



# SEAMAIL

Monthly Newsletter from  
Software Engineers Association

Volume 1, Number 5 - 6 | May - June 1986

## 目 次

|                        |        |       |         |  |    |
|------------------------|--------|-------|---------|--|----|
| 編集部から                  |        |       |         |  | 1  |
| SDIをめぐる誌上討論            |        |       |         |  | 2  |
| AIとSDI                 |        |       | 馬渡 恭三郎  |  | 2  |
| Computer Wars          |        |       | 野村 行憲   |  | 3  |
| SDIの意味                 |        |       | 大西 亮一   |  | 4  |
| Parnas 氏への反論           |        |       | 村井 進    |  | 5  |
| シグマからParnas エッセイを読む    |        |       | 久保 宏志   |  | 6  |
| SEA Forum April '86 から |        |       |         |  | 8  |
| 基調講演                   |        |       | 岸田 孝一   |  | 8  |
| パネル討論                  |        |       | 関崎 邦夫   |  | 11 |
| 長岡の雪に埋もれて              |        |       | 酒匂 寛    |  | 16 |
| 飲み屋の塩                  |        |       | 高波 治夫   |  | 17 |
| 田舎のコンピュータ              |        |       | 芝原 雄二   |  | 18 |
| メモリ・ダンプ解析の自動化!?        |        |       | 小須田 正孝  |  | 20 |
| 「若手の会」ワークショップへの招待      |        |       | 堀江 進    |  | 22 |
| 現状における再利用支援ツール         |        |       | 阿部 正平   |  | 24 |
| 書評                     |        |       |         |  | 27 |
| 総会報告                   |        |       |         |  | 28 |
| 幹事会報告                  |        |       |         |  | 29 |
| 分科会および支部活動             |        |       |         |  | 30 |
| 環境分科会                  | 再利用分科会 | 管理分科会 | A I 分科会 |  |    |
| 教育分科会                  | 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 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)：白井豊 野辺良一 藤野晃延

ネットワーク分科会(SIGNET)：鈴木弘

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. 5-6 昭和61年6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

5月は、年次総会やら設立記念フォーラムやら、行事がたてこんだため、ついに船便が1ヶ月以上遅れてしまいました。そこで編集長の独断で、5-6月合併号ということにさせていただくことにしました。4月号で少しがんばりすぎた反動かもしれません。ゴメンナサイ！

内容も、第4号にのせた次号予告とは、少し変わりました。誌上フォーラムのテーマは「STAR WARS論争」、SEA Forum April' 86のレポート、会員からのエッセイや、各分科会での発表成果等、これまでより、ちょっとだけバラティエに富んだものになったと思います。

ただ、残念ながら、プログラマ・インタビューは、今回もお休み。担当のインタビューが新会社設立の雑用に追回されていて、なかなか思うように時間の余裕がとれません。どなたか、お手伝いくださるボランティアはいませんか？（もちろん、インタビューをで、会社をではありません）

### 誌上フォーラム

今回の誌上フォーラムは、D. パルナスのSDI批判エッセイ（Star Wars Papers）をとりあげました。原文は、CACM誌の85年12月号ののっています。また、あまり上手でない翻訳が、岩波書店の「世界」86年6月号にのりました。

5人の会員の方々の意見に対する反論やコメントを、どしどし編集部（SEA東高円寺事務所）におよせください。

### Forum April' 86

April フォーラムは、4月24日（木）に、東京の機械振興会館で、約80名の参加者で行われました。第1回SEAワークショップの報告を兼ねたもので、「実践的ソフトウェア開発環境を考える」が基調テーマでした。当日行われた基調講演と、パネル討論を編集部でまとめました。これからも、こうしたイベントに参加できなかった会員へのサービスの意味で、編集部の力がおよぶかぎり、記録をのせるようにします。

なお、今後企画されているイベントの一覧は、巻末36ページに掲載してあります。

### 会員からの投稿

夏のワークショップ・チェアマンの堀江さんの呼びかけを含めて、会員からの投稿を5篇収録しました。SEAMAILは、みなさんの雑誌です。エッセイ、書評、情報紹介、技術メモなど、なんでもけっこうです。どんな原稿をおよせください。

この間の総会に出席された方々には、どんなものが書けそうか、アンケートさせていただきました。いずれにせよ、すこし原稿の在庫をためないと、船便が遅れる危険はつねにつきまといます。

みなさんの積極的なご協力をお願いします。

### 分科会からの成果報告

分科会が本格的に活動してきています。今回は、再利用分科会から、阿部さんに、研究会での発表内容をまとめていただきました。地方会員、また分科会に参加していない会員への情報提供を目的として、今後もこうした小論文をできるだけ掲載していくつもりです。

7月号の発行は、いまのところ、7月第1週を目標にしています。さて???

## A I と S D I

馬渡恭三郎

シャープ

## 1. はじめに

SEAに入会したおかげで、バルナス氏のSDIに関する技術的エッセイを読む機会をえた。近頃めずらしく、読後さわやかな気分になった。話題のSDI構想に不可欠な、大規模ソフトウェア・システムの非現実性についての、論旨の明快さもさることながら、個々のエッセイを通じて、氏の良識を感じとれたからであろう。

ひさしぶりに、漬物石のような日本語のマニュアル（広辞苑）をひいてみた。良識は、bonsenseの訳語で、「社会人としての健全な判断力」、見識は「物事の本質を見通す、すぐれた判断力。また、ある物事の本質についてのしっかりした考え・見方」とあった。

## 2. SDIは不可能か

8つのエッセイを分析的に読んでみると、ソフトウェア・エンジニアリングの実状についてのすぐれた解説ではあるが、SDI不可能説を立証するというでもないようだ。ただし、敵の出方によって、SDIシステムに対するリクワイメントおよびハードを含めたシステム自体がどうなるかが予測不可能であるという指摘は、SDIシステムの致命的なウィークポイントをついているように感じられる。

そもそも、核兵器を提供した科学者たちに、今度はそれを無能力化して、安心して眠れるようにしろなどと要求するのは、無理難題以外のなにものでもない。戦略というからには、生きた強力な相手のある話である。目的は防衛なのだから、ドカンと先制攻撃で終わりというわけにはいかない。相手にも知恵はあるし、考える時間もある。スパイの出る幕もあるだろう。

SDIによって平和をもたらすためには、敵の戦略をすべて察知して対処するためのシステムの開発を、ある期間分先行して行わなければならない。それには、大人と子供位の差がなければ不可能である。政治ではなく、技術の問題である。

## 3. ソフトウェアの信頼性について

伝統的なエンジニアリングに較べて、ソフトウェアがなぜ信頼できないかという氏の論旨は、数学的観点から

はうなずけるが、別な要因もあると思う。

ソフトウェアは処理手順の記述であって、重さもなければ厚みもない。通常のエンジニアリングでは、電気も含めて物体を扱う。これは硬さも強さもあるもので、加工に誤差はつきものである。物体を相手にした場合、許容誤差の概念がはいってくるのは、当然のことだ。

漬物石の重さが1キロ違っても、大勢に影響はない。それでも、たとえば航空機のエンジンなどは、1万時間のオーダーで単体テストを行っている。進水式で船がそのまま水底に行ってしまう、というおかしなことも起こる。動かしてみなければわからないのは、ソフトウェアだけではないようだ。

もう一つ気になるのは、積み重ねのことである。河原の石は積み重ねていくといつかは崩れるが、ソフトウェアはどうだろうか。規模が大きくなればなるほど、信頼性の向上は困難になるが、それでも、技術的な積み重ねで改良していけるものなのか、どうか。

時間と金の制限さえなければ、質的な転換（技術的ブレイクスルー）をとれないながら、進化していくものと考えたい。

## 4. 人工知能と真の知能

人工知能と人間の知能を較べるのは、造花と花を較べるようなもので、よく調べてみると、まったく違うものだというのは、名言だと思う。しかし、外見上限りなく本物に近い造花というものも、技術の進歩によって実現されていくのではないか、と思ったりもする。

バルナス氏のエッセイを通じて伝わってくるのは、やはりモジュール化と Information Hiding の手法である。最後のページで、氏はSDIのもたらす（実現可能な）希望のために、巨額の公費を使ってよいものかどうか、という疑問を投げかけている。自分はモラルや政治についての専門家ではないので、これには答えない、と書かれている。しかし、氏の最もいいたいのは、まさにこの点ではないかと思う。見せかけの判断ならば、AIを使ってもできる。しかし、良識の伴った判断は、真の知性にしかできない。

## Computer Wars

野村行憲

岩手電子計算センター

### 1. スペース・シャトルの受難

私は The Space Shuttle Operator's Manualを持って  
いるほど、ミューハーなスペース・シャトル・ファンであ  
る。だから、1月28日朝の悲劇的な事故の様子がテレ  
ビで報道されたときの驚きは、いまでも忘れない。事故  
原因の究明が進むにつれて、高い信頼性を誇っているは  
ずの複合コンピュータ・システムが、実は、センサがな  
いハード・ウェアの故障には知らんぷりを決め込むとい  
う、いわば当然の欠陥が表面化した。この事故は、はか  
らずもパルナス氏が予見していたように思う。

### 2. コンピュータ戦争

われわれの日常生活にコンピュータが浸透して来てい  
ることは、いろいろな現象で実感される。しかし、戦争  
(防衛)分野へのコンピュータの応用については、話で  
きくだけであり、こうした事件をきっかけに、あらため  
て考えると、背筋が寒くなるのを感じる。たしかに、軍  
事競争がいろいろな技術の発展に寄与していることは認  
めるものの、それが本来の目的に使用されないことを切  
に希望する。システムが信頼性に乏しいとすれば、なお  
のことである。

だいたい、SDIのいうイニシヤティブな防衛って、  
いったいどんなものなの。まさか、インタラクティブの  
Iではないんだろうね。もし、どうしても戦争がしたい  
なら、人間抜きでコンピュータだけでやれば、結果はは  
やいし、だいいち、だれも血を流さずにすむじゃない。

### 3. ソフトウェアのテストに完全はない?

世の中には、命令の数が百万をこえる巨大なシステム  
があるそう。私は個人的には、そんなに大きなシステ  
ムは経験していないが、しかし、数千〜数万行のシステ  
ムにおいても、ふつう、そのテストは完全には終わらな  
い。プログラマは、「虫」たちにあざわらわれながら、  
何を、どのように、どれだけテストすればよいか、悩ん  
でいる。

テスト環境では完璧な動作をしていたプログラムなの  
に、リリースしてすぐに虫が見つかるケースは多い。テ  
スト環境をいかに現実に近いに近づけるかが課題であるが、ど  
んなに近づけても、けっして現実とは一致しない。現実

の環境を使ってテストできるのは、ごく少数の幸運なケ  
ースに限られる。まして、SDIの場合には、それを考  
えることすら許されないだろう。

### 4. 開発方法論はどこへ?

伝統的なプログラマの一人である私は、これまで構造  
化・モジュール化をセッセと取り入れ、生産性・信頼性  
の向上に満足しつつ、自動プログラミングの進展に期待  
し、AI技術やAdaなど新言語の動向を見守ってきた。  
パルナス氏のいうようにこれらの効果が否定されるとし  
たら、何に希望を見出したらいいのだろうか。

しかし、答えは簡単で、私が実際にそうした問題にぶ  
つかる前に、先を行くだれかが解決策をみつめてくれて  
いて、私はそれを利用するだけでよいのだ。でも、それ  
はいつのことなんだろう。

### 5. テクノロジーのCOPY

コンピュータに新しい仕事を教えるには、ソフトウェ  
アの入ったフロッピー・ディスクをCOPYするだけで  
いいのに、人間の場合はドキュメントと音声で伝達する  
以外に、有効な方法がみあたらない。この場合、問題な  
のは、転送速度の遅さよりも、受け手の理解力に左右さ  
れることである。「馬を川につれていくのはできても、  
その水を飲ますことはできない」と、古いことわざにも  
いわれている。もちろん、プログラマは、馬でも鹿でも  
ない。

古典的な方法論にしがみついている人たちが、世の中  
には意外に多いようだ。そのことに安心感を持ちはじめ  
ている私がいまなによりも手に入れたいのは、苦勞せず  
に新しい技術を転送できるゲートウェイだ。そして、ま  
ず最初にそれを、自分自身の頭に取り付けたい。

### 6. おわりに

私は、しばらくのあいだ(注意深く)静観しよう。で  
も、Star Warsが話題になるのは、アカデミー賞の審査  
委員会だけに限ってもらいたい。なぜなら、これから面  
白いコンピュータがどんどん出てきそうなときに、自分  
の命をバグで殺されるのはマッピラだからである。

その意味で、SDI諮問委員会からパルナス氏が脱退  
したことに、一抹の不安を感じる。

## SDIの意味

大西亮一

協同システム開発

### 1. はじめに

SDIについては、現在日本でも参加が検討されているが、大規模な未知の技術を必要とすること、さらには核抑止論上の問題などもあって、計画に対する賛否両論が渦巻いている。

バルナス氏の論文は、SDIにおける戦闘管理ソフトウェアの実現可能性・信頼性・品質・検証可能性などに対して批判を加えたものだが、SDIの構想それ自体が、同様の技術的問題を持っていると思われるので、ここでは、ソフトウェアの観点のみに留まらず、防衛システムとしてのSDIの得失を述べてみたいと思う。

### 2. 防衛システムとしての問題

過去において、米ソの相互確証破壊による永続的戦争状態を脱するためのさまざまな政治的・技術的模索がなされた結果、弾道弾迎撃ミサイルが出現した。しかし、防衛技術の不完全さ、迎撃手段の増加は、相手の攻撃兵器の増大を招くこと（ABMの出現は、核ミサイルの多弾頭化を促進した）になり、相互確証破壊突破のキイにはならないとされて来た。

SDIは、レーザー兵器・宇宙事業技術・センサー・コンピュータ技術等を新たにひっつけて、再びBMD論争に火を着けたが、その実現にあたっては、次のような技術的問題を克服しなければならない。

弾道弾防衛は3段階に分かれる。

第1段階は、弾頭を軌道に載せる打上げ段階で、ロケット排気の赤外線を検出し、レーザー兵器により破壊する。SDIでは、この段階での撃破に最重点が置かれている。キーになる技術は、発射の早期発見・センサー・照準等である。これらは、赤外線の放出を抑えたり、おとりミサイルを多数発射させる等の、単純な対抗策によって攪乱されやすい。また、レーザー発射のための大電力の瞬間的供給も大きな問題である。

第2段階は、軌道上に多数の核弾頭がばらまかれた状態で、これは、赤外線検出と運動エネルギー兵器（銃に類するもの）を用いて撃破する。バルナスが述べたように、この段階で、おとり弾頭や対抗装置の識別にAIの技術が要求されるが、センサージャマー（電波や赤外線

による攪乱）等により、第1段階よりも、さらに撃破が困難になるだろう。

第3段階は、核弾頭の大気圏突入段階で、この段階での撃破は、第2段階よりもずっと困難とされている。

以上の動作を、ICBM（大陸間弾道弾）なら30分以内に、SLBM（潜水艦発射弾道弾）の場合には15分以内に、行わなければならない。また、SDI支援装置自体がきわめて大規模であるため、敵の攻撃には大変弱い。さらに、多様化する核兵器にどう対処するのかという問題もある。弾道ミサイルには有効であっても、海や陸や空中さらには水中から発射される巡航ミサイル、レーダー無感応の爆撃機等には無力である。

### 3. SDI研究の意義

にも関わらず、SDIを推進する大きな理由は、早期警戒・宇宙戦闘・宇宙通信等の新しい技術要素をそれが含んでいるからである。ミサイル発射の確認は、レーダー設備と航空機および軍事衛星によって行われるが、宇宙空間からの監視が最も有効である。また、これらの宇宙兵器を破壊するための衛星攻撃等の研究も重要である。

さらに、宇宙空間を利用した軍事通信システムの研究も、技術的にはおもしろい要素を含んでいる。全地球規模に展開する戦闘を想定し、命令指導系統を統合化すること（いわゆるCCCCI）が研究されているが、今日の電波通信以外に、SDIに関連が深いものとして、ミラー衛星中継によるレーザー通信がある。これは、原子力潜水艦に対し、現在の効率の悪い通信に代わる有効な通信法として、以前から検討されている。

### 4. まとめ

SDIの技術的特徴を大まかに述べたが、バルナス氏が指摘した問題は、そのまま全システムに対してもあてはまる。米国の多くの大学や研究機関がSDIに不参加の意志を表明しているのは、同様の危ぐからであろう。それにも関わらず、西側諸国の多くが本計画に参加するのは、上述のような宇宙空間での軍事的活動の研究に大きな技術的意義を見出しているからにちがいない。

以上、SEAの趣きとは違った小文を載せましたが、いかがでしょうか？

## Parnas 氏への反論

村井 進

日本コンピューター・システム

### 1. はじめに

全体で8つの章から成るバルナス氏のレポートは、約10ページの長さで、最初の9ページは、技術的見地からのSDIプロジェクト批判であるが、最後の1ページ(第8章)は、明らかに、SDIOにかこつけたDOD批判である。正確に言えば、DODのソフトウェア・プロジェクト運営方法に対する批判である。私は、この最後の章に、非常に興味を引かれた。

このレポートの最初の部分でバルナス氏は、「政治的な意味でSDIを批判するつもりはない」と何度かことわっている。しかし、人間は、本当に自分がいいたいことについては、意図的にそれを否定しがちである。そして、最後にひとこと本音をもらしたりする。

### 2. 批判

バルナス氏のSDIO批判の要点は、第1に、ソフトウェア・プロジェクトの予算のかかなりの部分が、ほとんど技術的価値のないことに費やされているということ、そして第2に、予算が、その本来の目的のためではなく、他の目的に使用されている(つまり、できもしないプロジェクトをでっちあげて予算を獲得し、適当な理由をつけて他に流用している)ということであろう。

その理由として、氏は、予算を決定するDODの役人に技術者としての判断能力がない、また忙しくて勉強する暇もない、また、役人が相談する諮問機関は利益代表で占められており、最先端の研究者や経験豊かな現場の技術者が実質的には参加していない、といったことがらをあげている。そして、DODが予算をムダにしないためには、これらの研究者や技術者をプロジェクトの決定に参加させるべきだ、と結んでいる。

### 3. 個人的関心

ところで、なぜ私がこうした批判に興味を引かれるかといえば、それは、自分自身が過去4年間、同様な国家プロジェクトにたずさわってきたからであり、また現在もいくつかの国家プロジェクトの進行状況に興味を持っているからである。

結論的に言えば、私は、「バルナス氏の批判は理解できるが、しかし批判を向ける対象を誤っている」と感じ

ている。DODに対する批判は、いかにそれが正しくても、おそらく建設的な意義は見出せないであろう。事態を変えるために、建設的な方向へむかって自らを変えなければならないのは、むしろ研究者であり技術者であろう。

研究者や技術者と、政治家あるいは政治的判断を行う役人とは、かなり縁遠い人種であると思う。というよりも、かなり離れた所に自分の土俵を持って生きている、という気がしてならない。それぞれの土俵の中でのケンカは、もちろん土俵の持ち主が強いに決まっている。だから、どちらも相手の土俵で勝負したがない。

### 4. 反批判

問題は、国家予算の使い方はだれが決めるべきか、という点に集約されるであろう。私は、予算は役人が決めるべきだと考えている。少なくとも、「決定権」を技術者に渡してはならないと思う。それは、政治的決定権を、「軍人」に渡してしまった(取られてしまった)がゆえに突入した第二次世界大戦の二の舞を引き起こさないためである。

では、決定権なくしてプロジェクトに参加するには、技術者はどうすればよいのか。それは、技術力とともに「交渉力」を身に付けることである。役人や政治家と技術者のあいだの、交渉力の差とは何か。それは、「証拠を突き付けられても、知らぬ存ぜぬ覚えておらぬと、しらをきるのが政治家で、自分の意志が通らぬと、いさぎよく身を引くのが技術者だ」という点につきると思う。しかし、いさぎよく身を引いたあとで意見を発表しても、あるいは酒を飲んで気炎をあげても、結局は何も変わらないのが世の中である。

私は何も、技術者全員が政治家になれといっているのではない。しかし、少なくとも国家予算の使い方に文句をつけるつもりがあるのなら、相手の土俵にでかけて行って一戦まじえるくらいの、技術力以外の力を身に付けてほしいと思うのである。

端末という窓から世界をにらむか、あるいは端末という井戸で遊ぶカエルになるかは、まあ本人の勝手でしょうけど、おっといかん、また極論を吐いてしまった。

## シグマから Parnas エッセイを読む

久保宏志

情報処理振興事業協会

### 1. はじめに

シグマ・システム開発本部の一員として、これらのエッセイをどう読んだかを綴る。共感を覚えるところもあれば、異議あり、といたいところもある。もっと深く知りたいと思うところもある。これらにおさまりきれない感想もいだかせる。総じていえば、読物として非常におもしろかった。

文中で8編のエッセイを参照するときには、[1], [2], ……の表記をもちいる。

### 2. 共感を覚える

「プログラマたちがプログラムを書くとき、計算機がそれを実行する順序に合わせて思考する」という意見表明がある[3]。

「セミナーや研究会でどんな高度な方法を勉強した人でもそうだ、メンテナンスのときには特に顕著である」とつけ加えている。思いあたるところではあるが、いわれてみると、この指摘は新鮮だ。この傾向は、COBOLでプログラムを書く人に、もっとも顕著ではなかろうかという気がしてくる。ちょっとした実験でたしかめられそうである。誰かやってみないか。

このような思考規範から人々を解放するには、フォン・ノイマン型のアーキテクチャから中途半端に離れただけの方法論ではだめである。ICOTの淵さんたちがいうように、論理プログラミングのようなところに飛躍しないとイケないのかも知れない。「AI研究者たちのプログラミング・パラダイムには学ぶべきことが多い」という玉井哲雄さんの指摘(本誌2月号)にもつながる。

「現実の開発プロセスから生みだされた現物が技術移転に有効である」という意見表明がある[4]。

これには多くの人々が同意するだろう。設計方法論がどんなに立派であっても、それだけではいい設計はできない。似たようなシステムで幅ひろい体験を積む必要がある。これは、おおくの人々が体験的に得ている知見と一致する。

しかし、いくら実体験が有効であっても、それに頼るだけでは能がない。先人たちの足跡を、現物があればそれを追体験することで、ある程度実感できる。その考え

があつて、シグマでは、いろいろな開発事例を、いいもわるいもすべてデータベースに集めていきたいと考えている。

現物による技術移転を実践すべく、著者の手で進行中のSCR(Software Cost Reduction)プロジェクトは、たいへん興味深い。誰か紹介の労をとってくだらないだろうか。シグマでも、大学人にその気があれば、似たようなことを依頼できそうに思える。

「ルール型プログラミングは人を誤らせる」という意見表明がある[5]。

この意見は、すでに本誌2月号で、宮本勲さんが紹介しているので、くわしくは触れない。ルール型プログラミングの手法でつくられた有名なシステムに、VAXの機器構成を設計するシステムがあるが、「わかりにくくてメインテナンスにこまるだろうから、在来プログラミング手法で書き直したほうがよい」といっている。記憶にとどめたい意見である。

### 3. 異議あり

「プログラミング言語のちがいやソフトウェア開発環境のちがいは、生産性・品質に大きな差をもたらさない」という意見表明がある[4,6]。

この意見はこまる。

プログラミングのテクノロジーは言語化しうるし、開発環境に転化しうるという、シグマ・プロジェクトの基本仮説を否定するものだからである。

この意見を一般的に正しいと受けとるのはまちがいだと、思っておくことにする。環境という言葉で表されることがらを、せまくなりすぎているようにも、感じられる。

気に入らないからといって、反発してもはじまらない。実体験から導かれた意見らしいからである。素直に、シグマへの問題提起として受けとっておきたい。

### 3. もっと知りたい

「プログラムの正しい構造を見つけるのが、むずかしいことが証明されている」という意見表明がある[1]。

それはそうだろうが、あるプログラムの構造が正しいかどうか、正しくないとされるのはなぜかを、プログラ

マに教えることはやさしいはずだ。これを教えてくれるような開発環境を持つことができれば、一つの進歩であろう。GOTOがあるかないかを調べてくれる程度ではつまらない。もっとインテリジェントでなければ進歩には寄与しない。誰か、この方向で進歩のプログラムを示してくれないものか。

「記号論理のプログラミングへの応用は進歩したが、進歩は緩慢である」という意見表明がある[1,7]。

ほとんど無知にひとしいので、詳しく知りたい。どこまできているか、そこまでくるのにどれくらいかかったか。どんな経路をたどったか。いまはどちらの方向を向いているか。これからどんな経路をたどりそうか。記号論理をほんのちょっと知っているだけでも、効用があると思っている方なので、どんなに進歩が緩慢でもあきらめたくない。IEEE Softwareの1985年1月号で読んだ On Formalism in Specifications程度のことを、現場に普及定着させるだけでも、大きな進歩である。

努力の方向を指し示すステートメントを、誰かつくってくれないものか。

#### 4. 感想はほかにもある

「政治的な立場を表明するものではない」としている。

しかし、「お上」に逆行行動に出たことは、たしかである。日米の社会の差を感じさせる出来事である。もちろん、アメリカの方が健全であるという意識が、筆者にはある。政治の世界にいないわれわれは、これを読んで自らを戒めるのが正しいであろう。

[8]はテクノクラート批判の文章である。ナショナル・プロジェクトを企画する人、それにむらがる人、それを率いる人に、ぜひ読んでもらいたい文章である。日本のテクノクラートの方が、アメリカよりずっと優秀であるとは信じている。それでも読んでほしいと思う。

#### 5. おわりに

日本人の学者に、日本のソフトウェアの現場の状況に対する批判をこうしたエッセイにまとめてもらいたい。希望するタイトルは「事務処理アプリケーションのソフトウェア的側面」である。焦点は、第三次金融オンラインに合わせてほしい。

会員各位と意見交換するのに、おもしろそうなトピックを拾うことにつとめたつもりである。筆者の読みまちがいを指摘することをふくめて、多彩な意見がこれから誌上にぎわすことを期待している。

### 今後の誌上フォーラム

特定の論争テーマを選んで、会員相互が意見を交換しあう誌上フォーラムは、これまでに、1月号「シグマ」、2月号「AIとソフトウェア工学」、3月号「プログラム定年説」、そして今月の「SDIに関するバルナス・エッセイ」と続いてきました。幸い、読者からの評判もいいようです。次号では、「ソフトウェア産業論」を取りあげる予定で、すでに何人かの方々に原稿を依頼してあります。

その次の8月号は、21人セミナーの講師の方々のポジション・ペーパー特集のため、フォーラムはお休みしますが、9月以降に予定している論争テーマは以下の通りです：

#### ★「情報処理技術者試験について」

毎年おこなわれている情報処理技術者試験について、賛成／反対、試験の内容や方法の改善、合格者へのフォローアップなど、さまざまな角度から議論してみたいと考えています。

#### ★「パソコン・ソフトをめぐって」

権利保護、不法コピー、プロテクトにからむ技術的なことがら、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。

#### ★「女性技術者から（へ）の提言」

SEAの会員のなかでは、まだ5%ぐらいですが、最近どの職場でも、女性技術者の比率はふえてきています。そこで、ソフトウェアの世界における女性技術者の問題を広く考えてみたいと思います。男女いずれの立場から論じていただいてもかまいません。

これらのテーマへの読者からの投稿をお待ちしています。原稿はSEAMAIL仕上がりで1ページ分（約2千字）、編集部（東高円寺の事務所）まで、どしどしお寄せください。できればワープロ出力が望ましいですが、手書きでも結構です。

また、これまでに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おります。

April Forum 特集

## 基調講演：環境構築技術の動向

岸田孝一

ソフトウェア・リサーチ・アソシエイツ

## 1. 変化への予感

環境というキーワードは、80年代前半のソフトウェア技術を、たしかに特徴づけています。NBS（アメリカ連邦政府標準局）の主催によって、最初に環境をテーマとするワークショップが開かれたのは、1980年春のことでした[1]。そして、その翌年あたりから、世界中のあちこちで、さまざまな目的を持った環境構築プロジェクトがはじまっています。

私自身がJSDのSMEFプロジェクト[2]にまきこまれたのも、ちょうど同じ頃でした。この計画の「たてまえ」上の目的は、アプリケーション・メンテナンスのためのツール開発ということになっていましたが、しかし、そのほんとうの狙いは、UNIXをベースとする対話型の支援環境を、日本のソフトウェア業界に持ちこむことにありました。

さて、80年代も半ばを過ぎ、SMEFを含めて、一応はさまざまなプロジェクトの成果も出揃った現在の時点で、あらためて周囲を見回してみると、これから何かが大きく変わるのではないかという予感が、ひしひしと感じられます。

社会的な背景としては、まず、コンピュータ普及度の一層の加速にともなうソフトウェア・ニーズの指数拡大があり、そしてまた、一方では、アプリケーションの複雑化にともなう、伝統的な開発パラダイムの崩壊現象があちこちで観察されています。

経済的な観点から眺めて見ると、ハードウェア技術の進歩によって、これまでの汎用大型機やスーパーミニとは肩を並べる程度の高性能パーソナル・ワークステーションが、ほとんど実用化の一手前まで来ていますし、それらはしかも、LANやWANによって連結されて、統合的な分散型環境を構成しようとしています。

そして、その上に、ソフトウェア・エンジニアリングの過去数十年の成果や、AIまたは知識工学の実用化によってもたらされた多くの新しい技術上のアイデアが、ようやく実践的なツールとして実を結ぼうとしているわ

けです。

## 2. 何が変わるか？

ワークステーションやネットワークの進歩は、単に開発環境のみならず、われわれが作り出すソフトウェア・プロダクトの形態や性格に大きな影響をもたらすことが予想されます。つまり、これからは、多種多様なマシンをネットワークでつないだヘテロな分散型システム全体をコントロールする統合的なソフトウェアこそが、われわれの主要な製品になると考えるべきでしょう。いま目前にある分散型開発環境の構築は、そのための1つのケース・スタディにすぎないのです。

ウォータフォール・モデルに代表される古い開発パラダイムが現実のソフトウェア開発プロセスに適合しないことは、すでにいろいろな人々によって指摘されています。おそらくこれからの新しい環境は、ソフトウェアの進化や、プロトタイピングなどの概念を取り入れた、よりフレキシブルなパラダイムを支援するものでなければなりません。

環境の自動化の程度は、ここ2-3年のうちに、めざましく進むものと考えられます。それは、単にエンジニア1人あたりの端末やワークステーションの台数がふえるというだけでなく、それらを通じて各人が利用できるコンピュータ資源の量が、飛躍的に増大することを意味します。今日の開発環境における最大の問題は、一般のアプリケーションにおける同様に、データベースの支援が貧弱だということですが、この欠陥は早急に解決されるでしょう。

ネットワークの普及は、プロジェクト・マネジメントのあり方や、ソフトウェア開発組織の形態に大きな変化をもたらすはずで、極端な集中型社会である日本の場合、その影響は他の国よりも大きいかもしれません。単に、東京で受注した仕事を地方のソフトウェア工場で消化するといった形態だけでなく、プロジェクト・チームそのものが最初から地理的に分散し、しかもそのパフォ

ーマンスが1ヶ所に集中した場合よりもはるかによい、といった新しいマネジメント・スタイルが、かならず登場するでしょう。

ネットワークは、また、これまでどちらかといえば切り離されていたリサーチ・コミュニティと産業界とを、うまくつなぎあわせるブリッジの役割を、果たしうはずです。それにより、AIその他の基礎技術の成果の産業界へのトランスファは、一層早まるでしょう。

### 3. ふたつのワークショップ

この半年のあいだに、それぞれ環境をテーマとして行われた2つのワークショップがあります。1つは、昨年の11月にアメリカのコロラド州ボルダーで開かれたRMISEのワークショップ[3]、そしてもう1つは、われわれが長岡で開いた第1回SEAワークショップ[4]です。

これら2つのワークショップは、期せずして、お互いに相補的な関係にあります。

すなわち、前者は、NBSワークショップ以来5年の時間的变化をふりかえり、また、今後1990年代へ向かっての技術の進歩をにらみあわせて、現時点でできるかぎり正確なテクノロジー・マップを描くことを目標にしています。一方、SEAのワークショップは、参加メンバーそれぞれが、自分自身いま抱えている環境改善の問題点を持ちよって、いわばボトムアップ型のアプローチで、問題解決の方向を探ろうという試みでした。

おそらく、実際には、これらの2つの視点を組み合わせて、われわれ1人ひとりが、自らの足場とこれから進むべき道筋について、正しい技術的オリエンテーションを行う必要があるでしょう。

### 4. 個々の技術要素

環境を構成する個々の技術的要素に関して、現状と今後の予測をまとめれば、およそ次の通りです：

#### (1) ツール

環境とは、一言でいえばツールの集合体です。したがって、これまで、ツールの機能の分割やその再構成、あるいはインタフェイスの標準化などについては、いろいろな実験が行われ、一応の成果をあげてきました（たとえばTOOLPACK[5]）。しかし、まだ未解決の問題が数多く残っており、ひきつづき試行錯誤による模索が必要です。

また、現状でそれなりの機能の実現されていて、それらを強化することにより実用化が図れるものとしては、メタツール、プロトタイピング、テスト支援環境、バージョン管理、マルチ言語支援環境などがあげられます。

#### (2) 分散化

次の3つの問題があります。

すなわち、第1に、分散型システムの開発や進化を支援するために、環境のなかにどんな機能が必要か。第2に、環境がそなえている機能それ自体を、物理的または論理的に、どうやって分散させるか。そして第3に、分散型環境を用いて分散型システムを開発する場合、ホストと開発環境との使い分けをどうすればよいか。

これらは、おたがいにからみあっていますが、とりあえず、われわれにとって、もっともプライオリティが高いのは、2番目の問題でしょう。

#### (3) DB支援

環境におけるデータベースを考えるさいに、まずわれわれがぶつかる困難は、適当なデータ・モデルがまだみつかっていないという事実です。それは、いいかえれば、ソフトウェア開発プロセスそれ自体が、まだ、きちんとモデル化されていないことを意味しています。

とはいえ、現在すでに、いくつかのデータベース・ツールは存在するわけですから、それらを環境のなかに組み込んで、いろいろな実験を行うことから、まず出発するのが、解決の糸口でしょう。

#### (4) 拡張性

ソフトウェア・システムとしての環境は、進化論的に見れば、いわゆるE型プログラムの1種であり、その機能や構造は、時間とともに変わって行きます。

したがって、環境のアーキテクチャを考える場合に、その拡張性を重視しなければならないことは、いうまでもありません。このことは、特にユーザ・インタフェイスのデザインにおいて大切です。また、環境のラビッド・プロトタイピングに関しても、最近多くの実験が行われ、成果を上げつつあります。

#### (5) AI

環境へのAI技術の応用については、多くのことがいわれていますが、私自身は、あまり期待を持っていません。たとえば、ソフトウェア開発用のエキスパート・システムは、まだしばらくは、夢の領域にとどまるでしょう。

しかし、バルナス氏がSDI批判のエッセイで述べた

ように、AIコミュニティが発明したいくつかのプログラミング技法(メモリ管理、プロトタイピング、オブジェクト概念など)は、環境の構築にさいして、十分有用だと思います。

### 5. 横断的ファクター

上記の個別的要素以外に、環境開発にあたって考慮すべき、いくつかの横断的要因があります。

それらは、たとえば統合化の戦略をどうするかとか、さまざまなプロトタイピングのアプローチをどう使い分けるかとか、環境(あるいは個々のツール)のもたらす効果をどんな指標で測定し評価するか、といったことがらです。

また、ソフトウェア・ライフサイクルの全体を1つの環境で支援しようとした場合、ユーザ・インタフェイスの主要な要素である言語を単一化するか(たとえばADAにおけるANNA)、それとも複数の言語を多面的に支援するのかも、きわめて大きな問題です。

ハードウェアの革命的な進歩が、これからはしばらくは続くものと予想すると、マン・マシン・インタフェイスをどう扱うかは、特に環境の拡張性に大きな影響を与えるでしょう。

環境の開発や導入にさいして、漸進的な進化をとるのか、それとも一挙に過激な革命を図るのかも、現実問題としては、大きな意志決定要因になります。それは、組織の風土や周囲の条件を考慮しながら、慎重に決められなければならないかもしれません。

### 6. 環境革命の意味

最近、政治学者の丸山真男さんが福沢諭吉の文明論之概略を解説した本(岩波新書)がベストセラーになっています。偶然、この冬海外出張をしたとき、往復の機内で、幕末・維新関係の書物を何冊か読みました。われわれが、いまやろうとしている環境革命は、ある意味で、いわゆるひとつの文明開化・明治維新にたとえることができるかもしれません。

そうするとUNIXは黒船にあたり、現在のメイン・フレーム中心の業界地図は、くずれゆく徳川幕府だといったらいいすぎでしょうか(笑)。そういうおまえ自身は何者かと聞かれたら、死に損なった吉田松陰だとしても答えておきましょう(爆笑)。

革命家としてのわれわれに必要なのが、強烈な求心的

イデオロギーであることは、いまさらいうまでもありません。松陰は、あるとき、高杉晋作他の同志に対して、次のような批判の手紙を書いています：

一君たちは功業を成すつもり、僕は忠義をするつもり。

個々のプロジェクトの成功・失敗よりは、その思想的純粹さを大切にしようという精神が、ここにあります。

また、かれの弟子だった久坂玄瑞が土佐の武内半平太に送った手紙の一節には、次のような表現があります：

一この大義さえ成就すれば、失礼ながら、おたくの藩も、うちの藩も、滅亡したってかまわない。

いま、アメリカのUNIXネットワーク上を飛び交っている電子メールのなかには、こうした幕末の志士たちの精神的連帯感を思わせる調子のものが、しばしば見られます。おそらく、そうした既存の組織の壁をつきやぶる自由なエネルギーの交流によって、いつか世の中が変わってゆくのだと思います。

SMEFにせよ、SIGMAにせよ、あるいはSTAR/SEIにせよ、ナショナル・プロジェクトの果たすべき役割は、勝海舟が長崎や神戸で開いた航海訓練所のように、草の根の志士たちの精神をめざめさせる保育所の範囲を越えるものではありません。

### 参考文献

- [1] L.J.Osterweil, "Software Environment Research Direction for the Next Five Years", IEEE Computer, April, 1981.
- [2] 協同システム開発, 「SMEFテクニカル・レポート」(1・6), 1981-1985.
- [3] W.E.Riddle, L.G.Williams, "Software Environment Workshop Report", RMISE TR-85-001, 1985.
- [4] SEAワークショップ・レポート「実践的ソフトウェア開発環境」, 1986.
- [5] L.J.Osterweil, "TOOLPACK - An Experimental Software Development Environment", IEEE/TSE, June, 1983.

## April Forum 特集

## パネル討論：実践的開発環境の構築をめざして

コーディネータ：安間 文彦（情報処理振興事業協会）

パネリスト：熊谷 章（パナファコム）

佐藤 隆（インテック）

佐藤 千明（長野県協同電算）

盛田 政俊（神戸コンピューター・サービス）

## 1. はじめに

パネル討論の前半では、それぞれのパネリストから、長岡ワークショップでのグループ討論プロセスの発表が行われた。その内容は、SEAMAIL 4月号に紹介されている各グループ・コーディネータの報告とかなり重複するので、ここでは省略し、以下、パネル後半の質疑応答を中心にレポートする。

## 2. ワークショップの成果をどう生かすか？

安間：長岡での議論は、各グループそれぞれに盛り上がり、パネリストのみなさん、現場の臨場感を伝えるために苦勞をなさっていたようですが、いま、パネリスが発表された内容に関連したことがらや、報告書の中味を見ていただいての質問、あるいは、みなさんの現場での問題でも結構ですので、何かご質問、ご意見がありましたらどうぞ。

質問：今回のワークショップから得られた成果（いろいろな情報やヒント、あるいはアイデアなど）を、それぞれの職場のなかで、どのように生かしていくかという点について、お聞かせ下さい。

熊谷：私は、メーカーの人間ですので、他の方々と若干立場が異なると思われそうですが、今回、いろいろな人たちとお話した結果、メーカーとソフトウェア・ハウスとでは、ソフトウェアを作る場合のものの考え方とか、既存のソフトウェアやツールの利用法とかに、大きな違いがあるということを感じました。

具体的にいえば、われわれメーカーの人間は、必要なものはすべて自分の手で作るしかないと、頭から思いこんでいますが、ソフトウェア・ハウスの方の話を聞くと、世の中にあるもので使えるものは、なるべく有効に利用するということでした。

それから、ワープロなどの使い方もだいぶ違うようです。われわれ、つまりメーカーでソフトウェアを作るとき

には、ワープロをあまり必要としませんが、ソフトウェア・ハウスの場合には、ワープロが重要な位置を占めているようです。

本来、ソフトウェア作るという点では、メーカーもソフトウェア・ハウスも変わらないと思っています。したがって、現状における違いの原因がどこにあるのか、一番重要なことは何かを考え、これから、われわれメーカー・サイドにおいても、ソフトウェアやシステムを作る道具とか、開発環境について、真剣に考えていかなければならないと感じました。

そのための具体的なケース・スタディとして、いろいろな会社の方のお話を聞きましたので、今後いろいろと参考にさせていただこうと思っています。

佐藤（隆）：私の職場では、現在、PC 9800をターゲットとして、C言語でアプリケーションを開発していますが、作成したプログラムの中で共通的に使える部分を、ライブラリとしてUNIXマシン上で管理することを考えています。

また、ハードウェアに依存しない部分は、UNIX上のC言語で開発しようというのが、今回のワークショップに参加する以前からの構想でしたが、ワークショップでの、みなさんとの討論を通じて、その方向性が正しいことの確認ができました。

佐藤（千）：今回私がワークショップに持って行った問題は、汎用大型機のTSS環境下での不満をどう解消するか、ということでした。

たとえば、環境が変わると、エディタがその都度変わり、覚えるのがたいへんだというような問題があります。このことに関しては、メーカーでは何もしてくれないので、自分で適当なツールを作成し、今までのエディタのスタイルをできるだけ引き継いで行こうと、考えています。

日本語処理が弱い、ビジュアル・インタフェイスがな

いという問題もあります。この点については、ユーザとしてできる範囲は限られているので、どのようなハードウェアがあるのかを自分で調査し、完全なワークステーションとはいえないが、グラフィック・インタフェースを持ったものを導入し、ホストと結びつけた環境を開発していこうと思っています。

今回のワークショップの討論は、このような問題解決の方向を見出す上で、たいへん役に立ちました。また、汎用機上でのゲストOSとしてUNIXが乗る予定になっていますので、オリジナルOSとどのようにインタフェースを取って行くか、ワークショップで聞いたUNIXユーザの方々の話を参考にしながら、これから検討したいと考えています。

私のようなユーザの場合、ソフトウェアの実行環境としてはではなく、ソフトウェアの開発環境として、どうしてもマイクロ/メインフレームの結合を考えていかなければならないと思います。

盛田： 私のセクションは、バンキング・システムの開発を、COBOLを使ってやっています。大型メインフレームで動かすソフトの開発を、いきなりワークステーションやパソコンで何とかするというのは、難しいと思います。

私たちが抱えている最大の問題は、なんとかして自前の開発環境を作りたい、いいかえれば、メーカ・オリエンテッドな技術から離れて、ユーザ指向の開発環境に少しでも近づきたい、ということです。

今回のワークショップで得たものを、社内にどのように生かしていくかということですが、それは、参加したメンバーひとりひとりを通じて、紙に書かれたレポートや口頭の報告によって、長岡で得た知識やインパクトが少しずつ広まって行くと思います。

安間： 私はワークショップのコーディネータだったわけですが、今回の参加者は、現場の第一線で活躍されている方々が多く、それぞれが持ってこられた問題も、かなり具体的なものでした。ワークショップの効果の一つは、自分の持っている問題を、他の会社の人に話すということで、それはそれなりにストレスの解消になったと思います。もう一つは、討論を通して、他の会社の技術レベルや抱えている悩みが分かることだと思います。

### 3. 開発専用環境の分離

質問： ターゲット・マシンとワークステーション（開発環境）の分離の問題ですが、現状はどの程度分離して

開発を行っているのでしょうか？

佐藤（隆）： グループ討論のなかでは、そうした話題は、とくにでなかったと思います。

当社の場合ですと、大型計算機の場合には、開発専用のマシンを購入しても、しだいに業務量が多くなって、いつのまにか業務用になってしまうという結果に終わります。つまり、比較的空いているマシンが狙われるということで、なかなか完全には分離がうまくいかないと思います。

パソコン関係のソフト開発でも、現状は、開発マシンの上でプログラムが動いても、なんの保証にもならないということと、実機以外の物を買うのはコストがかかり過ぎるということで、実機上での開発を行うケースが多いと思います。

ただし、2年前のプロジェクトですが、ソース管理とロードモジュールの作成までを汎用計算機で、テストは実機で行ったという例はあります。

これからの話としては、汎用機の世界ではどうか分かりませんが、パソコンの世界では、ソース管理の問題とか、スピードの問題とかがあり、また、実機が変わるたびにまたあらためて環境をセットアップしなければならないという問題もありますので、開発マシンとターゲット・マシンとは、完全に分かれてゆくと思います。

安間： 実際にソース/ターゲットの分離は、現在ちょうど過渡期にあると思います。現実には、パネリストの方々がどのような環境で開発を行っているかを教えていただければ分かると思います。

盛田： 私の会社はいわゆる計算センターでありまして、商売用のメインフレーム・マシンをTSSで開発にも使っています。しかし、われわれがそこで開発しているソフトは、その環境で動くものではなくて、お客様のハードウェア環境で動かすものです。

いまわれわれのところでは、J-starでソフトウェアのドキュメンテーションを作成する実験を行っていますが、これらのワークステーションはまた、LANのゲートウェイを通してホストにつながっていますので、その上でターミナル・エミュレータを動作させて、ホストのプログラム開発も行っています。

このような形で、いまJ-starが7台、TSS端末が6台、パソコンが12-3台あります。

これから実験しようと考えているのは、現在のホスト・マシンの使い方には、いろいろ面倒な規制があって、

プロジェクトごとに独自の環境を作ろうと思っても、そんな使い方は困るとか、OSのバージョンが異なるとかで、なかなか思うようにいかないで、ネットワークを通じて外部の計算センターにアクセスし、J-starとゲートウェイを通して、こちらにも自分たちの環境を作っていこうと思っています。

あるときは、外部のホストにアクセスし、あるときは社内のホストにアクセスするというように、複数のホストを、ワークステーションを中継基地として自由に使い分け、本来の開発マシンとターゲットの分離とは異なる形かも知れませんが、そうした分散型の環境を作っていくと考えています。こんなふうに考えて行くと、なんとなくシグマようになってしまいますが(笑)。

#### 4. 先進的環境と現実とのギャップ

質問： 私は、汎用機を利用して図形処理とか構造計算のソフトウェア開発を行っています。最近の新聞雑誌をにぎあわせているUNIX環境や、高性能ワークステーションなどのレベルと、実際に、われわれが用いているツールや環境のレベルとは、だいぶ開きがあるように感じられるのですが、果たしてそうしたギャップはあるのか否か。

また、一般のプログラマのツールに対する意識(どのようにして先進的な技術を自分のものにしていくか、自分に合ったものにアダプトしていくか)に、どちらかといえば遅れがあるのではないか。もしそうだとすれば、どのようにして意識を盛り上げていったらよいかを、お聞かせください。

佐藤(千)： 実際には私は、与えられたメインフレーム環境の中で、いろいろくふうしてツールを作ったりしてきましたが、それは別にUNIXとは関係ありません。UNIXは、いま、たしかに注目を集めていて、それはいろいろなツールが揃っているからだと思いますが、自分なりの開発環境を整えようとするれば、やはりその上に新しいツールを追加しなければなりません。そして、UNIXが人気があるのは、そうした環境のカスタマイズがきわめてやりやすいからでしょう。

しかし、おのおののユーザは、現在なんらかの開発環境を持っていますし、その実態や改善計画を無視することもできませんので、いくらUNIXがいいからといって、いきなり全面的にそれに乗り換える訳にはいかないと思います。

私自身も、最初はUNIXにカルチャーショックを受

けましたが、いまでは、それだけがすべてではないと気をとりにおしています(笑)。ただし、UNIXのよいところはどしどし取り入れていかなければならないと思っていますし、それをどうやったらよいか、模索している最中です。

ワークステーションに関しては、単にプログラミングだけをするなら、TSS端末で十分だと思います。ワークステーションの特徴は、ユーザ・インタフェイスがすぐれており、ものを表現する仕掛けがいろいろ用意されている、ということです。したがって、プログラミング段階のほかにシステム分析や設計のフェイズまでをサポートするようにしなければ、ワークステーションの真価は発揮できないでしょう。ただ、これらの段階の作業は、ワークステーションだけでは無理ではないかと思っています。

それから、ツール利用の意識の問題ですが、テクノロジー・トランスファー般ともからみますが、無意識の内にツールを使うような環境が必要なのではないでしょうか。それがあたりまえになり、昔風のプログラムの作り方で開発ができないような、また、どんどん新しい仕掛けを吸収していかなければ落ちこぼれていくような開発環境を作っていく、そして、そのさい別に高度なものを使っているという意識がないように、もっていくことだと思います。

安間： 盛田さんのところでは、ドキュメントの作成にワークステーションを使っているということですが、どの程度効果が上がっていますか。

盛田： 導入してから1年程経過しましたが、ドキュメンテーションに関する問題がすべて解決したということはありませんが、だんだんそれなしには作業がやりにくくなってきました。最近では、鉛筆で紙に書くということとはなくなり、メモ書き程度をもとに、ワークステーションの前で思案しながらドキュメントを作っています。物の書き方とか、仕事の順番とか、ライフスタイルがたしかに変わってきていると思います。

佐藤(隆)： 私の個人的なケースですが、ワープロに凝って残業が多くなり、会社にとっては不幸なことになっています(笑)。

ただワープロの機能だけを使っているのではよくないと思います。端末エミュレータを通して計算結果を取り入れるとか、プログラムの実行結果を編集してドキュメンテーションに反映させるとかしなければ、意味がないのではないのでしょうか。

安間： ちなみに、私のいるシグマでは、UNIXマシンにつながったパソコン端末を一人1台の割りを使っており、あらかじめフォーマットを決めて、プロジェクト運営に必要な各種のドキュメントを作っています。

#### 5. AIをめぐる

質問： さきほど、岸田さんの講演では、AIの効用には否定的なニュアンスでしたが、それはなぜでしょうか。

岸田： 基本的に、ソフトウェア開発は人間の知恵を必要とするむずかしい仕事であり、機械にまかせるわけにはいかないというのが、私の考えです。ただ、長岡での討論でもいいましたが、ソフトウェア・データベースについて、AIまたは知識工学を応用した改善は期待できそうな気がします。

つまり、でき上がったプロダクトの記述だけを入れるのではなく、それが作られてきた時間的経過もきちんと記録できるようなDBがでてくると思います。たとえば、設計時において、いくつかの代替案があった場合、それが、いつ、どんな理由で、それらのうちのどれを選択したのかということが、DBに記録されている。このことが、メンテナンスや再利用をうまくやる上でのポイントになるでしょう。

ここで必要になってくるのは、開発プロセスを構成するさまざまなアクティビティや（中間）成果物の関連を説明するしかるべき数学的モデルであり、それにしたがって、コンピュータが必要な情報を管理してくれる時代が、やがてくると思います。すでにそうした研究は始まっており、その成果が実用化されるようになれば、AIもまんざら捨てたものではないと思います。

安間： 熊谷さんは、AI関係のお仕事をされているということですが、いかがでしょう。

熊谷： AIといっても、範囲が広すぎてはつきりしない。なかにはLispかProrogでプログラムを書けば、それがAIだという人もいますが（笑）。

一般に、コンピュータのカテゴリは、マン・マシン・インタフェイスのよしあしで分けることができると思います。こうした観点からすると、汎用機的环境は遅れていて当然だと思います。もちろん、これまで何十年かの汎用機の歴史のなかでは、それなりの進化はありましたが、最近起こっている変化は、むしろ革命といったほうがよいくらいで、そのことはInterlispやSmalltalkの環境を使っていると、はっきりわかります。

これからは、フロントエンドの仕事は、そうした高性

能のワークステーションで行い、処理速度が要求される仕事をバックグラウンドの汎用機で行うようにすれば、よいと思います。

質問： 熊谷さんが開発されたLIPSでは、Lisp, Ingres, Prolog, Smalltalkというそれぞれのコンポーネントは、1台のマシンの上で動くのでしょうか。盛田さんがさっきおっしゃったように、ネットワーク内にCPUを分散させておいて、必要な時にアクセスするということではいけないのでしょうか。

熊谷： いいえ、1台のマシンに統合化されています。ひとつの理由は、名前をLIPSとつけてしまったので、分割のしようがない（笑）。

ひとつのジョブは、ふつういくつかのプログラム・コンポーネントから成り立っています。ある部分はLispで書く那么简单で、ある部分はSmalltalkが便利だということがあるでしょう。そうしたマルチ・ランゲージの開発をなんとか支援したい。それには、1つのタスクの中から、自由にLispやSmalltalkを呼べる環境があればよい、という単純な発想で作られたものです（笑）。

CPUを分散してしまうと、マルチタスクで動くことになり、それでは既存の環境と何も変わらなくなりますので、ひとつのマシン上で動くようにしたわけです。

#### 6. おわりに

安間： さて、時間もなくなってきましたので、最後に、第1回SEAワークショップの感想をお願いします。

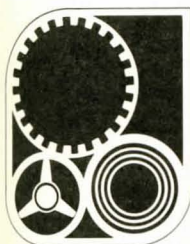
熊谷： メーカ、ソフトハウス、エンドユーザなどのさまざまな方と立場を越えて討論ができました。これから、おのおのの果たす役割りを、SEAを通して、討論できればよいと思います。

盛田： 組織や会社の壁を越えて、なかなか社内では相談出来ないような疑問とか問題をぶつける場として、また、ふつうでは手に入らないノウハウを得る場として、SEAのワークショップはきわめて貴重だと感じました。

安間： これからもいろいろな問題解決の場としてSEAを活用していただければと思っています。今後、どのようなテーマでワークショップを開けばよいか、アンケート等を通じて、みなさんの御意見をお聞かせ下さい。本日はお忙しいなか、熱心に討論に参加くださって、どうもありがとうございました（拍手）。

（レポータ：関崎邦夫）

# CALL FOR PAPERS



**9<sup>TH</sup> INTERNATIONAL  
CONFERENCE ON  
SOFTWARE  
ENGINEERING**

## FORMALIZING AND AUTOMATING THE SOFTWARE PROCESS

MONTEREY, CALIFORNIA, USA    30 March-2 April 1987

### CHAIRMAN

**William E. Riddle**

software design & analysis, inc.  
PO Box 3521  
Boulder, CO 80303 USA

### PROGRAM CHAIRMEN

**Robert Balzer**

Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, CA 90292 USA

**Kouichi Kishida**

Software Research Associates, Inc.  
1-1-1, Hirakawa-cho, Chiyoda-ku  
Tokyo 102 JAPAN

### TOOLS FAIR CHAIRMEN

**Larry E. Druffel**

Rational  
1501 Salado Drive  
Mountain View, CA 94043 USA

**Jack C. Wileden**

Computer and Information Sciences  
University of Massachusetts  
Amherst, MASS 01003 USA

### TUTORIALS CHAIRMAN

**Richard Fairley**

Wang Institute  
Tyng Road  
Tyngsboro, MASS 01879 USA

### LOCAL ARRANGEMENTS CHAIRMAN

**William M. Murray**

General Dynamics Corporation  
12101 Woodcrest Executive Drive  
St. Louis, MO 63141 USA

**Theme:** Over the last decade, two important myths have gradually been discarded: that the waterfall model describes the software lifecycle and that code is the only product of that cycle. Significant systems arise by a evolutionary process rather than by implementing a carefully constructed specification in a single sequence of development phases. Code defines execution but provides an inadequate basis for understanding, and hence, maintaining and/or evolving, the implemented system.

These insights have shifted attention toward paradigms which recognize software development as an iterative design (development) process and techniques for recording the process itself to document and understand the resulting implementation. Formalizing the intellectual activity provides the basis for a new generation of development tools which automate and/or effectively support portions of the design process.

**Objectives:** The main objective of the conference will be presentation and discussion of progress in formalizing and automating the software process through provision of better models, methods, and tools, and identification of critical issues to be addressed. A secondary objective is to broaden participation beyond traditional Software Engineering to include the Artificial Intelligence and Database technologies needed to model, store, manipulate, access, reason about, and automate portions of the software process. The final objective is to provide wider access to the best work in this expanded field by republishing outstanding papers, summarizing important specialized workshops, and presenting summaries of recent advances in particular areas.

**Original Papers:** Authors are encouraged to submit papers that directly address the formalization and automation of the software process. Theoretical papers should indicate applicability. Papers about new developments should separate achievement from plans and evaluate usage. Papers reporting practical experience should include sound empirical evidence. Authors should be advised that conference time allocated to the newly incorporated workshop summaries, "Recent Advances In", and republished outstanding papers, will substantially reduce the number of accepted papers.

**Tools Fair:** A Software Tools Fair will be held in parallel with the conference to provide conference attendees with information about current software tools. Both experimental and commercial software will be demonstrated. In addition, the conference will include a special, separate track featuring presentation and demonstration of tools selected by the tools fair committee. Those interested in exhibiting in the tools fair, and especially authors interested in presenting a paper describing practice and experience with a particular tool in conjunction with a demonstration, should contact one of the Tools Fair chairmen.

**Submission of Papers:** Four copies (in English) should be submitted by 1 September 1986 to either Program Chair. Papers should be no longer than 6000 words. Full-page figures should be counted as 300 words. The paper should include a short abstract and a list of keywords indicating subject classification. Notification of acceptance will be sent by 1 December 1986. Camera-ready copy of the final version will be due 1 February 1987.

**Further Information:** For further information and/or a copy of the advance program when available, write to 9ICSE, c/o IEEE Computer Society, 1730 Massachusetts Ave., N.W., Washington, D.C. 20036 USA.

### SPONSORED BY:



ACM SIGSOFT



IEEE COMPUTER SOCIETY

### IMPORTANT DATES

|                          |                   |
|--------------------------|-------------------|
| Submission Deadline:     | September 1, 1986 |
| Acceptance Notification: | December 1, 1986  |
| Final Versions Due:      | February 1, 1987  |



THE INSTITUTE OF ELECTRICAL  
AND ELECTRONICS ENGINEERS, INC.

## 長岡の雪に埋もれて

酒匂 寛

ソフトウェア・リサーチ・アソシエイツ

### 1. 雪の東京から雪の長岡へ

去る2月19日から22日の4日間、新潟県長岡市のホテル・ニューオータニ長岡で開かれた第1回のSEAワークショップに参加した。前日、東京地方は珍しい大雪で、出発まえから雪見酒という不心得者もいたようだが、あいにく私自身はそんなに暇ではなかった。

ワークショップのテーマは、「実践的開発環境について」であった。開発環境というコトバはここ数年の流行語であるが、どちらかといえば、コトバだけが先行して、その割に実体がともなっていなかったきらいがある。そこで、「実践的」という形容詞をかぶせてみたら、というのが、企画を立てられた方々の発想らしい。

考えてみれば、これはなかなか皮肉なネーミングである。そもそも、「実践的」でない開発環境などありえない（というより意味がない）はずではないのか。もしそれが、われわれソフトウェア技術者をとりまく状況を象徴しているのだとしたら、私たちは、まだまだ、五里夢中の世界にとりのこされているのだ（そのうち、実践的ソフトウェア・エンジニアリングなどとだれかがいい出すのではないかと、内心あやしんだりしている）。

### 2. 雪見酒のあいまのできごと

さすがに東京のそれとは比較にならない豪快な雪（でも地元の人たちは、こんなのは雪のうちにはいらないうっていった）とおいしいお酒（ツブ生はよかったですね、熊谷さん！）に応援されて、ワークショップのプログラムは、つつがなく進行した。

われわれのグループは、リーダーのコーディネーションがよかったためか、それともメンバーのパーソナリティが符合したせいかわからないが、非常に刺激的な討論と相成った。

最初は、現在実用に供せられている環境やツールをすべて列挙し、それらの長所や欠点を比較・分析して、あるべき姿を探ろうなどという（大胆きわまりない）試みが開始されたのだが、「いったい、だれのため、なんのための環境か？」という疑問が提出されたあたりから、話は2転3転しはじめ、「結局、万人に奉仕する環境の構築は、構築者個人の満足できるレベルからしか始められない」とか、「人類愛が必要である」とか、「教養の

ない人は環境について語ってはいけない」とか、ほとんど哲学的とも呼べる段階にまで到達した（ああ、ますます実践から遠ざかって行く・・・）。

また、大型機上の「一応は環境と呼ばれているもの」の不自由さについての議論の最中に、ひとり我が道を行くバロース・アーキテクチャの話聞き、非常に感銘を受けたりした。その話は、ユーザ・インタフェースの議論というよりは、もっと一般的な、ユーザ・フレンドリネスに関するいくつかの重要な示唆を含んでいるように思われた。

といった具合で、こんどのグループ討論は、かつて経験した同種のもののうちで、もっとも楽しく、かつ有益なものであった。

最終日の全体討論のなかで出た話題のなかで一番多かったのは、なぜかといおうか、やはりといおうか、ドキュメントの作成・管理および教育や技術移転に関することがらであった。私にとってはもはや日常と化したUNIXによるオンライン・ドキュメント管理は、世間一般の目から見ると、まだSFの領域に属しているらしい。しかし、それは程無く常識となるのだ。これからが、真剣に考えなければならない時機である。

最後に、COBOLをどうするのかという質問が提出されたが、事態はもはや言語のレベルでどうこうできる状況ではないということで、ほぼ参加者全員の同意がえられたように感じられた。

### 3. 終わりに

全体に、自由な雰囲気になりあふれたワークショップであった。これもSEAという場のおかげであろう。SEAについて思うところはあるが、それはまた別の機会に。

[追記]

ワークショップのあと、私は長岡市営スキー場で遊びたいという有志に随行して行って、ロッジでストーブにあたりながら、J. P. ホーガンの「ガニメデの優しい巨人」を読んでいた。スキー場でそんな酔狂なことをしているのは、どうやら私1人のようであった。そして、私は酔狂が好きなのである。

## 飲み屋の塩

高波治夫

ピー・エス・アイ

料理屋とか飲み屋の入口に、きれいに塩を盛ってあるところがある。子供の頃から不思議に思っていた。よく、お葬式の帰りに、玄関で背中に塩をかけてもらい、不幸をしょいこまないようにする迷信(?)がある。あの「お清め」の意味かな、となんとなく考えていた。

酒を飲むようになってからは、風雅なものだとは感じつつも、一方で、「塩というのは、専売するくらいに、なくてはならないものだ。それを食用に供さず、飾りで使い、一晩で捨ててしまう、もったいないことだ」と、ささやかな疑問を抱きつつ、長年、飲み屋の「のれん」をくぐってきた。

つい先日、読み直した本がある。一度読んだら、絶対に読み直さない人もいるが、私は、暇なせいもあり、古い本をよく読み直す。安あがりだから、親から受け継いだ性格を、へんなところで感謝したりしている。

ところで、その読み直した本とは、陳舜臣の「小説十八史略」である。私にとって、世の中の評価とは逆に、十八史略は面白い本の部類に入り、「史記」はつまらない方に入っている。その十八史略の中に、「塩」のエピソードが載っていたのである。なんともお恥ずかしい話だが、前回までは読み飛ばしていたようだ。

晋の時代というから、かの有名な三国志の次の時代である。晋の礎は、あの曹操の部下として、諸葛孔明と争った司馬仲達が築いた。晋の時代になってから、後漢末から分裂していた中国が再統一されたのである。呉を降して統一が成ると、晋の武帝(司馬炎)がまっさきに命じたそう。『江南には美女が多いときく。五千の美女を選んで、後宮に入れよ』

こんな調子だから、武帝の後宮は、美女の数が一万をこえ、その一人ひとりの顔を覚えられないわけがない。「今夜はどの女のところに泊まるかな?」彼は不精者であつたらしく、迷路遊びなどに惚けず、後宮の中を羊の引く車に乗ってゆき、羊が止まったところの部屋に入ることにしたらしい。

皇帝の寵愛を受けようとした頭のよい女性が、自分の部屋の戸に竹の葉をさし、戸の前の地面に塩を盛った。竹の葉や塩は羊の好物だったので、必ずそこで車が止ま

るという仕掛けだ。水商売の店の前に塩を盛る習慣は、このエピソードに由来するらしい。

風雅におもえても、やはり縁起をかついでいたのである。馬の好物が砂糖とは知っていたが、羊の好物が塩とは知りませなんだ。

よく考えてみると、「縁起」だけではないのかもしれない。当初は縁起だったかもしれないが、ほかの利点もあったからこそ、1500年以上たった今でも、続いているのではないだろうか。一つには、黒い石の三和土に真っ白い塩が盛ってあるのは、美しい。二つに、塩は商売で使うものだから、いつでもある。三つに、儀式をすることで、身がひきしまる。四つには、塩を洗い流すと、塩水がまわりに散るから、雑草が生えない。等などをあげることができよう。

前置きが長くなってしまった。塩、つまり「ソルト」と「ソフト」、違いはルとフだけだ(ちなみに、英語では、SALTとSOFTで2文字違う)。「フ」をひっくりかえして90度回転させ、となりに揚子か何か棒を一本つけたのが「ル」と思えば、たいして違いはあるまい(こじつけか?)。

ソフトウェアをソルトウェアといいかえると、なんだか、水をかけたら溶けてしまいそうな、感じがする。われわれの作ってきたプログラムは、観念的には、変更自在の「ソフトウェア」であるが、その実、ちょっと問題が起こると解体してしまう「ソルトウェア」だったのかもしれない。このように自虐的に考えてしまうのは、かのトインビーがいう「後退の時代」に、私自身あたっているのだろうか。

「後退の時代」が、真の「ソフトウェア」を創るために避けることのできない、暗い手さぐりのパスならば、それを克服するのが、一度足をつつこんでしまった者の務めであるといえよう。

しかし、とみに白髪が目立つようになってきたゴマ「塩」頭に嘆きながらも、会社の人間には「近頃、技術屋が減って、経営者が多くなり過ぎた」などとうそぶかれつつ、「あたしや、文科系だから、しああんめ!」と、ケツをまくってごまかしている毎日である。

## 田舎のコンピュータ

芝原雄二

協同システム開発

私 : Gogaxtu no yasun ni kodon nu turete kaextude na.

「5月の休みに、子供を連れて帰るよ」

おやじ : Kuukou ni mukeke ixtuga?

「空港に迎えに行こうか？」

私 : Sogen kotu sende yokaga. Soi yori kodon ni ui to buta wo misextu kuiyai.

「そういうことはなくていいから、子供に牛と豚をみせてやってください」

おやじ : Kodonnnya ui to buta wo misuxtuga, wai nia konpyuuta wo misuxtudo.

「子供には牛と豚をみせてあげるが、おまえには、コンピュータを見せてあげよう」

私 : Konpyuuta!? Ittai Konpyuuta ti nai sasutto na?

「コンピュータ？ いったいコンピュータに、何をさせているの？」

おやじ : Naiden kaiden sigoxtu sasutto yo.

「いろいろな仕事をさせている」

私 : Naiden kaiden ti nai na? Titto misextu kurun na?

「いろいろって？ ちょっと見せてくれる？」

おやじ : Yoka do.

「いいよ」

ということで、ひさびさに鹿児島に帰省する機会に、おやじの勤めている、川辺地区農業共済組合に導入されたコンピュータを見せてもらうことにした。鹿児島に帰った次の日に、早速、コンピュータを見せてくれた。

私 : Doke atto na?

「どこにあるの？」

おやじ : Kon oke yatyo.

「この奥に置いてある」

私 : Dogen sutto na?

「どうやって動かすの？」

おやじ : Oi ga tukote misurai.

「私が使ってみせよう」

TOSBACシステム45というミニ・コン(CRT3台, PR3台)を、おやじ自らが、オペレータとなって、デモしてくれた。

私 : Fun no kote dekutto na?

「ほんとうにできるの？」

おやじ : Koi gamen ni iruxtu deke yo.

「この画面にしたがって、入力するだけだからね」

私 : Koi koumon no zunbu dekutto na?

「この項目の全部ができるの？」

おやじ : Tigo. Zenbu wa dekityoran.

「違う、まだ全部はできないね」

私 : Koi songai ti nai na?

「この損害っていうのは何？」

おやじ : Suitou, tatemon yara ironna attoyo.

「水稻、建物などさまざまある」

58才のおやじが、自分でコンピュータを操作するのを見て、内心ではよくやるなと思っていたが、夜、若い獣医さんたちをふくめた農業共済組合の人たちと酒を飲んだとき(酒といってもこちらでは焼酎である)、そのへんを聞いてみた。

私 : Ai masin na wake sya tukowanto na?

「あのマシンは、若い人は使っていないのですか？」

獣医 : Jimuin no si ga tukotyoiyo. Omengei oyaxtusan mo haime waapuro wo tukoti sitadon, wakesi ni makasita do.

「事務の人たちが使っている。あなたのおやじさんも、はじめは、ワープロを使おうとしたが、今では若い人にまかせたよ」

私 : Soi kiite, titta ansin sita do.

：「それを聞いて、すこし安心しました」

: Konpyuuta wo irete, raki naxtuta na?

「コンピュータを導入して、仕事は楽になったのですか？」

獣医 : Jimu ga waxtuzee raki naxtuta yo.

「事務が非常に楽になったね」

私 : Sofuto wa wagataxtu de tukutta to na?

「ソフトは、自分たちで作ったのですか？」

獣医 : Fukuoka'n sofuto-kaisya ni sin morota, ansya waxtuzee hamentukunne. Mei wo kuu konmo hikya suru de ya.

「福岡のソフト会社にしてもらった。あのひとたちは、非常にながらね、飯をたべるのもわすれるほどに」

獣医 : Oman mo toukyou de konpyuuta kankei no sigoxtu sityo hannai yaga, tetuya nando suxtuto ya?

「あんたも東京で、コンピュータ関係の仕事をしているそうだが、徹夜なんかはするの？」

私 : Atai ya sen do, Puroguramatyu sigoxtu ga isogasi baxtukai tyu ta, futoka maxtuge yaxtu do. jya don, sigoti hamettu kuto nextu kon mo kuu kon mo hixtukya suru do, sogan kotya dogen sigoxtu den axtu do ga na.

「私はしきいすよ。プログラマという仕事が、急がしいばかりだというのは、おききな誤解です。でも、仕事に熱中すると、寝食をわすれるというのは、どの職種にでもあるんじゃないですか」

獣医 : Sogan yai ne.

「たしかにそうだね」

焼酎を飲みながら、若い獣医さん以外にもいろいろな話を聞いたが、畜産関係についての一般常識を、私が持ちあわせていないため、なかなか理解できなかったが、そうして組合のみなさんは、コンピュータに対して、よい印象を持っているようである。

私 : Iman kinou ni manzoku sityoi na?

「ところで、いまの機能に満足していますか？」

獣医 : Unnya, dekun nara, moo titto ironna kotu yanmoro goyaxtu do.

「いや、できるのなら、もっといろいろやってもらいたいことがあるけど」

今のところ、直接獣医さんのために役立つ、家畜の診断書に関しての業務のソフトウェアは、まだ開発されていない

いそうである。もちろん、やらなくてはならないことが、たくさんあると思うが、こういう業務にたいしては、エキスパート・システムの応用ができる分野と思われる。

それにしても、私の高校時代からすれば、田舎も変わったものである。このようなコンピュータが導入されるなどとは、夢にも思わなかった。でも、若い人の少ない地方でこそ、今後、コンピュータが必要となってくるのではないかと思う。

また、われわれが、こういう地方に、コンピュータを普及させる手助けをするためには、畜産など農業に密着した、広い業務知識も必要とすることが、あらためて再認識させられた。

おやじ : Wai ya kogen ta ikiiki wakiido ga?

「おまえは、こういうのはすぐにわかるだろう？」

私 : 「……………」

こういう、するどいカウンター・パンチの言葉をもらった。おやじのいいぶんでは、おまえは、ソフトウェア会社に勤めているから、コンピュータは、プログラムを見てすぐ理解でき、自分で操作もできるだろう、ということであった。

私はそれにたいして何もいえなかった。というのは、このおやじたちが導入したコンピュータは、私が普段使っているものとは、まったく別のものである。だから、よく理解できないよ、という弁解をしようとした。しかし、それをいっても始まらないと思った。

このことは、ソフトウェアにかかわっている人たちは、何度か経験しているのではないだろうか。自分の毎日やっている仕事を、違う分野の人たちに、十分な説明ができないというのは、すこしさびしい気もする。

そんなわけで、普段職業をきかれると、私は、コンピュータ関係と答えることにしていたのだが、このおやじの言葉は、ずいぶんと私にはきいたようである。

つまり、よく考えると、その弁解は、自分の不勉強を意味しているのではとってしまったのである。せめて、事前に情報をしれいしていけばとおもったのは、あとの祭りであった。まことに genne (恥かしい) 思いであった。

## メモリ・ダンプ解析の自動化！？

小須田正孝

日本コンピュータ研究所

### 1. はじめに

いま、最先端の技術を駆使して、最先端のソフトウェアを作っている人たちがたくさんおられます。しかし、フォーマルな設計仕様にもとづくプログラムの検証や合成は、まだそれほど普及しているわけではなく、昔ながらのメモリ・ダンプを参照しながら最先端ソフトウェアを開発している姿を職場で見たりするとき、何かアンバランスさを感じているのは、私だけでしょうか。

デバック／保守工程でいまだに使われているメモリ・ダンプに着目したツールは、比較的地味であるらしく、論文や発表等あまり見られませんし、一部を除いて製品も少ないように思います。そこで、ほんのお遊び程度に考察してみたいと思います。

### 2. メモリ・ダンプとの付き合い

ソフトウェア危機が話題になってから久しい現在、さまざまなソフトウェア開発ツールが研究され、発表されてきています。さらに、その開発支援ツールが積極的に開発工程に組み込まれ、自動化に向かっての努力が続けられています。ソフトウェア部品やツールの流通促進を意図したSIGMAプロジェクトの成果も、これから大いに期待されるところです。

さて、そのテスト、デバック、および保守工程を支援する自動化ツールの中で、処理装置で実際にプログラムを走らせて中断／再開、動的情報表示、等の支援をするデバッガがあり、また、そのときどきの静的情報をスナップしておくメモリ・ダンプがあるわけですが、前者については、動的にデータを変更させる機能や、トレース機能等の支援サービスを強化することにより、いろいろとよいツールがくふうされています。この傾向は今後も続くでしょう。

しかし、メモリ・ダンプに関しては、あまり自動化がなされておらず、従来から、処理装置に依存したメモリ内容だけが出力表示され、開発者が自らの職人的勘と知識を頼りに、それを解析している場合が多いのではないのでしょうか。

デバッガが今後充実して行けば、メモリ・ダンプは不要になるという期待もあるでしょうし、情報としては、

これほど必要かつ十分なものが他にないからでしょう。

また、メモリ・ダンプを開発の一助としている人間の数が限られており、それぞれの機種ごとに孤立しているので、もしかしたら便利なツールが作られていても、その情報が伝わらないのかもしれない。あるいは、原始的でも最善な方法として職人的に解析しているのが現状なのではないでしょうか。

大型機、小型機を問わず、特にシステムの中核的プログラミングを行う場合には、当面、メモリ・ダンプは必要不可欠でしょう。これから、このダンプ情報とうまく付き合っていくためには、なんらかの自動化を進めて行かねばならないと思います。その夢を描いてみますと……

### 3. 理想のダンプ解析プログラム

デバッガが、プログラムの動作を順プロセスでフォローし、テスト活動を支援していくのに対して、ダンプのなかには、ある不良事象が発生した結果しか残っておらず、その情報から、プログラムの動作を逆プロセスでフォローしなければならないという宿命があります。

であれば、その逆プロセスを行うツールを作ればよいのですが、現実には、そう単純にはいきません。一般に、プログラムは、ある条件を与えれば、順プロセスとしての結果は1通りしかあり得ないのですが、メモリ・ダンプのように結果のみが与えられたときには、逆プロセスをたどると、複数通りの答えが存在するからです。

したがって、複数存在するプロセスをひとつずつ検討するため、全ケースをツールに出力してもらう必要があるでしょう。さらに、このツールには、それぞれのプロセスとその根拠を表示してもらいつつ、妥当か否かをダンプ内の結果で裏づけながら、ひとつずつ消去してもらうことになります（原因が複数存在する場合もあるのはもちろんです）。そして、その都度、裏づけとなる根拠や結果をツールに表示させます。

ここで、このツールには、次のような経験的知識データベースが必要になってきます。

(1) メモリ・ダンプの採取時の現象：ひとくちにダンプといっても、異常状態が発生した時に、オペレー

- ディング・システムが、プログラムの異常を察知して出力した場合のダンプもあれば、システム自身も察知できずにいる異常状態を、スナップ的に採取した場合のダンプもあります。その振り分け方を、あらかじめ知識として入力してやる必要があります。結局、「何が問題／異常か」の判断基準を、あらかじめ、パターン化し設定しておくことです。もちろん、新たな追加や蓄積が可能でなければなりません。
- (2) 原因と根拠、それぞれの裏づけとケース生成／消去時等に用いる過去の関連パターン像、不良歴や関連知識についての情報。
- (3) テスト対象処理プログラムの内部仕様知識：ここで急に、テスト対象プログラム構造に依存した解析知識が必要になり、汎用性がやや遠退きます。ソース・コードからそうした知識を自動的に引き出すことは、また別の難問です。
- (4) その他、上記(1)～(3)に関するメタ知識も必要になります。

ツールの機能的構成は次の通りです。

- ・経験的知識データベース
- ・逆プロセス推論機構
- ・推論裏づけ機構
- ・不良歴／関連知識情報獲得機構
- ・メモリ・ダンプ

#### 4. おわりに

以上、お気づきの通り、古風なメモリ・ダンプに、無理やりAI技術を適用して、今はやりのエキスパート・システムに変身させようという試みです。しかし、このやり方では、たとえツールができたとしても、それを動かす前に必要な知識を与えるのに、大変な労力が必要となるでしょう。

私自身は、いま、職人的(ベテラン)知識を基に、ダンプの中をごく断片的に限られた範囲内でチェック検索できるツールを試みておりますが、方式として文字通り限界があり問題点もあります。

道は遠く、このエキスパート・システムがもしできあがったとして、さらに修正・修復情報も出力表示できることが望ましくなります。こうなると、メモリ・ダンプ内の情報の域をでしてしまうかも知れません。いや、それよりもその頃に、はたしてメモリ・ダンプそのものが存在していますかどうか。

### 秋のセミナー・ウィーク予告

9月3～6日に、機械振興会館(東京)で、秋のセミナー・ウィークを開催します。今回のセミナーは、これまでと少し趣向を変えて、一方通行の講義だけでなく、各講師3時間ずつのセッションのうち、約半分の時間を自由討論にあてることを予定しています。

いわば、次の21人の講師による連続ティーチ・インのスタイルです。

- ☆ネットワーク： 村井 純  
寺本 雅則  
長井 剛一郎
- ☆環境： 岸田 孝一  
熊谷 章  
盛田 政敏
- ☆AI： 玉井 哲雄  
斎藤 信男  
横井 俊夫
- ☆再利用： 大木 幹雄  
水谷 時雄  
村井 進
- ☆設計： 伊野 誠  
臼井 義美  
落水 浩一郎
- ☆管理： 堀江 進  
深瀬 弘恭  
松原 友夫
- ☆その他： 吉村 成弘  
佐野 稔  
辻 淳二

開催要領の詳細は、本誌8月号(7月末発行予定)で、上記各講師のポジション・ステートメントとともにアナウンスします。御期待ください。

(セミナー委員会)

## 「若手の会」ワークショップへの招待

堀江 進

日本電気ソフトウェア

### 1. はじめに

4回目の夏のプログラミング・ワークショップ（略称：若手の会 in 盛岡）が、「プロジェクト・プランニングのケース・スタディー主としてテクニカル・マネージメントの立場から」をテーマとして、今年も9月中旬に開催されます（詳細は35ページを参照）。

はじめてこの募集要項を見られる方は、「えっ、盛岡でワークショップ!?」とふしぎに思われるかもしれませんが、旧ソフト協時代以来の実績は、輝かしいものがあります。

### 2. どうして好評、若手の会?

このワークショップは、同じ時期、同じ場所で、これまでに3回開催されており、今年で4回目をむかえます。これまでの開催年月、基調テーマ、参加者数、チェアマンは、次の通りでした。

第1回（58年8月）、特定テーマなし、47名、岸田孝一

第2回（59年8月）、もしも私がプロジェクト・リーダーになったら、50名、野村敏次

第3回（60年9月）、これからのソフトウェア開発を考える、61名、三浦信之

スタッフを含む参加者の合計は、延べ158名にのびます。このように好評な理由としては、次の3点があげられます。

- (1) 参加者の問題意識が高く、しかも、若くて熱意がある。
- (2) 若い人を育てようというプログラム委員、事務局の人たちの努力と熱意がある。
- (3) 企業の若手に対する期待が大きい。

ここで、注意しておかなければならないことがあります。応募者の全員が参加できるのでなく、過去3回の実績では、そのおおよそ半数しか、参加できないことです。

応募者は、自分自身の経験をふまえた具体的な意見や提案をまとめたレポート（これを、ポジション・ペーパーといいます）を、申込のさい提出する必要があります。このレポートを、プログラム委員たちが審査し、意見や

提案が明確な人や発想がユニークな人を選び出して、参加者を決定します。

だから当然、全体会議やグループ討議は、白熱したものになります。さらに、何ものにもかえがたい若さがありますから、フリーの深夜討議（サンドウィッチ、酒、つまみ等つき）は、盛岡の静寂とは不釣り合いなほど、盛り上がります。

そのため、決して若くはないチェアマンやプログラム委員たちは、睡魔（酔魔、吸魔）との戦いで、つくずくと若い人との体力差を感じるそうです。今回のプログラム委員には、このような状況にめっぽう強い業界第1線で活躍中の「熱中型」のプレーイング・マネージャが集まりました。

### 3. なぜ盛岡か?

ワークショップのロケーションに必要な条件は、日常の業務や喧騒からはなれ、フレッシュで、爽快な気分にあふれることです。さらに、空気、食べ物、そして、酒がこのうえなくおいしいこと。盛岡が、これらの条件をすべて満たしていることは、過去のワークショップで証明済みです。

### 4. どうしてテーマに「テクニカル」がつくのか?

今回のテーマ設定には、2回目のワークショップ「もしも私がプロジェクト・リーダーになったら」での反省が生かされています。プロジェクト管理というテーマは、多くの人たちの共通の、しかも切実で大きな問題です。

2回目のワークショップでは、討議もさまざまな立場から、参考になるおもしろい意見がたくさんでています。しかし、対象範囲があまりにも広すぎて、内容の掘り下げが、いまひとつ足りませんでした。

- ・ユーザの現地での開発のため、マシンタイムが夜しかとれず
- ・実はコンパイラにバグがあり
- ・チームのメンバは若い人がおおく、どうしても経験不足で
- ・etc

論議のたどりつく先は、このような環境、体制、そしてスキル問題から、根性論、動機づけなどの精神論でした。

今回は、討議がこのような方向に展開されないようにするために、いろいろな工夫を凝らすことを考えています。その1つがこの「テクニカル・マネジメント」という副題です。

先に述べた3つの状況などは、プロジェクトの当初からわかっていたはずで、私たちがしなければならないことは、与えられた仕事を、与えられた期間で、与えられたメンバーで、与えられた状況のなかで、いかにうまく遂行するかということです。このような立場から、技術的にどうすれば、プロジェクトが成功するかを考えてみたいと思います。

見える管理、量と質のモニタリング、定量的進捗管理などのキーワードが浮かんでくるのではないのでしょうか。プロジェクト管理をどうシステム化できるかについても、討議する必要があるでしょう。

## 5. ケース・スタディあれこれ

このようなことから、いくつかのケースを設定し、それにたいしてどう取り組み、解決するかを、討議したいと考えています。たとえば、...

### (1) ケース1：ユーザ戦略

先日、大病院の受付業務が電算化された。サービス本番当日に出会った私は、受付で2時間以上も待たされたが、受診するためのカードを受け取ることができなかった。受付に並ぶ患者の列は、ずーと向こうの...

割り込みや例外の多い受付業務（サービス業務）と、一般の事務業務では、電算化する場合のアプローチが違ふことを、このシステムをつくったSEたちは知っていたのだろうか？ユーザの要求を聞き出し、仕様変更を少なくする技術はどうだったろうか？あなたなら、どうアプローチしますか？

### (2) ケース2：新規開発プロジェクトの戦略

リーダーになるのはある日突然です。なんの心の準備もなく、辞令ひとつで、メンバからリーダーへ変わるのです。そんなあなたが担当するのは、新規開発プロジェクトで、担当者9名全員がはじめてのメンバで、しかも、あなたにとって初めての分野のシステムです。

さて、メンバのスキルを把握し、プロジェクトを成功させるためには、どう攻めますか？

### (3) ケース3：改造プロジェクトの戦略

既存プログラムの改造という仕事は、新規開発の仕事にくらべると、人気がなく見栄えがしません。しかし、システマティックな開発をする人と、そうでない人に、この仕事を任せた場合、いろいろな意味で大きな差がきます。期間、改造量、バグ件数だけでなく、メンバのスキル、チームのパワーなどにもその差がでます。この違いは、いつもチームで動くか、それとも担当者に仕事を分割するだけで個人任せにするかの違いです。

さて、5年前に作られた50Kのプログラムの改造を任された。改造後も、5年間は毎年拡張することが予定されている。当然（！！）のごとく、ドキュメントは、マニュアルとリストだけしかない。しかも、初めての機種で、初めての言語です。5人の若いメンバと一緒に頑張ってどう攻めますか？

### (4) ケース4：ファイアマンの戦略

ソフトウェアの仕事に限りませんが、仕事というものは、問題なく終わってあたりまえです。ところで、一旦、火がついたプロジェクトを、いかに早く消し、立て直すかも立派な技術です。

さて本題は、納期を2ヶ月後に控えているにもかかわらず、まだ、プログラムはビクッともしないプロジェクトがあります。ファイアマンとして、あなたが任命されました。さて、どのような情報を収集して、どのように動きますか？

### (5) ケース5：だいじょうぶかい？への戦略

よくある話から。上司がリーダーをつかまえて話しています。「あと2ヶ月だが、XXXかね？」XXXの部分には、「だいじょうぶ」、「どう」とかがよく使われます。リーダーの回答のパターンは、「あと少しですから、頑張りますよ」、「ええ、まあなんとか」。多くの場合は、このようなやりとりで終わります。

さて、納入を1ヶ月後に控え、コーディングも完了し、一部のプログラムも動いている。「どう、だいじょうぶ？」という上司の問いにたいして、「かくかくしかじか、こうこうだから、だいじょうぶ」と証明し、上司を安心させ、納得させることができますか？

#### 【念のため】

このワークショップは「若手の会」であり、一般参加者は物理的年令30才以下に限られています。もし、あなた自身が、残念ながらこの制限にひっかかる場合は、周囲の若手を勧誘してください。よろしく。

## 現状における再利用支援ツール

阿部正平

協同システム開発

### 1. はじめに

この原稿は、再利用分科会の討論会で発表したものに、加筆したものである。

ソフトウェア開発における生産性ならびに品質の向上を目的として、これまでに種々の方法論とともに、多数の支援ツールがつけられている。

一般的に、製品の開発において、再利用が容易にかつ大幅に可能となれば、生産性や質の向上は明白である。ソフトウェアの開発も現在では、新規開発よりも保守が大部分を占めるようになり、そこで、「既存の対象物（プログラム等）を再利用」、あるいは「再利用しやすいものとして作り再利用をはかる」ということのための技術ならびにツール化が、現実的なアプローチとして重要視されている。

討論会で発表したのは、「現状における再利用支援ツール」と題して、各種開発支援ツールのうち、特に再利用を意識しているものを選び、共通した考え方、ツールの体系、今後の方向性などを示した。そのさいには、ツールを中心とした分類をおこなったが、本来の目的は、「再利用技術」にあり、したがって、再利用の対象、やり方（何をどう再利用するのか、再利用の基準は何か、現状でのツールはどこまでカバーしているかなど）といった観点からまとめ直した。

### 2. 再利用の対象

ソフトウェア開発という作業は、実際の製造にかかる作業と、それを効率よく進めるための管理作業に分けられる。これらの各々において再利用を考えることができるが、ここでは主として前者で考える。

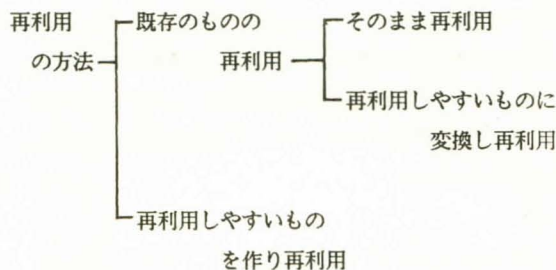
ソフトウェア製造において、再利用の対象となるものには、機能、機能の実現手段（アルゴリズム）、各種データなどである。

これらは、プログラムあるいはデータ・ファイルとして表現され、紙あるいはディスク等のコンピュータ記憶媒体という形態をとっている。実際に再利用をおこなうさいには、対象物が何であるか、どのような形態であるかが、再利用の容易性などにたいするおきかな要因とな

る。

### 3. 再利用の方法

前述のように再利用の方法には、「既存のものを再利用」と「再利用しやすいものを作りそれを再利用する」の2通りに大別される。



#### 3.1 そのまま再利用

われわれのまわりには、過去に開発したプログラムやデータが大量に存在する。これらの中には、再利用するに値しないもの、できるかどうか不明のものなどいろいろあるが、何とかして再利用したい（利用効果が明白なもの）と思われるものも少なくない。

そのようなものを再利用するための機能として、次のようなものがある。

#### ◆複写（コピー）

もっとも単純な再利用である。

#### ◆変換

- ・対象物を異機種／異OS／他プログラムへの移行のための変換（データの互換性が重要）
  - ・他の言語への変換
    - ソース・プログラムの変換
    - オブジェクト・コードの変換
  - ・形態の変換
- たとえば、紙上のドキュメントをイメージ入力（文字／

イメージ認識)によってEDPファイルへ直接変換

#### ◆編集

対象物の編集であり、例えば、ドキュメントのワード・プロセッシング、プログラムのエディティングなどである。

このような機能を実現しているツールとしては、OA機器やOAツール(コピー・ツール/ワープロ/コンパクタなど)と呼ばれるものに多い。

### 3.2 再利用しやすいものへの変換

既存のものにたいして、ある基準(たとえば、部品化など)をもうけて判断し、再利用しやすいものに変えていく。

現状では、この判断を人間が行うしかない。ただし、変換を支援するといったレベルではSEA/Iの部品生成機能などが該当する。

### 3.3 再利用しやすいものの製造

再利用しやすいものを作ってから、それらをうまく再利用するために、まず、次のような再利用しやすいものを作るための機能が考えられる。

#### ◆標準化

対象物の作り方、表現方法などを標準化(設計標準、プログラミング標準などの設定)することによって、できあがったものを再利用しやすくする。

ハード・メーカ等が提供している開発支援ツールなども、もともとは開発手法等の標準化(たとえば、STEPS, SPDS, SDEMなどをもとに、SEA/I, EAGLE, ADAMSといった支援ツール化)ベースとしている。

#### ◆部品化

ソフトウェアを部品化し、データベース化して再利用をはかる。新たなソフトウェアは大部分を、その組み合わせで作るという方法である。要求仕様、設計仕様、プログラムに対する部品化が考えられるが、現状では要求仕様レベルの部品化は遅れている。現状での部品は、過去の経験から次のようなものに分けられる。

・パターン部品: 骨組みをパターン化

・モジュール部品: 主として機能などの汎用/専用モジュール化

部品化では、部品の性質/尺度を決めること、ソフトウェアを部品に分割/抽象化の基準を決めることが必要である。現状では、このような課題があるため、部品化ベースの開発支援ツールは多数作られているが、ユーザ側に部品を作らせるよりも、ツール内部に標準部品を内蔵して、ユーザにこれらを利用させているもののおおい。支援ツールの例としては、標準化のところで示したツールが該当する。

#### (1) ツールの傾向

##### ◎支援範囲の拡大

ライフサイクル全体の支援

より上位フェーズへの支援(要求仕様など)

ホスト・マシンの拡大(汎用機からパソコンまで)

##### ◎自動合成

要求定義からソース自動生成へ

##### ◎知識ベース化

##### ◎視覚化

日本語

グラフィックス

##### ◎利用者別の支援化

(例)

専門家向け SEA/I本体

エンド・ユーザ向け SEA/Iアプリカ

#### (2) 部品ベース・ツール調査のまとめ

##### ◎標準部品

スケルトン: 約20種

モジュール: 100-200(汎用/分野個別)

##### ◎部品生成・修正

新規作成 エディタのレベル

既存SWの 構文エディタのレベル

モジュール部品化

##### ◎部品管理

部品登録(部品名などをキー)

部品検索

部品合成

ディクショナリ

##### ◎支援範囲

製造（プログラミング）フェーズを中心に  
設計→検査（含む、デバッグ）→（要求仕様）  
管理

#### ◎自動化

ドキュメント自動生成（ソース→設計書）  
入出力画面・帳票の視覚的な定義から  
ソースの一部を自動生成  
穴埋め方式によるソース・プログラムの半自動合成

#### ◆言語の高級化

言語を高級化することによって、より抽象度の高い機能などを実現することができ、再利用しやすいものを作ることができる。

再利用のメカニズムを言語処理プロセッサの中に内蔵することで、内部的にも下位レベルの命令スケルトン群が再利用される。とくに、簡易言語などはノンプログラムを目指しており、作られたものが再利用しやすいだけでなく、容易に作成／変更ができるという効果もおおきい。

更に、再利用を容易化するために、ワープロ、簡易言語などのOAツールの傾向として、データベース、ネットワークなどとの統合化（ファミリー型、一体型など）が進んでいる。

### 3.4 再利用機能

再利用しやすいものが出来上がれば、これらを多数蓄積し、再利用するための次のような機能が必要である。

#### ◆登録／管理

対象物を登録しておくための管理機能、すなわちデータベース管理機能である。かつては、単なるライブラリ化程度のものであったが、現状では、リレーショナル型を中心としたデータベース管理システムに変わっている。そして、段々と知識ベース化の傾向にある。

#### ◆抽出／検索

管理と同様に、データベース管理機能の一部である。しかし、再利用の目的で抽出／検索を行うためには、検索キーにたいして知的さ（たとえば、あいまい参照など）が必要である。

#### ◆加工

目的とするものを作り上げるための加工機能としては、合成／編集などがある。合成に関しては自動化が急速に進められている。利用者の意識的な再利用から、システムの内部に再利用の機構を装備し、無意識のうちに自動的に再利用する方法である。

- ・ 要求仕様書→設計書
- ・ 設計書→ソースプログラム

とくに、現状では部品ベースによるプログラム（主としてソースプログラム）の自動合成をおこなうものが大部分である。また、これらのツールは、ユーザとのインタフェースの高度化（グラフィックスを用いた視覚性向上／簡易操作性など）も、ハードウェアの高度化とともに図られつつある。

### 4. おわりに

近年において、ハードウェア技術の進歩はめざましいものがあり、短期間（数ヶ月～1年位）のサイクルで、高性能、かつ低価格の新しいハードウェアが、次々と製品開発されている。

そのために、ハードウェア主導型となり、ソフトウェア開発者はそれらハードウェアに対応したソフトウェアを、せかされるように開発する状況にある。しかし、このような状況に対して、ソフトウェアの再利用技術が確立され、必要なツール等が整備されれば、もっとも効果的な威力を発揮するであろう。

SIGREUSEの部会では、このようなソフトウェア再利用技術を開発の現場において、有効活用できるような具体的な形に整理していきたいと考えている。

## 書 評

## ソフトウェア開発支援環境

-UNIX から AI アプローチまで-

ディヴィッド・R・バーストウ／ハワード・E・シュロープ／エリック・サンドウェル共編 岡田正志監訳  
TBS出版会、定価13,800円

ソフトウェアの開発も新たな転換期をむかえようとしている。グラフィックスやネットワークを利用したワークステーションによる開発の浸透、部品化によるプログラムの再利用可能性の増大、仕様の未確定や変更にも対処できるインクリメンタルな開発技法の研究の進展、AI的アプローチの採用による矛盾のないシステム作りや、作業負荷の軽減への挑戦などは、その兆候となるものである。

このような話題をあつかい、過去10年間に発表された論文のなかで、特に重要と思われるものを28編抽出し、加筆、修正などを加え、5つの構成に整理して、全体を統一したのが本書である。

第1編は、対話型開発への動機づけと特徴、第2編は、特定の言語や状況に対応した環境、第3編は、さまざまな環境の問題点の考察、第4編は、人工知能の応用、第5編は、開発環境の展望という展開になっており、たんなる論文集でなく、読み物としての配慮がなされている。

ここに収録された論文は、現在進められているソフトウェア開発環境（広義のプログラミング環境）についての研究、および開発プロジェクトの思想や概念の基礎となるものであり、ソフトウェア開発上の問題点を解決する糸口を示すものである。

原著はたいへん好評を得ており、環境について一通りの知識を得る格好の書として位置づけられている。少なくとも本書にかかれた項目を理解せずに環境を語ることができないと述べてもいいのではないだろう。

ソフトウェア開発環境の構築には地道な努力と経験や実験にもとづいたノウハウの蓄積という膨大な投資が必要である。しかし、世界の主要な研究の成果を把握し、その実績を土台として環境の展開をはかることができれば、非常に有効である。本書に収められた考察の重要なポイントのいくつかを紹介する。

- ・コンピュータ・アプリケーションの多くは、完全なソフトウェア仕様を作るほど十分に理解されていない。思考探索型による開発は、従来の開発方法論よりも柔軟な設計ができるので、この状況には向いている。
- ・ソフトウェア・システムを実現するための言語について検討する場合、単に言語そのものについて考えるよりも、むしろ対話型システムとしての開発支援環境について検討することが重要である。
- ・プログラムは単なるテキストではなく構造をもっており、開発支援環境で支援すべきである。典型的な例としてはプログラムの構文構造があり、これは対話型開発支援環境の構文エディタで利用できる。
- ・グラフィックの先進的な機能は、ユーザ・インタフェースに非常に有益である。このことは、画面エディタやパーソナル・ワークステーションの高解像度グラフィックス機器に反映されている。
- ・対話型開発支援環境は、開発過程の種々の面を支援すべきである。この点については、Interlispのプログラマーズ・アシスタントがよい例といえる。
- ・人工知能は、プログラマに対する支援を非常に強力にする可能性を秘めている。
- ・ソフトウェアの開発は、新しいプログラムを作るのではなく、すでにあるものを修正する作業になる。

本書は、これらの有益な示唆を詳細に論じ、現状をふまえて、今後のソフトウェア開発を予見した好書である。

(編集部)

## お知らせ

SEAMAILの1号、2号、3号、4号の在庫が若干あります。  
ご希望のかたは、1冊につき700円  
(送料込み)の切手を同封のうえ、事務局までお申し込みください。

## 第1回総会報告

ソフトウェア技術者協会としての、第1回総会が、5月17日の午後2時から東京の機械振興会館の地下3階会議室で、41名の会員が出席して、午後4時までの2時間、熱心におこなわれた。

### 1. 出席

当日の出席者は41名であった。

また、事務局に届けられた委任状は、白紙委任52名、議長委任30名、個人委任9名、計91名であり、これによって第1回総会が成立していることが、報告された。

### 2. 60年度活動報告

5月8日時点での会員状況が報告された。この数字は、25ページの幹事会報告を参照されたい。

85年9月の設立準備期間から、86年3月までの活動が報告された。

### 3. 60年度収支報告

60年度の収支報告が以下のようなされ、公認会計士である吉村会計監事の報告書が代読され、承認された。

#### 収支合計

|        |           |
|--------|-----------|
| 収入合計   | 2,494,184 |
| 支出合計   | 2,430,388 |
| 61年度繰越 | 63,796    |

#### 貸借対照表

|     |         |      |         |
|-----|---------|------|---------|
| 現預金 | 63,796  | 正味財産 | 383,796 |
| 敷金  | 320,000 |      |         |
| 合計  | 383,796 |      | 383,796 |

### 4. 61年度幹事

61年度幹事候補者が、60年度幹事会から提示され、満場一致で決定された。

代表幹事と常任幹事の挨拶がおこなわれ、鈴木代表幹事からは、会員の増加をおこなっていきたいことが、岸田常任幹事からは、機関紙SEAMAILの月刊発行を死守したいこと、盛田常任幹事からは、関西地区での活

動を拡大したいことが、それぞれ抱負としてのべられた。

なお、61年度幹事は、本号の裏表紙に掲載されている。また、幹事の追加については、幹事会に一任することが決議された。

### 5. 61年度予算案

61年度予算案が提示され、61年度は60年度よりおおきく拡大されている。そのために、会員獲得、イベントに努力をすることが示され、満場一致で決定された。

#### 収入

|      |            |
|------|------------|
| 会費   | 11,810,000 |
| 事業収益 | 6,332,500  |
| 合計   | 18,142,500 |

#### 支出

|        |            |
|--------|------------|
| 人件費    | 5,000,000  |
| 事務所維持費 | 3,935,000  |
| 通信費    | 3,360,000  |
| 印刷費    | 4,800,000  |
| 予備費    | 1,047,500  |
| 合計     | 18,142,500 |

### 6. 分科会・支部報告

現在活動をおこなっている、分科会、支部の世話人から、60年度の活動報告、および61年度の活動目標が報告された。また、現在発足が待たれているAIとネットワーク分科会も、早急に発足することが提案され、活動を開始することになった。

分科会・支部の活動内容、今後の予定等は、SEAMAILに毎号掲載されているので、そちらを参照されたい。

## 幹事会報告

第4回：4月18日

第5回：5月8日

会 場：いずれも機械振興会館

議題：

### 1. 会員状況

会員数は、4月18日時点においては、正会員339名、賛助会員5社であった。5月8日には、正会員342名に、賛助会員は8社に増加した。

12月からの正会員の増加は、月平均30名になっている。

また、前回の幹事会でも議題になった賛助会員の募集については、担当の幹事をきめて、パンフレットの作成をおこない、勧誘活動をおこなう。

### 2. 会員更新状況

200名いるボランティア会員の更新状況は、4月18日時点では、約10パーセント、5月8日では、約20パーセントであった。

これは、会員の更新案内をSEAMAILに同封しているため、会員が更新時期をみのがしていたり、SEAMAILが船便になっていたという、事務手続きの不備とおもわれる。そのため、時期をみて、個人宛に、更新の案内をだすことにする。また、メール・ラベルに更新時期を追加することになった。

### 3. 総会

5月18日におこなわれる総会までに、役員の候補を決定すること、60年度の会計報告、および61年度の予算案を作成することになった。

代表幹事の候補者は、常任幹事のなかから選任することになり、鈴木常任幹事が、満場一致で選任され、また、常任幹事の追加候補としては、盛田幹事が選任された。

60年度決算は、事務局から提出され、会計監事によって監査がおこなわれ、その結果、承認された。また、イベントによる収支は、各イベントごとにおこなうこととし、収益がでた場合、それを事務局へ寄付することが決定した。

また、61年度の予算案が提出され、若干の手直しをして、総会に提出することが決定された。

これら決算と予算案は、24ページの総会報告に掲載されている。

### 4. エイプリル・フォーラム

4月24日（木）におこなわれた、「実践的ソフトウェア開発環境を考える」をテーマにしたフォーラムの報告が、杉田セミナー委員長からあった。

約70名の参加者があって、成功裡に終了したが、テキストに使用した「長岡ワークショップ報告書」の印刷費が予定よりおおくかったため、財政的には若干の赤字をだしたが、長岡ワークショップでの収益で相殺された。

なお、フォーラムの記録は、SEAMAILの今月号に特集されている。

### 5. 設立記念フォーラム

5月19日（月）の午後、東京農林年金会館パストラルでおこなわれる、「ソフトウェア技術者に期待する」の準備状況が説明され、幹事が協力して、宣伝にあたることになった。

このフォーラムの講演とパネル討論の記録は、SEAMAILの次号に特集される予定である。

### 6. イベント

今後のイベントとして、9月に「秋のセミナー・ウィーク」、10月の情報化月間にあわせたイベント、AIフォーラム、再利用フォーラム、等が月に1回の割で予定されていることが発表された。

また、盛田幹事から、関西地区でのイベントをおこなうことが提案され、その可能性が検討され、積極的におこなうことになった。

これらイベントは、SEAとしての活動の活性化と、事務局の財政基盤確保のためにも、是非とも成功させたい。

### 7. 分科会・支部報告

各分科会・支部の活動状況が、各世話人からあった。幹事会と分科会・支部間の情報を密にするために、世話人を幹事にする件が提案され、前向きに検討することになった。

### 8. 次回会合

日時：6月17日（火）18:30 - 21:00

会場：機械振興会館 地下3階B3-1

（野辺良一）

## 環境分科会 (SIGENV)

## 第5回会合報告

日時：昭和61年5月14日（水）19：00～21：30

場所：機械振興会館 地下3階2号室

出席者：11名

内容：長岡ワーク・ショップのレポートに収められている、30編のポジション・ペーパーをもとにしてに意見交換を行った。

## 1. 意見交換

意見交換のなかで、だされた意見をいくつか記す。

- ・環境という言葉が使われているが、環境とツールを、異なる内容をもつ概念として使い分けているとはいえない。
- ・UNIXが開発用ツールとしてすぐれていることを、自明のこととしている議論にはついていけない。従来からの仕事のやり方に染まっている面もあるが、UNIXの勉強をしたくても、その時間が与えられなくて、反感やら疎外感をいだいている会員もいることを知っておくべきである。
- ・ソフトをエンド・ユーザにつかってもらう立場にあるソフトウェア技術者が、比率の上では多いはずなのに、ワークショップのレポートをみる限り、ユーザは何を求めているのか、それにどのように対応すべきかへの関心がうすいように思える。

## 2. 次回以後の研究会予定

## 第6回月例会

日時：昭和61年6月18日（水）19：00～21：00

場所：機械振興会館 地下3階第2号室

テーマ：ドキュメントの再利用支援環境

世話人（連絡先）：岡本吉晴（三菱総合研究所 情報技術開発室）03-270-9211

## 第7回月例会

日時：昭和61年7月16日（水）

場所：機械振興会館 地下3階2号室

テーマ：未定

世話人（連絡先）：引地信之（SRA技術本部）

（久保宏志）

## 再利用分科会 (SIGREUSE)

## 活動報告

## 1. 第2回編集会議

4月9日（水）7：00～9：00

・目次案の検討

暫定的目次案を決めた。

・記述方針

体系的な解説書に豊富な具体例を加えたものとし、単なるhow-to物や論文集にはしない。また、ツールに関する記述には使用評価も盛り込み、記述内容は編集委員会が責任をもつ。

## 2. 第2回研究会

4月23日（水）7：00～9：00

ソフトウェア再利用概念の整理に関する検討をおこなった。Freemanの再利用対象の階層的分類、およびBiggerstaffの再利用方法の分類にしたがって議論を行ったが、議論百出し結局発散してしまった。この件に関しては、第3回編集会議で叩き台を作成し、第3回研究会でもう一度討論を行い、第3回討論会でその結果を報告する。

## 3. 次回以降の予定

以下に次回以降の予定を記します。各会とも参加は自由です。また、参加の連絡は特にいりません。参加費として、会場費代に各会ごとに千円徴集させていただきます。

## 第4回研究会

日時：6月11日（水）7：00～9：00

場所：機械振興会館 B3-9

連絡先：大西亮一 JSD(503-4981)

## 第4回討論会

日時：6月25日（水）7：00～9：00

場所：機械振興会館 B3-1

連絡先：青島茂 SRA(234-2611)

世話人は、会運営上の雑用を引き受ける人のことと考えています。現在世話人は、阿部正平、村井進、青島茂、大西亮一が引き受けておりますが、世話人を引き受けてもよいと思われる方は、ご連絡下さい。

（村井進）

## 管理分科会 (SIGMAN)

## 第2回研究会報告

第2回研究会を、5月9日(金)に、8名の参加者でおこなった。

当日は、JSDの芝原から、コスト管理の面から収集した「コスト・データ」の紹介があり、そのあとで、発表内容についての討論をおこなった。

この発表内容と討論をまとめて、近々、SEAMAIL誌上で紹介をします。

今後の、管理分科会の研究会の予定は、以下のようです。みなさんの参加を、待っています。

当面、事務局として、芝原が担当し、会計も兼ねていますが、世話人希望の方は、芝原まで御連絡下さい。

連絡先：協同システム開発(03-503-4981)  
(芝原雄二)

## 管理分科会第3回会合

日時：6月13日(金) 6:30~8:30

場所：JSD会議室

テーマ：デザインレビューに関するデータの紹介

報告者：前島 仁

参加費：1,000円

## 管理分科会第4回会合

日時：7月11日(金) 6:30~8:30

場所：JSD会議室

テーマ：プロジェクト管理上の問題について

報告者：土崎直樹

参加費：1,000円

## A I 分科会 (SIGAI)

## A I 分科会発足

A I 分科会が、発足することになりました。これまで、アンケートに回答をくださったかた、電話等で連絡をくださったかた、おまたせしました。当面の世話役もきまり、第1回の研究会を、以下のようにおこないます。

## 第1回月例会

日時：6月24日(火) 18:30-20:30

場所：機械振興開館 6階68号会議室

テーマ：「A I 技術のテクノロジー・マップ」

白井豊(構造計画研究所)

参加費：1,000円程度(会場費として)

発表が終わったあとに、今後のA I 分科会の活動方針、世話人の追加等について、参加者による討論を行いたいとおもいます。ふるって参加してください。

参加は自由ですので、会員以外のかたの参加も歓迎します。会員の近くにいるかたで、A I に興味をもっているかたなどへの宣伝をしてください。また、参加のための連絡は、とくにいりません。当日、会場へおこしてください。

A I 分科会では、月例の研究会の他に、A I フィールドでの言語(LISP, Prolog, Smalltalk)の勉強会も予定しています。詳細がきまり次第、SEAMAIL誌上でおしらせします。

## 問い合わせ先

協同システム開発(03-503-4981)野辺良一

## 教育分科会 (SIGEDU)

## 1. 第2回分科会

5月15日(木)に、10名の参加を得て第2回会合を開催しました。

テーマは「新入社員教育について」として、日本電子専門学校情報学部河村一樹さんから話していただきました。ちょうど今は、新入社員教育の真っ最中ということもあって、担当されているスタッフの方が中心に集まり、活発な討論がおこなわれました。主な話の内容と討論のポイントは次のようになります。

## 教育の目標

プログラムの能力

カリキュラム事例

教育効果の測定

新しい教育展開

なお、この分科会の内容はいずれSEAMAILにまとめて掲載する予定です。

## 2. 次回(第三回)分科会の予定

- ・日時: 6月24日(火) 18:30-20:00
- ・場所: 機械振興会館: 6S-1部屋
- ・参加費: 1000円
- ・発表者: 大浦 洋一(シーイーシー)
- ・テーマ: 実践的な技術移転の試み
- ・内容: ソフトウェア開発の現場で既存の作業手法の改善や、新しい技術を導入することについて、いわばケーススタディとして講師の経験を発表していただき、参加者と一緒に今後の効果的な技術移転の方策を探る。
- ・プログラム
  - 18:30-19:10: 発表
  - 19:10-20:00: 討論&意見交換
- ・参加資格: 参加自由
- ・連絡先: 杉田 義明(tel:03-239-5473)
- (参加については事前連絡は不要です)

## 3. 会員の募集及び世話人について

技術移転や教育に関して興味を持っている方の教育分科会への参加をお待しております。

希望者は分科会に直接来ていただくか、事務局または世話人の大浦(シーイーシー)、杉田(SRA)までご連絡下さい。

## 横浜支部

第1回、第2回のSEA Yokohama DOC(横浜支部)のミーティングが、4月28日、5月30日に開催されました。SEAMAILによるアナウンスが遅れ、会員の方には残念ながら「後の祭り」となってしまいました。

そんなわけで、Y-DOCの結成式は、紹興酒と広州風家庭料理を前に、中華街の一角で厳かにとりおこなわれ、今後の運営を、次のような基本方針でおこなうことになりました。

## 1. 運営方針

Y-DOCは毎月一回、その月の最終週の金曜日におこなう。

内容としては、いわゆる学識経験者の人を招いて講演をして戴く、あるいは自分たちが持つ問題について「文殊の知恵」を出し合う等、いろいろなことに果敢にチャレンジするSIGANY(Special Interest Group on ANYthing)としての活動を行う。

## 2. 世話人

熊谷、林、藤野、松下の4人がとりあえず世話人となる。世話人とは、開催場所の確保、資料の準備、会員への連絡などの雑用を行うボランティアのことで、今後、さらにボランティアをつのりたいと思います。

## 3. 今後の予定

## 第3回Y-DOC

日時: 6月27日(金), 18:30~21:00

場所: 神奈川県政相談センター 903号室

(横浜西口から徒歩3分、三越の裏、川渡る)

参加費: ￥1,000円程度(会場費、茶代として)

内容: 支部として先輩の「関西支部」から講師を招いての講演を予定しています

盛田政敏氏: 「ドキュメントの機械化」

その他: 自由討論

申し込み: 参加自由、事前に連絡は要りません

## 問い合わせ先

電話03-496-8521(内線450) 藤野までどうぞ

(10:00~12:00, 13:00~PM.17:00)

## 関西支部から

桜の花も散り果てて新緑の季節となりました。初夏を思わせる爽やかな日差し（紫外線入り）と深山に煙る春の長雨（放射能入り）が、目まぐるしくうつろう季節を感じさせてくれる今日この頃です。この陽気に大阪市内を流れる堂島川（近年はドブ鯨川と呼ぶなり）にも釣り人の姿がちらほら見られます（一体どんな魚が生息しているのでしょうか）。中之島にはうまく相手を釣ってきたアベックがハデに出没するようになってきました（一体どんな手を使ったのでしょうか）。夜になれば堂島川にはこうもりが、近くの北新地には蝶や堂島すずめが飛来します。ところでSEAのメンバーは一体どこへ行ったのでしょうか。ひょっとしたら月例会の案内が届かずにあてもなくさまよっているのでしょうか？

関西支部の月例会は、この堂島川と北新地に挟まれたすばらしい環境にあるSRA関西支社のミーティング・ルームでおごそかに行なわれています。討議の内容は、最近のソフトウェアに関する話題を中心に、集まったメンバーの見近な問題を上げたり、次回の研究会の計画を立てたりというような非常につまらん事柄を、もとえ、非常に役に立つ議論をかわしています（春らんまんの世間を離れて、こんな所で、こんな時間に、こんな話をしている者の気が知れんと思われる方も多いことでしょう。実は我々もフトそう思うことが・・・）。

このようなシビアな形態の会談は、大体2時間程度で終わり、たいていそのまま梅田方面へ向かって地下街に潜りこみ、蝶どころか蛾もない居酒屋へ、こうもりのようにそっと入ります。しかしながら、ここでこそソフトウェア開発環境の大切さとともにソフトウェア討議の環境の重要性を実感として体験できるのです。つまり、前半の会議形態で（シラフで）具にもつかないことを恥も外聞も無く喋れる人と、後半の環境になって一気に盛上がる人がいます（たまに収拾のつかない状態に陥る人もいますが、相手にしないで下さい）。こういう状況を冷静に観察すると、各個人のスキルと性格の相違が如実に現れるため、ソフトウェアの開発における人材の配置と作業の進め方を考える上で大いに参考になります。

さて、このような恵まれた環境でこれまでに必死で討議した結果、次のような画期的な結論を得るに至ったのです。つまり、今後のソフトウェアに関する論議はおじんやおぼん（大阪弁でお年寄りのことをいいます）が、ガ

タガタ言っているアカン、関西人の身上であるフレッシュな思考とどわいたん（大胆）な発想をもって、ソフトウェア理論を根底からひっくり返さなアカン！矢でもテッポでも持ってこい！！（注：関西では本当に持って来ることがあります）。従って、新年度を迎えたのを機会に、SEA関西では今までのくら〜いイメージのこうもり群団を一掃し、SEAにふさわしくおらからで新鮮な会話と、かもめのような爽やかなSE諸氏を迎えたいということになったのです。

とにかく、ソフトウェアに従事している方、従事しようと思っている方、従事したが失敗した方、恨みを抱いている方等、すこしでもソフトウェアに関心のある方はどしどし参加して下さい。特におもしろい人、若い人、美しい人、賢い人は堂島に集合して下さい（その他スケベな人は中之島へ、野蛮な人は甲子園へでも行って下さい）。

SEAの月例会では、今後その都度テーマを決め、テーマに関心のある人に広く参加していただくと考えています。事務所や研究所に閉じこもって必死で仕事をしている人にとっては、一服のカンフル剤、暇で時間をもてあまして人々には、暇潰しの絶好の場所になるでしょう。きさくなソフトウェア談義を続けたいと思いますので気軽に参加してください。待っています。

（追記）

（）でコメントを挿入することによって、理論の展開が非常に明確になるというプログラム開発のモハンとなるような見事な文章になったと自負しています。

（臼井義美）

### 次回月例会のご案内

日時：6月11日（水）18：30～20：30

場所：SRA関西支社

大阪市北区堂島1-5-17 堂島グランドビル6F

tel 06-344-2611

テーマ：「つながらないネットワーク」

連絡先：同上 阪井または瀧波迄で

## SEA Forum July '86

## 実践的再利用技術を探る

ソフトウェア技術者協会(SEA)  
再利用分科会

SEA再利用分科会では、このたび、その活動の一環として、次のようなフォーラムを計画いたしました。ソフトウェア生産性向上を目的として、具体的なツールの開発や導入を担当されている技術者・管理者の方々の、積極的なご参加をお待ちしております。

## 開催要領

- (1) 日時: 昭和 61 年 7 月 11 日 (金) 13:30 - 17:00  
 (2) 場所: 青年会議所会館 (東京都千代田区平河町 2-14-3 TEL: 03-234-5601)  
 (3) プログラム

基調講演1: ソフトウェア再利用技術の現状と方向 (13:30 - 14:20)

村井 進 (日本コンピューター・システム)

ソフトウェア生産性と信頼性を向上させる手段として、再利用に関する数多くの技法が提案され、実際に適用されてきた。それらの技法は、パターン化とスケルトン化、および部品化から発展してきたものとして整理できる。ここでは各ジャンルに属する具体的なツールを紹介しながら、技法に関する系統的な整理を行う。

基調講演2: オブジェクト指向と再利用 (14:20 - 15:10)

大木 幹夫 (日本電子計算)

ソフトウェア再利用の方向は、究極的にはフォーマル・スペック、自動合成、オブジェクト指向言語の方向に分けることができる。ここでは、講師の経験にもとずいて、オブジェクト指向言語の再利用性について述べる。

パネル討論: 実践的再利用環境の構築をめざして (15:30 - 17:00)

コーディネータ: 寺田 賢二 (ソフトウェア・パートナー)

パネリスト: 阿部 正平 (協同システム開発)\*

青島 茂 (ソフトウェア・リサーチ・アソシエーツ)

新森 昭宏 (インテック)

(\*は交渉中)

(4) 参加費: 1 名当たり 10,000 円 (SEA会員), 15,000 円 (一般)

(5) 申し込み及び参加方法: 所定の申し込み票にご記入の上、下記宛てお送り下さい。折り返し受講票及び請求書をお送りします。  
 ただし、定員(100名)になり次第、締め切らせていただきますので、あしからずご了承下さい。

〒 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

FAX 03-262-9719

問い合わせ: SEA事務局 (03-312-3256) または forum専用電話 (03-238-9909) までお願いします。

## SEA Forum (July '86)参加申込書

氏 名: \_\_\_\_\_ (ふりがな: \_\_\_\_\_) 会費扱い (〇印で指定): 会員, 一般

会社名: \_\_\_\_\_ 部門: \_\_\_\_\_ 役職: \_\_\_\_\_

住 所: (〒 \_\_\_\_\_) \_\_\_\_\_

TEL: \_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_ (内線 \_\_\_\_\_)

備考(s): \_\_\_\_\_

## Call for Position Papers

## 第4回 夏のプログラミング・ワークショップ

## プロジェクト・プランニングのケース・スタディ

—主としてテクニカル・マネジメントの視点から—

主催：ソフトウェア技術者協会（SEA）

後援：（社）情報サービス産業協会（JISA）

早いもので、夏のプログラミング・ワークショップ（通称・若手の会 in 盛岡）も、4回目を迎えます。今年は、議論をより手ごたえのあるものにするために、具体的な事例をとりあげて、ソフトウェア開発プロジェクトのキックオフ時における技術面のプランニングが持つ重要性について、参加者全員で考えてみたいと思っています。

このワークショップは、若手技術者どうしの有益な経験交流と自己啓発の場として、これまで高い評価を受け、毎年定員の2倍以上の参加希望者が殺到して、スタッフにうれしい悲鳴をあげさせています。今年もまた、それぞれの職場の次代をになう若きエースたちの、熱気あふれる討論が期待されます。迎えるつプログラム委員会スタッフには、業界の第1線で活躍されている「熱中型」プログラミング・マネージャの方々にお集まりいただきました。

多くの若手技術者からの、意欲的なポジション・ペーパーをお待ちしています。

## 開催要領

1. 期日： 昭和61年9月10日（水）午後から13日（土）正午まで（泊4日）
2. 会場： ホテル東日本（岩手県盛岡市）
3. 定員： 25名
4. 参加資格： 年令30才以下、実務経験2年以上のソフトウェア技術者
5. 費用： 65,000円（SEA会員、JISA会員）、75,000円（一般）  
現地集合・現地解散とし、期間中の宿泊費（朝食付き）および資料代を含みます。
6. スタッフ： 実行委員長： 堀江 進 （日本電気ソフトウェア）  
プログラム委員長： 松尾 正敏 （ソフトウェア・リサーチ・アソシエイツ）  
プログラム委員： 臼井 義美 （日本電子計算）  
熊谷 章 （パナファコム）  
野村 行憲 （岩手電子計算センター）  
深瀬 弘恭 （アスキー）  
盛田 政敏 （神戸コンピューターサービス）
7. 申し込み方法： 下記の申込書に記入し、ポジション・ペーパー（ソフトウェア・プロジェクトの計画や運営に関して、自分自身の経験を踏まえた具体的な意見や提案をA4判2～5枚にまとめたもの、なるべくワープロで）を8部コピーのうえ、昭和61年7月15日（火）までに、プログラム委員長宛にお送りください。申し込み多数の場合には、プログラム委員会でポジション・ペーパーを審査の上、7月末までに参加者の決定を行ない、結果を各自に通知します。

申し込み先： 〒102 東京都千代田区平河町1-1-1

ソフトウェア・リサーチ・アソシエイツ 環境開発本部  
松尾 正敏 (TEL:03-234-2611)

## 第4回 夏のプログラミング・ワークショップ 参加申込書

氏 名： \_\_\_\_\_（年齢： \_\_\_\_ 才） 会費扱い（○印で指定）： 会員（SEA, JISA）、一般

会社名： \_\_\_\_\_ 部門： \_\_\_\_\_ 役職： \_\_\_\_\_

住 所：（〒 \_\_\_\_\_） \_\_\_\_\_

TEL： \_\_\_\_\_ - \_\_\_\_\_ - \_\_\_\_\_（内線 \_\_\_\_\_）

備考(s): \_\_\_\_\_

## SEAこれからの活動予定

- 1986 -

### SEA会員状況

#### ◆6月(June)

11日(水)

関西月例会(SRA関西支社)

再利用分科会第4回研究会

13日(金)

管理分科会(JSD)

18日(水)

環境分科会第5回月例会(機械振興会館)

6月24日(火)

AI分科会第1回研究会(機械振興会館)

教育分科会第4回研究会(機械振興会館)

6月25日(水)

再利用分科会第4回討論会(機械振興会館)

6月27日(金)横浜支部第3回Y-DOC

(神奈川県政相談センター)

昭和61年5月24日現在の入会会員状況は以下の通りです。

正会員—356名

賛助会員—10社

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、事務所へ早い時期にお知らせください。

#### ◆7月(July)

1日(水)法的保護分科会第1回研究会

(ソフトウェア流通促進センター)

11日(金)

SEA Forum July(東京・青年会議所会館)

「実践的再利用技術を探る」

管理分科会第4回(JSD)

16日(水)

環境分科会第6回月例会(機械振興会館)

### 法的保護分科会(SIGSLP)発足

第1回会合は7月1日。

テーマは、「UNIXのライセンスをめぐる諸問題」

問い合わせは、今月の世話人、能登末之(SRA-03-239-5472)まで。

#### ◆9月(September)

3日~6日

秋のセミナー・ウィーク(機械振興会館)

10日~13日

夏のプログラミング・ワークショップ(岩手県・盛岡)

#### ◆10月(October)

1日~2日

国際フォーラム(東京・東条会館)

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

年令 \_\_\_\_ 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