも、技術を習得する以上に有益な講演および討議がなされることと思います。

ソフトウェア技術者協会（SEA）の設立主旨の中に、「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く場>であることを目指しています」がありますが、このセッションはまさにその主旨を具体化したものです。参加者との討議の結果が期待できます。

セッション G1

システム監査とEDP部門

吉村 成弘

1. システム的アプローチの必要性

コンピュータ犯罪に関する紹介記事や単行本は、すでにたくさん見受けられる。日本の事例はあまり多くはないが、米国のものでは、かなり詳細なレポートも公表されている。しかし日米を問わず、そのすべてのレポートには、肝心なところが抜けているように思われる。それは、「システムのどこにどのような問題があったのか」という点である。

システム監査に関する出版物についても、同様な問題点を指摘できる。それらの多くは、米国の文献の翻訳や紹介にすぎないものであり、「システム設計上の対応」について言及しているものは、ほとんどないのが実情である。

つまり、コンピュータ犯罪やシステム監査に対するシステム的アプローチが決定的に不足しているのである。私は、このセミナーでは、典型的なコンピュータ犯罪の幾つかを取り上げて、

- ・システムのどこに問題があったの
- ・同じことが我々のシステムでも起こりうるのか
- ・防止策はあるのか

について検討を加えたいと思っている。

2. 問題はどこから来るのか

問題の原因はどこにあるのだろうか。ひと口に言えば、それは昔ながらの内部統制に関する考え方と、EDPシステムの実態との、ギャップからくるものである。それらを列挙すると、次のようになるであろう。

- (1)データがその発生場所から直ちにインプットされるケースが多くなり、事務処理がもはや内部統制の連鎖ではなくなった。
- (2)内部統制担当部門の不在
- (3)自動化の落とし穴
- (4)効率化の追求と本質的に矛盾する内部統制の原則
- (5)不正の問題に対して全く無関心なSE
- (6)末端から中心のファイル（会計帳簿等）に対するアク

セスが容易であるなどの全く新しい業務処理方式の一般化。

3. 適切な対策はあるのか

システム監査問題がクローズアップされるのは、常に大きなコンピュータ犯罪の発生を契機としている。しかし、システム監査は、コンピュータ犯罪に対しては極めて間接的な効果しかないものである。もともと現在の一般的な監査は、犯罪を直接的に摘発することを目的とするものではない。それはいわば統計的な信頼性を基礎とする間接的な統制手段にすぎないものである。

コンピュータ犯罪への対応を含めた、システムの信頼性・安全性の問題は、いずれにしてもシステム部門の自覚と主体的な対策にまつしかないものであろう。

そのポイントを指摘すると、次のようなことになるであろう。

- (1)EDPシステム（特にオンライン・システム）における、新しい内部統制の考え方及び原則の確立
- (2)コンピュータ犯罪に対するシステム的アプローチ
- (3)コンピュータ犯罪を含めて、システムの信頼性・安全性に対する意識の高揚と統合的・具体的な対策の実施

以上の点に関しての講演および討論をふまえて、システム監査について、システム部門は、いかにして対応すべきかを参加者とともに探ってみたいと思います。

(公認会計士)

セッション G2

ソフトウェアの法的保護

佐野 稔

1. 「アイデア」と「表現」

本年1月1日から施行された改正著作権法により、プログラムがその適用対象の中に取り込まれることになった。これによってプログラムが、とにかくも法的に何らかの保護を受けることになったわけである。

ここで問題となるのが、「保護の内容」である。つまり、あるプログラムに着目した場合に、プログラムの何が保護されるのか、また、どのように保護されるのかという問題である。これを別の側面からいえば、他人の書いたプログラムに対してどのようなことをすれば、著作権侵害としてその責任を問われることになるか、ということもできよう。この問題をつきつめて考えていくと、プログラムにおける「表現」と「アイデア」との境界はどこに引けるのかという問題に帰着する。

周知のとおり著作権法は、「表現」を保護する法律であって、その表現を通して伝達しようとしている表現内容（「アイデア」）を保護するものではない。この「アイデアを保護せず表現のみを保護する」という法原則は、日本のみならず世界共通の著作権法上の大命題である。

コンピュータのプログラムを著作権法の枠内に取り込むということは、とりもなおさず上述した「アイデアを保護せず表現のみを保護する」という原則をプログラムにも適用するということの意味する。その当否についてはいろいろと異論のあるところであるが、改正法が施行された今の段階となつては、当面この原則を前提としてプログラムの保護範囲を考えざるをえない。

2. プログラムにおける「アイデア」

プログラムは論理の結晶である。論理的展開に功拙の差はあるにせよ、およそともに動くプログラムであるならば、そこに具体化されている各ステップの連鎖には、プログラムが考えた推論過程がコンピュータ言語を通じて表現されている。この場合、所与の入力データをもとにして所期の出力データを得るには、どのような処理をすればよいのか、ということが上で述べた推論過程あるいは論理の展開である。この意味での推論過程がコーディングをする者にとって、必要な程度に特定されている

ならば、あとは「どのような言語でどのようにコード化するか」という作業が残るだけとなる。この関係に着目すると、プログラムにおけるアイデアとは、「コーディングに必要な程度に入力データから出力データを得るまでの過程の論理展開を示す情報」であり、表現とは「かかるアイデアにもとずいて現実にコーディングされた成果物」ということができよう。

3. 米国の裁判例

上記の見解に対して裁判所はどのような態度をみせているのかは大いに関心のあるところである。日本には事例が見当たらないので、米国での裁判例をもとに考察してみよう。

(1) ウェラン対ジャスロー事件

この事件は、ようするに原告ウェランがEDL言語（フォートラン系）で書かれたプログラムを、被告がBASICに書き換えたという事件である。この事案に対してペンシルバニア連邦地方裁判所は「プログラムにおけるアイデアとは、当該プログラムの果たす機能、つまりこのプログラムにいかなる処理をさせるかという側面がアイデアある」という考え方を示した。

(2) ウィリアムズ対アード事件

この事件は、一般的な書物の中で原告が示したデータ処理手順の解説記事をもとにして、被告がこれをBASICでプログラム化したという事案である。この件に対してマサチューセッツ連邦地方裁判所は、前述ウェラン対ジャスロー事件の考え方を更に進め、元になる資料が一般的な書物であったとしても、その中に記載されている処理手順を使ってこれをプログラム化すれば、それは元の書物の表現を侵害したことになるという見解を示した。

以上のように米国では、考え方そのものと思われる性質のものであっても、いったん紙に書いた形を通せば最早他人が同じ考え方を紙に書くことは許されなくなるという方向でプログラムの保護を行おうとしている。しかしその当否は大変疑問であるばかりか、危険でもある。

(コンピュータ法務研究所)

セッション G3

SE道(どう)について語り合おう

辻 淳二

1. はじめに

本セッションのテーマ「SE道」についての、私の問題意識がどの程度であるのかを知っていただくために、拙著「情報戦略のこころ」(企画センター刊、昭和60年)の巻頭に書いた一文を転載させていただきます。

2. 「情報戦略のこころ」から

(前略)最近、金融証券業界で長く情報システム部門のキーマンを務めておられ、筆者が常日頃多くの刺激を受けている友人からていねいな手紙をいただいた。金融自由化の流れに沿っての事業の拡大と管理の統合化の要請に伴いシステム化テーマがかかってないほど山積し、判断・行動の拠所をどこにおいたらよいか迷う中で、「〇〇道」ということを考えさせる、との内容であった。それを読んで、筆者ははっと胸をつかれる所があった。

日本では剣道・柔道・茶道・書道・武士道など一芸の世界には「道」という言葉がついていて、各界の達人たちは鍛錬・修業を続けることで心技ともにその道の「奥義・極意」を極めたとされている。友人の言う「〇〇道」とは、この種の「道」が情報システム分野にも例えば「SE道」「プログラマ道」といった形で存在すること、そしてますます繁忙を極めるこの分野の中核の関係者にとって究極の拠所となるのは、「求道の心」ではないかということなのである。(中略)

さしずめ筆者は「情報プランナー道」を究めなければならぬのであろう。この友人から受けた触発により、1つの大きな目標を与えられたような気がしている。

友人は、これについて、手紙の中でいくつかのヒントを与えてくれている。その第一は、伝統的な諸芸の世界の「道」は、いずれも「形から入って心に通じる」という点で共通していること。その第二は、その心とは茶道では「お客をもてなす心」、柔道では「弱きを援ける心」とされ、それぞれに表現は違っているが行き着く所は同じらしいということ。どうやら、我々の世界の「道」においても、奥義はこれらに共通の所にありそうだ。

有力なヒントを与えられると、次の手がかりに行きつきやすくなる。友人の手紙のあとしばらくたって、筆者

は書店の店頭でサントリー(株)鳥井道夫副社長の近刊書「商いルネッサンス」に出会い、第二の触発を受けることになる。(中略)

書中で同氏は、仕事をする姿勢に関連して「道」を論じ、「道」の心得をしっかりと身につけていけば、おのずからその心得に沿って行動できる」として、「道」の会得の重要性を説いておられる。そして、「道」は茶道や書道と同じように仕事やビジネスにおいても必要で、その本質は仕事に対する「さとり」ではないかと考察されている。ここで、この「さとり」という言葉は、気くばりや思いやりを越え、センスやカンとも違って、ノミといえはツチ、タバコとくればマッチというように”物事に自然体で対応しながら相手のこころに叶っているあり方”という深い意味合いで使われている。

これまた、極めて印象的なヒントである。今後一段と忙しくもなり、重要な役割をも担うであろう情報システム分野の人たちにとって、「道」を究める姿勢はやはり大切なことのように思えてきた。(中略)

今回、本書の刊行の前に、このような触発を受けたことは誠に貴重なことだった。筆者は、いつ究められるかわからないほど遠いが、少し手がかりをつかみ始めたように感じる。我々、すなわち情報システム分野で働く人にとつての「道」の追求を、本書刊行後のテーマとして長く暖めて行きたいと考えている。

3. 呼び掛け

その後1年ぐらいの時がたっていますが、残念ながらまだ確かな前進はしていません。しかし、十人位の友人たちで月1回会合をもっている研究会で共同研究したり、いくつかの文献をあたったりという形で、少しずつ足場を固めつつあります。

そこで、このセミナーでは、私が最近の勉強成果についてレポートし、それを叩き台に参加者の皆さんと自由に討議を交わし、何か「確かな手がかり」を得たいと考えています。双方向で、率直かつ中身のある議論ができたらと期待しています。

(辻システム計画事務所)

A I 分科会 (SIGAI)

1. 第2回月例会

第2回月例会を、7月10日(木)に、第3回月例会を、8月7日(木)に、それぞれ開催しました。

第2回は、MRIの西山さんに、「ロジック・プログラミングについて」ということで、今話題になっているPrologについて、その特徴、長所、欠点を話していただきました。そして第3回は、電総研の新田克己さん、「手続法表現システムKRIP」ということで、エキスパート・システムの試作について話していただきました。

2. 次回以後の月例会予定

第4回月例会

日時：昭和61年9月11日19:00-21:00

場所：機械振興会館 B3階 9会議室

会費：1,000円程度(会場費として)

テーマ：「エキスパート・システム構築
ツールESHELL」

山本(東洋情報システム)

第5回月例会

日時：昭和61年10月9日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を中心にして」

坂下(ASTECC)

連絡および問い合わせは、協同システム開発(03-503-4981)の野辺までどうぞ。

管理分科会 (SIGMAN)

第4回の月例会を、東高円寺のSEA事務所でおこなった。今後の月例会の会場として、東高円寺のSEA事務所でおこないたいと思っています。5回以降の日程等は以下の通りです。参加をお待ちしています。

また、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。

第5回会合

日時：9月19日(金)19:00~21:00

場所：SEA東高円寺事務所

話題：プロジェクト管理上の問題について

報告者：土崎直樹

教育分科会 (SIGEDU)

7月30日に、第 回会合を開催しました。テーマは、今業界でも問題となっている「中堅技術者教育」ということでしたが、発表内容はSEAMAIL誌上に掲載します。

次回の第5回会合は、

日時：9月12日(金)19:00~21:00

場所：機械振興会館6F-

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)で開催する予定です。詳細が決まり次第、皆様にお知らせいたします。

お、教育分科会への問い合わせは、杉田(SRA-03-234-2611)、または中園(BSC-03-501-4164)迄。

再利用分科会 (SIGREUSE)

再利用分科会は、毎月第2、第4水曜日に、機械振興会館(東京・神谷町一地下鉄日比谷線)で、討論会を19時から6F-62で開催しています。参加は自由ですので、再利用に興味のあるかたはどうぞおいで下さい。

8月27日は、「部品化に基づく再利用システム」がテーマです。また、9月10日には、パネル形式の小討論会を企画しています。詳細は、村井(日本コンピューター・サービス-03-433-8171)までどうぞ。

法的保護分科会 (SIGSLP)

第2回の会合を、8月5日(火)にソフトウェア流通促進センター会議室で行った。

次回第3回は、9月2日(火)に18:30から、同じソフトウェア流通促進センター会議室で行う。法的保護分科会の会合は、毎月第1火曜日です。そんなわけで、

「一火会」という名称をつけました。

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7回月例会

環境分科会の月例会が7月16日(水)に開かれました。あいにくの雨でしたが、事務アプリケーションにおける自動生成系のケース・スタディ 言える内容で、活発なディスカッションが行われた。この内容は、SEAMAIL誌上で発表したいと思っています。

2. 今後の予定

環境分科会の月例会は、第3水曜日の19:00から、機械振興会館で行います。

第8回月例会

日時：8月20日(水) 19:00~21:00

場所：機械振興会館B3-9

話題：ソフトウェア開発環境としてのメインフレーム TSS

問い合わせは、松尾(SRA-234-2611)まで。

ネットワーク分科会 (SIGNET)

世話人の鈴木弘さんが、社業で忙しく、なかなか実質的なスタートができません。会員の方からも問い合わせがありますので、どなたか世話人に参加して頂けたらと思います。SEA事務所または、鈴木(03-382-6761)まで連絡下さい。

関西支部

去る7月19日に第5回の研究会を行いました。内容は、松本さんから「ダメプログラムの見分け方」、東京からの応援講演で熊谷さんから「愛(AI)の序章」、そして山田さんからは「マッキントッシュの世界」という、メッチャ3本でした。それぞれ、ソフトウェアに関わっている者にとっては、興味深いものでしたが、誌面の都合上、内容の紹介は別途にします。

以下に、今後の関西支部の活動予定を掲載します。

SEA関西9月度月例会

日時：昭和61年9月17日(18:30-20:30)

場所：SRA関西支社会議室

テーマ：システム設計用紙の比較

SEA関西10月度月例会

日時：昭和61年10月8日(18:30-20:30)

場所：SRA関西支社会議室

テーマ：未定

SEA関西第6回研究会

日時：昭和61年9月20日(14:00-17:20)

場所：日本能率協会研修室

テーマ：プログラム自動生成の現状と動向

関西支部へのお問い合わせは、臼井美義(06-488-6021 日本電子計算)にどうぞ。

横浜支部 (Y-DOC)

第4回のY-DOC月例会は、関西支部から常任幹事でもある盛田さんを招いておこないました。

さて、第5回の会 は、ちょうど中華街の老舗「Hei-Chin-Rou」が改装を終え、モダンになって営業を再開したので、以下のような納涼討論と会食会にしたいと思います。

第5回Y-DOC「納涼討論&会食会」

日時：8月29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

費用：1名¥10,000-(会食費、資料代等)

内容：「UnixとMacの世界」佐原伸

会食：「Hei-Chin-Rou」(TEL.045-681-3001)

20:00~22:00を予定

連絡先：8月25日(月)迄に、藤野(045-622-2163)

までご連絡下さい、15名に達し次第、締め切ります。

第6回会合

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

テーマ：「AI: Now and Then」伊野誠

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
- ・定量的マネジメント-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ください。

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

The 1
a) Ex
tools.
fee w
interi
b) To
prese
will b
tools

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

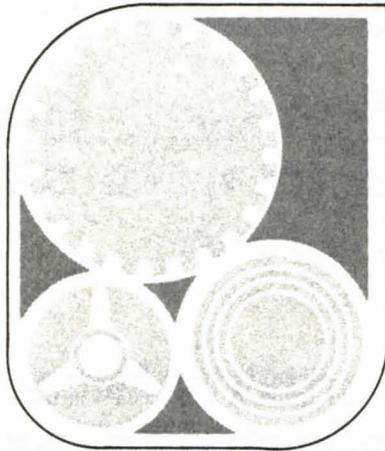
Tools Fair is being held in conjunction with the International Conference on Software Engineering. The purpose of the Tools Fair is to provide conference attendees an opportunity to view a variety of software development tools that are becoming available, both from the research community and from industry.

The Tools Fair will have two components:

- 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.
- 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

SEAこれからの活動予定

- 1986 -

◆8月(August)

20日(水)

環境分科会第7回月例会(機械振興会館)

27日(水)

再利用分科会第6回討論会(機械振興会館)

◆9月(September)

2日(火)

法的保護分科会第3回会合

(ソフトウェア流通促進センター)

3日~6日

秋のセミナー・ウィーク(機械振興会館)

10日~13日

のプログラミング・ワークショップ(岩手県・盛岡)

10日(水)

再利用分科会第7回討論会(機械振興会館)

11日(木)

AI分科会第4回月例会(機械振興会館)

17日(水)

環境分科会第8回月例会(機械振興会館)

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

特別フォーラム

「ソフトウェア技術と産業の未来を考える」

(東京・東条会館)

9日(木)

AI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~24日

UNIXワークショップ86 in 横浜(ホリデイ・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)

AI分科会第6回月例会(機械振興会館)

SEA会員状況

昭和61年7月30日現在の入会会員状況は以下の通りです。

正会員--432名(先月比60名増)

賛助会員--11社

	地域分布>		<男女分布>	
	勤務	居住	男 =	404
岩手 =	2	2	女 =	28
新潟 =	4	4	<年齢分布>	
栃木 =	1	2	20-24 =	18
茨城 =	2	3	25-29 =	88
埼玉 =	3	37	30-34 =	20
千葉 =	8	33	35-39 =	115
東京 =	296	171	40-44 =	47
神奈川 =	26	85	45-49 =	27
長野 =	5	5	50-54 =	5
富山 =	1	1	55-59 =	6
静岡 =	3	3	60以上 =	4
岐阜 =	2	4		
愛知 =	7	5		
滋賀 =	0	1		
京都 =	4	8		
大阪 =	43	33		
奈良 =	0	2		
兵庫 =	14	18		
福岡 =	2	2		
長崎 =	1	1		
鹿児島 =	1	1		

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。

ソフトウェア技術者協会入会申込書（正会員）

（フリガナ）

氏名： _____

勤務先名： _____

勤務先住所：〒() _____

勤務先TEL： _____

自宅住所：〒() _____

自宅TEL： _____

連絡先（どちらかにチェックしてください） 勤務先 自宅

年齢 ____ 才 性別（男・女） 血液型（A・O・B・AB）

会費： 入会金3千円 + 年会費（入会より1ヶ年分*）7千円 = 1万円

ソフトウェア技術者協会入会申込書（賛助会員）

会社名： _____

（フリガナ）

代表者： _____

住所：〒() _____

電話： _____

連絡担当者：

（フリガナ）

氏名： _____

所属： _____

賛助会費： _____ 口（1口 5万円） ただし入会より1ヶ年分*

<申込書送付先>：〒166 東京都杉並区高円寺南1-5-4 高円寺サンハイツ404
ソフトウェア技術者協会

<会費振込先>：三菱銀行本店公務部
普通預金口座 No. 0004830
口座名：ソフトウェア技術者協会
なるべく添付の振込用紙をお使い下さい。

*： 61年2月26日の幹事会での細則決定による



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