



SEAMAIL

Newsletter from Software Engineers Association

Volume 4, Number

12

1989

目 次

事務局から		0
ソフトウェア・シンポジウム'89 のミニパネルから	渡邊 雄一	1
HyperScore Tool Kit Ver-5.2 (Smalltalk-80 上の音楽ツール)	森口 岳雄	11
UNIX VS MS-DOS	横山 博司	19
シリコンバレーを楽しむために - その2 -	大神原 薫	21
SEA のネットワークについて		31
会員の声		35

事務局から

世界中がなぜか慌ただしかった1989年もそろそろ暮れようとしています。「師」は12月にしか走らないようですが、人手不足のSEA事務局は、1年中走り通してました。「月日は百代の過客にして、行き交う人々もみなこれ旅人なり」とかの松尾芭蕉さんがいっています。今年SEAに新しく入会した人は約180人、それとほぼ同じ数の人たちが退会していますので、実質上の会員数はあまり変わりませんでした。まさに、「年々歳々花相似たり、歳々年々人同じからず」の心境です（もちろん、「花」とは私のこと！）。

今日は12月13日。昨夜今年最後のイベントである「年忘れ自由討論フォーラム」が賑やかに終わったばかりです。SEAも設立以来満4年、いろいろな活動も定着してきて、東京では8月を除いて毎月フォーラムがありました。定期的な会合も、これまでの「ソフトウェア・シンポジウム」、「環境」、「教育」、「若手」、「信頼性」（これは年2回）の他に、「テクニカル・マネジメント」と「プロセス」の2つが増えました。合計すると年間8回もの大きなイベントが開かれたことになります。

さらに、もう1つ。天安門事件のあおりを受けて第3回日中シンポジウムが来年に延期になり、やれやれと思ったとたん、気まぐれな某代表幹事殿の発案で、秋の京都の「国際シンポジウム」、しかも全部英語で討論をするなどという無謀な企画が飛び出しました。それもなんとか無事に終わったようです（報告書や会計処理の後始末はまだですが）。

今年のもう1つの特筆事項は、なんとSEAMAILの第12号が年内に印刷できることです。来年もみなさんからの原稿をお待ちしています。各イベントの責任者の方、報告書よろしくお願いします。また、ときどき行なうアンケートにも、なるべく多くの方々にお答えいただきたいと思います。

機関誌の送付、毎月のSIGおよび支部の活動案内、そして会費更新のお知らせを含めると、会員1人あたり年間20数回の郵便を事務局からお送りしていることになります。

$$1700 \times 20 = 34,000;$$

$$34,000 / 365 = 93;$$

なんと1日平均百通近くのお手紙を出している計算です。そのうち、麹町郵便局から感謝状が届くでしょう。これだけの回数、宛名ラベルを貼っていると、まだ1度もお目にかかったことのない方も、すっかり古いお友達のように感じます。会費更新用の返信ハガキには、なんでも結構ですから、ぜひ一言メッセージをお書きください。

来年もみなさまにとって良い年でありますように！

(From Hayabusa-Chō, with LOVE)

ソフトウェア・シンポジウム'89でのミニパネルから

— ソフトウェアプロトタイピングの適用に関する技術的課題は何か? —

渡邊 雄一

0. はじめに

「ソフトウェア・シンポジウム'89」(SEA, JISA 共催)で行われた「プロトタイピング」というミニパネルの内容を紹介します。このセッションは、メインスピーカの宮本勲先生[ハワイ大学]の講演が約1時間、それを受ける形でコメントの佐原伸[SRA], 藤野晃延[FXIS]両氏の意見発表が併せて30分、計1時間半のパネルでした。司会者たる私の力不足から、特に敵対するような意見が飛び交うこともなく、パネルとしての面白み(?)には欠けましたが、技術的には非常に興味深いものでした。

以下、3氏のお話を講演記録風にまとめます。1章から5章までが宮本先生の講演です。

1. ソフトウェア・プロトタイピングの現状

まず現状ですが、沢山のプロトタイピングの仕掛けが出てきているように思います。それらを幾つかに分類して並べてみました。

1.1 プロトタイピング・ツールの現状

1.1.1 古典的なツール

これはYACとか、ALOEとかの古いツールです。これらは、ある意味で言語システムのプロトタイピング・ツールだといえましょう。

1.1.2 一般的なツール

これらは、アプリケーション・システムをプロトタイピングするためのもので、かなり増えてきているように思います。まず、ERA, UP, PWBおよびKIDSは、我々のところ[ハワイ大学]で作ったシステムで、後で少し詳細に説明します。それ以外では、例えば、Naval Postgraduate schoolのCAPS[2]ですが、これはプロトタイピングを記述するPSDLという言語があって、それを変換してスケジューラなどで絵を書いていって見せてくれるといったシステムです。またオレゴン州立大のOSU[3]というシステムですが、これはユーザ・インタフェースのプロトタイピング(勿論グラフィックカルなもの)です。そしてそのグラフィックカルな記述から、多少ダイナミックな記述を見せるために、実行可能なプログラムに変換して実際の振舞いを見せています。

それから、最近の小さなコンピュータ、NEXTであるとかMacであるとかには、システム自体に特にユーザインタ

フェースを簡単に作れるように、Interface BuilderあるいはPrototyperという名前のものが用意されています。それから、BuxtonのやっているMenulayとか色々ありますが、この中で比較的有名なのは、WassermanのRapid/Useではないかと思います。これは基本的にはState Transition Diagramでユーザ・インタフェースの記述をやってそれを見せるというものです。これらは、多分にビジネスオリエンテッドなシステムの開発を支援するものですが、一応汎用といってよいでしょう。

1.1.3 特定アプリケーション向けツール

プロトタイピングというよりも“速く実行可能なものを作る”という意味合いが強いのですが、第4世代言語風のものとかプログラムジェネレータなどが、幾つかあるようです。分散システムや知識ベース・システム向けのものは比較的少ないようです。まだまだこれからといったような状況ではないでしょうか?

またこの他にも、直接プロトタイピングという言葉は使われていなくても、システム分析や設計のフェーズで直接/間接に人間の作業を助けるいわゆるCASEツールは沢山存在します。

1.2 それらの評価は?

現状は「新しいプロトタイピング・ツールがどんどん出てきつつある」というところです。ところが「それらは本当に役に立つのか?」という評価がまだあまりきちんとされていないように思います。ごくごく一部なんですけれども、80年代の初期にBoehmがプロトタイピングをしているの/していないのを、同じアプリケーションを別のグループにやらせて比較評価しているものが1つだけあります。

それから我々のところでは、学生がキャンパスの外に出まして、だいたい4~5人くらいのグループで、約半年位かかるプロジェクトをやらせているのですけれども、そういったところで使った評価があります。学生は素人ですから、初めて実際のシステムを作るので、必ずしも評価のベースになる最適なものだとは思わないのですが、特にユーザインタフェースなんかをプロトタイピングして、それをクライアントに見せて、要求定義をやって開発していくのに、(非常に大雑把なんですけれども)大体1900~2000人時掛かっていたシステムが、1500~1600人時で

終えられるようになってきました。「それはどうしてか？」といえば『開発途中の段階での要求仕様の変更のが極端に少なくなっている』ということによります。そういうことで、大雑把ではありますが『かなり役に立つ』といってよいと思います。

1. 3 いくつかの所見

我々の経験も踏まえて気づいた事柄を列举します。

・プロトタイピングは新しい技術であり、ツールのサポートも完全ではなく、いろいろ問題がある。それは大雑把に分けて、作業手順の問題、それから純粋に技術上の問題、それからツールの問題、それからプロトタイピングの作業の管理、そして測定/評価といったことです。

・プロトタイピングについては、次の2つのタイプがある。

- 1) “本番システムの部分的で重要なところを先行して作るということ”ということの主目的とするもの
- 2) 要求仕様それもフレキシブルな要求仕様を何らかの形でモデル化し実現して、さらに詳細な要求仕様を導き出すもの

・プロトタイプ(あるいは workmodel)を作る方法には次の2つのタイプがある。

- 1) 本番と同じようにプログラミング言語(勿論できるだけ high-level なもの)を利用し、その言語の実行環境を使って、プロトタイプ(あるいは Workmodel)を動かす
- 2) プログラミング言語ではなく、何かモデルを記述する別の言語を考えて、シミュレーションやアニメーションといった技術を使って、動的な振舞いを見せる

・ユーザインタフェースのプロトタイプは非常にポピュラーである。

・要求仕様をプロトタイプする場合と、設計をプロトタイプする場合がある。時には両者が混ざり合っていることもある。

2. プロトタイピング/ツールに関する技術的問題

2. 1 作業手順

一口に『プロトタイピングのパラダイムはまだ標準になっていない』ということがいえると思います。例えば[1]で幾つか紹介したものや、ほかにもたくさん作業手順のモデルがありますが、どれ1つとしてまだ一般に広く受け入れられてはいません。

2. 2 技術的な問題点

2.2.1 要求仕様の規定のされかた

ソフトウェアの要求仕様(ここではプロトタイプすべきものに限定せず、ターゲットシステム全体の要求仕様を指

す)がいかに与えられるか、ということです。場合によっては、かなりきっちりとしたものが与えられ場合もありますし、そうじゃないこともあります。非常に単純なモデルは、ソフトウェアの要求仕様はかなりきちんとした形で与えられるのだという仮定で、技法なりツールが供与されるというものが非常に多い。それから、そういう風に仮定していても、プロトタイプすべき要求仕様に関する情報をどうやって集めるか、またどういう恰好にするのかは、ほとんど議論されていないことが多いようです。

またプロトタイピングのメソッドロジー、あるいはツールの使用が、従来からのシステム分析あるいは要求分析定義というアプローチと遊離している場合も少なくありません。そのようなことから、役に立つ/立たないといった議論が多く出てきているように思います。それから、プロトタイピングをするために、あるいはそのプロトタイピングする対象となる要求仕様は、どういう形でもらわなければならないかが、明確でないことが多い。どういうタイプの要求仕様が対象となるのか?、対象とすべきなのか?、場合によってはユーザインタフェースだけであったり、あるいは状態遷移図で書けるようなところであったり、個々のアプローチによって違います。必ずしも、プロトタイピングの対象が完璧ではないといえますし、さらにいえばプロトタイピングすべき大事なところが、他に一杯あるような気がします。

それからプロトタイピングの作業のステップの中で、どういったアプリケーションシステム(あるいは複雑・大規模なシステムでしたらどういう部分が)プロトタイピングに向いているのか、あるいは向いていないのかという判断をしなくてはいけないのですが、それがまるっきり抜けているところがあります。ですから、何でもかんでもプロトタイピングすればよいというわけではなく、場合に応じてそれをリコメンドする技術/方法論が必要といえます。

要求仕様ではなく設計をプロトタイプすることも往々にしてあるようです。ただ、これらのモデルというのは、実はもう少し複雑でしてお互いに関連していて、色々なモデルのエレメント間の関連に対するエレメント、この要求仕様はシステムの中を見るところとここが関連している、ユーザインタフェースのところを見るところとここが関係しているという「関連」に関する要求仕様になるわけです。それは、実はこれらのモデリング技法以外に、リンケージスペシフィケーション言語というのがあります。簡単にいいますと first order logic, set operation, pass expression というそれが組み合わさっている言語だと考えて思っして下さい。それで、

関連を書いて行くということになります。まあいずれにしても、作業手順あるいは要求仕様というものが、どういう風に与えられるかというところで、いろいろ問題があるみたいです。ですからユーザインタフェースだけの要求仕様が全てではないということです。それから1ショット的に考えるのではなくて、従来からのシステム分析なんかの手順の中に良く入り込まないと必要な要求仕様がもらえないということがあると思います。

2.2.2 何がプロトタイピングされるのか？

これが特に大事だと思うんですが、ターゲットシステムのどういう観点をプロトタイプすればよいのか？。で、ユーザインタフェースをプロトタイプするというのは、非常に簡単ですけれどもコンピュータ・システム、アプリケーション・システムというのは、一般に考えても特別なものを考えても、ユーザインタフェース以外にもいろいろ大事なところはあるわけです。

例えばビジネスシステムでしたら、特徴となるようなものは、たとえばユーザインタフェースであるとか、I/Oのフォームであるとか、データベースといったようなものに対する要求仕様がかなり大事な位置をしめるということです。またリアルタイムシステムなどでしたら、むしろ性能であるとか、コントロールであるとか、それから他のシステムとのインタフェースとかデータベースとカリカバリとか信頼性といった要求仕様あるいは観点です。で、こういう風に簡単に見ましても、アプリケーションによって特徴的なところ、あるいは大事なところというのは少しづつ違うわけです（ユーザインタフェースだけではないということです）。

それらの見方からしますと、コンピュータシステムにはいくつかの本質的なコンポーネントがあります。specificationのレベルでは、static（静的）に見ますと入力、出力、それから入力から出力を得る変換 function、functionの属性、性能、信頼性、精度、functionを実行する execution agentなどです。要求仕様というのは、それぞれに対して基本的にあるし、これらの間の関連に関して要求仕様というのはある。

これらは静的に、個々に要求仕様があるわけです。これらの間の「関連」であるとか、あるいはダイナミックに時々刻々とシステムが動いているときに、これらのものが互いに関連し合いながら処理が進んでいくわけですが、そういった物に対する要求仕様（あるいはデザイン）は、実は静的な物なので、それほど難しくありません。またシステムやプログラムを理解するときも同様です。

しかし dynamic にどういう風に動いていくのかを想像したり、デザインするのが難しい。それをもう少し細かくいいますと、システムの内部の振る舞い、外部から見るところの振る舞い、あるいはユーザとシステムとの間での振る舞いというものを、こういったものが、代表的な「動的な」振る舞いではないかと思います。

それで、既存のプロトタイピングツールやその環境を眺めてみると、とても沢山の問題があります。それを列挙すると、以下のようになります。

- 1) プロトタイピングをして確認しなければいけない、あるいは要求仕様間でいろいろ矛盾がある／ないといったことを解決しなければいけないわけですが、大部分のものが、ごくごく一部分の観点だけをプロトタイピングするようになっている。
- 2) プロトタイピングのメソッド／ツールなど多くのものが、ただ単に1つの観点しか扱えず、そして多くの場合、それがユーザインタフェースであるということです。ですから distribution システムであるとか knowledge-base とか いかいものには役に立たない物が多い。
- 3) 今の逆ですけれども、いろいろなシステムの大事な観点をプロトタイピングするものは少ない。「プロトタイピングする」ということは何かを確かめるためにやるので、すけれども、そういうマルチプルな view（複数の観点）に基づいたものは少ない。
- 4) 機能の属性として最初に誰でもが考えるのが効率／パフォーマンスということなんですが、それを扱えるプロトタイピングのテクニックというのは非常に少ない。それから大部分のものが、単純なプロトタイピングとかやなくて、往々にしてシステムの静的な観点をモデルする。あるいはユーザインタフェースにしても、いわゆる紙芝居的な1枚づつ絵が変わってゆくような物が多い。
- 5) 扱える観点とか、動的な振る舞いの扱い方がきめ細くないものですから、そのプロトタイプがあたかも本番システムと同じように使えるとか触れるとか振る舞いを見せてくれるといったことが少ない。このため、何をプロトタイピングすべきか？といった時に、動的な振る舞いをプロトタイピングするものが少ない。これは非常に大きな問題だと思います。
- 6) 複数の観点からモデル化しようといったときに、当然1つのものを見ているわけですからその複数の観点の関連といったものをきちんと持たないといけません。独立している所は独立しているし、関連もっているところは関連もっているようにプロトタイプ（あるいはワークモデ

ル)を作らなければいけない。あるいは、システムの中の
 こういふ観点とユーザからみたさういふ観点、その関連
 に関する要求仕様については、もう殆どのものがモデル
 化できないという非常に残念な状況があるようです。

- 7) 要求仕様というのは、大きな項目できている要求仕様も
 ありますし、基本的な要求仕様から re-compose されて
 出てくるような要求仕様もあるわけですし、まあ基本的
 には階層系になっているわけですが、そういう関係が
 扱えないもの、それから要求仕様間の dynamic な関係
 が扱えないもの、さらに要求仕様の詳細化が扱えない
 といった技術的な問題もあるようです。モデル化という
 技法で問題になるわけですが、
- 8) プログラミング言語で書くというときには、いくら
 high-level でも、こういった階層系のあるものから
 derive されたようなものは書けないはずで、先程いい
 ましたように、分散システム、KB システムに対するプロ
 トタイプ化の技法は少ないというか、今のところ問題
 外のような気がします。

2.2.3 プロトタイプのどの側面をためすのか？

プロトタイプをしたという時に「じゃあ、何をチェッ
 クするのか？」というのが非常に大事です。目的もなしに、
 ターゲットシステムの部分的なものをインプリメントして
 しょうがないわけですし、どういったことがチェックで
 きるのか／しなければいけないのか、ということを理解す
 ると問題が色々あるようです。それらを列挙すると、完全
 性、一貫性、修正可能性、テスト容易性、設計と要求仕様の
 独立性、トレース能力、システムとユーザとの間の伝達性、
 システム内部のモジュラリティなどといったことです。

それで、そういうことが全部できればいいんですけども、
 現状がどうなっているかを述べてみます。

- 1) プロトタイプをする仕掛けに formality が足りない
 といいますが、formal なアプローチが少なく、その少
 なさ故に、分析能力が弱いような気がします。プログラ
 ミング言語では syntax チェックがあった。その内容的
 なもの、意味的なものを、実際 dynamic-test とかして
 チェックして保証するわけですが、プロトタイプ
 でも同じことが必要なわけです。
- 2) syntax 的なものすらもチェックできないものがある。
 ですから、semantic なものは、ほとんどされないことが
 多い。ただ、ユーザインタフェースのプロトタイプ化
 などは、盛んです。それはユーザが見れば自分が欲しい
 ものであるかどうかは分かるわけです。ですから、
 ユーザ・オリエンティッドなチェックというのは、行われ

ているといえます。ですけれども、プロトタイプ化の
 ところで、プロトタイプ化の仕掛けとしてどこまで
 チェックできるか、ユーザ(人間)がチェックしなければ
 いけない、また責任を負わねばいけないところはどこな
 のか？という、そのところの区切りがはっきりしていな
 いことが多い。

- 3) いろいろなチェックすべき評価基準があるわけですが、
 どれをツールでやって、どれを人間がやらなければ
 いけないのかをはっきりしていないことが多い。ですから、
 プロトタイプをユーザが実際に見て試すときに、場合
 によってはワークモデルの記述が根本的に間違っている
 なんていうところまで(人間が)見てやらなければならない。
 まあ、良くできたシステムならそういう単純なあや
 まりというのは、システムがサポートして間違いの無い
 ようなものにあらかじめして、ユーザは意味的に自
 分が欲しいシステムであるのか？ということをチェック
 することに専念できるわけですが、
- 4) たとえそのツールでチェックできるとしましても、何を
 チェックするのか？。評価基準というものがチェックす
 るプログラムのなかに埋まってしまっていることがあ
 って、別の項目をチェックしようといったところまで拡張
 できない。理想的にはチェックすべき criteria に関する
 記述があって、その書かれている内容に従ってワークモ
 デルを調べてゆくアクションプログラムのようなものが
 あって、別々になっていると便利なわけですが、現実には
 そういう風にはなっていないことがほとんどです。
- 5) timing (efficiency ですけれども) そういふものを
 チェックできない。
- 6) formality が足りないが為に、consistency や complete-
 ness のチェックもできないことが多い。
- 7) プロトタイプを作るという過程で、色々な decision
 making を既にしてしまっているということが問題です。
 プロトタイパーという人がプロトタイプを作ったとしま
 すと、その人の偏った解釈、あるいは decision making
 の結果しか見られないということが非常に多いようです。
 それは非常にまずいような気がします。ですから、要求
 仕様があった場合に、(それらの間に)お互いに矛盾があ
 ろうと、不完全であろうと、それをそのままワークモデル
 として map するということが非常に大事です。要求仕
 様というのは、プログラムが specification に対して正し
 い／正しくないというのと違ひまして、正しい要求仕様
 とか間違った要求仕様というのは無いんです。「この人
 がいつていることと、この人がいつていることが矛盾し

ている」とか、「同じ人が昨日いっていることと今日いっていることが違う」とかというようなもので、ツール支援環境ではそういったものをワークモデルに map して「あなたの昨日いったことと今日いったことはこういう風に矛盾していますよ」「どちらが正しいのですか?」「両方とも間違っているのでしょうか?」「それとも両方とも正しいのでしょうか?」ということを聞くようなしかけなんです。ですから、システムの方で勝手に decision making をしたり、矛盾を解いてしまったんではいけないのです。

- 8) 書く内容によって分析できる内容が違う。また書いていなければ、分析のしようが無いということです。現状では、シングル view ベースで formality が足りない為に、プロトタイピングのツールの方でチェックする項目が非常に少ないように思います。

2.2.4 如何にワークモデルを作成し実行するか?

- 1) 先ほどいいましたように、2つの代表的な作り方があるように思います。ひとつは、プログラミング言語とその実行環境のメカニズムを使って、本番システムを先行して作成するというものです。この場合はプログラミング言語ですから、high-level でならシntaxのチェックはしてくれるわけです。ところが、そのプログラムは従来からのやり方のようなテストをしないと、その言語で書いたもの自体が正しいかどうかということが確かめることができません。ですから、セマンティックなチェックというものは弱い。一方、ワークモデルを書くために言語を考えて、あるいはダイアグラミングの仕掛けを考えて、分析したり、シミュレーションしたりするというメカニズム、そういうものを使うやり方がありまして、こちらの場合はもちろん formal さの程度によってちがいますが、formal であればあるほど、その分析できる可能性というものは多くなるわけです。ところが、その行き過ぎは困るわけです。書きにくさとかいったものが出てくるわけです。いずれにしてもそういう2つのタイプがあります。

それで、モデル言語を考えてシミュレートするような場合には、よいところもあるのですけれども、せっかくプロトタイプを作っているのに本番のシステムを作るときに流用するというのが、ちょっと弱いということがあります。ユーザインタフェースをいろいろ定義しますけれども、それをプログラミング言語で表現するところのつなぎがちょっと弱いのです。

- 2) ワークモデルを作る“しかけ”というのが、ほとんどの場

合、スクラッチから作られておりまして、プロトタイピングのしかけ自身の変更であるとか、メンテナンスが非常に大変であります。

- 3) 往々にしてマシンディペンデントなものとなります。
- 4) それからテキストオリエンティドなもの以外がユーザ・インタフェースに(グラフィカルなものも含めて)含まれますと、しかけ自体が非常に大きかったり、あるいは単純な方がいいのですが「重い」のが多いようです。
- 5) メタツールであるとかジェネレータとかを使って(システムを)実現しているプロトタイピングの環境は少ないようです。
- 6) だいたい例題などに出てくるのは、規模の小さなシステムをプロトタイプするものであったり、あるいは、しかけなどが紹介されている場合でも、必ずしも大きなシステムを構築する際に本当に適当なしかけであるのか否か、分からないことが多い。これは大きな問題だと思います。大きなシステムを記述する、プロトタイプする、という場合は、記述でも、階層的に記述できなければならない。「階層的」とはどういうことか?、といいますと abstraction であるとか、re-composition であるとか、partition, composition あるいは simplification とか projection とかいう、いわゆる logical な抽象度のレベルを、上下左右にいったりするものです。こういうものが機能として備わっていないものが多い。ですから、大きなシステムが書けない。
- 7) real-time あるいは distributed system に便利のようなしかけが少ないように思われます。

これは私の経験から来ているんですけど、プログラミング言語で要求仕様をプロトタイプがいったん解釈して、いろいろなディシジョン・メイキングをやってプロトタイプを作るのは、あまり良くないように思います。それよりも、executable な要求仕様の表現法というのを考えて、そのなかに矛盾があらうと、不完全であらうと、要求仕様は要求仕様として表現してやって、シミュレートしたりアニメートしたりしながら、ユーザが本当に欲しいものというものにきちんとできた方がよいんじゃないかと思います。そこにはプロトタイプバのディシジョン・メイキングが入っていませんから。

モデル言語を使った場合に、本番システムを記述する言語と違うわけですから、流用ができないのですけれども、それはなんからの形で translator のようなものをおいて、reuse (再利用) ということを考えなければならない。

それから、もう少しグラフィックスを使った方がよいの

ではないかとも思います。また場合によってはビジュアルプログラミングのようなテクニックも簡単にワークモデルを作るという点で有効でしょう。それから、プロトタイピングの作業手順というものをサポートするために、プロセスプログラミングのようなものを利用してゆくのもかなり重要ではないかとも思います。

それからやはり要求仕様ですので、formal なモデルも informal なモデル/情報も、とりまぜて扱わなければいけないと思いますし、そういったことでは natural language (自然言語) というのはなんらかの形で処理した方がよいのではないかと思っています。

3. 我々 [ハワイ大学] のアプローチとツールの開発

3.1 ERA

ALOE システムのグラフィックス版です。非常に小さくてコンパクトなんですけれども、マッキントッシュの上と、それからテキストの Smalltalk-80 の上で動作します。もし「ダイアグラムを作りたい」という時に、そのダイアグラムのルール/シンタックスを知っているグラフィックスエディタをジェネレートする道具です。ですから、例えばデータフローダイアグラム (DFD) を書くルールを知っているエディタが欲しいとしますと、その DFD のノードのアイコンと、そのノードとノードをつなぐ関係を文法として捉えて、このシステムに入れますと、そのルールを知っているグラフィックスエディタができます。そのエディタは文法を知っておりますので、ただ絵を書くのではなくて、ちゃんとシンタックス的に正しい絵が自然と書ける。〔この ERA はパブリック・ドメインとして公開しています〕

3.2 UP (User interface Prototyping tool)

88年の ICSE (シンガポール) に出展しました。特徴は、ユーザインタフェースのプロトタイピングが非常に簡単にできることです。システム的には、process model, tool behavior model, product model というのがありまして、それぞれがユーザから見える。ちょっと複雑なシステムのように聞こえるかもしれませんが、まず、ターゲットシステムがカバーしなければいけない作業の手順をプロセスモデルで書きます。ついで、作業ステップを実は自動化しないといけないわけですから、それをサポートする機能を2番目のモデルで書きます。それから、ターゲットシステム全体が、ユーザとインタフェースしないといけないし、また個々のツールファンクションがユーザとインタフェースしないといけないので、そのあたりをプロダクトモデルあるいはユーザインタフェースのフォームモデルというもので

モデル化する。この3つのモデルは各エレメント毎、各エレメント間で、両方向の link がはられていまして、その言語は先ほどもいいました、first-ordered logic set-operation pass-expression ということになります。静的にはそういうものを用意しておきまして、UP システムの方は、その link 関係をインタプリットしまして、その目的があたかも本番のときに使用されているような順番で振る舞いを見せてゆくものです。〔近々パブリック・ドメインにする予定。マッキントッシュの SE 以上で動作可能〕

3.3 PWB (Prototyper's WorkBench) [4]

ツール展示で見ていただいているものは、実は PWB1 といいまして、ほぼ2年前に終えたプロジェクトの結果です。今現在は、PWB2 というのをやっております、2年間のうちのちょうど半分の約1年が経過したところです。正直いいまして、PWB システムは少し大きいシステムです。これに対して、先の UP システムは、比較的簡単にビジネスシステムであるとか、小〜中規模のものをプロトタイプするときに便利なものです。特に重要な技術というのは、シミュレーションやアニメーションであるといったようなものです。PWB2 では要求仕様の分析/定義のところを、便宜的に3つのフェーズに分けているんですけれども、全部カバーします。それからモデル記述がたくさん増えています。名前だけをいちおういいまして、needs analysis のところでは objective task constraint model という技法が使われます。それから、システムの基本的な概念化をしなければいけないんですけれども、それには conceptual process model あるいは conceptual data flow model というのが備えられています。それからデータ構造の定義のところは、conceptual なデータモデル化の技法が使われます。それからもう少し詳細になりまして、function model、それから user interface モデル。それから詳細なデータモデリング。その後、少しシステムの中に対する要求仕様を詳細に議論するために functional な structure のモデル。それから、IOPM と書いてありますが、Internal Operational Profile Model、システムの中のシーケンシャル/コンカレントなプロセスの動き/コントロールの動きに対する要求仕様を表現しようというわけです。技法としましては、ペトリネットの技法を extend したものを使用しています。それから、システム外から見るステートといいいますか、External Operational Profile Model というのもあるのですが、技法としましては、状態遷移図で書きます。そのシステムと他のシステムとのインタフェース、あるいはユーザとのインタフェースというものを総括的に書くものです。

ですから、EOPMの一部が、ユーザインタフェース部分として扱われていると考えて下さい。

それからパフォーマンスに対する要求ですが、例えばレスポンスタイムが2秒という要求仕様のシステムと、レスポンスタイムが1秒というシステムとでは、システムの中身ががらりと変わってしまいます。ですから、パフォーマンスに対する要求仕様も扱えないといけないわけで、我々のところでは、queing network などを使いまして、性能に関する要求仕様を表現しようとしております。これから付け加えていくものなのですが、やはりシステムの信頼性というものもきちんとデザインして要求仕様で明確にデザインして、場合によってはフォールトレラントなシステムにするとか、あるいはハード/ソフトを2重に持つとか、いろいろなシステムの中身が要求仕様によって変わってきますが、そういったものを記述できるようにしています。

3.4 KIDS

一番最近に作ったもので、知識ベースシステムのプロトタイピングツールで、ピッツバーグでの ICSE でツールプレゼンテーションを行いました。ユーザインタフェースの部分とか、プロセスモデル、ツールのところは UP システムをベースにしておりまして、知識の表現と、インファレンスメカニズムおよびそのアクセスの部分がきちんとプロトタイプできるようにしております。知識はプロダクションルール的な表現、デシジョンテーブル的な表現、それからプロローグ的な表現の3つが可能になっております。それぞれ細かく入れるわけではなく、グラフィカルなオブジェクト・オリエンティッドなインタフェースがありますので、なにも prolog でいちいち書くような感じではありません。大体基本的にはマウスを使いまして、そういったことが簡単に書けるようになっております。

4. 1990年代はどうか?

80年代の前半からプロトタイピングに関することがいわれはじめて今日まで来たわけですが「はたして技術的な革新はあったのだろうか?」という問い掛けに対しては『No』といわざるを得ないと思います。ですが、近い将来に向けて、よいやり方ができるいろいろなタネはいうものは出てきているのではないかと思います。

技術的な問題がある程度分かってきたということも勿論ですし、それ以外に様々な環境が少しずつ変わってきているようにも思います。例えば昔と比べて、色々なグラフィックスのルーチン、あるいはユーザインタフェースなどが比較的簡単に実現できるようになってきました。また“手続き的”なもの以外に、“オブジェクト・オリエンティッ

ドな環境(含む言語)”が増えつつある、あるいは広まりつつあるというのもプロトタイピングのよいタネであると思います。さらには、ソフトウェアエンジニアリング全般に関係してですが、プロセスプログラミングのような技法/技術を多数の人が提案するようになって、それらをプロトタイピングのところで積極的に使える状況にあります。

それからやっぱり一番大きいのは、「手作業でやってきた仕事を CASE ツールなどを使ってやっていくんだ」といった考えが非常に広まってきていることが、大きな助けになっているような気がします。

また、プロトタイピングのツールそのものを見ましても、昔の紙芝居程度のものから、よりきめの細かいシミュレーションであるとか、動的な振る舞いなどに注目したものも出てきている。そのようなことから『驚異的な技術的な革新は無かったといわざるを得ないのですが、かなり希望を持てる状況になっている』と思います。

5. 終わりに

これからは、従来からの要求仕様の分析/定義の技術とプロトタイピングが、もう少しインテグレートされないといけないと思います。それは論理的な枠組みでも、物理的な枠組みとしてもです。またプロトタイピングに限らず、何でも同じなのですが実際に使うユーザの環境というのは千差万別です。ですから、1つの仕掛けが全ての場合に有効だということはまず有り得ないわけですので、いろいろと事情の違う環境に適合したプロトタイピングのしかけというものを準備しないとけません。具体的には、例えば adaptation というような概念で、プロトタイピングの環境を生成するような仕掛けをあらかじめ作成しておき、それを個々の環境に特有なパラメータを入れて、その特定の組織等で使えるプロトタイピングの環境というのが準備されないといけない。これは、プロトタイピングとして遊離しているのではなく、要求仕様全体について、それらをどう集めながら、定義しながらやっていくか、それらを融和した仕掛けであり方法論であることが必要であると思います。

それから個々で作られたプロトタイピングあるいは、場合によっては要求仕様のスペックということになるかもしれませんが、そういうものが本番システムの開発というところで、できるだけ流用できるように考えていかなければなりません。またこれからは、こういう仕掛け自体も、やっぱりもっともっと知的にならないとしょうがないのではなからうか?という気が致します。

こういったことが実現できれば『技術的な革新がプロトタイピングに於いてもあった』といえるのではないでしょ

うか？

6. コメンテータとの討論

〔佐原〕宮本先生の話は非常に整理されていて、コメントのしようがないというのが本当のところですね。といいますのは、プロトタイピングは宮本先生が最後におっしゃられた方向に行くのが正しいだろうと私も思っているためです。ただ、具体的にプロトタイピングをどう進めていったらよいのか？ということに疑問があります。

あと、Rapid / Use とかいったツールを実際に使っているのですが、今あるプロトタイピングツールというのは、先ほど宮本先生もお話になったと思うのですが、全て適用領域が限られているというか、制限があるわけです。今プロトタイピングを実際にやろうとすると、私の場合ですと、まずハイパーカードを使ってユーザインタフェースをチェックして、そしてそれを CASE ツールの方でもう少しフォーマルな形で仕様を定義してみて、それを今度は CASE ツールの中にあるプロトタイピングツールを使って動かしてみるとかいうことをやらざるを得ない。そうすると、一旦覚えてしまえば楽なのですけれども、それらを覚える知識量が大変という意味で負荷が多くなります。そして技術的にもこの部分を変更するためにこのパラメタをどうやったらよいのだろうか？といった膨大な言語を覚える努力が必要になって、プロトタイピングがなんとか上手くできるというのが状況です。

そのような観点からすると、宮本先生のおっしゃられた方法に進んでいくためには、『何がブレークスルーとなりうるのか？』ということをもうちょっと話して頂ければと思います。

〔宮本〕個々のしかけについて説明していないので、1つ1つの作業手順をどういう観点を1番目にやってどういう風なやり方をやる、というものを用意しました。逆にそういうものがないと学生にはやらせられないわけですが、それでも個々のシステムには、方法論があると理解して頂けるとよいと思います。

〔佐原〕先ほど、プロトタイピングには大きな変革の可能性があるのだけれどそれが見えていない、というふうにおっしゃいましたけれど、先生にはその一筋の光が見えているのではないかと思いますのですね。

〔宮本〕我々のところでも経験の足りないものもありますし、アプリケーションの分野によってはほとんど知らない分野もございます。ただ一つ気をつけていることは「非常にフレキシブルにプロトタイピングの仕掛けを作っている」ということです。ですから、モデルがいくつもあります

けれども、それらモデルを変えることも、またその中の記述を変えることも簡単にできるようにしてあります。色々な試行錯誤が簡単にできるように心掛けています。

ということは何かといいますと、現在デモで見えているものや雛形としてシステムに入っているものは、プロトタイピングの機能であるとか、手順であるとかは思っていない。フォーマリズムの記述の仕方を変えられるだとかといったような仕掛けをし経験を積んでおけば、近い将来それらを実現する仕掛けや方法論が分かるのではないという気がしております。

〔藤野〕宮本先生のやっておられるプロトタイピングシステムのチェックポイントを全部拾い上げてみると、現実のシステムとプロトタイピングが目指すものとの関係がとれなくなってしまう。例えば、さきほどのパフォーマンスの問題でも、Completeness (完備性) を現実世界で証明しうるかというところではないかと思えます。また consistency (無矛盾性) についても同様に現実世界では証明できないのではないのでしょうか？。そうすると、先生のおっしゃっているプロトタイプ・システムというのは、まだ現実世界にはないシステムであるのかなあと感じてしまったのが1つ。もう一つ、プロトタイピングのゴールをどこに設けるのかといった問題があるのではないかと思います。それで、先生のプロトタイピング・システムのゴールは、要求仕様を作ったユーザあるいは顧客みたいな人が、容認できる／できないといったことが評価のポイントになるのか？、それともさっきいったようにフォーマルに評価基準といったものを出すようにするのでしょうか？

〔宮本〕要求仕様として consistent なもの、設計者が設計を進めていくうえで必要なレベルでの completeness をきちんとやろうというのが狙いです。ですから、現実世界での consistency であるとか completeness といったものではありません。要求仕様が色々な人から出てきたときに、現実には声の大きな人の要求仕様が勝っているような気がするんですけど、その辺を公平に扱うとか、あるいは開発コストであるとか運用コストであるとかフィジビリティなんかの問題をからめての要求仕様を評価していかなければならないんです。しかし現在はそういう仕掛けがない。だから、我々のプロトタイピングの環境でしたら色々な要求仕様が矛盾があればあったでもモデルとして記述できるし、またそれをシステムの振る舞いとしてアニメートするようなことも可能である。そして矛盾する要求仕様があった場合に、システムはそこで止まってしまうので、ユーザには矛盾が何かが容易に分かる。

〔藤野〕そうすると、先生のアプローチというのは、SREMなんかでやっているように要求仕様がきちっと書けるならばという前提に立っていますね。そして、それに対してのサポートツールとしてのプロトタイピング・ツールということですね？

〔宮本〕え〜と、そうではないつもりなんですけれども、とにかくむちゃくちゃな要求仕様でも何も落とすことなくワークモデルとして記述して、それを見てユーザ画面では、併せてこんな図をいれたらとか、といった新しくより詳細な要求仕様を引き出す役割をするわけです。一方 SREM などでしたら、軍事システムを相手にしていますから、コンサルティング会社のようなところが要求仕様 spec を出すだけのプロジェクトを起こしてしまうんですね。それで、要求仕様 spec が出てくると別のコントラクターがデザインしてインプリメントしようというプロジェクトを起こすわけです。ですから、要求仕様としてかなりのところまでそろいます。でも、一般的には揃わないですね。ですから、むしろ SREM とは逆の方向だと理解して下さい。

どんなすごい分析者（システムアナリスト）でもエキスパートと呼ばれている人でも、きちんとコンパクトな形で consistent に complete に要求仕様を表現するということは、多分無理なことだと思います。ましてや、エンドユーザが自分の欲しい物すら表現できないのです。そうしますと、むちゃくちゃでもなんでもよいですから、それは例えば現状のシステムの不満でも、あるいは漠然とした新しいシステムに対する希望でも何でもよいんですけれども、そういったものをモデル化して次に発展させようということです。

〔藤野〕続けて質問してしまうと、特にパフォーマンスのクライテリアのチェックのところなんですけれども、例えば非常に大きなシステムの臨界性能みたいなものをチェックしようとする、ほとんどプロトタイプではできないと思うのです。例えば、さきほどいったリアルタイムシステムでも、何マイクロ秒といった程度のものを扱おうとする場合などですと、様々な計測のためのルーチンなどが入ってしまうと、オーバーヘッドの方が大きくなってしまいます。ということは、臨界性能でプロトタイピングするというのは非常に難しい場合が多いと思います。それから、大きなデータベースに対するある一時点でのトランザクションに対する処理のフィジビリティみたいなものをプロトタイプでチェックするというのは難しい面があると思います。そういった点はどの様にお考えでしょうか？

〔宮本〕それはおっしゃる通りだと思います。ただ、実際の現実のシステムに近い環境（含むプログラミング言語）で部分的にアルゴリズムを試してみてもスピードを測るとかというのは1つの方法です。また、もう1つモデルでやる場合があるのですけれども、この場合は、今おっしゃられたようなことはほとんどできません。ただ、できることは何かといいますと、いろいろな機能に対する要求仕様がありますが、システムのレスポンスタイムのようなものでしたら、そのレスポンスタイムが、ユーザから何か出て、ユーザにその結果をもらうまでに、いろいろなファンクションが関わり合うわけです。システムの中と外で、例えばレスポンスタイムが2秒としたときに、いろいろなファンクショナルな要求仕様にどう割当てられてゆくかというシステム内部の詳細なところのファンクションに対する問題と、全体に対するそれと矛盾が無いかな否かといったことをチェック出来ますし、ここが大事です。簡単な例をいいますと、銀行のオンラインシステムのレスポンスタイムが1秒でなければいけない時にシステムの中でディスク I/O を100回もやったらもうシステムとして間違いです。要求仕様はどちらが間違いという訳ではないんですけれども、1秒というレスポンスタイムに対する要求仕様がもっと詳細なファンクションレベル（例えばディスク I/O とか種々の演算など）での性能の要素との矛盾しているということがチェックできます。

実際はプログラミング言語を使ったりしていろいろとやっても確かめられることはかなり限られているわけで、やってみないと分からないという所はあるのですけれども、1つのアプローチとしましては、要求仕様のところで例えば50%くらいエラーの誤差範囲でその性能を保証する。次に基本設計の段階でその誤差率を±20%位に抑えていく。それはシステム設計では、もう少し情報が入るわけです。それを基に、プロトタイプを作ればよいわけです。それで詳細設計のところまでできますとさらに補完する情報があるわけで、誤差が±10%位でパフォーマンスを保証していく。このようなアプローチが現実には重要になってくると思います。

〔藤野〕質問に答えて頂けたので、コメンテータとしての私の意見を述べたいと思います。昨今、現実世界のシステムを考えてみると、最近のシステム開発は正直いいまして「要求仕様を書くまでが大変だ」という状態になってきていると思います。1度要求仕様が書けてしまうという状態になると、その先は様々な実際には現実の state of practice の技術とどういう風にマッピングがされるかという solution

domain へのマッピングの作業ですね。(実際にはデザインになってくるとは思います)そして要求仕様を作っている時というのは、抽象度のレベルが物凄く高く、ユーザ(顧客)自身もあり正確には意識していないし、特に昨今のシステムへの要求仕様というのは、現時点での人間の作業を機械化するというようなものではなくて、全く世界中に存在していないような全く新しい形の、つまり戦略的なシステムを作ってほしいという要求仕様がかなり多いわけです。そうすると、先程おっしゃったように要求仕様を出してもらって、それをワークモデルとして作成しテストするのはなくて、逆に現実の世界で今求められているのはそのような形で解、つまり要求仕様を含めてそこから提案されるというような形のものが要求されだしてきているのではないかなと思うんですが、そこらへんをどうお考えになりますか?

【宮本】それは非常に単純でして、誰かが「こういう物を欲しい」と不完全であろうが矛盾していないのが、最初に(要求仕様が)無いと駄目なわけです。それはプロトタイピングという環境では、あるいは概念の枠組みではどうしようもありません。最初の minimum な要求仕様はないと、プロトタイプできませんので。

そういうこともありまして、従来からのシステム分析/定義とのうまい融和というのはやっぱり必要になってきてまして、皆さんそれぞれおやりになっていると思うんですが、ブレーンストーミングであるとか KJ 法であるとか、新しい発想を新しいコンピュータの利用方法を色々と create するのを試すためだと思うのですが、そういったものの融和が必要です。そして、何か(要求仕様にあたるものが)出てきたならば、それをプロトタイピングの方でちゃんと色々な観点から保障していく、ということだと思います。

【藤野】一応、コメンテータの立場としては、私のところで行っているプロトタイピングというのは、どちらかというとか概念的なモデリングに近いわけです。そして実際、そうしたモデリングをやっていく時に、自分たちの経験が少ないので、トレードオフのファクターを評価できていないわけです。経験が乏しいために、モデリングの試行錯誤を続けることによって、トレードオフに関する知識を自分自身が得ていくという感じです。そのような形でプロトタイピングというのを私たちは使っています。特に私たちの環境では、Smalltalk-80 を使っているということもあって、プロトタイピングというとはほとんど概念モデルを作るために、あるいは自分の考えているモデルが果たして正しいかどうか

を確認するために使っていることが非常に多い。

先程、宮本先生もおっしゃっていましたが、ふつうプロトタイピングといった場合には、現実世界ではどちらかというと要求仕様ははっきりして、その先の設計フェーズのトレードオフのチェックのようなものに使われるのが、現状ではやはり多いのかなと思います。もしそういう形で、プロトタイピングというのが要求仕様フェーズ以降に、あるいは要求仕様フェーズで役に立つのであれば、かなりそれはエンジニアリング的なアプローチでも現時点でかなり使えるものが、かなり多いと思うので、できるだけそういったものをアプライしていくことが肝心なのではないか、という気がしました。

私が1つコメントとしていいたいのは、日本では、実際に使ってそういった経験をしていくというよりは、こういう所(ソフトウェア・シンポジウムのような場)でこういう話を聞いてみようとか、購入する前に調査してみようという話が長すぎてしまって、実際の経験だとかを交流していくというステージになかなか移っていけないのが問題だと思います。

【参考文献】

- [1] COMPUTER, IEEE, Vol.22, No.5 (1989.5) rapid prototyping の特集号である。そのうち、指摘のあったシステムに関するものは以下の2つである。
- [2] Luqi: Software evolution through rapid prototyping, pp.13-25
- [3] Lewis, T.G, et al: Prototypes from standard user interface management systems, pp.51-60
- [4] PWBに関しては、色々と資料があるが、入手しやすいものとしては、ソフトウェア・シンポジウム'88 論文集 pp. 415 - 418。また、近々発行される予定の SEAMAIL「プロトタイピング特集(2)」でも、開発に携わっておられる塩谷さん(SRA)の解説講演の記録が掲載される予定です。

HyperScore Tool Kit Ver- 5.2 (Smalltalk-80上の音楽ツール)

富士ゼロックス情報システム(株)
森口岳雄

はじめに

Smalltalk-80上の音楽ツールであるHyperScore Ver5.2の調査報告をする。作者のStephen Travis Popeは、Parc Place Systems社に勤めるコンピューター技術者であるが、ウィーン音楽アカデミーやモーツァルトテウムなどで学んだ現代作曲家でもある。彼の勤めるParc Place Systems社はXEROXのパロアルト研究所でSmalltalkの研究・開発に携わっていた研究者らがXEROXから独立してできた会社で、Smalltalk-80はParc Place Systems社の商標になっている。HyperScoreはPope自身が作曲を行うのに使用しているツール群であり、Smalltalk-80の音楽用プログラム・ライブラリでもある。

HyperScoreの動作環境

Smalltalk-80はオブジェクト指向言語の代表格として高く評価され、従来の手続指向型言語とはまったく異なる概念をもつインタラクティブなプログラミング環境を提供している。Smalltalk-80は現在、多くのワークステーションで動いているが、HyperScoreの使用には、MIDIのドライバやSmalltalk-80の値段のことを考えてMacintoshを薦める。他の機種はSmalltalk-80からRS-232Cを通してMIDIインターフェースに出力するようになっているが、MacではMIDIドライバを組込んだSmalltalkの仮想マシンができあがっており、Smalltalkの環境から直接MIDIデータを出力することができる。PopeもMacを愛用しているようである。

メモリは、最低4メガバイトは必要であろう。私はMac-2にメモリを8メガ搭載し、大型モニター・ディスプレイで使用している。

EventListエディタ

EventListとはMusicEventを扱うためのものであり、どのような音をどのようなタイミングで演奏するかのデータを保持している。これについては後で詳しく説明する。このEventListをグラフィカルに表示したり修正したり演奏したりするのがEventListエディタである。EventListエディタはHyperScoreで最もよく使われるウィンドウで、ウィンドウの形に色々なバリエーションがあるが、標準的なものを図1に示す。

見ておわかりのように、中央の領域は縦軸に音程を、横軸は時間を表している。c3,c4,c5...と書かれてるのは音階の下で、横線は黒鍵の位置を表している。HyperScoreのマニュアルでは、このような記法をHauer-Steffen記法と呼んでいる。この領域は、スクロールバーを用いて上下左右にスクロールすることができる。右の領域に縦に白まるや黒まるや三角が並んでいるのは音色(Midiのチャンネル)を選択するボタンである。左側のff,f,mf,mp,p,ppが縦に並んでいるビューは、音の音量を決めるサブ・ビューである。いちばん左には八分音符や十字などの形をしたボタンが並んでいるが、これらは編集や演奏に使う。

音を入力する時は、音色と音量を決め、出したい音の位置へマウスをもってゆき、そこでマウス・ボタンを押し、押したまま横にずらして音の長さを決めてボマウス・ボタンを離す。演

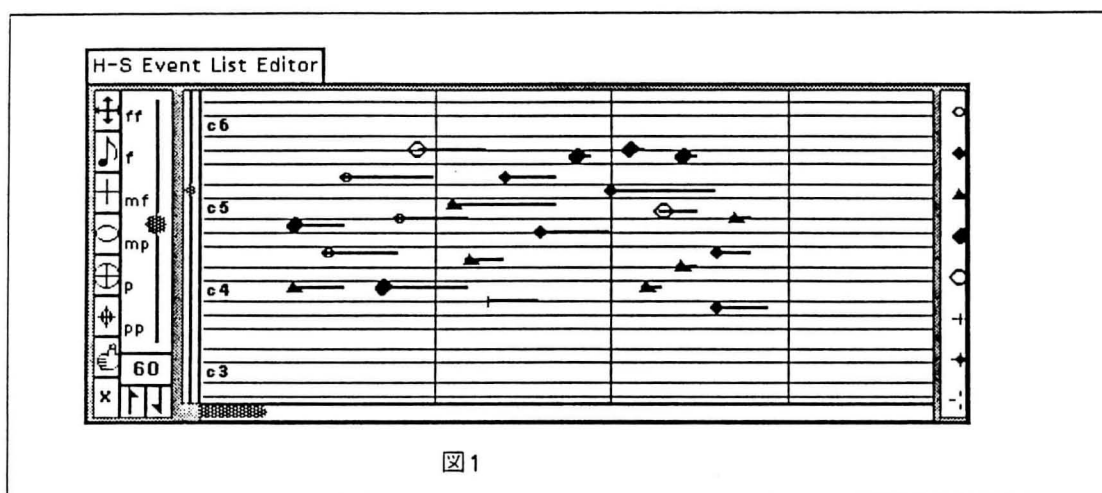


図1

奏は、ウインドウの左にある八分音符の形が描かれたボタンの位置でマウスをクリックすると、このEventListエディタの中にある全ての音がMidiアウトから出力される。

HyperScoreのクラス階層

音程、和音、旋律などの音楽情報も、HyperScoreでは全てオブジェクトとして表現され、それぞれの振る舞いがクラスに記述されている。そして、オブジェクトにメッセージを送ることにより実行がなされる。例えば、和音を表すオブジェクトにplayというメッセージを送ると、和音は自分の和音構成にしたがって演奏を始める。

HyperScore Tool Kit Ver-5.2には、約70のクラスがSmalltalkの環境に加わるが、音楽情報そのものを表しているクラスはMusicMagnitudeとそれに属するクラス、および、MEventとそれに属するクラスである。これらのクラスのクラス階層を次に示す。これら以外のクラスは、MIDIによる演奏のサポートや、グラフィック・ユーザー・インターフェースなどに用いられる。

```

Object()
  Magnitude( )
    MusicMagnitude( 'value' 'type' )
      Amplitude( )
      Duration( )
      PField( 'name' )
      Pitch( )

Object()
  MEvent( 'duration' 'name' 'props' )
    NoteEvent( 'voice' 'pitch' 'loudness' )
      EventList( 'events' 'links' 'index' 'tempo' 'modifiers' 'code' )
        EventGenerator( )

```

```

Cloud('density')
  DynamicCloud('startEvent')
  SelectionCloud()
    DynamicSelectionCloud()
Cluster()
  Chord('root' 'inversion')
    Arpeggio('delay')
Roll('number' 'delta' 'noteCuration')
Trill()

```

MusicMagnitudeとそのサブ・クラス

MusicMagnitudeはMagnitude(尺度を表すクラス)のサブクラスとして定義されている。valueとtypeという2つのインスタンス変数(内部状態)があり、valueは値そのものが、typeにはvalueの型が入る。このクラスはこのサブクラスであるAmplitude(音量)、Duration(長さ)、Pitch(音程)、PField(その他)の共通概念を記述した抽象クラスである。Amplitude、Duration、Pitchのインスタンスを生成するにはそれぞれのクラスにvalue:というメッセージを送るが、引数には様々な型が許されている。

Pitch value: 60	; Integer型	(MIDIの中央のド)
Pitch value: 440.0	; Float型	(周波数)
Pitch value: 'a5'	; String型	(5オクターブ目のラ)

Amplitude value: 64	; Integer型	(MIDIのベロシティ)
Amplitude value: 0.71	; Float型	(割合。71%)
Amplitude value: 'mf'	; String型	(メゾ・フォルテ)

Duration value: 500	; Integer型	(500ミリ秒)
Duration value: 1.5	; Float型	(1.5秒)
Duration value: 1/4	; Fraction型	(1/4拍) ???

また、他の型へ変換するメッセージ(asMidi,asHertzなど)も用意されている。

クラスMEventとNoteEvent

MEventはMusicEventに関する抽象クラスで、duration,name,propsの三つのインスタンス変数を持っている。

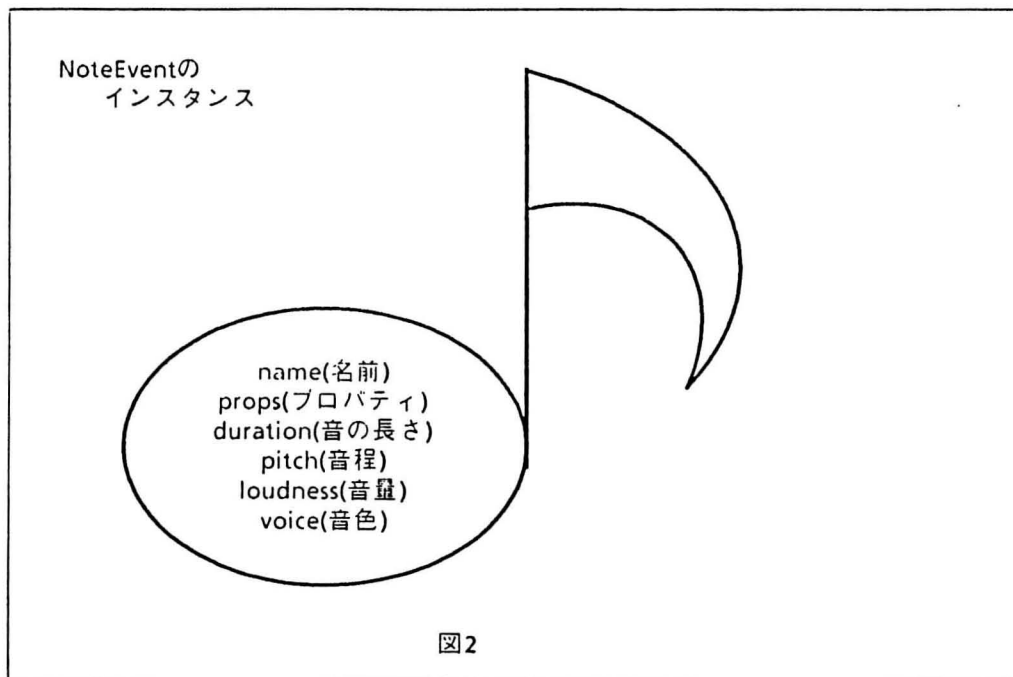
MEventのサブクラスにNoteEventというクラスがあり、このクラスは単音を表すクラスであり、voice, pitch, loudnessという3つのインスタンス変数が付け加えられている。

NoteEventのインスタンスは次のようなメッセージで生成される。

```

NoteEvent dur: 1000 pitch: 36
NoteEvent dur: 1000 pitch: 36 ample: 100
NoteEvent dur: 1000 pitch: 36 ample: 100 voice: 2

```

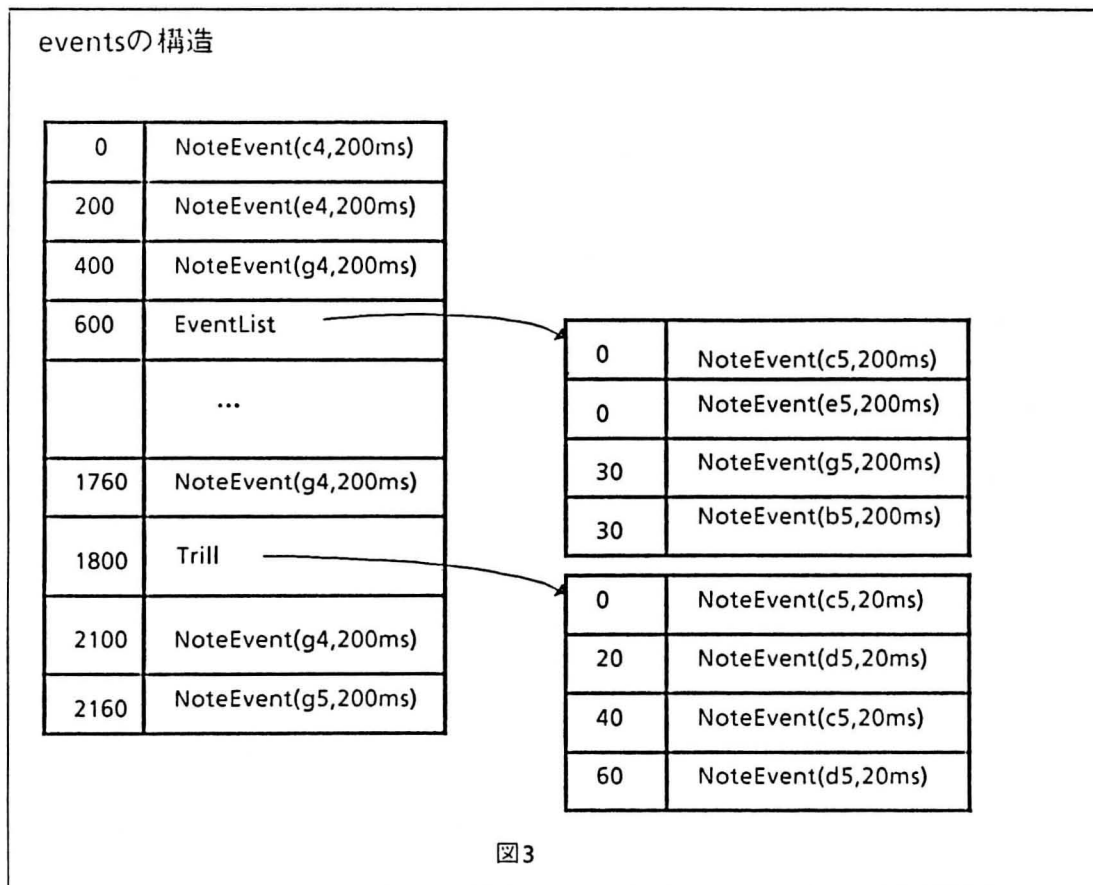


音量や音色を省略した場合はデフォルト値がとられる。また、これらのメッセージの引数には、数値以外に様々な型を用いることができる。

NoteEvent dur: 1000 pitch: 'c5' ample: 'mf' voice: 2

クラスEventList

EventListはNoteEventのサブクラスとして定義され、events, links, index, tempo, modifiers, codeの6つのインスタンス変数がある。なかでも一番重要なのはeventsというインスタンス変数で、ここに音の情報とその音をだすタイミングが対となって格納されている。ここにしまわれる音は、単音(NoteEvent)であってもEventListであってもよい。また、後に説明するが、EventListに属する和音(Chord)やトリル(Trill)であってもよい。



EventListおよびそのサブクラスのインスタンスに、playというメッセージを送ることで演奏され、また、editというメッセージを送るとEventListEditorが開く。

EventListのインスタンスは、それぞれに名前をつけて登録したり、その名前で参照することができる。また、EventListのeventsの中には単音のみならずEventListを入れることもできるので、曲を階層構造で表すことができる。たとえば、バイオリン、フルート、チェロの3つのパートからなる曲では、それぞれのパートをEventListとして表し、曲全体のEventListの中に3つのパートのEventListを持たせることができる。また、パートごとのEventListの中にも、小節やフレーズごとにEventListを、持たせることもできる。このようにEventListは曲全体としても表すことができ、またメロディーの一部としても表すことができる。

EventListのサブクラス

EventListのサブクラスの主なものに、Roll(連打)、Trill(トリル)、Chord(和音)、Arpeggio(アルペジオ)、Cloud(??)がある。図4にこれらをEventListEditorによってHS記法で表したものを示す。

The screenshot displays the EventListEditor interface for a file named 'demo0404.w3'. The interface is divided into two main sections: a piano roll on the left and a parameter editor on the right.

Piano Roll: The piano roll shows four sub-classes: Roll, Trill, Chord, and Arpeggio. Each sub-class has its own staff with musical notation. The Roll staff shows a continuous sequence of notes. The Trill staff shows two notes alternating rapidly. The Chord staff shows a single chord. The Arpeggio staff shows a sequence of notes played in order.

Parameter Editor: The parameter editor on the right contains the following settings:

- Roll:** ((Roll length: 2000 rhythm: 80 note: 60) ampl: 100) edit
- Trill:** ((Trill length: 2000 rhythm: 60 notes: #(60 62)) ampl: 100) edit
- Chord:** ((Chord majorTetradOn: 'c4' Inversion: 2) duration: 1000) edit
- Arpeggio:** ((Arpeggio majorTetradOn: 'f3' Inversion: 2) duration: 1000) delay: 100; edit

Below the piano roll, there is a section labeled 'Arpeggio' with a staff showing a sequence of notes (c6, c5, c4, c3) and a '60' value.

Figure 4 is indicated by a small box with the number 4.

図を見てわかるように、Rollは同じ音程の音が連続している。また、RollのサブクラスのTrillは、二つの異なる音程の音が交互に連続している。Trillは音をどのように刻むかなどの振る

舞いはスーパー・クラスのRollにある記述を利用し、TrillクラスではRollとの差分のみが記述されている。

同様に、Chordは複数の音を同時に演奏するものであり、ChordのサブクラスであるArpeggioは複数の音を、ずらして演奏するものである。Arpeggioは和音構成に関する振る舞いはChordの記述を利用し、Arpeggioクラスにはタイミングをずらすことに関する記述のみが追加されている。

Cloudは現代作曲家のPopeらしく、乱数を発生させてフレーズを作るクラスである。でたために乱数を利用するのではなく、音程や音量の幅を制限したり、フレーズの始めと終わりで条件を変化させたりすることにより、いろいろ変わったフレーズを生成することができる。

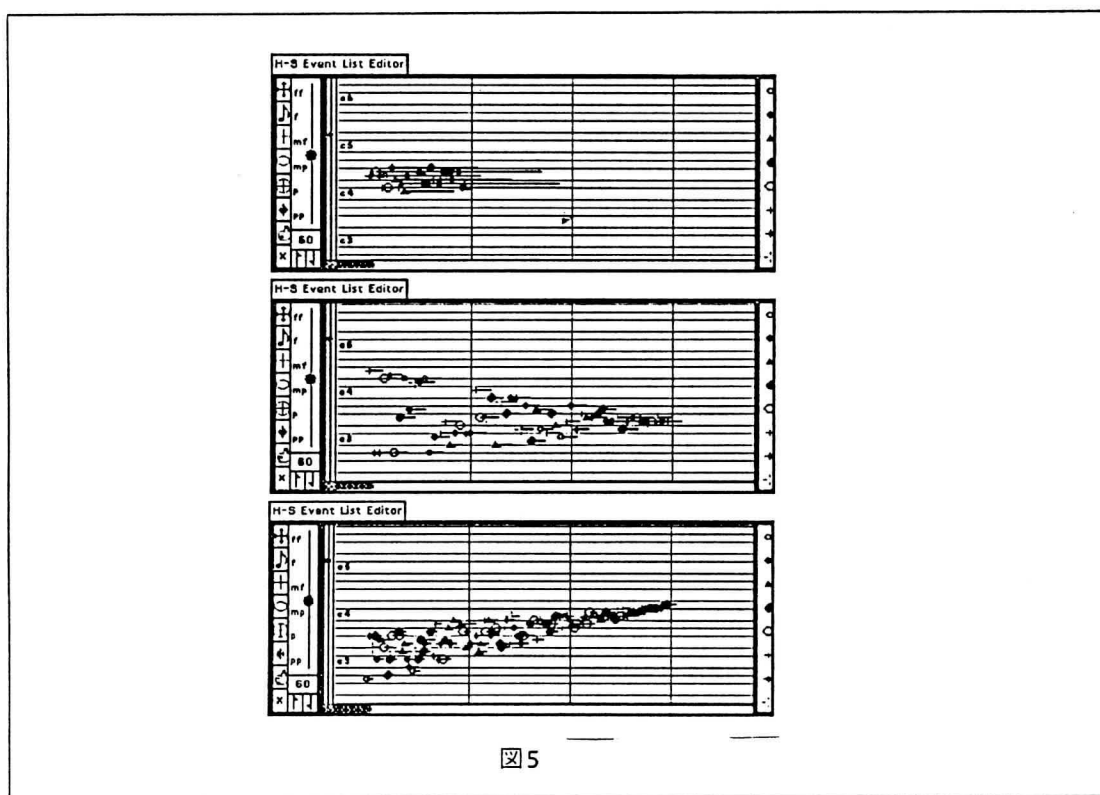


図 5

これらのクラスはみなEventListのサブクラスなので、playやeditなど共通のメッセージで演奏したり編集したりすることができる。またこれらはEventListのeventsの中に含ませることができる。

AdagioParser

CMUで開発された楽譜記述言語Adagioで書かれたスコアをHyperScoreに取り込む為にAdagioParserというクラスが用意されている。Adagioの記述方法はBASICなどでおなじみのMMLとよく似ている。「ドレミファソラシド」をAdagioで書くと

```
!tempo 300  
c4; d; e; f; g; a; b; c5
```

となる。詳しくは知りたい方は「CMU Midi ToolKit handbook」を参照していただきたい。

AdagioParserはAdagioで書かれた曲を読み込んで、Scoreというクラスのインスタンスを返す。このScoreのインスタンスにasEventListというメッセージを送るとEventListに変換したものを返してくれるので、これにplayやeditなどのメッセージを送ることにより曲を演奏したり修正することがができる。簡単なメロディーや既存のスコアを入力するときは、Adagioで記述し、それをEventListに変換するのが良いであろう。

最後に

HyperScoreの特徴は、EventListなどにみられるデータ構造の美しさ、それとSmalltalk-80環境のどこからでも音楽に関するオブジェクトが容易に扱える点である。

一般に市販されている音楽ソフトの中にはHyperScoreに比べ、入力方法や演奏機能に優れているものがいくつもあるが、これらは用意されている機能しか使うことができない。HyperScoreはSmalltalk-80の環境が控えているので、プログラムによって旋律を生成したり合成するなど、いろいろな実験を行うことができる。また、ユーザーがHyperScoreのツールの機能を拡張したり、AIの手法を用いた作曲・編曲システムや楽譜印刷システムなどに利用することも可能である。

私もHyperScoreのプログラム・ライブラリを使用していくつか新しいツールを作成してみたが、プログラム作成にはBASICやC言語では味わうことのできない快適さがある。

ただ残念なのは、Popeが一人で作っているためかVer5.2になっても未完成の部分が多く、またバグもいくつか残っている。アメリカではHyperScoreユーザー会が発足し、情報交換の場となっているが、日本にもユーザーを増やしHyperScoreをより良いソフトへと発展させていきたい。

UNIX VS MS-DOS
ソフトウェアの開発環境は、どちらがよいか。
現場からはっきり言ってしまう

横山博司
SRA(関西)

制御システムの開発において、かねてより、開発環境に気遣いながら、データ収集に励んできた、集大成として、ここにまとめたものである。

データの収集は、1985年からのあしかけ5年間に及ぶもので、この間に担当した、システムの中から、厳選した、3件について簡単明瞭に示す。

また、そのうち、メンテナンスをしたときのデータも掲げて、判断材料の一つとする。

すべてのシステムについては、以下に示す特徴を持っているので、格好の比較対象である。

- ①MMI(マンマシンインターフェース)であること。
- ②ターゲットマシンは、インテルのI8086シリーズを使用したマイコンのシステムである。
- ③マルチタスクによるリアルタイム処理をしている。

表1 をご覧下さい。

表に示された、「プログラミング」とは、ソース入力から一部単体テストまでの作業を示し開発マシン上で行なった。また「システムテスト」は、ターゲットマシンで行なった。

さて、結果を見てみると、バグ発生率と生産性の両方とも、AシステムよりBシステム、BシステムよりCシステムの方がよくなっている。

しかし、興味深いことにシステム全体の生産性を比較してみると、新規にしても、メンテナンスにしても、ソース修正とテストが同一マシンでできるほうが、約2倍効率がよいという同じ傾向を示している。

(Cシステムの生産性/Aシステムの生産性 $\div$ Cシステム改造の生産性/Aシステム改造の生産性)

表1 開発環境と信頼性、生産性

システム	開発期間(月)	投入工数(人)	開発環境		ターゲットCPU	実効ステップ数(s)	システムテスト バグ数(b)	バグ発生率(b/s)	生産性(s/m) STEP/MM
			プログラミング	システムテスト					
A 新規	11	63	UNIX/TEST_TOOL	iRMX+Hi	i8086	14396-PL/M	169	1.174	229
B 新規	8	76	MS-DOS=ICE=iRMX	同左	i80286	21140-C	216	1.022	278
C 新規	12	92	MS-DOS/CPMGR	同左	i80386	45680-C	381	0.834	497
A 改造	4	6	UNIX	iRMX+Hi	i8086	14413-PL/M	9	0.062	2402
C 改造	7	13	MS-DOS/CPMGR	同左	i80386	56735-C	82	0.182	4364

開発工程は、すべて、基本設計からシステムテストまで。

改造システムの実効ステップ数は、改造後のシステム全体の数を示す。

PL/M ; PL/M-86 言語

C ; Lattice-C 言語

その差は、どこから出てくるのか調べてみると、

(1) 開発マシンと、ターゲットマシンとの距離。

Aシステムは、FDDにて、ソースマシンから、ターゲットマシンに運んだ。

Bシステムは、ICEを経由して、GP=IBで、接続されている。

Cシステムは、開発マシンそのものが、ターゲットマシンである。

(2) コンピュータの処理速度。

①開発マシンの処理速度。

A システムは、1-CPU に20台の端末が接続されていて、レスポンスが遅い。

B, C システムは、スタンドアロン。

C システムのCPU は、B システムの約10倍の処理速度を持つ。

(それぞれ 16bit, 32bit CPU)

②ターゲットマシンのCPUの違いによる、1回のロード、実行の処理速度。

PL/M言語とC言語を比較するという大胆なことをしているわけだが、およそC言語の方が、1ステップで書ける機能は、PL/M言語より多く(例えば、for文は、C言語にしかなく、1ステップとして数えられる。)、同一機能のアルゴリズムの場合だと、C言語の方が、少ないステップ数でこ

と足りる。
仮に、これはあくまで、仮定の話だが、AシステムをC言語で作ったとすると、システムのステップ数は、少なくなり、結果的に生産性を下げることになる(投入工数が変わらないという前提で)。すると、数字の上での生産性の差は、大きくなる傾向があり、今回の主旨である開発環境の比較結果は、正しさを増すことになるので、ご容赦願いたい。

また、経験から、慣れたプログラマーだと問題はないが、不慣れたプログラマーだと、汎用性の高いC言語の方が、PL/M言語より悪戦苦闘するようである。(どうしてコンパイルエラーになるのか、とか、コンパイルは通っているにもかかわらず、実行結果が期待通りでないときの、原因がつかめない、など。コンパイラーのエラーが不親切だったり、甘かったりするところからくるようだが。)

各システムのメンバー構成は、経験者、未経験者の比は、余り変わらないので、C言語の方が、生産性の低下への影響が多い。にもかかわらず、A、B、Cの順に生産性が、高くなっているということは、開発環境の比較結果は、これも正しさを増す傾向を示している。-数字がどれくらいのものかは、よくわからないけれど。

さて、公平さをきすため、表2に、改造対象部分のみについてのデータを示す。

表記されていない数値は、すべて表1と同じ。

改造実行ステップ数は、削除数+追加数+更新数である。

(改造による影響を受けるが、修正不要な部分は含まれない。)

表2 改造部分のみのデータ

システム	改造実行ステップ数(s)	バグ発生率(b/s)	生産性(s/m) STEP/MM
A改造	695-PL/M	1.295	115
C改造	13565-C	0.604	1043

実は、この結果を見て、ぎょっとしてしまった。

今までの開発環境についての話の裏付けにはなかったが、なんと、Aシステム改造と、Cシステム改造では、バグ発生率で2倍、生産性で9倍もの差がついてしまっている。

しかも、Aシステム改造は、バグ発生率、生産性ともに、新規開発より下がっているではないか。

そういえば、システムに関係しなかった新人が一人加わった。投入工数が少ないだけに、大きく響いているのだ、きっと。それにひきかえ、Cシステム改造は、新規開発時の担当者がメンテナンスを行なった。うん、納得。それにしても、慣れとは、恐ろしいものなのだナァ。やっぱり、メンテナンスは、難しい!!

《お詫び》

タイトルに大きなことを言って御免なさい。UNIXが、ターゲットマシンであれば、またそれは、それで、結果が出ただろうし、しかも、エデッティングマシン+若干の単体テストにしか使っていないので十分なパワーを引き出しているとは言えない。

この場合、タイトルが中身を示していない良い例(?)なわけで、開発マシンと、ターゲットマシンとの関係についての話として理解してもらえれば、結構なのです...(でもまあ、UNIX, MS-DOS共に内容にはでてきたわけで、かすっているでしょ)

シリコンバレーを楽しむために

— その2 —

大神原 薫

San Jose, California

1. ショッピング

海外旅行（出張）の土産物と言うと、有名ブランドのバッグや時計やアクセサリに、酒とタバコとチョコレートと、およその相場は決まっているようで、これらを空港の Duty Free Shop で手っ取り早くかき集めてしまうのも効率的で良いのですが、時にはアメリカ人が普段の買い物をしている店で土産物を探してみるのも一興ではないでしょうか。それで今回は、まずアメリカ風の買い物ができるところを集めてみました。観光客相手の店では買えない物が、きっと見つかることでしょう。

1-1. ショッピングセンタ

いくつかの店が集まっている所という意味でショッピングセンタというタイトルを付けましたが、これには、有名デパートや各種専門店が一つの大きなビルに集まっているショッピングモールと、スーパーマーケットやドラッグストアを中心にファーストフードやレンタルビデオなどの店が軒を連ねるショッピングセンタとに分けることができます。

ショッピングモールの方には、多くの衣料品店や靴屋、宝石店、書店などが集まっていて、中にはけっこう有名な店もあるようですので、ウィンドーショッピングだけでも十分楽しめます。

一方スーパーマーケットを中心としたショッピングセンタの方は、日用雑貨や食料品など、生活に密着した物を扱う店が多く、従って派手さはないものの安くて実用的な品物を買うことができます。

それではいくつか特徴的なショッピングセンタをご紹介します。

1-1-1. Valley Fair Shopping Center

営業時間 月～金 10:00～21:30

土 10:00～21:00

日 11:00～18:00

ここは、Software Etc. と言うソフトウェアショップの

紹介のところで既に出てきたショッピングモールですが、道順を簡単に説明しておきましょう。フリーウェイ 880 を南へ行くと、やがて Stevens Creek Blvd. への出口がありますので、これを西行きへと出ます。この時にはもう右側に駐車場と大きな建物が見えているはずですので迷うことはないでしょう。

細長い2階建ての建物の中がちょうどアーケード街のようになっていて、各種の専門店がずらりと並んでいます。圧倒的に多いのはやはり婦人服の店で、40店以上もあり、以下、靴、紳士服、時計/宝石と言う順ですが、これらに分類されない特徴的な専門店の方が多く、またモールの中央部にはファーストフードの店が集まった一画もあって、家族連れで一日中でも楽しめる場所です。

大手のデパートとしては、Emporium, I.Magnin, Macy's, Nordstromがあり、大抵のブランド物は手に入りますし、また日本では売られていない Made in USA の物も多いようです。

モール内にはとにかくいろいろな店が集まっていて、どこがよいかなどというのは解説の範囲を越えた個人のセンスの問題になってしまいますが、あえて筆者の好みを言わせてもらえば、The Nature Company, Brookstone, Cutlery World などが楽しい店だと思います。The Nature Companyは名前の通り、自然科学に関係した物（地球儀、望遠鏡、鳥や動物の写真集、鉱物、化石、おもちゃなど）を売っています。また Brookstone はアイデア商品やギフト用品、Cutlery World はナイフなどの刃物の店です。いずれもブランド品とは無縁ですが、見るだけでも十分に面白い店です。

ところで、お土産にチョコレートを考えているのであれば、Godiva Chocolatier をお勧めします。日本では See's Candies の方が有名だそうですが、Godiva のミニトチョコは絶品です。

音楽、特にジャズやロックが好きな人には、CD などはいかがでしょうか。同じ物は日本でも手に入るかも知れま

せんが、\$15~20くらいなので日本よりは少し安いでしょう。

デザイナーズブランドのベルト、バッグ、財布などは Macy's や Nordstrom などのデパートで買えます。同じような物は空港の免税店でも買えますが品揃えはこれらのデパートの方が圧倒的に豊富です。価格もデパートの方が安いように思いますが、同一の品物について値段を比較した訳ではありませんので断言はできません。ごく普通に見かける物では、ベルトが\$25~50、バッグが\$100~250、財布はかなり幅があって\$30~200といったところです。

1-1-2. Valco Fashion Park

営業時間 月~金 10:00~21:00
土 10:00~19:00
日 11:00~18:00

ショッピングモールはどこでも同じような構造で、同じような店が並んでいるのですが、気に入った店があるかどうかや店の配置、雰囲気などで好き嫌いができるようです。筆者の場合は、特に理由はありますが、この Valco Fashion Park が一番気に入っています。

場所は、前述の Valley Fair Shopping Center から Stevenson Creek Blvd. を西へ約15分程、 Laurence Expwy. とフリーウェイ280をくぐってから2つめの信号 (Wolf Rd.) を右へ曲がったところにあります。

San Jose の中心部からフリーウェイで行くのなら、880 南行きから280北行き (San Francisco 方面) に入り、3番目の出口 (Wolf Rd.) を南へと出たところです。

ブランド物でなくても品質が良ければよいという方には、Berman の皮製品がよいでしょう。デザインはシンプルですが、大変しっかりしていて、値段もあまり高くはありません。

自然科学に興味のある方には、Natural Wonder や The Science Shop をお勧めします。まさか解剖用のカエルを買って帰る人はいないでしょうが、珍しいおもちゃや本、ちょっとしたアクセサリなど、選ぶ人のセンス次第では良い贈り物が見つかると思います。もっともこういう物は、もらう方の人にもかなりの趣味とセンスが要求されますが、...

一方、余りかさばらない土産物としては、ボールペンやキーホルダーなどがよいようです。Things Remembered というギフト用品の店では、Cross や Sheaffer などのボールペンが\$15くらいから買えますし、また無料で名前を彫り込んでもらうこともできます。キーホルダーは、Key Ring とか Key Chain と呼ばれて、例えばデパートの

Men's Accessory の売り場に行けばデザイナーズブランドの物が\$20くらいから置いてあります。あるいは、Victorinox のアーミーナイフの小さい物 (\$20弱) をキーホルダー代わりのお土産にするのも良いかも知れません。

日本でも話題になったらしい、音に合わせて踊る花のおもちゃのアメリカ版も見かけました。こちらは、植木鉢の部分が瀬戸物でできていて日本の物より高級感がありました。

パズルの好きな方には、世界で最も難しいと言われるジグソーパズルはいかがでしょう。これは、表と裏に同じ図柄が90度回転して描かれていて、さらにカッティングの工夫によってエッジの部分で表裏を判別することができないようになっているものです。日本でも販売されているかどうかは分かりませんが、\$15くらいで、ギフトショップやおもちゃ屋、ゲーム専門店などに置いてあります。

土産物として買った物が、実は Made in Japan だったと言うのはよくある事ですが、この傾向は、高級スカーフや食器などに特に多いようです。確かに、陶器や絹織物は日本ならではの物ですから当然と言えば当然かも知れません。ほかに、衣料品などは台湾製や韓国製が多く、電気製品ではシンガポール製というのよく見かけます。今のアメリカでは Made in USA の物を見つけるのが難しいのかも知れません。

そんな中であって、センスと実用性を兼ね備えたアメリカ製品と言うと、少々かさばりますがタオルなどはいかがでしょう。質の良い物はけっこう高価ですが、カラフルなハンドタオルと香りの良い石鹸を組み合わせるとラッピングしてもらえば、立派な贈り物になります。

なお、どこの店でもそうですが、包装は極めて簡素です。特に要求しなければ品物をレシートと一緒に袋に入れて渡してくれるだけです。ギフト用のラッピングは自分でやるかお金を払ってやってもらうかのいずれかになります。デパートにはギフトラッピングのコーナーがあり、箱、包装紙、リボンなどを選んでラッピングしてもらうことができます。あまり器用でない人にはお勧めでしょう。(\$5~10くらい)

1-1-3. Eastridge Mall

営業時間 月~金 10:00~21:00
土 10:00~19:00
日 11:00~18:00

前述の2つのショッピングモールと基本的には同じですが、この Eastridge Mall の中央部に BASS Ticketmaster というチケット売り場がありますので紹介しておきます。

案内所といった感じのカウンタなのですが、ベイエリア（San Francisco 湾沿岸地域）のスポーツや演劇の切符が手に入ります。ほかのプレイガイドで売り切れになっていても、ここに来れば手に入る事がありますので大変重宝です。

なお、このBASS Ticketmaster では、Lottoと呼ばれる宝くじも売っています。この Lotto は、買うと言うよりも投票すると言うべきでしょうか。売り場においてある紙がマークシートになっていて、1から49までの数字から適当に6個を選んでマークをつけます。係りの人にこのシートを渡して\$1払うと、選んだ数字と日付がプリントされたチケットをくれます。このチケットの6個の数字が、後日発表される6個の当たりの数字とすべて一致していたら一等賞、6個のうち5個が一致していたら二等賞という具合で、当たっていればチケットと引き換えに賞金がもらえるというものです。賞金の額は Pari-Mutuel の式とかいうもので決めるそうですが、なにせ自分で数字を選ぶという方式ですから当選者がいない事もありうるわけで、その場合には賞金は次回へ繰越になります。先日筆者が見たときには、一等は\$25,000,000でした。また、6つの数字にマークを付ける代わりに QuickPick というところにマークを付けると、コンピュータが適当に6つの数字を選んでくれます。

この Lotto は、ガソリンスタンドやスーパーマーケットなどいたるところにありますので、興味のある方はぜひどうぞ。

この Eastridge Mall は、フリーウェイ101を南へ（Los Angeles方面）向い、680/280とのインターチェンジを過ぎて約2マイル、Tully Rd. を東へ約1マイルほど行ったところの右側にあります。

1-1-4. Yaohan

675 Saratoga Av., San Jose

(408) 255-6699

営業時間 年中無休 9:00~20:00

すでに、本シリーズ第1回の Kinokuniya のところで触れましたが、比較的最近できたスーパーマーケットで、日本の食料品などを売っています。フリーウェイ880から280北行きに入り、最初の出口 Saratoga Av. を南へ出るとすぐに右側に見えてきます。

この Yaohan を中心に、ほかに約20店ほどの店が集まってひとつのショッピングセンタになっています。

当然の事ながら日本人客が多く、また店内の掲示板には"XXX売ります"といった広告なども貼ってあったりして、駐在員やその家族にとってはちょっとしたオアシスになっているようです。

ただし、日本から輸入した物はかなり高価です。もともとカリフォルニアは物価が安くて、色々な物の値段から考えると\$1は200円くらいに相当するというのが実感です。から、例えば、缶コーヒーが\$1.5、カップヌードルが\$1.4、納豆100gが\$2.3などは少々高すぎるという気がします。その半面、アメリカ国内で得られる肉、野菜、果物などは一般のスーパーマーケットとほぼ同じかやや高いくらいなので、うまく組み合わせればまあまあと言ったところでしょうか。

日本の風邪薬や胃腸薬などもありますので、アメリカ製が体に合わない人などにはよいでしょう。また、日本のテレビ番組を録画したレンタルビデオもあります。日本ではけっこう楽しく見ていた番組でも、\$2.5を払わなければならないとなると思わず考え込んでしまいますが、それだけの価値のある番組が少ないということなのでしょう。

1-1-5. Nihon-machi

営業時間は店によって違う。

San Jose にジャパントウンがあるなどというと驚かれる方も多いかも知れませんが、実際にあるのです。

San Jose 空港からだと、まず1st St. まで出るのがよいでしょう。1st St. を南へ進むとやがてフリーウェイ880の下をくぐります。その後、HeddingやTailorとの交差点を横切って、次のJackson St. で左折します。すぐに、2nd St. 3rd St. と交差し、次の4th St. の辺りはもうNihon-machi です。

特に駐車場はありませんのでその辺の道路に路上駐車することになりますが、歩いてみて回ったところで1時間とはかかりません。

正直なところそれほど広くもなければ、特に目につく店もなく、また極めて美しいというわけでもありません。どちらかといえば地元商店街とでも言うような雰囲気ですので、観光名所にはなり得ないだろうと思われれます。ただ、それだけ気楽な雰囲気であることは確かで、例えば滞在が長くなって和食が恋しくなってきたときなどには訪れてみる価値があります。特に、Nagasaki のうどんと Gonbei の揚げ出し豆腐がオススメです。

ところで、Los Angeles の Little Tokyo にしても、San Francisco の Japan Center にしても、さらに規模はぐっと小さくなりますがこのSan Jose の Nihon-machi にしても、日本の雰囲気を十分持っている店が多いのに街全体にはほとんどなく違和感があってもなじめないと言う気がするのは筆者だけでしょうか。個々の最適化は必ずしも全体の最適化にならないというシステムインテグレーション

ヨンの鉄則がここにも生きているように思います。

例えば、San Francisco のチャイナタウンなどは、もしかすると本当の中国とは全く違うのかも知れないけれども、なぜか本物の中国を味わったような気にさせてくれるところがあって、目的もなく出かけて行っても場違いな感じを受けることはありません。これは恐らく観光地化という統一された意識で街が作られているからでしょう。中国人が持っている本物の中国の記憶をむき出しにするのではなく、観光客が見れば中国的だとイメージするような雰囲気を作り出しているために、一度も中国を訪れたことのない者にも違和感を与えないのだろうと思います。これは、真実を伝えるためには必ずしも真実をもってする必要はないというプレゼンテーションのテクニックの応用例とも言えるでしょう。

この San Jose の Nihon-machi も、日本情緒を売り物にした観光地とするのか、日本製品を供給するショッピングセンタとするのか、あるいは人と情報の交流を狙ったコミュニティセンタにするのかというような、方向性のある意識で街作りに取り組まなければならない時期に来ているのだと思います。

ともあれ、そういう難しい事は誰かに考えてもらうことにして、ここでは蛇足として（シリコンバレーからは離れてしまいますが）おいしい物を食べたいという方のために、San Francisco の Japan Center の紀伊国屋ビル2階のサッポロラーメンの店（名前は忘れてしまいました）と、Fisherman's Wharf の Tokyo Sukiyaki を紹介しておきます。アメリカで和食なんてと思っている方にはあえてお勧めはしませんがいずれもなかなかのものです。

あ、思い出した。確か「札幌屋」です。あまり愛想は良くありませんでしたが、ラーメンが井でなく播り鉢で出てきたのでびっくりしたのを覚えています。

1-2. 専門店

ショッピングモール内にも数多くの専門店がありますが、一般的にあまり価格は安くはないようです。もし買いたい物のはっきりしていて、なるべく品揃えの豊富な店で選びたいとか、なるべく安く買いたいなどというのであれば、そういう専門店は人に尋ねるかイエローページで探すかのいずれかでしょう。ここに述べるいくつかの店も、筆者がそうやって探し出したものです。

1-2-1. San Jose Camera

1600 Winchester Blvd., Campbell
(408) 374-1880

営業時間 月～土 10:00～18:00

日 休み

日本製のカメラはアメリカではけっこう安く手に入ります。もしカメラでも買おうかと思っているのなら、まずはこの San Jose Camera をのぞいてみるのがよいでしょう。

フリーウェイ 880 を南へ下り、280 とのインターチェンジを過ぎて約 2 マイル、最初の出口 Hamilton Av. を西行きに出ると、やがて Winchester Blvd. と交差します。San Jose Camera はこの交差点の手前右側にある茶色の建物です。恐らくカメラの在庫はシリコンバレーでは最も豊富でしょう。アクセサリなども十分揃っています。価格は次に述べる L. Z. Premium の方が少し安いかも知れませんが、品揃えはこちらの方がずっと上です。

店内のカウンタには、一ヶ所だけ一切の説明やデモンストラーションをしないという窓口があります。これは自分の欲しい物を正確に知っている人にスピーディなサービスを提供するための窓口なのですが、裏を返せば他の窓口ではけっこう詳しく説明してくれるということになります。カメラ用語は日本語も英語も共通で、意味も一意に対応しますから、筆者はよくここで英会話の練習をしました。アメリカの商店にしては珍しく店員の商品知識が豊富です。

1-2-2. L. Z. Premium

1162 Saratoga Av., San Jose
(408) 985-9718

電気製品、時計、カメラなどの店です。少々狭苦しく、あまりきれいでもありませんが、価格は十分納得できます。多少のネゴも可能で、ほかの店で値段を調べておいて、あそこではいくらで売っていたなど言えば少なくとも同じ値段にはなります。

場所は、Yaohan から更に南へ 1 マイルほどのところで、フリーウェイ 880 南行きから 280 北行きに乗り換えて最初の出口の Saratoga Av. を南行きへと出、Yaohan を右手に見ながら更に行くと、Williams Rd. との交差点を過ぎてまもなく、左側に、あまりきれいではない店の並んでいる一画が見えてきます。その一番南の端にこの L. Z. Premium があります。San Tomas Aquino との、信号のある交差点の 50m ほど手前になります。

1-2-3. Oshman's

4080 Stevens Creek Blvd., San Jose

営業時間 月～金 10:00～21:00

土 10:00～18:00

日 11:00～18:00

かなり大きなスポーツ用品店で、在庫の豊富さ、価格、店の広さなど、十分納得できると思います。場所は、既に紹介した Data Source というコンピュータショップ（前回は参照）の隣です。フリーウェイ280を Saratoga Av. で北行きへと出て、最初の交差点が Kiely Blvd. ですのでこれを左折します。すぐに大きな右カーブがあってまもなく Stevens Creek Blvd. と交差しますので今度はこれを右折し、San Jose Nissanの前を通り過ぎると右側にあります。他にも、Palo Altoなどに姉妹店があるようです。

筆者はゴルフをやらないので分かりませんが、ゴルフクラブはアメリカでは極めて安いそうなので、品揃えの豊富なこの店で選んでみてはいかがでしょうか。春先のスキー用品クリアランスセールや、秋口のキャンプ用品セールなどが狙い目ですが、ほとんどいつでも何らかのセールをやっています。テニスラケットのガット張り替えはかなり上手です。また、各種のウェアや靴の種類も豊富なので、あまりスポーツをやらない人にもお勧めできます。

なお、スポーツ用品の店には、このほかに The Big 5 というチェーン店もあります。筆者の知っているかぎりでは、Stevens Creek Blvd. と Winchester Blvd. の交差点から西へ約50m ほど行ったところ（Valley Fair Shopping Center の西端から歩いて6〜7分くらい）と、Milpitas Town Center とにあります。

1-2-4. Best Products

550 West Hamilton Av., Campbell

(408) 374-6630

営業時間 月〜金 10:00〜21:00

土 10:00〜18:00

日 11:00〜18:00

既に、エレクトロニクスショップ（前回）のところで出てきましたが、あえてここで再掲します。

この店には、食べ物以外は何でもあるのではないかと思います。もともとカタログショッピングの店なので、常に新製品を置いているというわけではありませんが、セールの品は絶対にお買い得です。いろんな店を見て回るのが面倒な時は、この店に来れば、まず失敗することはないでしょう。

時計、ネックレス、ブレスレットなどが豊富で、ギフト用品、家電品、旅行鞆などはかなり安いようです。

ここでは、選んだ品物の商品番号を備え付けのオーダーフォームに記入してオーダーカウンタで入力してもらった後、ピックアップカウンタで商品を受け取ってお金を払うという方式です。店内を見て回るのが面倒ならば、オーダーフ

ームと一緒に備え付けてあるカタログで探してしまうのも手軽で便利です。テレフォンオーダーやメールオーダーも受け付けてくれますが、この場合、商品が手元に届くまでに2〜3週間かかるようです。

場所は、前述の San Jose Camera の前を通って Hamilton Av. を更に西へ0.5マイルほど行ったところで、San Tomas Expwy. と交差する直前の左側、Dennys の裏手にあります。

2. レストラン

さて、続いてレストラン編です。短期間（せいぜい10日くらいまで）の滞在ならば、毎日ステーキやシーフード、各国の料理などを楽しむこともできるのですが、2週間を過ぎるころから和食が恋しくなってくるようです。もし1ヶ月以上の滞在が予定されているのなら、ホテルなどではなく、台所のついたアパートを家具や食器付きで借りて、たまには自分で飯を炊くというのが健康にもよいかも知れません。

いずれにしても食事というのは、毎日のことでもあり、また体調にも直接関わることなので、楽しく、おいしく食べたいものです。そこで、割と気楽な雰囲気でおいしい物が食べられるところを中心にして紹介することにします。中には、多少狭苦しい店やあまりきれいではない店もあるかも知れませんが、味の方は折り紙付きです。なお、どんなレストランでも食事時は開いているだろうという仮定の下で、営業時間は割愛しました。なお、普通のレストランでは、ラストオーダーは10時頃です。

2-1. 和食

深田裕介氏が何かの本で、" 駐在員は自分が和食を食べたくて仕方ないので、日本から来た出張員を何とかして和食のレストランに連れて行こうと考えている " というような内容のことを書いておられましたが、実はこれには筆者も心当たりがあります。特に、滞在が1週間くらいになる人を食事に誘う時には、もうそろそろしつこい料理に飽きてきているのではないかと自分の経験に照らして気を遣ってしまうので、ついつい和食を提案することが多くなります。（もちろん、一方では、自分の欲求もあるわけですが、）

中には" わざわざアメリカまで来て和食なんか食べたくない " と思っている人もいたりして、この種の気遣いがよけいな世話になることもあります。量の多さやしつこい味付けのために消化不良になりがちなので、あまり強情を張らないほうが身のためでしょう。また、何を食べたい

かと聞かれて、何でもいいですか、お任せしますなどと答える人が異常に多いのですが、本当に No Idea だったらせて、昨日はXX だったのでそれ以外の物を、というふうに言ってみてはどうでしょうか。何でもいいと言っておいて提案された物には難色を示すという人が実際にいるから困り物です。

また、和食のレストランでは、日本風の雰囲気のために、支払いの時にチップのことを忘れてしまうことがあるようです。チップなんてさげなく渡すべき物で、要求されて渡すのは大変カッコ悪いので注意しておくべきでしょう。

それでは、何はともあれ、和食の店をいくつか紹介することになります。

2-1-1. 北浜

974 South Saratoga-Sunnyvale Rd., San Jose
(408) 257-6449

フリーウェイ 880 南行きから 280 北行きに乗り換えて 4 目の出口 Saratoga-Sunnyvale Rd. (旧名 De Anza Blvd.) で降り、南へと向かいます。この辺りが Apple Computer の本拠地で立ち並ぶ Apple のビルの群れを見ながら約 1.5 マイルほど行くと、左側にこの北浜があります。割と地味な看板なので見落とししてしまうかも知れませんが、K-mart という大きなスーパーマーケットの向かいにあるということを記憶しておけば、簡単に見つけられると思います。もし、Rainbow Dr. との交差点まで来てしまったら行き過ぎですので引き返してください。

たいていどのレストランでも、値段はそれほど心配することはありませんが、一般的な傾向としては和食はやや高いようです。更に寿司はもうすこし高くなります。あまり調理の手間のかかっていないネタをメシの上に乗せて握るだけなのに何で高いんだとアメリカ人はよく文句を言いますが、言われてみればその通りで返答に窮してしまいます。それはとにかくこの北浜では、標準的なディナー（飲み物や肴も合わせて）は 1 人当たり \$20~30 くらいでしょう。カウンタに座って寿司だ、刺し身だと食べまくと \$50 くらいになることもあります。

ところで、和食の店には必ずと言ってよいほど Teriyaki なる物がありますが、多くはビーフ、ポーク、チキンなどに焼鳥のタレを付けて焼いた物で、ブリの照り焼きはほとんど期待できません。実は筆者はこの Teriyaki を和食の範疇にいれることには少なからず抵抗があります。後で出てくるカリフォルニア巻が、アボガドという得体の知れない西洋野菜をうまくアレンジして、寿司という代表的日本料理に新しい味をもたらしたのに対し、Teriyaki という

のはステーキあるいはバーベキューというアメリカ料理に日本風のソースを持ち込んだ物だと言うからです。でも、決してまずいわけではありません。下手なステーキソースよりずっといいし、ご飯にもよく合いますから、一度試してみたいかがでしょうか。

肉料理の話題のついでにもうひとつ。スーパーなどで売っている肉は、一番高い物でも 1 ポンドが \$6~7 くらいで買えますから、しゃぶしゃぶやすき焼きなどは自分で作ったほうが絶対に得です。短期間のホテル住まいの場合は無理かも知れませんが、もし自炊設備があるのなら、Yaohan で薄切り肉を買ってきてやってみるとよいでしょう。

さて、話題を北浜に戻して、もし 2 人以上で、かつ大変空腹な場合には、スペシャルコースがお勧めです。刺し身、寿司、天ぷら、Teriyaki などがセットになっていて、2 人前で確か \$25 くらいでした。

2-1-2. 鮨

1335 North First St., San Jose
(408) 275-0744

San Jose 空港から Airport Pkwy. を通って 1st St. に出ます。1st St. で右折して約 1 マイルほど行くとフリーウェイ 880 への入り口があって、その手前、右側にこの鮨があります。隣に Genji というレストランやその駐車場があるのですぐに分かるでしょう。店の造り、価格、雰囲気など、ここはカリフォルニアの典型的な Sushi-Bar です。入り口を入ると右手にカウンタ、左手にテーブルがあり、けっこういつも賑わっています。San Jose 中心部のビジネスエリアという立地条件のためか、昼食時などは大変な混雑になります。

標準的な握り鮨は 2 つで \$2.5 くらいで、他の食べ物と比べればかなり高価ですが、それでもけっこうアメリカ人に人気があるのは、低カロリーの健康食品というイメージがあるせいだと思います。

カウンタに座って好きな物を限界まで食べたとしてもせいぜい \$40 くらいで、\$50 以上食べたらかきっと店の人から呆れられることでしょう。

さて、どこの Sushi-Bar でもカリフォルニア巻きというのがあるはずですが、伝聞ですので真偽の程は分かりませんが、このカニ蒲鉾とアボガドを巻いたカリフォルニア巻きは Los Angeles のある Sushi-Bar で客の要求によってアドリブで作ってみたのが起源らしく、最近では日本でも市民権を獲得しつつあるという噂を聞きますが、ともあれ、アボガドの口当たりも良く、淡泊な味わいで、寿司の新しい

い味覚として十分評価されるべき物だと思います。

2-1-3. 富喜鮨

4119 El Camino Real, Palo Alto
(415) 494-9383

Palo Alto では一番の Sushi-Bar だそうです。まずフリーウェイ 101 の北行きに乗り、San Jose 空港から 10 マイルほどの San Antonio Rd. で西行に出ます。およそ 2 マイルほどで El Camino Real と交差しますからこれを右折して更に 2 マイルほど北に向かって行くと、道の右側にこの富喜鮨があります。

和食のレストランに行くと驚くことは、けっこう多くのアメリカ人で賑わっていることと、そのほとんどが箸を使っていることです。決して上手というわけではありませんが、箸を和食を楽しむ小道具であるかのように気楽に使っているのを見ると、けっこう好感が持てます。日本ではナイフやフォークの使い方をテーブルマナーなどと言ってうるさく規制したりしますが、それはそれで様式美のひとつの表現方法であるにせよ、アメリカ人が箸を使う時ほどの気楽さを持てれば食事がもっと楽しくなると思います。

なお、この富喜鮨には寿司ばかりでなく、天ぷら、雑炊などの各種の日本料理もありますのでぜひどうぞ。

2-1-4. ながさき

230 Jackson St., San Jose
(408) 297-0153

San Jose の Nihon-machi にあります。レストランというよりは大眾食堂といった感じで、気楽で安くてなかなかおいしい店です。メニューも豊富なので毎日でも飽きることはないでしょう。

今回、ショッピングセンタ編でも触れたように、Nihon-machi は、Jackson St. に沿って、4th St. から 6th St. くらいまでがメインストリートになりますが、このながさきは Jackson St. 沿いの 5th St. と 6th St. の真ん中あたりにあります。

さて、これはただ単に運が悪いだけなのかも知れませんが、筆者はまだアメリカでおいしい天ぷらに巡り合いません。どこで食べても、衣がベタベタしているかカラカラに揚げ過ぎているかのどちらかなのですが、そんな中で、このながさきと、San Francisco の Nikko Hotel の弁慶の天ぷらはかなりおいしいほうだと思います。

また、このながさきとは道を挟んで反対側、4th St. と 5th St. の間に、権兵衛というレストランもあって、こちら

もかなりお勧めです。味はなかなかのものでボリュームも十分、鯖の塩焼き、納豆、カレーライスなどなど、極めて庶民的な味覚が思い切り楽しめます。割とアメリカ人にも人気があるようで、食事時には行列ができることもしばしばです。

2-2. アメリカ料理

アメリカに来たのだからアメリカ料理を食べたいという方は多いのですが、では、何がアメリカ料理なのかというとこれはけっこう難しいと思います。アメリカ自体が 200 年余りの歴史しかなく、伝統的な料理が成立していないこと、移民によってできた国なので各国の料理が正統な状態で受け継がれていることなどがその理由です。だから、アメリカにあるすべての料理がアメリカ料理だと言ってしまっても別に構わないと思いますが、それでも強いて挙げれば、ステーキ、ハンバーガー、フライドチキン、ポークアンドビーンズなどがアメリカらしい食べ物だということになるでしょうか。

マクドナルドやケンタッキーフライドチキンなどのファーストフードのおかげで、ハンバーガーやフライドチキンはややもすると軽く見られることがあるようですが、なかなかどうして、ちゃんと作ればそれなりの御馳走になります。例えば、San Francisco 空港の近くにある Airport Marriott Hotel の 1 階のカフェテリアの Marriott Burger などは、バンズはさっくりと軽い歯応えだし、ミートパティは香りよくジューシーに焼き上がっているしで、これはもうほとんど官能的です。また、ずいぶん遠くになってしまいましたが、Los Angeles から北へ車で 1 時間ほどのところにある Summerland という小さな街の Summerland Cafe というレストランで食べたハンバーガーもただ物ではありませんでした。一方、フライドチキンは Los Angeles の Knott's Berry Farm のレストランにとどめを刺します。

前置きが長くなってしまいましたが、とにかくここでは、誰が見てもこれはアメリカだと納得してもらえそうなレストランを紹介することにしましょう。

2-2-1. T. G. I. Friday's

10343 North Wolfe Rd., Cupertino
(408) 257-9600

別に金曜日でなくてもこの店に来れば思わず神に感謝したくなることでしょう。いかにもアメリカな雰囲気です、しかも安くおいしくて、もう何も言うことはありません。

場所は、既に本シリーズで紹介した Valco Fashion

Park というショッピングモールの中にあります。と言っても、モールの建物の中ではありませんが、とにかく敷地内であることは確かです。

フリーウェイ 880 南行きから 280 北行きに乗り換えて 3 つ目の出口 Wolfe Rd. を南へ出るとすぐに Valco Fashion Park がありますが、この店は右側の最初の駐車場入口を入ったところにあります。モールの中からだと、Nordstrom というデパートの1階の電気製品売場のところから裏側へ出ればすぐ右手に見えます。

前菜ながら極めつけと言うべき物がポテトスキンです。少し厚めに身を焼いたじゃが芋の皮の部分にチーズとベーコンを乗せて焼いた物ですが、ビールのつまみにはこれ以上の物はないでしょう。メニューには載っていませんが、チキンを使った物、ピザ風の味付けをした物などもあって、頼めばそれぞれをいくつかずつ取り合わせた物も作ってくれます。

店内は余り静かではありませんが、若い人達が多く楽しい雰囲気です。金曜日の夕方などはかなり混むそうで、少し待たなければならぬかも知れません。もっとも、良いレストラン（すなわち安くておいしい）はどこでも大変混んでいて、30分くらいの待ち時間は普通です。実際、長く待たされた店で期待を裏切られたことはありませんが、もしかするとそれは“空腹は最上のソース”だったからかも知れません。

2-2-2. Lonnie's Bar-B-Que

1595 Franklin Rd., Santa Clara
(408) 243-7752

セルフサービスなので、むしろファーストフードと言うべきでしょうか。あまり広い店ではないし、お世辞にもきれいとはいえませんが、ここのリブバーベキューは最高です。

場所は少し分りにくいので、San Jose の地図を買って確認するとよいでしょう。一例としてたびたび出てきた Valley Fair Shopping Center から行く場合を述べておきますと、まず、このショッピングセンタの裏から北の方へ延びる Monroe St. に入ります。これを 1.5 マイルほど行くと Homestead Rd. との交差点の次に、左へとはいる Franklin Rd. がありますので、ここで左折します。2 ブロックで Jefferson St. ですが、ここにこの Lonnie's があります。

バーベキューはソースが手や口の周りについて、ほとんど歌舞伎役者みたいになってしまいますが、そういうことは余りにしないで、ただひたすらむさぼり食うというの

がおいしい食べ方です。

2-2-3. The Fish Market

3778 El Camino Real, Santa Clara
(408) 246-3474

名前が示すとおり本来は魚屋さんですが、レストランもやっていて、シーフード中心のおいしい店... なのだそうです。

だそうですと書いたのは、実は筆者はまだここで食べたことがないからです。何度も行ったのですが、いずれもあまりにも待ち時間が長かったため諦めてしまいました。きっととてもおいしいのだろーと思ひます。もしどなたかここで食事をされた方がありましたらぜひお話しをお聞かせください。

2-2-4. Sundance Mine Co.

1921 El Camino Real, Palo Alto
(415) 321-6798

シリコンバレーでステーキを食べるならここでしょう。特にプライムリブは絶品です。なかなか落ち着いたムードですし、値段もそれほど高くはありません。スタンフォード大学にも近く、大学観光の後に立ち寄ってみるのもいいと思います。Tap Beer（樽のビール）の種類が豊富ですので、びんのビールと飲み比べてみるのも楽しいでしょう。

San Jose から行くのなら、フリーウェイ 101 北行きで San Antonio Rd. まで行き、これを西行きへと出ます。約 2 マイルほどで El Camino Real と交差しますのでこれを右折して北へ向かいます。約 1 マイルほど行った右側にありますが余り派手な看板は出していないので、番地に注意しながら行くとよいでしょう。

出てくるステーキはかなり分厚く、ちょっとした感動を味わえることは間違いありません。

2-2-5. Scott's

185 Park Av., San Jose
(408) 971-1700

San Jose ダウンタウンにあるシーフードのレストランです。空港からだと、Guadalupe Pkwy. か Coleman Av. を南へ下ります。Coleman だとそのまま Market St. につながりますが、Guadalupe だと Market St. への出口で出ることになります。この Market St. を南へたどると Fairmont Hotel の向かいぐらいに Park Av. があり、これを右折して 100m 程の立体駐車場の6階にこの Scott's があります。

ここは料理もさることながら、おまけで出てくるパン（

Sour-dough (サワードゥ) が最高です。この Sour-dough は、San Francisco の隠れた名物だそうで、高貴な香ばしさとわずかな酸味があります。これは食べ始めると切りがなくて、メインディッシュが出て来る前に満腹になる恐れが十分ありますので気を付けてください。

2-3. その他

各国の料理は実にバラエティ豊かで、しかも、オリジナルの味と雰囲気をも十分に残しているのです。あちこちと食べ歩くのは絶対に楽しいと思います。割と多いのがメキシコ料理と中国料理でしょうか。

San Jose のイエローページで Restaurant の項目を見ると、アルファベット順のほかに、国別のレストランのリストが載っていて大変便利です。

2-3-1. Chef Chu's Restaurant

1067 North San Antonio Rd., Palo Alto
(415) 948-2696

San Francisco の China-town の Four Seas (四海飯店) に勝るとも劣らない味の中華料理店です。けっこう混んでいますので、待たなければならないことを覚悟しておくといよいでしょう。

まずはフリーウェイ101を北へ行き、San Antonio Rd. で西行きに出て約2マイルほどで El Camino Real と交差しますが、この Chef Chu's は交差点を渡った左側にあります。交差点を渡ったばかりで左折するのは大変なので、この交差点でまず左折してからすぐ右側の駐車場にはいるのがよいでしょう。

餃子、焼売、春巻、酢豚などはあまりにもポピュラー過ぎていまさらという感じがしますが、絶対に試してみるべきです。ほかにはレモンチキンと炒三鮮(何と読むのかわかりませんが)がおすすめです。ただ、炒三鮮に関していえば、神戸元町の中国酒家の方が少し上でしょうか。

中華料理の鉄則ではありますが、特にこの店については、4人以上で行くことをお勧めします。中華料理は色々な物を少しずつ食べるのがおいしいので、そのためにはどうしても4人は必要だからです。北京ダックは予約が必要ですが、それほど高くはなかったはずですので、一度試してみたいかがでしょうか。

2-3-2. Lock Chun Restaurant

4495 Stevens Creek Blvd., Santa Clara
(408) 249-2784

San Jose 中心部付近にはたくさんの中華料理のレストランがありますが、とりわけこの Lock Chun がおいしいと

思います。どちらかといえば日本人好みの味付けだと言えるかも知れません。ここでは木須豚(ムッシュポーク)がおすすめです。豚肉のほそ切りを野菜と一緒に炒めた物を、ちょっと甘目の味噌を付けたパンケーキに包んで食べるもので、木須牛、木須鶏などのバリエーションがありますが、豚が一番です。

場所は、フリーウェイ880から280北行きに入り最初の出口の Saratoga Av. を北行きに出た後、最初の交差点(Kiely Blvd.)を左折し、大きな右カーブを抜けてまもなく Stevens Creek Blvd. にぶつかるのでこれも左折します。約 200m 程で右側に何件かの店が並んでいるところ(Woodham Shopping Center)がありますが、その一番西の端にこの Lock Chun があります。看板には楽群居と書いてあります。

2-3-3. El Mesquite

3136 Stevens Creek Blvd., San Jose
(408) 249-1400

名前からも分かる通り、メキシコ料理です。

フリーウェイ880を Stevens Creek Blvd. で西行きに出て、Valley Fair Shopping Center を右手に見ながら 0.5 マイルほど行くと Winchester Blvd. と交差してまもなくの所、左側にあります。

内部は日本のファミリーレストラン風です。筆者自身は余りメキシコ料理に慣れていないせいか、味付けはどの料理も同じように感じるのですが、Fajita (ファヒータと読む) と Brito がなかなかのものでした。それぞれ色々なバリエーションがあるのですが、特に Fajita はチキンがおいしいと思います。

なお、この店の隣に、既に紹介した The Big 5 というスポーツショップがあり、また向かいには Wherehouse というビデオ/レコード店があります。

2-3-4. Seoul Restaurant

(所在地、電話番号は調査中)

韓国料理店です。あまりきれいとか明るいとかいう店ではないのですが、テーブルで炭火で焼くカルビは、ビールにもご飯にもよくあって、思わず食べ過ぎてしまうことでしょう。Dinner として頼むと十数種類のキムチの小皿が出てきて、これがまた楽しめます。

所在地と電話番号が分からなかったのですが、道順は次の通りです。まず、フリーウェイ880を北(Oakland方面)へ行くとフリーウェイ101とのインタチェンジを過ぎて5〜6マイルで、Calaveras Blvd. (州道237号線)への出口

があります。この Calaveras Blvd. を東行きに出て、右側車線をそのまま進むと、2番目の信号のところで、左へカーブする Calaveras Blvd.と別れて、まっすぐに進む Serra Wayへと入ります。右側に Serra Shopping Center を見ながら直進するとやがて Main St. に突き当たりますが、その突き当たりの正面、やや左側にこのレストランがあります。

ここでのお薦めは、テーブルで焼く Dinner の焼肉（特にカルビ）とビビンバでしょう。なお、同じカルビでもメニューには価格の違う3種類のカルビが載っていて、実はテーブルで焼く Dinner というのが一番高いのですが、ここは迷わずこれを選ぶべきです。

2-4. ついでに

シリコンバレーという範囲からは外れてしまいますが、San Francisco と Los Angeles の有名店をいくつか紹介しておきます。ガイドブックにはこの手のレストランがそれこそ数限りなく紹介されていますが、ここに挙げた店ならきっと満足できるはずです。

2-4-1. Lowrie's Prime Rib

まさかこの店の載っていないガイドブックはないでしょう。Los Angeles のダウンタウンから Wilsher Blvd. を西へ進み、Beverly Hills の La Cienega Blvd. で右折したところのすぐ左側にあります。メニューは極めてシンプルで、メインは4〜5種類のプライムリブだけです。このプライムリブがおいしいことは言うまでもないのですが、付け合わせのヨークシャープディングとベイクトポテト（要はじゃがバター）がこれまた最高です。

オーダする時には前菜やスープ、サラダなどを頼むだけで、メインのプライムリブはコックさんが席までワゴンを押してきて、目の前で切り分けてくれます。

2-4-2. The Chart House

San Francisco の Fisherman's Wharf にあるレストランで、シーフードのほかに肉料理もあります。蟹が好きな人はアラスカのキングクラブ、肉の好きな人にはプライムリブ、魚の好きな人にはキングサーモンなどはいかがでしょう。ここのサラダバーの充実ぶりは筆者の経験の中では文句なく一番で、野菜やフルーツが心行くまで味わえます。前菜にはシュリンプカクテルとポテトスキン、デザートにはマッドパイ（3人でやっと食べられるくらいのアイスクリーム）がお薦めです。

ここは、店内のインテリアも素晴らしくゆったりと落ち着いた雰囲気です。食事を楽しむことができます。

なお、もし Lake Tahoe へ行くことがあれば、South Lake Tahoe の姉妹店に行ってみるのもよいでしょう。夜景の美しい高台にあって眼下にはカジノ街のネオンが広がり、最高のムードです。

2-4-3. Scoma's

かなり有名なシーフードレストランで、食事時はかなり待たなければならないと思います。もちろん待つだけの価値のある店であることは言うまでもありませんが、...

San Francisco の Fisherman's Wharf にありますが、少し奥まったところなので、詳しい場所は誰かに聞くのがよいでしょう。また、姉妹店が Sausalito にもあって、車のある人にはこちらの方がお薦めです。遠く対岸に浮かぶ San Francisco の夢のような夜景を見ながらの食事は、忘れられない思い出になることでしょう。

いくらでもおかわりできる Sour-dough というパンは、実際、いくらでもおかわりしたくなる味ですし、また、ボストンクラムチャウダは誰が何と言おうとここが一番です。メインディッシュには、各種のシーフードパスタやロブスターテイルが良いでしょう。病みつきになる可能性があります。

2-4-4. Four Seas (四海飯店)

San Francisco の China Town にある中華料理店です。China Town のレストランをすべて試したわけではありませんが、筆者の口にはここが一番合うようです。

各種点心類の小皿をお盆に乗せて、売り子さんが店内を徘徊しているので、これで飲茶を楽しんでみてはいかがでしょうか。多少グロテスクですが鶏の足は不思議な味わいでした。あるいはまた、たくさんの種類の餃子を食べ比べてみるのも面白いかも知れません。満腹するまで食べて1人\$20くらいです。

3. 次回予告

さて今回のレストラン編、お楽しみいただけましたでしょうか。他にも名店はいくらでもあると思いますが、もしご存じの方がいましたら SEAMAIL 編集部までお知らせください。そういう情報は共有してこそ価値があるというものです。

ところで、今回は、一応最終回の宿泊／観光名所編です。一応というのは、更にその後、もしかしたら、場合によっては、番外編があるかも知れないからなのですが、先のことは分かりませんので、ひとまずは次回をお楽しみに。

SEA のネットワークについて

— SIGNET 活動再開 —

1. 会員アンケートから

10月月末の会員アンケートで、SEAのコンピュータ・ネットワークについて、会員の方々の御意見をおたずねしました。レスポンスは思ったより悪く、全部で十数通の回答しか帰ってきませんでした。昨年の夏に行われた開発環境調査の結果からすれば、SEAメンバの内の半数近くがすでにjunet等のネットワークを利用しているはずなのですが、寄せられた回答のほとんどは、まだそうした条件に恵まれていない方々からのものでした。

おそらくすでにネットワーク環境にいる方々は、その恩恵を十分に享受していて、「いまさらSEAネットなんて!」ということなのかもしれません。しかし、SEAにとっての問題は、いまネットワークの外にいる会員をどうしたらこの世界に引き込むことができるかを考えることでしょう。あとで述べるように、SIGNETも活動を再開しました。ぜひ、みなさんの知恵を出しあって、SEAの活動の場を拡大して行きたいと思います。

福田 充利

PC-VANにCUGとして発足するとか。でも、金がかかるな。やはり、4回線位で運用して欲しい。WWIUなんのでもあるしい。FAX配送サービスがあると非常によい。

折田 豊

SEAのBBSはパソコン通信のほうがよいと思う。ハード的には、パソコン1台買ってBBSソフトを自作して、動かしてしまう。内容は、SEAの行事案内、バカヤローコーナー、こんなソフトのあんなバグ(私は、これが一番ありがたいと思う)、オンライン分科会等があると面白いのでは?。

junetに接続するのは、結構むずかしい。まず、Unixマシンが必要である。最近安くなってきたとはいえ、100万円以上する。直接利益を生みださないマシンに大金をかけられるのは一部の会社だと思う。派遣で出向している人なんかは全然ダメでしょうね。また、junetは、ふつうのパソコン通信とは違って、接続する相手を自分でさがさないといけない。SEAのネットワークをjunet上に置くと、接続先の紹介をSEAが行わないといけなくなる(私にもぜひ紹介して!)

パーソナルワープロでもアクセスできるパソコン通信で

あれば、Unixを使えない人でもアクセスできる。しかし、パソコン通信でやると決めた場合、問題になるのが回線コストである。草の根ネットみたいにすると全国規模のSEAのメリットが生かせない。回線コストを安くするためには大手ネットにCUGとして登録するのが一番よいと思う。どのくらいのコストがかかるのか調べたことがないので、大きな声で進められないのが残念だが、...

junetはプロフェッショナルが議論するのに向いていて、パソコン通信は素人のメディアだそうだが、ほんとうだろうか? パソコン通信でだって議論はできるはずだ。どんなメディアでも使いこなせるのが本当のプロではないのか?

佐原 伸

既存のネットワークに皆で加入するのがよい。

中山 聡

JUST-PC手順等を使ったエラーフリーな通信をベースにさまざまな情報の交換を実現させたい。

中山 照章

NIFTY等の商用ネットワークに、CUG(Closed User Group)を作り、その中で活動する。または、SEAフォーラムを作る。適当なネットワークを乗っ取る。FAXメールを利用する。

山本 博行

電子会議というかSIGというか、比較的長期にわたってあるテーマで、突っ込んだ討論のできる環境が整備されるとよいですね。

野々下 幸治

一番簡単なのはNifty等の商用ネットのCUGを利用することだと思います。junetは利用できる環境にある人が少ないので問題だと思います。独自のホストを持つものいいのですが、Try-PとかDDX-Pなどが利用できるようにしないと遠隔地の人の電話代が大変です。その点商用のネットであれば複数のアクセスポイントを持っています。パソコン通信であれば接続側も会社でパソコンは所有しているでしょうから、モデムを購入するだけの安い費用ですみます。ぜひ何らかの形で、会費相互がネットワークで情報をやり

とりでることを実現して欲しい。そのようなことができれば、SEAMAILの発行も年4回程度にしてもいいのでは。また、毎月の分科会の案内も全員に出す必要なく、パソコン通信が利用できない人だけに送ればよい。月々のホストの運用費も月額いくらかでも取ってもいいのでは。

船切 誠

junetもいいのですが、やはり、つなぎにくいのが当社の現状です。働いている社員は「プロフェッショナル」を目指していても、会社が「メンテナンスが大変」「リスクが大きい」という判断をするのです。製造業の会社は、まだそんなにプロのソフト屋を必要としないのかもしれない（当社だけでしょうか）。SEAは非営利団体ですから、公共のネットワーク上で活動してもいいのではないのでしょうか。「プロらしくない」のは確かかもしれませんが、この場合、実績で考えてみてはどうでしょう。「本当にSEAのネットワークが必要」という証明をしてみませんか？

ところで、もしも多くの苦難の末にjunetへの接続を勝ち取った方がいらっしゃいましたら、ぜひその過程での御苦労話をうかがいたいのです。なんだか、他人のフンドシで相撲をとっているみたいですけど、私自身の環境でjunetにつながる光を見つけるきっかけになるかもしれません。どうか力をお貸しください。

うちの会社でも以前junetに接続しようかという話はあったらしいのですが、やれ「メンテナンスがわずらわしい」とか「通信経路を確保しておく責任がある」とかいった、実に消極的な話で流れてしまったようなのです。そうした大変さの見返りとして、それ以上のメリットが導き出せるはずなのですから（ですよ）。

いまはNIFTY-Serveに個人的につながっていますが、もしjunetに入れたら「いや、SEAのネットはやはりjunetでなければ！」などというかもしれません。

小林 俊明

パソコンネットにしてほしい。会社には自分のワークステーションもLANもあるけど、junetにはつながっていない。マシンの深夜運転と通話料の問題で会社がOKを出さない。パソコンネットなら自分で参加できる。

小林 順二

幹事会での話の中に「だってあれは素人のためのメディアで、我々プロフェッショナルが技術的な議論をするには向いていない」との発言があったそうですが、会員の多くが持っているまたは持ちやすいメディアを使用すべきである。

弘法筆を選ばずのたとえもあるように、道具をどう使うかの問題だと思う。「プロフェッショナル」の自負があるのはいいのだが、そこには反面、権威主義の匂いがして、いささか嫌悪感をおぼえました。

中込 昌吾

パソコン等を用いて新たに自前のネットを構築するのは、費用の面からも体力面からも難しいと思う。そうすると、どこかのネット上にBBS(News Group)を設けることになるが、その内容をOpenにするかCloseにするかによって、junetあるいはTeleStar, PC-VAN, NIFTY-Serveといった商用BBSを利用するかが決まるのではないかな。

権熊 敏朗

いま考えられる形態としては、

- (1) junet
- (2) パソコンネット
- (3) 手仕事ネット - FAXや電話を使う、

あるいはSEAMAIL自体。

があります。それぞれ置かれている環境によって、利用できる形態は異なります。(1)とか(2)だけでは、それを利用できない会員は除外されてしまうでしょうから、どれも利用できるにしたほうがよいでしょう。まず、運用規則とかガイドライン的なものを考えてみたらどうでしょうか？

ここでひとつ申し上げておきたいことがあります。それは、世の中には、パソコン1台自由に買えないソフトウェア会社だってゴマンとあるのです。ソフトウェア業界というのは、もともと労働力の切り売りによって成り立っているところですから、自分の会社に座席すら持てない人だってたくさんいるのです。これは過去のはなしではなく、現在でも依然として続いているこの業界の現実です。このような環境にいるものがどうして「会社を説得してjunetに入る」ことができるのでしょうか？SEAは個人の資格で参加する組織なので、会社の施設を頼らずに、個人でも（少し位無理をすれば）できる手立てを考えるべきだと思います。

村田 英峰

社内でjunetに入りたいといっている者がわずか5人位しかいない状況で、SEAの存在を知っているものもヒラにはほとんどいない。逆に管理職の中には妙に毛嫌いしている人もいて、細々と活動している状況なのです。何というか、技術者意識の強い人間が少ないので、なかなかやりにく

いのです。

以前 junet の ROS(Read Only Site) の問題が議論されていたと思いますが、今の状況では、たとえ加入できたとしても ROS になってしまうという恐れを感じています。が、どこかにホスト局を解説するか、既存のネットの上に Forum を作るかしたほうが、利用しやすいかなという印象を持っています(お金の問題もあるでしょうが...)。某社のように、社員がみんな junet の id を持っていて名刺にそれを印刷しているのを見ると、アコガレとヒケメを感じるのです。

2. SIGNET 活動再開

ながらく休眠状態にあったネットワーク分科会(SIGNET)が、あらたに久保宏志さん(常任幹事、富士通)という強力な世話人を得て、今月から活動を再開しました。以下は、12月6日に行われた再キックオフ・ミーティングのための呼び掛け文です。

SIGNET メンバー各位殿

(1) 世話役が一人増員になりました。

富士通の久保です。青島さんと一緒に SIGNET の世話役を引受けることになりました。SIGNET はここしばらく休眠を続けて、もう十分休まれたでしょうから、またしばらく活発に活動しませんか。ということで、再開を働きかけるべく、久しぶりに SIG の会合を持ってもらうことにしました。これは、その開催案内に同封してもらうべく書いています。

再開後の活動をどう展開していくかは、みなさんと議論を重ねながら詰めていきたいと思います。世話役を志願した気持ちのようなものを書いて、議論のイニシャル・インプットにします。ネットワークのことに明るい訳ではありませんのでとんちんかんなことも含まれているかもしれません。

(2) 新世話役の基本のスタンス

私の立論は、SEA というソフトウェアのプロフェッショナルのコミュニティにとって、ネットワークは欠かせないものであるという前提にたっています。ネットワークのもつ可能性は計りしれないものがあります。しかし、そのポテンシャルをコミュニティが引出すためには、何はともあれ会員相互がネットワークでむすばれている必要があります。この認識から出発します。

(3) SEA の立場でネットワーク問題をどうとらえるか?

現在日本には様々のネットワークがあります。大きく分けると UNIX network とパソコン通信に別れるでしょう。UNIX network で私が知っているのは junet と juice です。パソコン通信は数えきれないくらいあります。商用の BBS、業界団体のネット、地域のネットなど無数といってよいくらいあります。

これらのネットに参加されている SEA の会員は相当数に上るのではないかと考えていますが、不幸なことにネットが相互に接続していないために、会員間でネットワークをうまく活用できていないという問題があります。この問題の解決、つまりネットワークの相互接続の実現が一つの課題です。

2 番目の問題は、ネットに参加していない会員の皆さんを促してネットに参加していただくために何をすべきかです。パソコンやワークステーションを会社なり自宅に所有し、モデムを所有することは各自の努力に待つしかありませんが、その状態からネットに加入する状態に移るための障壁は決して低くありません。特に地方の方にはそれが言えるでしょう。

3 番目は上の2つが解決してもなお残る問題です。これが一番大事です。ネットの将来をどのようなものにするか、そしてネットを使って何をするか、という問題です。ネットワークの持つ利用価値ポテンシャルをどうやって引出すかです。その価値を SEA のために生かす実践に結びつけていく必要があります。

(4) SIGNET の活動指針の提案

SIGNET は、この3つを同時進行的に研究していく場です。自らも3つを実践しながら周囲に実践を働きかけて行くボランティアが定期的に集う場でもあります。そんな場に SIGNET を位置付けてはどうでしょうか。

その際、次の方針を掲げたいとおもいます。その方針は、理論的にも最も正しい選択であると思われる。

ネットワークの相互接続は UNIX network を介して、これをおこなう。これが一番の問題の攻めかたの方針です。どんなに会員の数が多くても、いわゆるパソコン通信は、UNIX network の一個のネットワークアドレスで代表させることができるはずで、UNIX network のプロトコルをパソコン通信のホストに採用させる、といっても同じことです。

junet なり juice のホストを全国のあちこちに配置することでネットワークへの参加を促す。これを第2の問題攻略の方針にします。UNIX network は junet にしろ juice

にしろ、参加は必ずしも容易ではありません。しかし junet のホストなり juice のホストを全国のあちこちに配置することができれば、草の根 BBS に参加するのと同等の経済的負担で UNIX network に参加できるようになるはずです。

junet に参加して、広く産業界や学界の様々な分野でのネットワーク実践者とともに歩む。これを第3の問題の攻めかたの方針にする。junet は現在、Internet の一員になる方向に動いています。Internet のメリットを享受し、そのメリットが及ぶ範囲を拡大する。Internet の中に SEA の活動の成果を残して行く。こんな方向を追求することになります。

(5) SIGMA への期待をつなぐ

考えてみれば、ここに掲げた方針の下にソフトウェア技術者のためのネットワークをつくるべくスタートしたのがシグマプロジェクトでした。その現状も将来も十分に我々に伝わっているとは言えませんが、SEA にとって身近な存在になっているとは言えません。残念ながら、そうです。

シグマはいずれ身近なものになるであろうから、それを待とうという意見もあるかもしれません。それについては次のように考えます。

ネットワークは実践を通してしか、その進歩成長はありません。UNIX network の歴史を繙いてみればこの考えの正しさの歴史的証明を見つけることができるはずです。何か立派なものの出現を待つのではなく、現実的に可能なことを実行する、そこで遭遇する問題を克服して次の段階に進む、そんな歩みを積み重ねつつ、より立派なものに近づくのが正しいと私は考えます。そんな歴史観にたちたいと思います。

そうすることはシグマを否定することにはなりません。むしろ逆です。我々の側からシグマに近づくことになります。それだけシグマとの距離が縮まります。縮まっていった先にシグマがあれば、喜んでシグマの世界に吸収されることを選択しましょう。そのとき我々の使命は終わったことになります。

(6) 当面の研究課題

これからの SIGNET の大きな枠組みを描こうと思って、キーボードをたたいてきました。その思考の延長で当面の活動テーマを導いてみます。

- 月例フォーラムでネットワークをテーマにパネル討論を行なったことがあります。これの記録をレビューする。
- SEA の会員のネットワークへの関心度合いと、ネッ

トワークの活用実態をしるためのアンケート調査を行なっています。この結果を分析する。

- シグマネットワークの現状と今後の方向性について研究する。
- ネットワークの相互接続の技術論と制度論を研究する。
- UNIX network を低コストで実現するための技術資産とソフトウェア資産について研究する。

これらのテーマで今年度一杯くらいの時間をかけて、研究すれば、かなり具体的な活動計画に到達できるのではないのでしょうか。

(7) 再開第1回会合

再開第1回の会合を12月6日に開きます。そこでは、ここに書いたようなことについて議論ができればと思っています。最低限、今年度の活動計画を導くことはやってしまいたいと思っています。

ご参加いただく方は多いにこしたことはありませんが、最低限どれかのネットに参加している方に限定して議論の密度を高めたいと思います。

となるとオフラインで集る前に、オンラインでも情報を交換できます。会議に先立ってネットワークを活用することをやってみましょう。ネットワークの相互接続問題を実践的に考える機会を得ることにもなります。

最初のメッセージは、郵政省メールを使いますが、続くメッセージは E-mail を使いましょう。私は、以下のネットの間のゲートウェイならできます。どうぞこのゲートウェイもお使いください。

kubo@mfd.ks.fujitsu.co.jp

kubo@kubowb.se.fujitsu.co.jp

74360.3202@CompuServe.COM

PDG03202@NIFTY-Serve

h.kubo@MIX

VMD83316@PC-VAN

以上を私からの最初のインプットにします。フォローアップをお待ちしております。

ということですので、関心をお持ちの方はぜひ積極的に来年以降の活動に御協力ください。SIG の会合での議論の模様などは、あらためてこの誌面で報告していただくつもりです。

会員の声

折田 豊

最近、自分がいままでにやった事がないようなソフトを作った。プロジェクトの工程管理が大変だった。開発手法は、ウォータフォールモデルに従って(そのつもりで)行ったが、先が見えないので苦労した。そんな訳で、6-7-8月号は、まじまじと読んでしまいました。「プロトタイピング」や「CASE」って、本当に役に立ちますか。仕事に使ってる人にぜひ本音を聞いてみたい。

【編集部から】: どなたか、経験者の方、ぜひ答えてあげてください。

滝口 亨

長年システム開発の仕事に携わってきたが、それぞれに環境が異なるので共通の議論がなかなかできず、また応用も難しいことと痛感している。我々の仕事の本質は客商売であり、限られた時間と予算(人)で、どれだけ、要求を満たせるかという観点に立つての議論が必要ではないだろうか。今の SEAMAIL の記事には、客の声が全くなく身内だけで自己満足しているような気がする。

ビジネスとしての成功例をもっとのせてほしい。また、追求すべきではないか?

【編集部から】: とりあえず、あなた御自身のサクセス・ストーリーまたは失敗例を報告してください。

中山 照章

厚い SEAMAIL は読むのが大変です。関係者の皆様、お疲れ様でした。

木全 洋

SEAMAIL 6-7-8 合併号の環境ワークショップ特集は、読みごたえがあり、大変参考になりました。今後、「プロジェクト管理」や「ソフトウェア技術者のキャリアパス」の討論やアンケート特集をしてくださると幸いです。

【編集部から】: どんなアンケートにしたらいいか、原案を考えて事務局にお送りいただけますか?

山本 博行

最近の SEAMAIL、割と定期的に刊行されえるようになってきたようですね。編集部の方、事務局の方の努力の成果でしょう。私のような、ほとんど半分は冗談で生きて

る者にとっては、カタイ記事は少々消化不良になってしまっていますが、それでも毎号、結構楽しく読んでいます。

中山 聡

HyperCard のスタックウェアが MS-DOS や Unix の上で動けば、楽しいだろうな。

野々下 幸治

先日、子供の運動会を撮るために、8mmVTR メラを購入しました。最近、子供の日常の姿を撮るのが趣味になりつつあります。そこで気が付いたので提案したいのですが、SEA のセミナー等を VTR に撮って、地方の人やセミナーに行けなかった人に貸し出しを行なうというのはどうでしょうか。SEA の会員でも、VTR カメラを持っている人はかなりいるでしょうから、セミナーを録画するくらいの台数は集まるのではないのでしょうか。もちろん、私のも無償にてレンタルいたします。地方では、セミナーの記録の上映会もできるし、記録としては音声よりリアリティーがあつていいのではないのでしょうか。

【編集部から】: 大変いいアイデアだと思いますが、さて具体化はどうしたらいいのでしょうか。録画、ダビング、貸出し事務、そうしたことをやる人手もお金も SEA にはないのですが、どなたかビデオ狂の方、ボランティアしていただけますか。

村田 英峰

技術の教育はどうやるのがよいのだろうか、どうやって何を教えるべきか、悩んでいます。テキストを作るのも、再利用や改版のことを考えると、電子ファイリングのできるオーサリング環境がほしいなあと思うのです。Mac にスキャナやスピーカをくっつけて、MO-DISK で大容量を扱えるようにしたらいいかなと思っているのですが、会社のトップは某 F 社のほうばかり向いているので、異端視されそうだし、ぜいたくなオモチャに見られてしまうのもいやだし、なんとかうまくダマスいい方法はないでしょうか?

【編集部から】: 一度、教育分科会(SIGEDU)へ顔を出して、ご相談されたいかがですか。

椎熊 敏朗

最近よく考えていることといえば、環境が欲しい!(もち

ろん開発環境です), 情報が欲しい!, ネットワーク(junet)に入りたいよー, といったことです。私の会社は中古の16ビットパソコン1台買うにも大騒ぎするくらいド貧乏なところなので、理想的な環境など望むべくもありません。最近になって、自分の環境は自分で手に入れるしかないということに気がつきまして、生活を切り詰めて、せっせと貯金をしている最中です。MMCに預金できるくらいになったら、某NEWSでも買おうかと思っています。

ところで、さきごろ新聞やテレビのニュースをにぎわせた大手メーカの1円入札事件、みなさんどう思われますか。これは、ソフトウェアの価値はタダ同然だとみなしていることにほかならないのではないのでしょうか。業界に大きな支配力を持つ大企業がこうしたタワケタ行為をやり続けている限り、われわれソフトウェア技術者が正当な地位につくことはできません。

SEAのメンバのみなさんは、純技術的なことに強い関心がある反面、社会のこととか、経済のことなどは、多少苦手ではないかと思いますが、優れた技術やアイデアといったものは、公正な社会があってこそ初めて正当に評価されるわけですから、この問題について、真剣に考えてみてはどうでしょうか。今後も機会をみて、こうした社会問題とリンクした議論を誌面でたたかわせてみたらよいのではないかと思います。

【編集部から】: たしかに1円入札事件については、会員アンケートをやってみてもよいと思います。それと、ネットワーク・ウォーム事件。

中込 昌吾

ネットワークにアクセスするようになって感じることは、それは市外局番にかける電話代の高いこと。モデムと接続している回線はもっと安くしてほしい、というのはエゴかしら。

船切 誠

最近、ようやくαバージョン程度のシステムができたので、工場のほうで調整に使ってもらっています。で、当然100% だいじょうぶのソフトではありませんから、トラブル続出です。トラブルにまきこまれた人、パニックに陥った人の話を聞いてソフト(システム設計)の担当者に伝えるのですが、「そんなはずはI Hい!」。いやあ困った、こまった。

こうしたトラブル情報をうまく担当者に伝え、解決策を見出すためのシステム(秘訣)がどこかにないでしょうか。

最近「ひと」と話をするのがこわくて...

福田 充利

「あまり有益でもないんだが、ウー、枯れ木も山のにぎわいで(だれが枯れ木なの?) まあ入っとうかな」ということで入会しましたが、まだあまりお役に立っていません。よろしく。

小林 順二

フォーラムの内容についてもっと知りたいのですが、たとえば第9号の標準化フォーラムの報告など、事後アンケートの方が重視されていて、よく内容がつかめませんでした。

それと、各支部の所在地を知りたいのですが...

【編集部から】: フォーラムでの議論は基本的にインフォーマルでオフレコになっているので、くわしくはお伝えできません。御面倒でも足をお運びください。支部は「物理的には」存在していません。つまり、事務所とか専属の事務員がいるわけではありません(東京の本部も、事務所はjusと同居ですし、事務員は中島さん1人だけです)。世話人の連絡先は毎月の案内にのっていますので、そちらに御連絡ください。

岸田 孝一

先日、釧路の環境ワークショップ(雪が降らなくて残念!)でTool Interfaceの標準化について、ヨーロッパのPCTEやアメリカのCAISに対応する(あるいはそれを超える)ような試みを日本でもやるべきではないか、やるとしたらSEAぐらいしか場はないのではないかと、という意見が出ました。私個人としてもこの問題には興味があり、ぜひ手をつけたいと以前から思っているのですが、なかなか始められません。何人か同志を見つけられれば、SIGのかたちでスタートしたいと考えます。興味をお持ちの方はぜひ御連絡ください。E-Mail Addressはk2@sra.co.jpです。

ソフトウェア技術者協会 (SEA) は、ソフトウェアハウス、コンピュータメーカ、計算センタ、エンドユーザ、大学、研究所など、それぞれ異なった環境に置かれているソフトウェア技術者または研究者が、そうした社会組織の壁を越えて、各自の経験や技術を自由に交流しあうための「場」として、1985年 12月に設立されました。

その主な活動は、機関誌 SEAMAIL の発行、支部および研究分科会の運営、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、および内外の関係諸団体との交流です。発足当初約 200人にすぎなかった会員数もその後飛躍的に増加し、現在、北は北海道から南は沖縄まで、1650 名を超えるメンバーを擁するにいたりました。法人賛助会員も約 60 社を数えます。支部は、東京以外に、関西、横浜、長野、名古屋、九州の各地区で設立されており、その他の地域でも設立準備をしています。分科会は、東京、関西、名古屋で、それぞれいくつかが活動しており、その他の支部でも、月例会やフォーラムが定期的に開催されています。

「現在のソフトウェア界における最大の課題は、技術移転の促進である」といわれています。これまでわが国には、そのための適切な社会的メカニズムが欠けていたように思われます。SEA は、そうした欠落を補うべく、これからますます活発な活動を展開して行きたいと考えています。いままで日本にはなかったこの新しいプロフェッショナル・ソサイエティの発展のために、ぜひとも、あなたのお力を貸してください。

代表幹事： 岸田孝一

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

幹事： 青島茂 天池学 飯沢恒 岩田康 岡田正志 落水浩一郎 片山禎昭 川北秀夫 岸勝義 杉田義明 武田知久
田中慎一郎 玉井哲雄 玉川滋 中來田秀樹 中園順三 中野秀男 野村敏次 野村行憲 針谷明 平尾一浩
深瀬弘恭 藤本司郎 北條正顕 細野広水 盛田政敏

会計監事： 辻淳二 吉村成弘

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

分科会世話人 環境分科会(SIGENV)：田中慎一郎 渡邊雄一

管理分科会(SIGMAN)：相沢圭一 川北秀夫 芝原雄二 野々下幸治

教育分科会(SIGEDU)：大浦洋一 杉田義明 中園順三

ネットワーク分科会(SIGNET)：青島茂 野中哲

法的保護分科会(SIGSPL)：能登末之

C A I 分科会(SIGCAI)：大木幹雄 中谷多哉子 中西昌武

ドキュメント分科会(SIGDOC)：田中慎一郎 野辺良一

支部世話人 関西支部：白井義美 中野秀男 盛田政敏

横浜支部：熊谷章 林香 藤野晃延 松下和隆

長野支部：小林貞幸 佐藤千明 細野広水

名古屋支部：岩田康 鈴木智

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

SEAMAIL 編集グループ：岸田孝一 佐原伸 芝原雄二 関崎邦夫 田中慎一郎 中村昭雄 長井修治 成沢知子

野辺良一 藤野晃延 渡邊雄一

SEAMAIL Vol. 4, No. 12 平成元年12月25日発行

編集人 岸田孝一

発行人 ソフトウェア技術者協会 (SEA)

〒102 東京都千代田区準町2-12 藤和半蔵門コープビル505

印刷所 サンビルト印刷株式会社 〒162 東京都新宿区築地町8番地

定価 500円 (禁転載)



ソフトウェア技術者協会

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