



SEAMAIL

Newsletter from Software Engineers Association

Vol. 14, Number **4** August, 2004

目 次

編集部から		1
SESSAME による組込みソフトウェア初級実習教材の開発と運用事例	森孝夫 ほか	2
インターネット早期広域攻撃警戒システム WCLSCAN	鈴木裕信 ほか	12
ソフトウェアプロセス最適化マニフェスト	伊藤昌夫	21
概念設計についてのノート	岸田孝一	23
新刊紹介：玉井哲雄「ソフトウェア工学の基礎」	山崎利治	28
季節外れの札幌便り	熊谷章	37
新年度総会		47

ソフトウェア技術者協会

Software Engineers Association

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ー、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増加し、現在、北は北海道から南は沖縄まで、400余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20社を数えます。支部は、東京以外に、関西、横浜、名古屋、九州、広島、東北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： 荒木啓二郎

常任幹事： 熊谷章 高橋光裕 田中一夫 玉井哲雄 中野秀男 深瀬弘恭

幹事： 石川雅彦 大場充 落水浩一郎 窪田芳夫 小林修 小林允 桜井麻里
酒匂寛 塩谷和範 篠崎直二郎 新谷勝利 新森昭宏 杉田義明
中來田秀樹 野中哲 野村行憲 野呂昌満 端山毅 平尾一浩
藤野誠治 松原友夫 渡邊雄一

事務局長： 岸田孝一

会計監事： 橋本勝 吉村成弘

分科会世話人 環境分科会(SIGENV)：塩谷和範 田中慎一郎 渡邊雄一
教育分科会(SIGEDU)：君島浩 篠崎直二郎 杉田義明 中園順三
ネットワーク分科会(SIGNET)：人見庸 松本理恵
プロセス分科会(SEA-SPIN)：伊藤昌夫 塩谷和範 新谷勝利 高橋光裕 田中一夫 端山毅 藤野誠治
フォーマルメソッド分科会(SIGFM)：荒木啓二郎 伊藤昌夫 熊谷章 佐原伸 張漢明 山崎利治
オープンソース分科会(SIGOSS)：石川雅彦 岸田孝一 杉田義明 鈴木裕信 中野秀男

支部世話人 関西支部：小林修 中野秀男 横山博司
横浜支部：野中哲 藤野晃延 北條正顕
名古屋支部：石川雅彦 角谷裕司 野呂昌満
九州支部：杉田義明 武田淳男 平尾一浩
広島支部：大場充 佐藤康臣 谷純一郎
東北支部：布川博士 野村行憲

賛助会員会社：ジェーエムエーシステムズ SRA PFU テブコシステムズ 富士通
オムロンソフトウェア キヤノン 新日鉄ソリューションズ
ダイキン工業 オムロン 富士電機 ブラザー工業 オリンパス
リコー NTTデータ ヤマハ オープンテクノロジーズ
SRA西日本 日本総合研究所 SRA東北
(以上20社)

SEAMAIL Vol. 14, No. 4 2004年8月5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)
〒160-0004 東京都新宿区四谷3-12 丸正ビル5F
T: 03-3356-1077 F: 03-3356-1072 E-mail: sea@sea.or.jp URL: http://www.sea.jp
定価 1,000円 (禁無断転載)

編集部から

☆

今年のソフトウェア・シンポジウム 2004 in 岡山は、最終的に150人近い参加者を得て無事に終わりました。Proceedings CD を同封してお送りします。

☆☆

この号の巻頭には、この SS2004 で最優秀論文賞に選ばれた森孝夫さん他の論文と、同じく最優秀プレゼンテーション賞に選ばれた鈴木裕信さん他のアブストラクトおよび発表スライドを載せました。

☆☆☆

SEA-SPIN(プロセス分科会)世話人の一人である伊藤昌夫さんが、かねてから持論として展開されて来た SPO(ソフトウェア・プロセス最適化)を議論するための Web Page を立ち上げられました。

<http://www.process.jp> まずは、そのマニフェストをお読みください。

☆☆☆☆

次に載せた概念設計についてのわたしのメモは、昨年 TM Workshop での議論の産物です。SEA の周辺で行われているさまざまなイベントで何がどう議論されているのか、参加者以外への情報公開のひとつの形として、まだ未定稿ですがあえて掲載してみました。

☆☆☆☆☆

山崎さんからは、玉井先生が岩波書店から出された本の内容紹介を兼ねた書評をお寄せいただきました。

☆☆☆☆☆☆

そして、去年から北の大地・札幌の住人になられた熊谷さんに、雪国での暮らしの一端を示す日録を寄稿いただきました。猛暑の中で読むにふさわしい冬物語だと思います。

☆☆☆☆☆☆☆

岸田孝一 @ SEA Office

SESSAME による組込みソフトウェア初級実習教材の開発と運用事例

森孝夫 大西建児 西康晴
組込みソフトウェア管理者・技術者育成研究会
tmori@mbj.nifty.com

1. 要旨

組込みソフトウェア管理者・技術者育成研究会 (Society of Embedded Software Skill Acquisition for Managers and Engineers : SESSAME)[1]は、組込みシステム開発力を日本の産業競争力のキーにするという理念のもと、中級以上の技術者や管理者を10万人育成する基盤を構築すべく活動を行っている。

上記の活動の一環として、SESSAME では各種の教材開発を行っている。その中の一つが、本論文で紹介する初級者向け組込み実践教材「鹿威システム」である。

本教材は、組込みシステム開発に必要なコンピタンスをバランスよく育成できる総合的な教材を目指して開発され、その後各種のセミナーなどで教材として有益な効果をもたらした。本論文では、その開発から運用に至るまでの一連の経験と効果をまとめて説明する。

2. 総合的な教材の必要性

2.1. 近視眼的な視野から全体視野への移行

組込みシステム開発技術者に限らず、ソフトウェア技術者の教育を行う際の問題の一つに、技術者の視野が近視眼的になりがちであることが挙げられる。例えば、ソフトウェア技術者が、ハードウェア技術に対して興味、理解を示さなかったり、品質や営業、マーケティングなど開発以外の活動について興味、理解を示さなかったりするという問題である。

この問題が発生してしまう原因を心理的な側面から分析すると、大きく分けて2つのケースがあると考えられる。

第一は、未知の技術に対する心の壁が存在するケースである。人間は、自分の知っている範囲、もしくは仕事として担当している範囲内のみ技術を扱うことは楽しい、もしくはたやすい。逆に、総合的な視野を身につけるために、その周りの範囲の技術を扱おうとすると、ある壁を乗り越える努力が必要となる。その努力を開始、継続するためには、技術者本人の果敢な精神も必要

であるし、周囲のサポートも必要であろう。しかし現状ではそれらが不足してしまうケースが見られ、結果全体を見通す視野が育たない場合がある。

第二は、一つの事象、技術に熱中しすぎるケースである。この場合、本人は技術を深く追い求めていく果敢な精神にあふれていることが多い。ところがある特定の技術に熱中し過ぎると、視点がその分野に偏ってしまい、その結果全体を見通す視野が育たないということになってしまう。

これらの問題を解決するには、若年層の技術者に対する教育の方針が鍵となる。一般的な開発現場におけるキャリアの積み方は、最初はプログラミングなどの実装作業から入り、年齢が上っていくにつれて徐々に扱う範囲がシステム、市場、経営へと広がって行く。しかし、教育の観点から見ると、まだ経験が浅いキャリアの最初の時期にこそ、ソフトウェアに留まらないシステム開発全体を意識し、更には経営を見据えた製品企画まで意識することが、将来の視野を広げることにつながると考えられる。

できれば、このような教育は18歳前後の高校、大学教育で行うことが望ましい。ソフトウェア開発の目的、組込みシステムにおけるハードウェア技術の必要性などをこの時期に意識させ、その中で自分に合った道を選んで行けるように導く配慮が大切だと考えている。技術者が広い視野で技術全体を捉え、その中から自力でどの方向に進むかを決め、その結果ある特定の技術に熱中し極めて行くのは決して盲目的な行動ではない。

したがって、組込みシステム技術者の初級者教育では、自発的に取り組むことができ、かつ広範囲な内容を扱う教材の開発が求められるのである。

2.2. すりあわせによる実践を体験する教材の必要性

現在のソフトウェア業界は、オブジェクト指向、MDAに代表されるように、抽象設計が重視されている。このような考え方の流行は、ソフトウェアの開発に数学的な整理と美学を持ち込む意味で歓迎すべきである。

しかし、抽象化能力を育成するだけで、産業競争力に即つながらのかと考えると、多少疑問も残る。他に必要となる特性もあるのではないかと。特に組込みシステム開発では、日本人の性質のうち競争優位要因になりうる点として、チーム一丸となって開発を行うことができる組織的開発力の強みと、未定義でも前進できるという精神構造が挙げられる。変化する仕様に対応し、ハードウェアとソフトウェアを協調しながら設計していく「すりあわせ[10]」の力強さは、中国やインドとの価格競争に陥らないためにも、是非とも維持し、かつ次代へ継承することが必要と考えている。抽象化の力と、すりあわせの力の両方が身につけば、これほど心強いことはない。

とすると、教育現場においても、すりあわせながら実装していく力を体験してもらうのが望ましいと考える。組込みシステムの教材は、トラブルがいろいろ発生して、それを乗り越えなければ完成しないような教材の方がかえって役立つところが多いのではないだろうか。

また、組込みシステムでは、最終的にメカトロニクスとの「すりあわせ」も必要になることが多い。そのため、ソフトウェア、ハードウェア、メカトロニクスの実習をバランスよく盛り込んだ教材があるとよい。

3. 教材の狙い

3.1. 組込みシステム開発の裾野の拡大

組込みシステム以外のソフトウェアで十分に経験を積んだ優秀な技術者にとっても、組込みシステム開発の最初は壁を感じる、とよく耳にする。これは未知の分野に対する心の壁であろう。

また、若者の理科離れが叫ばれて久しい。国際教育到達度評価学会(IEA)[2]が実施した国際数学・理科教育調査によると、日本の中学生は、世界で数学5位、理科4位の学力を示している。それにも関わらず、数理系の学科の好き嫌いや、将来数学や科学を使う仕事に就きたいかといった学科に対する態度においては、参加国の中でほぼ最下位という統計が出ている。

裾野の拡大では、この両方の問題を解決したい。そのために、下記のような教材を開発することを狙いとした。

- ・ 小規模で負担が少なく、取り組みやすい教材
- ・ 完成したシステムが面白くて愛着が湧く教材

このような教材を完成させ、展示などを通して多くの人に興味を持ってもらうことにより、いろいろな人に組込みシステム開発の第一歩を踏み出してもらうことを狙っている。

3.2. 工業活動の全体を見据える教材

全体視野への移行の大切さは周知の通りであり、学校教育や企業のOJTでもそのための努力は惜しみなく進められている。

しかし、組込みシステムの教材に目を向けて見ると、現存する多くの組込みシステムの実習教材は、設計、実装に重点が置かれていることが多い。また、分析に関しては多くの優れた書籍が販売されているが、設計、実装まで連続的にカバーする教材は少ない。テストに関しても同様である。更に言えば、マーケティングや経営の視点を育成するための教材は、専門家向けには多く出揃っているが、組込みシステム技術者向けのやさしい入門用教材はなかなか存在しない。

そこで本論文では、次のような教材を提案する。

- ・ 教材に企画書相当の文書である、開発図書[7]を盛り込む。開発図書には市場分析の内容を盛り込み、製品企画から見直しを立てていく内容とする。
- ・ V字モデル[6]に沿った開発工程、すなわち、分析、設計、実装、テストの全てを、教材に盛り込む。

そして、この企画書を読んだ教材の使用者が、製品企画や市場分析の面白さを学び取り、その後の実習編でV字モデルの全工程を理解、実践できるようにすることを狙っている。

4. 初級組込み教材「鹿威し」の開発

4.1. 題材のきっかけ

鹿威しとは、日本庭園などでよく見られる、竹と石で構成された日本の伝統的な装置のことである。

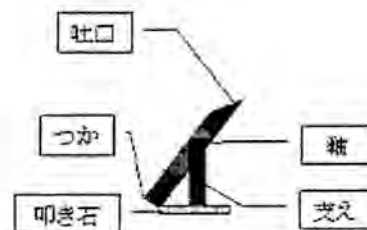


図1 鹿威し装置の機構

この装置は、給水、排水できる場所に設置して使用する。吐口に水が流れ込み、一定量以上溜まると動き出し、つかと叩き石が離れて水を排出する。水を排出すると、竹の重さで勢いよく元に戻る。この時につか叩き石が勢いよくぶつかり、心地よい打撃音響を発する。

動きは単純で親しみやすいが、メカトロニクスの面から見るとかなり奥深い装置である。この動きを力学的に解明してモデル化するのはかなり難しいからである。ただ、ハードウェア、ソフトウェアは単純で済むことが予想され、初心者向けの題材としてはちょうどよい難易度ではないかと考えた。

そこで SESSAME では、日本人に広く受け入れられやすい鹿威しの装置を、一般家庭に持ち込めるように小さくまとめ、教材化することを試みた。

4.2. ドメイン調査とマーケティング

組込みシステム開発においては、そのドメインのプロになることがエンジニア個人、ひいては組織的な競争力強化につながる。このことは、どのような製品を企画、開発する場合も同じであろうと考えられる。したがって、鹿威しのシステムを開発するためには、日本庭園や鹿威しのことに詳しくなる必要がある。

そこで、日本庭園に造詣が深い SESSAME の二上氏と坂本氏の両名が、自ら京都の詩仙堂に向き、この装置を家庭用にするために必要な事項を調査した。

この実地調査の結果、次のことがわかった。

- ・ 竹のつかと叩き石が発する打撃音と、庭園の音響効果が風流を生み出している
- ・ 置いてあると、竹に触って動かしたくなる衝動に駆られるという不思議な魅力がある
- ・ そのまま家庭に持ち込むには、音が大きすぎる
- ・ 水がかなり大量に必要となる

また並行して、家庭にこのシステムを持ち込む必要があるのかを調べるために、庭園のミニチュア・アクセサリ市場に関して、SESSAME メンバ間による小規模なマーケティングリサーチを行った。それにより、マンションなどでミニチュア庭園を造る小さなブームが発生していることもわかり、潜在的な需要があることも判明した。

以上の結果から、SESSAME では家庭用の鹿威しとして、循環水流型の小型鹿威しを企画することにした。この段階で、鹿威しシステムの開発コードネームを「SozeX」と名づけた。この名前は、鹿威しの別名が僧都あるいは添水ということに由来している。

4.3. プロダクトラインによる開発

プロダクトラインとは SEI/CMU (カーネギーメロン大学ソフトウェア工学研究所) にて定義されたソフトウェア開発のアプローチである。プロダクトラインとは、特定の市場のニーズを満たす特徴を共有する製品群のことを

を指す。

鹿威しシステムの開発でもこの考えを導入し、試作機 S000 から、量産初号機 S001、エコ運転対応機 S002、ライトアップ対応機 S003 までの一連のバージョンアップを、同一のコア資産で開発していくことを企画した。

企画内容の詳細は、開発図書[7]にて示している。

4.4. モデリングの方針

モデリングに当たっては、規模が小さいシステムであること、及び初級者の教育という目的を考え、DeMarco[9]が提唱する構造化分析、設計手法を用いることにした。教育用にはそちらの方がシンプルでわかりやすく、本質が伝わりやすい。そしてこれは SESSAME の初級者向け教育の方針でもある。

また、モデリングは下記のような方針で進めることにした。

- ・ コンテキストダイアグラムでは、ソフトウェアだけでなくシステム全体を考察する
- ・ DFD のプロセス名称は動詞句にこだわらない表現を用いる
- ・ プロセス仕様の表現は、文章や状態遷移モデルなどで示すこととする
- ・ 状態遷移モデルの複雑化を防ぐため、ムーアモデルを使用し、エントリアクションは存在するがアクティビティは使用しないこととする

4.5. 試作機 S000 型の開発

企画作成後、二上氏は試作機 S000 型の開発に入った。このバージョンは単に水を流して鹿威しを動作させるだけの単純なものであるが、試作エレクトロニクス版として、パソコンのシリアルポートからの指令を受信し、水流の開始/停止ができるようにした。これにより、水流をリモコンで制御した場合の使用性を調査した。

メカの部分は LEGO ブロックで作成した。



図2 LEGOブロックで作成された鹿威し

4.6. 初号量産機 S001 型の作成

S001 は、小型 CPU とソフトウェアを用いて実現する初の鹿威しシステムである。機能仕様は以下の通りである。

1. 赤外線リモコンにより、水流の開始/停止ができる
2. 運転忘れ防止のため、180 分運転状態が続くと自動停止する

この段階でシステムをモデリングする作業に入った。S001 型のモデルは、二上氏が作成した。コンテキストダイアグラムは次の図のようになる。

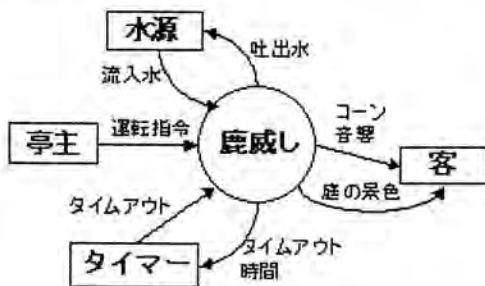


図3 鹿威しS001型のコンテキストダイアグラム

この鹿威しシステムを機能分割した DFD は次のようになる。

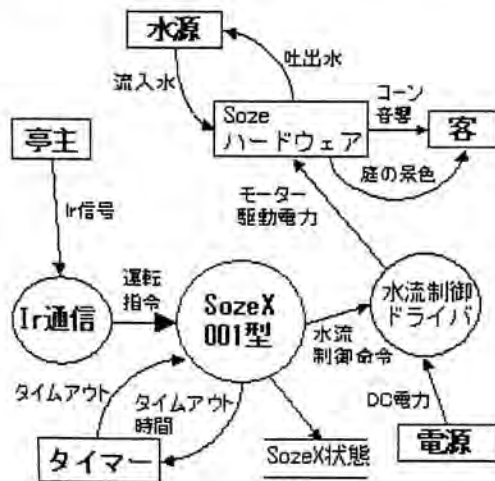


図4 鹿威しS001型のDFD0

図の SozeX001 型のプロセス仕様は、状態遷移モデルで表現できる。これを言葉を使って下記のように表し

た。

(SozeX001 型プロセス仕様)

SozeX 状態 = “待機中” の場合:

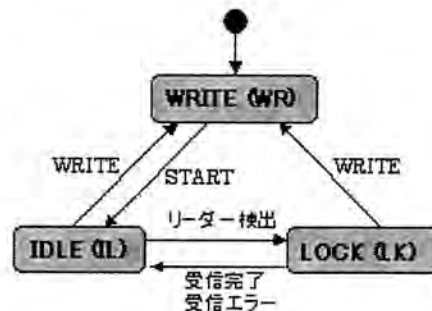
- 運転指令: = “運転” が着信したら、
- ・水流制御命令 = “On” を出力する
- ・タイマにタイムアウト時間を設定する
- ・SozeX 状態 = “運転中” として遷移する

SozeX 状態 = “運転中” の場合:

- 運転指令: = “停止” が着信したら
- ・水流制御命令 = “Off” を出力する
- ・タイマをキャンセルする
- ・SozeX の状態 = “待機中” として遷移する
- タイムアウト: = “Yes” が着信したら
- ・水流制御命令 = “Off” を出力する
- ・SozeX 状態 = “待機中” として遷移する

このモデルにしたがって、二上氏の手により、M16 マイコンによる S001 型の実装が行われた。

並行して、SESSAME の山崎氏と著者の一人である森が、共同して赤外線リモコンのプロセス仕様を作成した。その状態遷移モデルを下記に示す。



- WRITEモード = 受信機のパラメータ設定モード
- STARTコマンド = ホストから送信されるコマンド
鹿威し本体が動作可能になったことを示す
- WRITEコマンド = ホストから送信されるコマンド

図5 赤外線リモコン受信部の状態遷移モデル

このモデルにしたがって、森が赤外線受信素子を利用した受信ソフトウェアを作成した。

また著者の一人である西は、赤外線受信モデルの状態遷移パステストと状態遷移マトリクステストを設計した。また、赤外線受信ソフトウェアのソースコードに対するユニットテストの事例として、制御パステストの設計も

行った。そしてこれらを、初級者向けソフトウェアテスト教材としてまとめ上げた。

4.7. エコ運転対応機 S002 型の作成

S002 は, S001 の機能仕様に, 水資源を有効利用するための機能と, 故障検出機能を追加したシステムである。機能仕様は以下の通りである。

(S001 からの流用仕様)

1. 赤外線リモコンにより, 水流の開始/停止ができる
2. 運転忘れ防止のため, 180 分運転状態が続くと自動停止する

(S002 からの追加仕様)

3. 竹のつかと叩き石が離れている間は運転中でも水を停止する
4. 停止の際は, 竹が腐らないように, 竹の吐口から一旦水を排出してから停止するようにする
5. 運転中でありながら, 80 秒以上連続してつかと叩き石がくっついたままの場合は, 故障とみなしてユーザーに知らせる
6. 運転中でありながら, 20 秒以上連続してつかと叩き石が離れたままの場合は, 故障とみなしてユーザーに知らせる

上記の追加仕様を実現するため, つかと叩き石のところに接点スイッチを設けることになった。



図6 接点スイッチ

S002 型の最初のモデルは, 二上氏が作成した。コンテキストダイアグラムは S001 とほぼ同じため省略し, DFD の作成から入った。DFD0 は次のようになる。

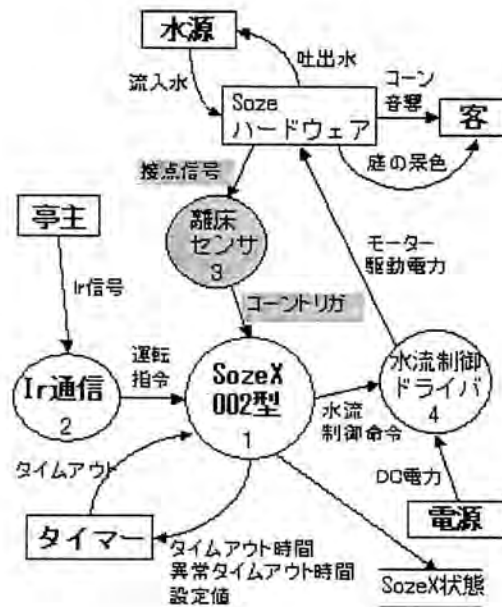


図7 鹿威しS002型のDFD0

SozeX002 型プロセスを分割した DFD1 は次のようになる。

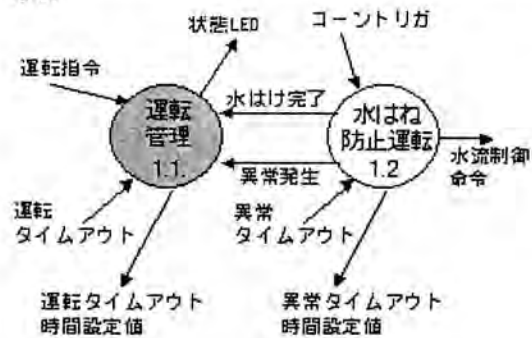


図8 鹿威しS002型のDFD1

このモデルでは, 水はね防止運転プロセスの起動と終了を, 運転管理プロセスが管理する。

運転管理プロセスのプロセス仕様は次のようになる。



図9 運転管理プロセス状態遷移モデル 第1版

水はね防止運転プロセスのプロセス仕様は次のようになる。

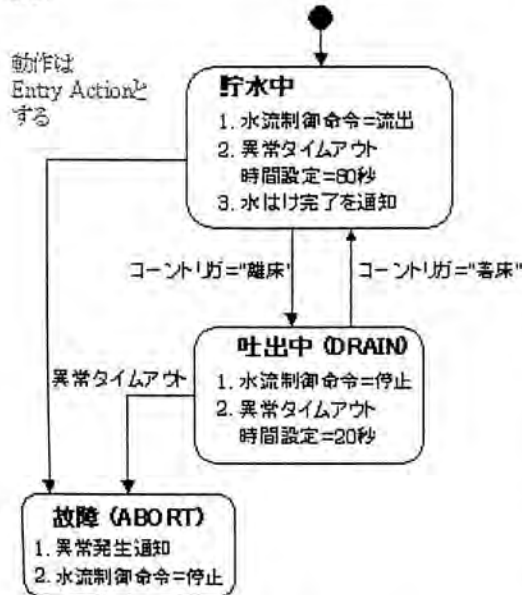


図10 水はね防止運転プロセス状態遷移モデル

このモデルに対して、SESSAME ではメーリングリストを活用した公開形式で、設計仕様書を対象にデザインレビューを行った。その結果、このモデルは機能仕様4.を満たしていないことが判明した。ユーザーがリモコンにより停止を行うと、水が途中まで溜まったまま水流が停止するからである。

これを改善するため、新たな運転管理プロセスの状態遷移モデルを作成した。それを次に示す。

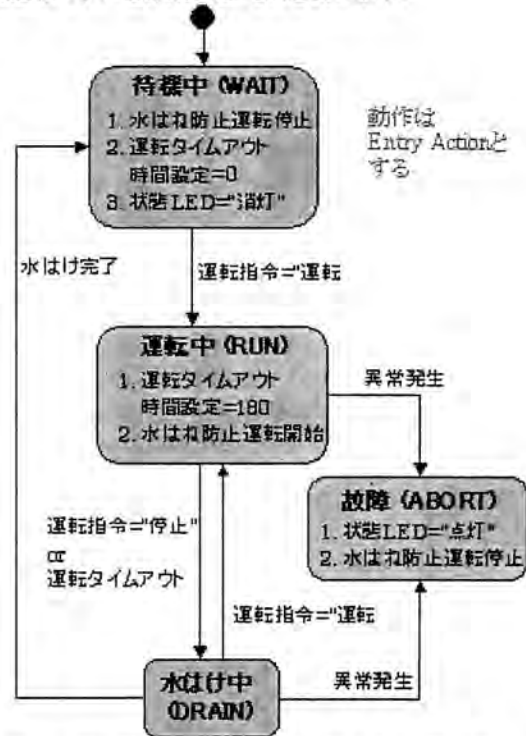


図11 運転管理プロセス状態遷移モデル 第2版

厳密にはこのモデルであっても問題は残っている。停止して竹から水が排出された後、一瞬水が注入される可能性がある。この問題は未だ解決されていない。しかし、セミナー開催という納期の制約があること、システムに搭載されるモーターの仕様上、実機では何の問題も発生しないという2つの理由から、S002はこのモデルの通りに実装され、セミナーや各ワークショップでのデモンストレーションでそのまま使用されていった。このことをセミナー受講者にも説明することで、現実問題における論理的不具合と実装における割り切り面でのトレードオフがあることを伝えることができた。

また、S002の開発では、ソフトウェアとメカとの「すり

あわせ」が発生した。LEGO で組み上げたシステムは、つかと叩き石が離れてすぐに水を停止すると、つかの部分がかまく持ち上がらないのである。そこで、つかと叩き石が離れても、1秒程度は水を流しつづけるように仕様を変更し、対応した。これは物理的制約により、実装工程でソフトウェアの仕様変更が発生するという事例となった。鹿威しはその性質上、教材化した場合もこのようなすり合わせは不可欠になるので、学習者もこのような体験ができる。このことは、教材として有用ではないかと思われる。

4.8. コンカレントエンジニアリングの遂行

S002 型は、3rd Open Sessame Seminar の開催まであと一ヶ月を切った段階から、セミナーの教材として使用するために開発が開始された。しかも、その頃はまだハードウェアも完成していない状況であった。

この状況を打開すべく、二上氏はメカとハードウェアの実験を進め、その裏で森は S002 シミュレータを作成してソフトウェアを実装し、更にその裏で、リアルタイムシステムとしてテストするためのテスト仕様を著者の一人である大西が作成した。

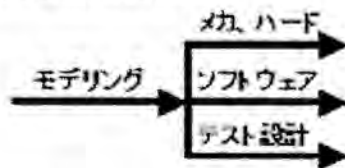


図12 コンカレントエンジニアリング

コンカレントエンジニアリングは結果として成功し、セミナーのリハーサルの日にはシステムテストまで無事遂行することができた。

これが成功した鍵は、下記の2つが挙げられる。

- ・ 小さなシステムでありながら、きちんとモデルベースで開発したこと
- ・ モデリングで抽出したイベントは、受信するための処理を除けば、非常に抽象化されていたこと

シミュレータ開発においては、上記の性質を最大限に生かし、イベント受信処理をソフトウェアの最下層に位置させるようにモジュール構造を設計した。このモジュール構造図を次に示す。

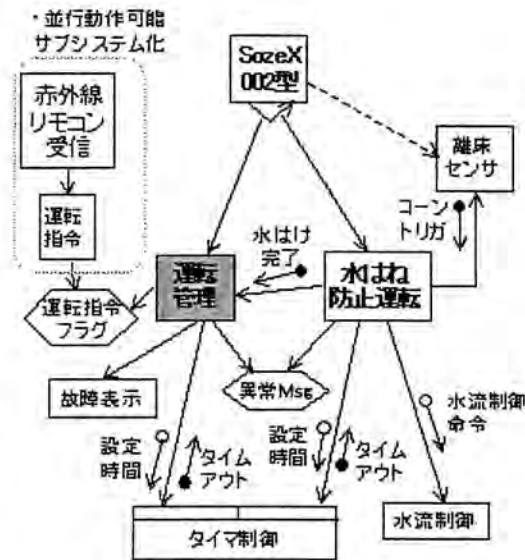


図13 鹿威しシステムモジュール構造図

つかと叩き石の接点スイッチからの入力、最下層モジュールである離床センサに押し込められている。また、リモコン受信モジュールは独立している。そして、この2つのモジュールにより、外部の具体的なイベントが抽象化される。これを利用して、その他の部分をシミュレータとして事前に作成し、擬似的に外部入力を与えることでソフトウェアが事前にテスト可能となった。また、シミュレータの作成により、実機ではほとんど再現できないタイミングのテストなどが遂行できるようになった。

これらの事例は、コンカレントエンジニアリング、及びハードウェアが存在しない状況でのシミュレート開発の事例として、各種セミナーで説明を行った。そしてその鍵は設計の洗練と上流工程の先行であることを示すことができた。

5. 運用の結果

5.1. 2nd Open Sessame Seminar in Osaka での運用

2003年6月に開催された2nd Open Sessame Seminar in Osakaは、SESSAMEが主催する初級組込み技術者向けのセミナーである。この時には、分析、設計の題材としてS001、S002型の本体モデルを使用した。また、S001型の赤外線受信モデルを、プログラミング及びテストの題材として使用した。

分析、設計の演習では、説明、実習、解説合わせて

それぞれ1時間ずつ、合計2時間使用した。受講者の多くは、「講義や、回答例の解説を聞くとわかるのだが、自分で実践するのは難しい」と口を揃えていた。また、構造化手法になじみのない受講者にとっては、DFDやStructured Chartにも戸惑いを感じていたようである。ただ、鹿威しのモデルを用いて、要求と仕様のモデルの違いや、仕様とはどこまでを指すのかを受講者に提示することには成功しているようであった。

プログラミング及びテストの演習では、説明、実習、解説合わせてそれぞれ30分ずつ、合計1時間使用した。題材が赤外線モジュールというやや難解な部分だったこと、時間が短かったことから受講者にとっては大変な演習だったようである。またこの演習でも、「聞くとわかるが自分では難しい」という意見は多く聞かれた。

以上から、初級者向けセミナーにおける分析、設計の教育では、題材は小さく、時間はしっかり取ることが必要であるとわかる。鹿威しであれば、S001の規模、レベルで十分で、それでも演習に時間を割いて本質的な分析、設計に集中してもらうのがよいと感じた。

5.2. 3rd Open Sessame Seminar での運用

2003年10月に開催された3rd Open Sessameは、SESSAMEが主催する中級組込み技術者、初級管理者向けのセミナーである。この時には、S002型の本体モデルが、初級コースのまとめ、及びリアルタイムシステムのテストの題材として使用された。

初級コースのまとめでは、構造化手法による分析、設計、実装からテストまでのまとめを、約1時間説明した。この時のアンケートからは、中級者向けには、S002のモデルで説明を進めても理解できるということが判明した。むしろ、システムの理解はできるから、分析、設計手法についてもっと突っ込んだ説明が欲しかったという声が多かった。もしこの要望に応えようとする、S002型を1~2日かけてゆっくり説明する必要があると考えられる。初級コースのまとめを1時間で行う場合は、あくまで復習コースとするのが妥当である。

リアルタイムシステムのテストでは、約10分間、実際に受講者がシステムを動かしてテストするという演習を行った。

結論としては、S002の規模になると、セミナーで行う実習型の題材としてはかなり大規模だと感じる。S002以降のバージョンは、企業内で時間を掛けて教育したり、座学でじっくり勉強するのに適した題材ではないかと考えられる。

5.3. 第7回 CEST セミナーでの運用

CESTとは、Consortium for Embedded System Technology[3]の略で、愛知県三河地区を中心とした産学共同の研究・開発活動を行う団体「組込みシステム開発技術研究会」のことである。この研究会では、活動を通じて会員の組込みシステム開発技術を向上させること、また会員企業の産業分野における競争力の向上を目指している。

この研究会は、定期的に技術セミナーを開催している。その第7回セミナーにて、鹿威しを題材に、組込みソフトウェアのテストに関して一連の解説を行った。

受講者の声としては、テストという分野に対する反応がよく、「初級者向けテストの重要性がわかってよかった」「教育のヒントが得られた」という声が多かった。

鹿威しに関しては、「この開発ストーリーはかなり実務に近いので面白い」という声や、「不具合が生じる仕様、設計をレビューやテストで実際に潰していく仮定が公開されてよい」という声が聞かれた。

5.4. JaSST2004 での運用

テスト技術者交流会(TEF)[5]が主催するソフトウェアテストシンポジウム2004(JaSST2004)にて、鹿威しS002型のシミュレータを題材にして、各界のテスト専門家に実際にテストを設計して頂き、現場でテストを行うというライブ形式の催しが開催された。この催しは、SESSAMEが主体となって、企画、運営を行った。

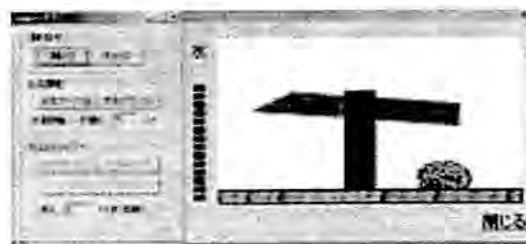


図14 鹿威しシミュレータ

このライブは受講者には非常に評判がよく、一部からは社内教育の題材としても使用してみたいとの声が上がった。

エラーシーディングしたシステムをテストの学習の教材に使うというのはありふれた発想であるが、シンポジウムの準備を通して実際の教材までまとめ上げたことには意味がある。また、テスト専門家を招聘し、実際にテストを設計しその知見を得られたことも有用であった。

また、社内教育の題材として鹿威しシミュレータが候補に挙げられたことは、このシステムがテストの教材としても適切であると判断された結果である。

5.5. 他団体との協調

名古屋大学の高田教授が主催するTOPPERS[4]の教育WGにて、鹿威し教材をRTOSを用いて実装する試みが実行され、見事成功した。これにより、本教材の仕様が、RTOSの教材としても使用できる目処が立った。またこの成果は、ET2003にて紹介され、多くの人の目を引く展示となった。

鹿威しはもともと愛着が湧く見た目をしており、人目を引きやすい性質がある。これがSESSAME以外の多くの団体でも実装、展示されたことで多くの人に認知された。これが功を奏したのか、その後もともと組込み開発の経験がない方々から「是非とも鹿威しを作ってみよう」との声を多数得ることができた。この動きが広がり、組込みシステム開発に対する心の壁を破る一つのきっかけになれば幸いである。

6. 教材の有用性

6.1. 全工程、全成果物を公開できる教材の重要性

現実の組み込みシステム開発では、V字モデルに従って、何の問題も発生しないままシステムが完成することはまず有り得ない。また、納期の問題もある。そのような問題との格闘、納期との格闘の中にこそ本来の工業活動に役立つ知見が存在するはずである。しかしそれは機密の問題からなかなか公開されにくい。

ところが、鹿威し教材の開発事例なら、それが可能になる。SESSAMEの有志は業務時間外にボランティアで成果物を作成していることから、各人の了解さえあれば成果物を洗いざらい公開できる。これは非常に大きな利点である。

第7回CESTセミナーを主催していた高田教授も、セミナーを聞いた後で、この教材が公開できることの有用性に関して言及されている。

どのあたりが公開されると有効なのか、ここではいくつか具体例を挙げておく。

(1) 納期との格闘

本教材の開発では、定期的なセミナー開催やイベントの開催に間に合わせるために、開発者はまるで本当の工業活動のような「納期との格闘」を強いられた。これが非常に良かった。納期との格闘を行うと、当然ながら問題が発生する。本教材開発においても当然のようにいろいろな問題が発生した。そしてこれらの問題が仕

様変更やアーキテクチャ変更による改善の事例を生み、教材開発の上で非常に役立つ。

(2) テスト教材の有用性

ソフトウェアテストの重要性を身を持って知るには、不具合が入っているソフトウェアを実際にテストしてみて、ブラッシュアップさせる経験を積むことが有用であろう。また、V字モデルに沿って全工程の成果物に対してテストすることで、真の有用性が理解できる。

鹿威しの教材であれば、全工程の成果物が公開されるので、それが可能となる。これを利用して、開発の全工程のテストに関する事例教材が完成した。

また、納期との格闘により、ある程度不具合が混入していたところに、実際にテストを実行し、不具合がどんどん浮かび上がってきた。これがテスト教材の有用性をより高めている。JaSST2004でのテストライブの試みは、バグを仕込んで教育に使用するという点で、この形の更なる発展形といえる。

(3) ハードウェア開発の遅れへの対処

組込みシステム開発では、常にハードウェア開発の遅れがリスクとして挙げられる。本教材開発でも、部品調達の問題が発生した。その状況下で迫り来るセミナーの日程に対処するため、シミュレータによるハードウェアとのコンカレント開発を行った。そしてこれがインクリメンタル開発の設計と、それにより可能となるコンカレント開発の貴重な事例となるとともに、リアルタイムシステムの動的テストを可能にする方法の事例ともなった。

6.2. すり合わせの体験

鹿威し教材の実装は、基板のはんだづけから始まる。そして、最終工程では、竹の絶妙な位置に穴を開けて、水量による動作と復元力のバランスを取る行為が待っている。この穴の位置は、数式を用いて導けると考えていたが、竹の材質や、節の位置、質的密度といった複数の要因を考慮した一般式を導くことが困難であった。そこに労力を費やすよりも、経験則という暗黙値を用いるのが現実解としては良いというすり合わせの事例だと認識するに至った。ソフトウェアだけでは制御しきれない物理世界を味わうことは、若年層の教育に大きな効果をもたらすであろう。

6.3. ドキュメンテーション

どのようなシステムであっても、その要求や仕様を正確に記述しようとする、ドキュメントの量は非常に膨大となる。そして、そのような正確な記述こそが、後工程をスムーズにする鍵である。このことは誰も理解している

ことだが、実際の工業活動ではなかなか実現できるものではない。

そこで SESSAME の有志は、開発図書[7]、及び SozeX002 ソフトウェア仕様書[8]にて、鹿威しシステムの要求や仕様を正確に記述し切ること挑戦した。DFD0 レベルでプロセスが 4 個、ソースコードにして実行数が 2300 行前後という小規模なシステムでありながら、その語数は企画書で 6300 語、仕様書で 6500 語という規模である。

この事実から、たとえ小さなシステムであっても正確な記述を行うためには大きな労力が必要であるということがわかる。また、後に SESSAME の他にもいくつかの団体がスムーズに鹿威しの実装に成功したことが、ドキュメンテーションの効果を物語っている。

6.4. 品質意識の向上

ISO/IEC9126 の品質特性を大別すると、内部品質、外部品質、利用時の品質に分けられる。

ソフトウェア技術者が、内部品質、外部品質のみの近視眼的な視野に陥らないようにするために効果的なのは、愛着が湧くシステムを構築し、実際に運用させてみること、そして運用時に苦労した点を品質特性と結びつけて意識させることである。特に、学生においてはモチベーションの基点として、この「愛着」がより大切になるであろう。

そこで鹿威しである。このシステムは、そもそも日本人の特性である「わび、さび」に訴えかける見込みをしており、非常に愛着が湧く。しかも、メカトロニクスは重力と静的構造を利用した構造であるから、時として水をこぼしたりするなどハプニングが続出する。これを何とかしようとする気持ちが、仕様と実世界をすり合わせる気持ちを生むという副次効果がある。

いくつかの団体で作成した鹿威しの内部仕様が、どれも少しずつ好みで変えられていることは、どの作成者もこのシステムに愛着を持って取り組んでいる現われであろう。品質意識の向上という面では狙い通りである。

初心者用、学生用の教材として、今後もこのような効果が現われることを期待する。また、教材としては品質関連の部分より強化して行きたいと考えている。

7. まとめ

初級者用の教材として、鹿威しは以下の内容をバランスよく含むことに成功した。

- ・ 企画、要求分析、仕様、設計、実装、テスト
- ・ ソフトウェア、ハードウェア、メカトロニクス

- ・ 日本人特有の「すり合わせ」
- ・ ドキュメンテーション
- ・ 品質意識

そして、企画、開発からシステムテストまでの経緯や成果物は全て公開可能である。また、規模が小さく、短時間のセミナーで用いるには適しており、自学自習でも無理なく学ぶことができる利点がある。

教材としては、約 1 年の歳月を経てようやくまとまりつつある。この教材は、教科書、ソースコードなどの大半のコンテンツはパブリックドメインであり、いろいろな方に有用に使える教材であると思われる。

また、この教材なら、企業人だけでなく学生や子供向けの教材としても発展させられる可能性がある。今後の展開としてはこれらも重要であり、また楽しみでもある。

謝辞 鹿威し教材を企画し、教材開発を推進した原作者の二上氏、教材開発に多大なご支援を頂いた坂本氏、山崎氏、吉澤氏をはじめ、この活動に協力・貢献頂いた SESSAME の多くの方々に、強く敬意と謝意を表したいと思います。

参考文献

- [1] <http://www.sesame.jp/>
- [2] <http://www.iea.nl/>
- [3] <http://cest.ich.tutkie.tut.ac.jp/CEST/>
- [4] <http://www.toppers.jp/>
- [5] <http://blues.se.ucc.ac.jp/swtest/>
- [6] Boehm, 1979, Guidelines for Verifying and Validating Software Requirements and Design Specifications, Euro IFIP 79, P. A. Samet (editor), North-Holland Publishing Company
- [7] 二上貴夫, SozeX 開発図書 r4.1, SESSAME
- [8] 二上貴夫, SozeX002 ソフトウェア仕様書 r1.2, SESSAME
- [9] Tom DeMarco. *Structured Analysis and System Specification*, Prentice-Hall, 1978
- [10] 藤本隆宏, 能力構築競争—日本の自動車産業はなぜ強いのか, 中央公論新社, 2003

インターネット早期広域攻撃警戒システム WCLSCAN

鈴木 裕信
鈴木裕信事務所
hironobu@h2np.net

石黒正輝
三菱総合研究所
masa@mri.co.jp

村瀬一郎
三菱総合研究所
murase@mri.co.jp

大野 浩之
通信総合研究所
hohno@ohnohlab.org

1. 要旨

インターネット上に設置されたセンサーに到着する IP パケットを計測・集計し、ベイズ推定に基づき広域的なネットワーク攻撃の活発化によるインターネットの危険状態を検知するシステム (以下 WCLSCAN) を作成し運用実験を行っている。本発表ではそのシステム概要、評価、実際のケース紹介、一般ユーザの利用方法を報告する。

2. はじめに

近年においてはシステムの脆弱性の発見から exploit コードや攻撃コードの出現までの期間は非常に短くなっており、0-day Attack のように脆弱性情報が公開される前に、その脆弱性を狙った攻撃も行われている[1]。一方でベンダー等による脆弱性対応も進んでいるため、攻撃側は脆弱性が有効である短期間に広域に攻撃する傾向が見られる。また Slammer や MSBlaster のように極めて短時間で一気に広がるようなワームが発生するため 24 時間 365 日体制の監視を続ける必要がある。本研究では、広域に影響するインシデントでは不特定の IP アドレスにたいしても攻撃 IP パケットが増えることに着目し[1][2]、その到着を計測しベイズ推定をもちいて危険度を計算し[3]、危険レベルを報告するシステムを作成した。

3. システム概要

3.1. WCLSCAN の処理ステップ

Step1: インターネット上に設置した複数センサーボックスで IP パケットの到着を記録し、暗号化した後、集積システムへ送る。**Step 2:** 集積システムはそれを復号化しデータベースへ記録する。**Step 3:** 計算システムは一定時間毎(現在 30 分間隔)で、データベースから情報を取り出し危険度を計算する。**Step 4:** 表示作成システムは計算結果を受け取り、グラフや見やすい表を作成し、公開用 Web サイトへ転送する。作成するために必要な情報はデータベースから取り出す。

3.2. 危険度の通知

Web サイト上に情報を転送し、PC などで参照が可能である。また簡易 HTML(I モード用)、HDML (EZWeb 用)を用意している[4]。筆者の場合は携帯電話からの確認回数の方が多い。危険度が高まったときメールで通知する機能もある。

3.3. ベイズ推定による攻撃検知手法

インターネット上の危険度は、時間とともに変化するため、観測に対してベイズ推定に基づき危険度推定値を繰り返し更新することにより、危険度の動的な変動に対応した推定を行う必要がある。本システムでは、ポートスキャン頻度をそのトレンド(移動平均)からの差として観測することにより危険状態を推定する。尚、ここで危険状態とは“ネット上の広域的な攻撃活動の活発化により自サイトへの攻撃による被害が発生する可能性の高い状態”と定義とする。

4. まとめ

ベイズ推定の手法が有効であるかどうかの評価実験を行った結果、有効であることがわかった。また Doomjuice[5]が発生した際には、ウィルスベンダーが警告を出す前に異常を検知し、すばやく危険な状態であることが認知できた。一般にも公開しているので、広く利用されインターネットセキュリティ向上に役立てば幸いである。

参考文献

- [1] 鈴木裕信, ポートスキャンログ分析からみたインターネットセキュリティの一考察, SEA ソフトウェアシンポジウム 2001 予稿集, 2001
- [2] <http://isc.incidents.org/>
- [3] 石黒正輝ほか, ベイズ推定に基づくインターネット攻撃検知システムの開発, SCIS2004 予稿集, 2004
- [4] <http://www.clscan.org>
- [5] <http://www.f-secure.co.jp/v-descs/v-descs3/doomjuice.html>

WCLSCAN

インターネット 早期広域攻撃警戒システム WCLSCAN

鈴木裕信^{※1} 石黒正輝^{※2} 村瀬一郎^{※2} 大野浩之^{※3}

※1 鈴木裕信事務所

※2 株式会社三菱総合研究所 情報技術研究部

※3 独立行政法人通信総合研究所 情報通信部門非常時通信グループ

WCLSCAN

概要

- 背景・モチベーション・ゴール
- システム概要
- ベイズ推定による攻撃検知手法
- 評価実験・ケース紹介
- 一般ユーザ利用方法
- 今後の課題
- まとめ
- デモ

背景・モチベーション

SS2001で
発表

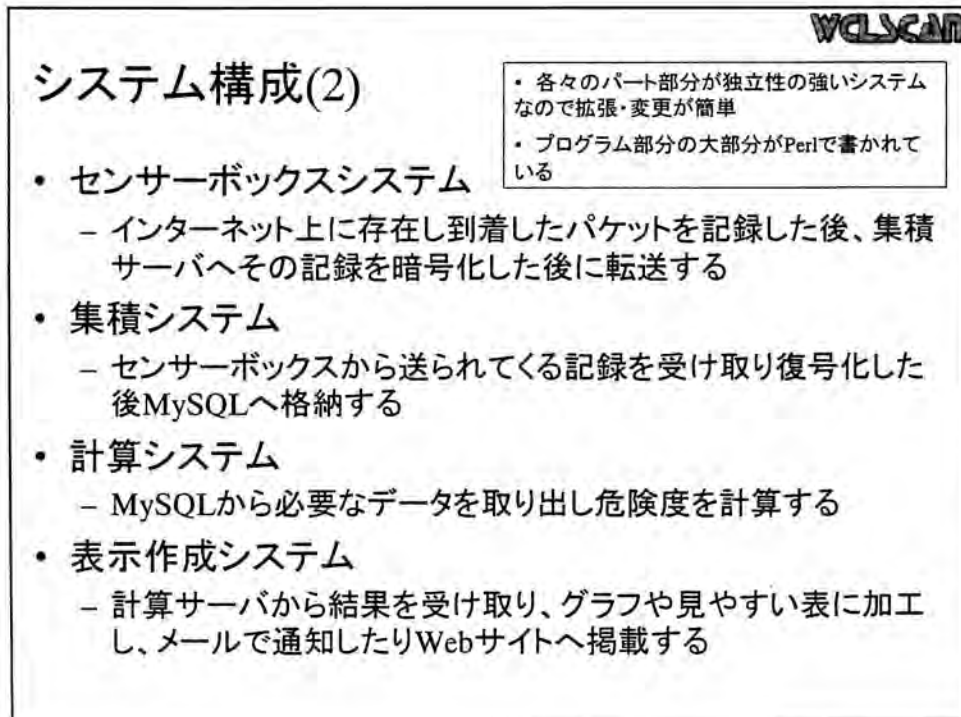
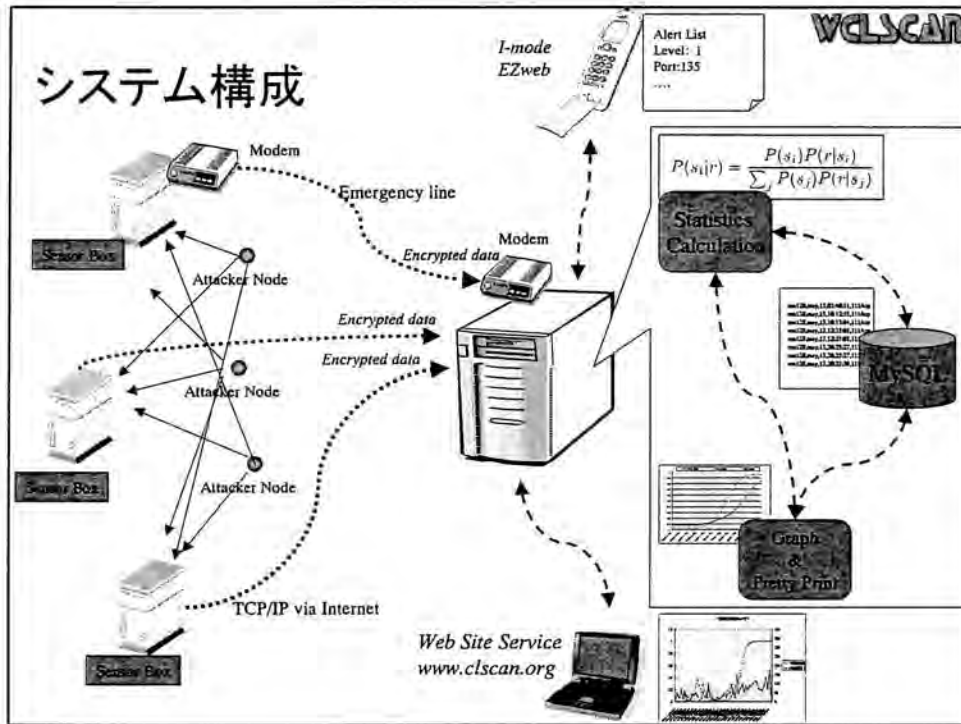
WELSCAN

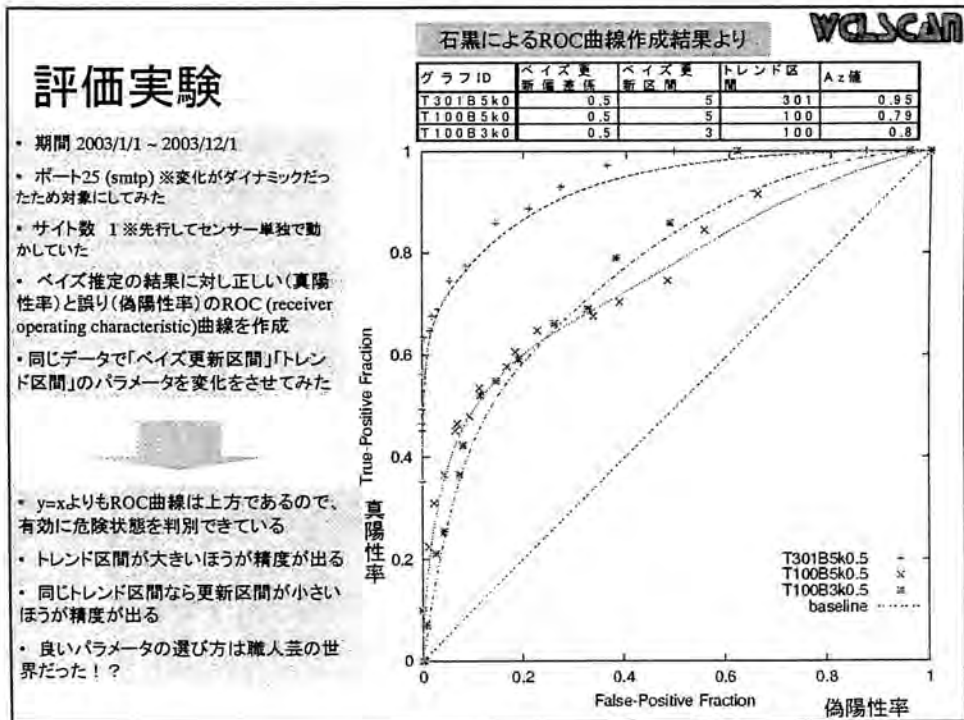
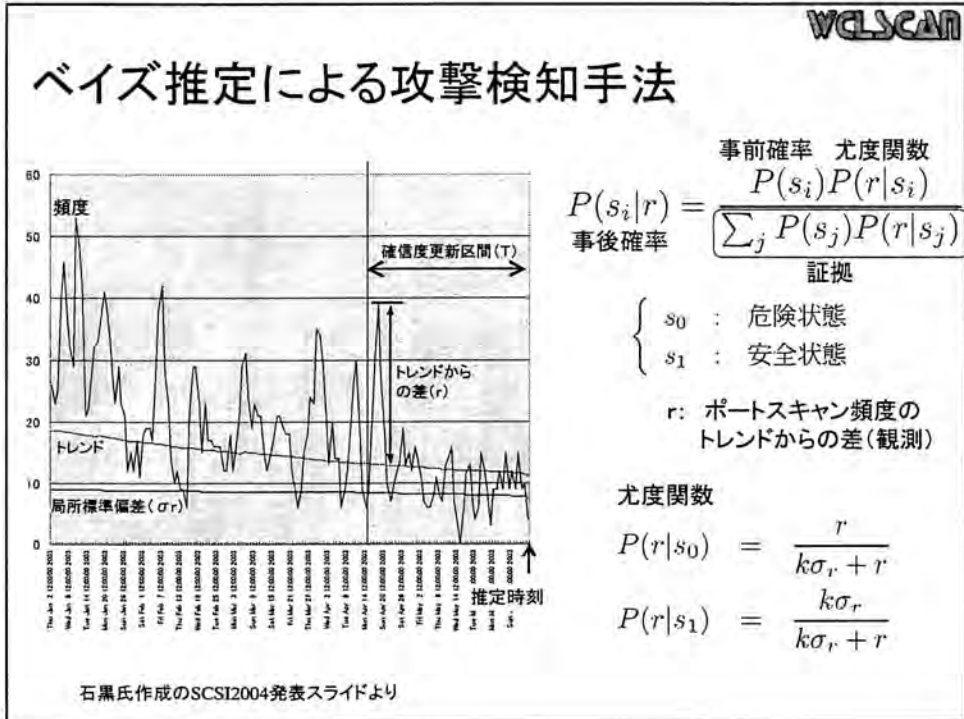
- 広域に影響するインシデント(アタック)では攻撃IPパケットが増える
 - 単体サイトだけだと局所的な偏りがあるかも知れない
⇒複数センサーでの検知が必要
- どのような攻撃が行われたかより攻撃が行われた兆候を見つけることが最優先したい
 - セキュリティ組織が警告するのと攻撃にはタイムラグがある⇒知識ベース方式では不十分
 - パケット統計を分析したら0-dayアタックらしきものを発見⇒シグニチャー方式では役に立たない

ゴール

WELSCAN

- 24時間365日自動攻撃検知する
 - 状況を診断するような専任オペレータは不要
- 広域なインターネット上で発生している攻撃を自動的に検知する
 - “広域情報収集／データベース化／自動化分析”の3つの機能からなり独立しているので拡張が簡単
- 自動的に解析し危険度を予想する
 - ベイズ推定(推測統計学)を使い危険度を計算させるので確信度が高い
- いつでもどこでも知ることができる
 - パソコンからだけではなく携帯電話からのチェックも可能なのでいつでもどこでもチェックができる
 - デスクトップ上から常時モニタリングができる

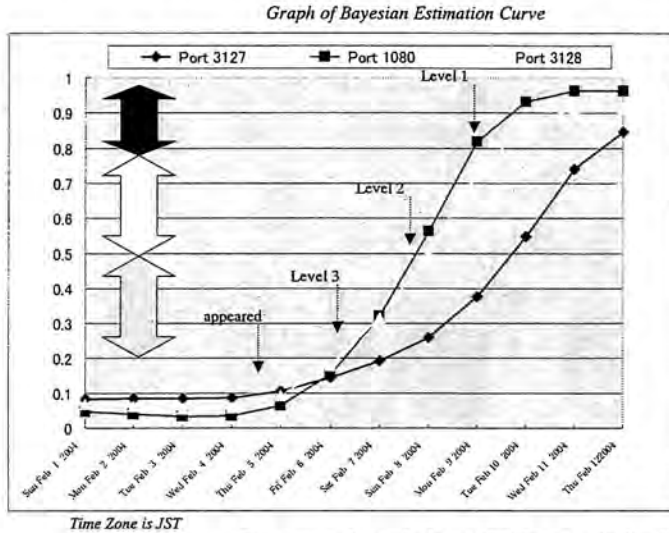






Doomjuiceケース まったく新しい攻撃が発生した場合

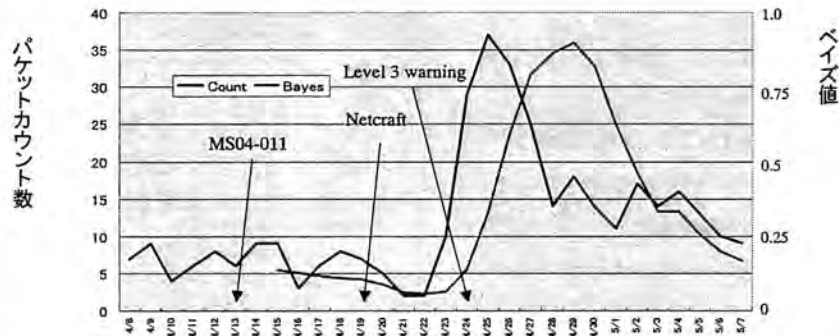
- 2004/2/7 (昼ごろ)レベル3に見たことのないポートへのアクセスが増加したとあった
- 2004/2/7 (午後4時) iWR 研究者間メーリングリストへMydoomバックドア狙いのポートスキャンが増えていることを報告
- 2004/2/9(日本時間)ウィルスベンダによるアナウンス
- 2004/2/10 (日本時間)メディアによるアナウンス
- 2004/2/11 @pliceによる注意喚起掲示



Detected activity of "Doomjuice"



SSL BOMBケース たぶん来るだろうと待ち構えていた場合

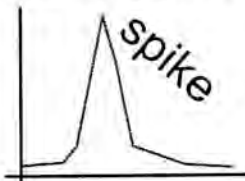
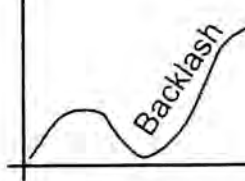


- SSL BOMB --- MS-IISへのDoSアタック
 - 2004/4/13 MS04-011でTCP/SSLの脆弱性を公表
 - 2004/4/19 Netcraftがexploitコードが公表されていると警告
 - 2004/4/24 IWRでlevel 3にリストアップされる
 - 予想通りの攻撃をセンサーが捕らえた

WCLSCAN

ベイズ推定の癖

- Spike effect
 - 一部センサーにスパイクが観測されたとき、それが全体の総数をひっぱってしまい、影響が後々まで出てしまう
- Backlash effect
 - 一度ベイズ値が上昇した後に減少した状態からは、ベイズ値の上昇が早い


WCLSCAN

一般ユーザの利用方法

- Web Brower from PC/PDA
 - www.clscan.org
- I-mode (NTT DoCoMo)
 - www.clscan.org/iwr/i.html
- Ezweb (KDDI)
 - www.clscan.org/iwr/ez.html

2004/2/19 19:31
IWR:
L P B C
1 80 0.88 117
1 135 0.84 287
2 901 0.69 9
2 443 0.50 2
3 1080 0.50 126
3 3128 0.49 130
3 12345 0.45 4
3 3127 0.42 209

携帯電話の
画面イメージ




- ほとんど場合、携帯電話からのチェックで十分
- レベル1発生時に携帯メール通知する機能は“ウザイ(煩わしい)”ので中止

WELSCAN


一般ユーザの利用方法 *Update!*

- Facial indicator
 - GNOME アプレットを開発
 - デスクトップ上から常時モニタ
 - Webサイト上のアイコン
 - パソコン上から
 - 携帯電話から


Smiley Uneasy Distress Panic




(^_^)v



(-_-;)



(-_-T)



(@o@#)

WELSCAN

今後の課題

- 基本システム作成及び評価終了しセンサー設置台数の拡大へ
 - データとして意味のあるものを検知するために独立した専用センサマシンを配置中(現在4台)
- データベースに格納されているログを使ってさらに色々な分析を検討中
 - www.sco.comへのDoSアタックの痕跡を見つけることもできた
- 本プロジェクト参加間でのコミュニケーション
 - 独立行政法人通信総合研究所情報通信部門非常時通信グループ・株式会社三菱総合研究所情報技術研究部・大阪大学大学院情報科学研究科・群馬大学工学部情報工学科・鈴木裕信事務所・ほか1社

WELSCAN

まとめ

- インターネット早期広域攻撃警戒システムの開発と運用を行っている
- ベイズ推定による危険度を計算し、その結果を一般に公開している
- パソコンからだけではなく、携帯電話からも確認できる
- 評価実験では有効であることが確認され、実際のケース紹介のように役に立っている
- さらに充実したシステムにするために現在活動中
 - 共同研究提案Welcomeです！！

WELSCAN

デモ

<http://www.clscan.org>

ソフトウェアプロセス最適化マニフェスト

伊藤 昌夫

(Nil Software)

SEA-SPINの活動が始まって以来、色々なことを考えてきましたし、MLでも発言してきました。しかし、いつも私自身は不完全燃焼のまま終わっていました。

それは主としてCMMに関してです（MLの最初の頃は特にその議論が多かったように思います）。当時は、ちょっと tricky な言い方でしたが、CMM はレベル 5 即ち最適化しか意味がないという言い方をしていたと思います。これは、レベル 5 の実践に意味があるというより、CMM が提示する階段そのものが極めて不自然に思えたので、階段そのものを否定するということだったのです。

それには、私が以前にお世話になっていた防衛産業においては、逆に問題ないという経験に基づいていたかもしれません（DOD-STD-1679Aで育ってしまうと実に良く分かる階段なのです。防御線を張っておくと、continuous model でも上記は同じと思っています）。

不完全燃焼を言い換えると、観念主導によるある種の嘔吐感といえるかもしれません（もちろん、Sartre的意味です）。これは例の日本版CMMに関する専門委員会の議論によって、私の中で最高に達したまま、未だに解消していませんでした。

日々、仕事をする中でプロセスについて考えるのですが、何が欠落していたかに最近ようやく気がつくことができました。それまでの議論には、日々のソフトウェア技術者が会おう問題に対する解答が何ら存在しない！ということなのです。そのような単純なことに今になってようやく気がつきました。

私が、そしてプログラムを作っている人が切実に欲しているのは、問題の解消であって、ラベル付けや新たな役に立たない作業を押し付けられることではないのです。

そのための実践的な場を作るとというのが、私の engagement になります。

次のWeb Page:

<http://www.process.jp>

が、そのための実験的な場所です。まだまだ実験である以上、中途半端なのですが、これから小泉毅さんや熊谷章さんほかの同士と一緒に、のんびりと作りこんでいこうと思っています。

この話は昨年末のSPINの会合や、JASPICのワークショップで話をしました。ある程度、完成した所で仲間を増やすようにしようかと思いましたが、編集部の岸田さんからMANIFESTを書いたらとお勧めがあり、ここに示します。

もし、何か感じる場所があり、興味を持っていただけるならば、いつでも気軽にお声をかけてください。

++++
SPO (Software Process Optimization) MANIFEST
++++

- (1) QCDを総合的に向上させるためには、プロセスの広域的最適化が必要である。
- (2) ソフトウェア開発プロセスはモノではなく、良否はない。適しているか否かだけである。
- (3) 開発プロセスについて考えることは、標準化を進めることではない。
- (4) 特定のキーワード(例えば、CMMIやISO9000)は、何も解決しない。開発現場主導で考えるこ

と、それは何を良くし、何を悪化させるか。ちなみに、SPOはキーワードではない。それは問題解決のための行動そのものであるから。

ソフトウェアを作ることはムズカシイ。

ソフトウェアを作る人に何か問題があるとしたら、それは先ずソフトウェア作り、そのものに内在している問題である。未熟さを、問題に対する容易な解として選ぶべきではない。

例えば、進捗の遅れを考える。進捗の遅れは、ソフトウェア技術者のナマケによるものか、或いは、見積もりが不十分だったからか。では、ナマケをなくすように技術者を叱咤激励し、監視すべきなのだろうか。或いは、技術者の行動を逐一記録することによって、我々は高精度の見積もり手段を得ることができるのであろうか。

SPOの立場では、それは否と考える。ソフトウェアを作るということは、多数の複雑さをもった現実をプログラムに転写する作業である。或いは、誰かが転写したプログラムの上に、更に別の要素を加える作業である。そこには、一片の誤りも許されない。神経の集中を必要とする。かつ、問題を解きほぐすのに多くの執念を必要とする。

これだけの複雑さを扱う作業を核とする産業が、かつて、存在しただろうか。

我々ソフトウェア技術者は、日々、転写すべき世界を発見する。或いは、どう転写するかを発明する。そうしている人の行動を単純にナマケと呼べるか。苦勞の末解に到達する振る舞いを見て、明日のあなたにどれだけ役立つだろうか。

プロセスを見る、或いはプロセスについてかんがえるというのは、個人の監視でも、新たな標語の導入でもない。技術者のために、この発見・発明に集中できる環境を作ることである。或いは、発見・発明を極力しなくて済むようにすることである。

様々な粒度を持つ閉世界で、活動に関与する全ての要素を明らかにすること。その関与の度合いを調整すること。或いは、開世界として、これまでにない要素を導き、問題自体を緩和すること。それがプロセスについて考え、行動することになる。

もちろん、それでも我々は Sisyphus でありつづける。転写すべき世界は変わるという意味で、<今のところ>技術者の住む世界は、Thanatosであろうとし続けるのだから。

そこに救いがあるとすれば、技術者がプロセスを通じて連携することである。そうすれば、石を運ぶ仕事も、もっと楽しくなるに違いない。世界の転写をする仕事など、減多にないしごとなのだから。

概念設計についてのノート

岸田 孝一

(SRA-KTL)

0. はじめに

このノートは、昨年(2000年)の10月2～4日に山形県遊佐町で行われた第13回テクニカル・マネジメント・ワークショップでの「概念設計」についての短いプレゼンテーションのスライドをもとに、とりあえずメモ書きしてみたものである。草稿をワークショップ参加者およびSEA 幹事の Mailing List に投げてみなさんからいろいろなコメントをいただいた。それを取り入れてもう少しまとまった形にしようと考えていたのだが、雑事に追われてなかなか手がまわらない。まだほとんど未定稿の状態だが、異論・反論・Objection をいただければ幸いである。

1. 設計の本質

ソフトウェア・システムとは、それが対象とする世界(それは、われわれを取りまく全世界の一部分でしかないが)を何らかのかたちでモデル化したものに他ならない。

設計問題に関してわたしが愛読しているネルソン・グッドマンの名著: *Ways of Worldmaking* (邦訳は「世界制作の方法」みすず書房)の冒頭に引用されているエルネスト・カッシーラーのことは:

— 記号を用いて、数え切れないほどの世界が無から創り出される

が示すように、世界のモデル化にはさまざまなやり方が可能であり、結果として複数の世界が同時並列的に創り出されることになる。M.M.レーマンのE型プログラム(現実世界のアプリケーションに組み込まれるプログラム)のプロセス・フレームワーク[*1]では、それを複数のシステム・ビューと呼んでいる。

[*1] 関連論文を集めたレーマン先生の Web Page は ; www-dse.doc.ic.ac.uk/~mml/

そうした複数のモデルあるいはビューの解釈として、ひとつの現実世界を異なる視点あるいは枠組みでとらえたものと考えるか、または、ネルソン・グッドマンのように複数の世界のバージョンが同時に存在すると考えるかは哲学上の未解決の課題である。わたし個人は、グッドマンの考えにほぼ同調したいと感じているが、そのことをいまは論じないことにしよう。

問題は、複数の世界(そのモデルあるいはビューあるいはバージョン)がどのようにして創られるのか、である。カッシーラーは、それらが「無から」創りだされるといったが、それは単なることばのあやであって、事実は異なるであろう。グッドマンが指摘したように、新しい世界のバージョンは、すでに存在しわれわれに与えられた古いバージョンを少しあるいは大幅に手直しするかたちで創られるのであって、決して無からではない。

何らかの記号(ことば)を用いて書かれたモデルあるいはビューを媒介する以外に、われわれは世界にアクセスする手段を持っていない。はじめにまずわれわれに与えられるものは、ひとつあるいは複数のモデルによって記述された世界(群)なのであり、それをいくつかの視点から眺め、分析的に考察することによって、われわれは新しい「世界」を創りだすのである。その意味で、「設計」とは既存の世界モデルを利用して、その上に構築される「再設計」(設計のやりなおし)なのである。少なくとも設計の出発点である概念設計には、そうした色合いが濃い。

2. 設計のプロセス

ネルソン・グッドマンは、多くの論議を呼んだその著書のなかで、かれのいう世界(あるいは世界のバージョン)を創りだすプロセスを構成する活動を次の5つのカテゴリに分類している:

- (1) Composition and Decomposition (構成と分解)
- (2) Weighting (重みづけ)
- (3) Ordering (順序づけ)
- (4) Deletion and Supplementation (削除と補充)
- (5) Deformation (変形)

何年か前、初めてこの本を読んだとき、わたしは、この分類は単に「世界制作」あるいは「設計」のプロセスを構成する諸活動をただそれぞれの性質に応じて種類わけしただけのものにすぎないと考えていた。しかし、その後、グッドマンが提起した問題をテーマに開催されたシンポジウムの記録 "Starmaking" (MIT Press, 1996) に再録された文章 "Words, Works, Worlds" を読み直してみても、その解釈は誤りだと感じた。この分類は、単なる設計活動のカテゴリ分けではなく、設計プロセスを構成する5段階のサブプロセスの論理的な順序（それはあくまで論理的な順序であって現実の設計が行われる時間的順序ではない）を暗示していると考えたほうがよい。グッドマン自身の文章は、絵画そのほかの芸術分野における世界制作の場合を数多く例示していて、ひとことも「設計」については言及していないが、わたしにとって、それらの例示は示唆に富むメタファのように感じられた。

2.1. 構成と分解

あらゆる設計は、無からの創造ではなく、何らかの既存の設計をベースにした再設計である。ひとつの新しいソフトウェアの設計を始めようとするとき、われわれに与えられるものは何かといえば、対象とするシステム（それはあるアプリケーションかも知れないし、ある機能を意図したパッケージかも知れない）のイメージなのであるが、それはきわめて曖昧で多義的な「仕様」と呼ばれる仮想的なモデルのかたちで提示されるのがふつうである。そうした仕様は、当然ながら記号（コトバ）を用いて記述されており、その背景には、おそらく、何らかの設計意図が隠されている。

したがって、「概念設計」と呼ばれるわれわれの仕事は、まず、この「仕様」という名の曖昧で暗示的な「設計（もどき）」を注意深く吟味し、それがどのような構成要素からなっているか、それらがどのようなアーキテクチャを構成しているかを分析することから始まる。

また、ほとんどの場合、そうした「仕様」は、既存の類似ソフトウェアを土台として書かれていることが多い。したがって、われわれの概念設計の作業は、それらの類似ソフトウェアの設計を再吟味し、分解することから始まることになるであろう。

この分解と「再」構成こそが、概念設計のスターティング・ポイントだと考えられる。

ソフトウェア開発方法論の短い歴史を振り返ってみると、たとえば、いわゆる構造化プログラミングの方法論は、既存のプログラムの機能的構成要素に着目して、それらを、連続・分岐・反復という基本構造に分解することから出発し、最終的には、ソフトウェアの機能(計算上の意味論)とは独立した構造パターンを見出すことに成功をおさめた。また、オブジェクト指向技法は、シミュレーションのためのプログラミング言語から出発して、ソフトウェアが対象とする世界を、データとプロセスの両側面を兼ね備えたオブジェクトの集まりとしてモデル化するというアプローチで、いくつかの分野でかなりの成功をおさめたが、そのひとつの理由は、対象世界の分解と再構成という概念設計の自然な流れにそのアプローチが自然なかたちでマッチしていたからである。

ただし、マイケル・ジャクソンが「ソフトウェア博物誌」で指摘しているように、構造化プログラミングの後継者として登場した SASD 技法やオブジェクト指向のアプローチは、もともと、コンピュータの内部におけるプログラム実行プロセスの意味論的側面を体系的に整理する手段としてのプログラミング技法に由来しているがゆえに、現実社会のなかのアプリケーション・プロセスを対象とする一般的なシステムの分析や設計に応用しようとした場合には、ある種の限界がある。

2.2. 重みづけ

概念設計の第2の論理的ステップは、前の段階で明らかにされた対象システムの構成要素に対して、それぞれ適当な重みづけを行うことである。

この「重みづけ」すなわち対象システムの構成要素のうちで何が重要であり何が重要でないかを評価することは、概念設計においてもっとも中心的な仕事である。それは、設計者が対象システムをどのようにとらえているかを示すからである。

世のさまざまな開発方法論のちがいは、このステップにおいてあらわれてくる。プロセスあるいは機能中心のアプローチ、データ中心のアプローチ、あるいはオブジェクト指向アプローチ、etc, etc. それらの各方法論のあいだには別に優劣があるわけではない。ただ単に概念設計の「重みづけ」ステップにおける考え方のちがいが（対象システムのどのような側面を重視して世界をとらえるか）をあらわしているだけである。

グッドマンはそのあたりのことを次のように述べている：

ひとつの世界（注：かれの用語「世界」は「設計」と読み替えることができよう）における重要なものたち（Relevant Kinds）が他の世界には見あたらない場合に、われわれは、ふたつの世界はおなじ種類のものから成り立っていて、しかし、それらが重要か重要でないかというそれぞれの評価にしたがって、異なるかたちに整理されているだけなのだと考えたほうがよいだろう。ひとつの世界において重要視されているものたちは、もうひとつの世界に存在しないというわけではなく、ただあまり重要でないものとして隠されているだけなのだ。ふたつの世界の相違は、構成要素の違いというよりは、それらに対する重みづけの違いであることが多い、そしてそうしたアクセントの違いはかなり本質的なのである。ある単語を発音する場合に、すべてのシラブルにアクセントを置くことは、どのシラブルにもアクセントを置かないのと同じことになってしまう。

ここでいわれている「世界」の違いは、ひとつの現実を眺める視点あるいは Frame of Reference の差ではないかというのが常識的な考え方であろうが、非実在論者のグッドマンは、そうではなくふたつの現実世界が同時並列的に存在するのだと主張する。哲学に疎いわたしは、まだかれの主張を全面的に納得し受け入れているわけではないが、.....

2.3. 順序づけ

設計の成果物は第三者（顧客 etc.）に見せて評価を受けなければならない。概念設計の第3のステップは、そうしたプレゼンテーションの大まかな計画を立てることである。

設計が、設計者自身の自己満足のために創られるのではない以上、このプレゼンテーション計画は、その設計を具体化させる上で、きわめて重要だと考えられる。

「あらゆる言説は、それが書かれた瞬間から、筆者の意図とテキストの意味とは合致することをやめる」と、ある哲学者が述べている（注：子安宣邦「事件としての祖徠学」<ちくま学芸文庫>の冒頭に引用されているP.リクルールのことば）。設計のプレゼンテーションにおいてもっとも注意をはらうべきは、設計者の意図とそれを「読む」第三者の理解とのあいだの乖離をなるべく小さくすることであり、そのためには設計を構成する要素の「順序づけ」を慎重に行う必要がある。

ソフトウェア設計において用いられるさまざまな図表（流れ図、木構造図、etc）の主な機能は、この順序づけを支援することなのである。

「およそ、言に類あり、世あり、人ある」というのは、富永仲基[*2]の鋭い指摘であるが、言語表現のスタイルが「世および人」すなわち時代や環境によって変化することは、たとえば現在のUMLブームの状況をみても明らかである。しかし、より重要なのは「類」すなわち言語それ自身が持っている変化への要因であろう。

ソフトウェア設計のさまざまな表現図式の特質を、富永が仏教の経典や儒教の四書五経に対して行なったように、かれのいう「五類」すなわち「張・偏・泛・磯・反」のカテゴリ分けに照らして分析してみたら、きわめておもしろい結果が得られるものと想像される。

[*2] とみなが・なかもと。18世紀初めに浪速の商人たちが作った私塾「懐徳堂」が生んだ天才的思想家。32才で夭折したが、荻生徂徠の言語哲学的方法論を批判的に発展させ、宗教思想史を題材としてかれが発案したユニークな分析アプローチは、周囲に大きな影響を与えた。主著は「出定後語」および「翁の文」（岩波書店刊「日本思想体系」第43巻。現代語訳は中央公論社刊「日本の名著」第18巻）。江戸思想史に富永が占める位置の重要性についての解説としては、宮川康子「富永仲基と懐徳堂－思想史の前哨」（ベリかん社）がすぐれている。

2.4. 削除と補充

概念設計におけるこの論理的ステップは、やがてくる具体化（Implementation）に向けての準備を行なうものである。前のステップで選択された設計の諸要素のうちで、不要なものを削除し、また新たに必要要素を追加することが、ここでの主な活動になる。

もともとあいまいで不確定な要求仕様から出発する概念設計の作業においては、このステップにおける活動

は、そうした仕様の曖昧さあるいは不確定性を少なくする働きを持つ。

プロトタイピングは、そのために用いられる有効な技法のひとつであろう。

目的とするシステムについてユーザのイメージがはっきり固まっていない場合には、仕様は「あれもこれも」と過大に膨らむ可能性が高い。したがって、新しい要素の「補充」はなるべく控えめに、不要と思われる要素の「削除」は大胆に行うことが望ましいだろう。

2.5. 変形

概念設計の論理的最終段階は、設計の基本的なアイデアをブラッシュアップし、それを第三者に性格に伝達するために、それぞれの要素に「変形」を加えて、プレゼンテーションを仕上げることである。

そのさいに注意すべきは、先にあげたリクルの指摘にあるように、どのような表現も作者の手から離れた瞬間に、とんでもない誤解を招く可能性があるということであろう。つまり、そうした誤解もまた理解の一種でしかないという居直りの精神が設計者の側には必要とされるのである。

きわめて逆説的なニュアンスを含むこの「変形」作業の例を文学の分野で拾えば、和歌や詩などでよく見られる「本歌取り」の作品群があげられよう。美術の分野における極限的なサンプルは、ポップ・ラウシェンバーグが著名画家のデッサンに対して1個の消しゴムを用いて行った作業、大きな話題を呼んだ「消されたデクーニング」がある。ソフトウェア分野において世間的な成功を収めた例は、Unixの変形としてのLinuxだろうか？

3. 創造的設計とは？(結びに代えて)

今回のワークショップの1週間前に、大学時代の友人たちと毎年開いている絵のグループ展があった。ここ数年、エッチング・プレスで作った版画もどきを原材料とし、コラージュその他の手法による加工を加えて、アクリル絵具およびメディウムで仕上げるというプロセスで作品を創っているのだが、昨年はSARSの影響で秋のISFST会議が中止になり、例年夏に中国で開かれているPCミーティングがなくなって、かなり時間の余裕ができ、実際に手を動かし始める前に「どんな手順でどのような作品を創るか」についてのいわば「概念設計」をする時間がとれた。

そのことが、ワークショップを前にグッドマンの文章にもう一度目を通してみようという気持ちを起こさせたのである。

絵のグループのメンバーで建築学者の島田良一が、その著書「かたちに見る造形の構成」(鹿島出版会)の巻頭エッセイで提示している「かたち先行仮説」が、ソフトウェア・システムの概念設計を考えるさいに気にかかっている。

島田の指摘は、シドニー湾に建つ有名なオペラハウスのデザインは、「オペラハウス」というものの機能から生まれたものではなく、父親の仕事の関係で造船所で幼年時代をすごした設計者の「かたち」の心象風景から浮かび上がってきたものだということであった。

ソフトウェア・システムの概念設計において、そうした「かたち」のイメージというものの働く余地はあるのだろうか。もし、あるとしたら、その「かたち」とは何か？

シドニー・オペラハウスの例は、「設計」というより「意匠」の問題ではないかというコメントを何人かの方からもらったのだが、そうしたコメントに、わたしは少し違和感を感じている。

ネット上の和英辞典で「意匠」を引くと"design"という単語が出てくる。しかし"design"を英和でひくと「設計」とか「図案」とかは出てくるが「意匠」は出てこない。「辞書とはコトバの定義をあたえるものではなく、コトバが人びとによってどう使われているかを記録しただけのものである」ということに照らしていえば、この現象は、「設計」についての考え方が、ものを作る人の視点(ソフトウェアの場合にはプログラマの視点)にかたよっていて、具体的なもの作り以前の「どんなものを作るか」それを「どんなふうに見えるようにするか」という視点がなおざりにされているのではないかということを示しているかのように思われる。

ソフトウェア以外の設計、たとえば建築設計や家具の設計においては、「どうやって作るか」以前に、「どんなものを作るか」すなわち意匠の設計がきわめて大切だと考えられている。おそらく、わたしが違和感を感じた

原因は、プログラマになる以前に、抽象絵画を描くための方法論をパウル・クレエの造形講義[*3]で勉強したせいであろう。

[*3] Paul Klee, "Das bildnerische Denken": 美術史を専攻した友人の卒業研究の手伝いで、500ページもあるこの分厚いドイツ語の書物の私的日本語訳を作る作業には、ほぼ1年半ほどの時間がかかった。邦訳（「造形思考」新潮社）がでたのは10年以上も後の話である。どんな外国語も辞書を頼りに忍耐強く100ページも読み進めば一応はマスターできるというのが、このボランティア作業のもうひとつの収穫だった。その成果の応用の1つとして始めた英文翻訳のアルバイトでプログラミング入門書を訳したが、わたしがプログラマになるきっかけだった。

60年代の終りから70年代の初めにかけて、わたしがまとめあげた構造化プログラム設計のアイデアは、他人の書いたさまざまなプログラムを読んで、「どうしてみんなはこのような形にプログラムを書くのだろうか」という心に浮かんだ疑問を、抽象絵画の画面構成法と比較しながら思案した結果であった。Dijkstra や Jackson あるいは Warnier などの人たちの構造化プログラミングもやはり同じように、プログラムの「意匠」に着目して考え出されたものだと思う。

構造化とは機能的モジュール化のことだという解釈が一般に広く行き渡っているようだが、それは誤解である。構造化プログラミングが目指していたのは、プログラムが対象とする処理の意味論的機能のモジュール化ではなく、それとは無関係な意匠の構造化であった。そのあたりが、SASD や OO の方法論と構造化プログラミングの思想とを区別する本質的なポイントなのである。

シドニー・オペラハウスの設計は、「建物をどうやって作るか」とか「この建物の機能は」とかいった視点ではなく、「どんな建物を作るか」という発想から生まれたものだと思う。「概念設計」(Conceptual Design) というコトバからわたしが連想しているのは、ソフトウェアの設計においてそうした「意匠」の設計とはいったい何だろうかということである。

島田のエッセイの冒頭には、かれの主張する「かたち先行仮説」について、次のような文章がある：

— わかりやすい言葉でいえば「かたちを先に考える」あるいは「かたちを独立して考える」ということである。もっと身構えていいおせば、かたちだけを独立して教育したり、研究したりできるということである。また、この仮説は、もう少し欲張った内容を含んでいて、建築設計や芸術制作の過程においても、機能とかテーマとかよりも、かたちが先行して選ばれるのではないかということも含んでいる。機能の分析や構造の選択の結果として、かたちが決まるのではない。かたちが選ばれてそれから構造や機能との調和が図られる方が、おもしろいものができるということを考えているのである。

建築や絵画のように「かたち」があるものとは違って、もともと「かたち」がないソフトウェアに、そんな仮説を適用するのは無理な話だという声が聞えてきそうだが、書かれたプログラムの「かたち」に興味を抱いてプログラマ生活をスタートしたわたしには、建築家であるこの友人の主張が、きわめて暗示的に響くのである。

新刊紹介
玉井哲雄, ソフトウェア工学の基礎
岩波書店, 2004

山崎利治

待望の玉井先生の著書「ソフトウェア工学の基礎」が上梓されました。出版社が岩波書店なのでちょっと立ち寄った書店では目に付かないかも知れません。そこで紹介したいと思いました。ソフトウェア技術者はもちろんのこと、ソフトウェア開発やソフトウェアを利用する部門の管理者、またソフトウェアを扱う企業家の必読書です。むしろ、今世紀を生きるひとすべてに読んでいただきたいと思いますがそのわけはあとで書きます。特筆すべきは著者が企業においてソフトウェア開発の実務を経験され、また、ソフトウェア工学の発展に尽力され、さらに、大学と企業の両方でその面の教育を行ってこられたことです。その著者による教科書ですから紹介にも自然ちからが入ります。

さて表題ですが、「ソフトウェア工学の基礎」というのは計算科学のことかと考えるひとはへそ曲がりです。「ソフトウェア工学通論」で、その本格的な教科書です。その特徴は

- 40年近く経過したソフトウェア工学の成果のすべてを教育者と実務家の視点から取捨選択している
- 解説が平明で、自明ではない技術的内容が直ちに理解できる
- 豊富な内容を短い頁(270頁)に収めている
- 原理的、系統的な解説が読者の理解を深め、新発見を促す
- 誤解しやすい箇所を周到に注意している

などです。いちいち例示はできませんが、ぜひ実際に読んで確かめてください。

内容を具体的に紹介するべきでしょう。大きく3部に分けていいと思います。第1は序論、あるいは、超工学で、つまり、対象、目的、計画などが明かにされます。第1章から第3章までの3章46ページがそれぞれです。第2は本論で、つまり、多分著者がこれだけは理解し、実用してほしいと考えられた(と思える)主題「モデル化」です。第4章から第10章までの125頁です。第3は特論(本論だと叱られるかも知れませんが)で、残り3章87頁がそうです。ここでは設計技法、検証技法、ソフトウェアの保守・発展が解説されます。

序論

第1章は「ソフトウェアとソフトウェア工学」です。まず用語ソフトウェアを説明します。計算機の金物を指す用語「ハードウェア」に対比して使われた用語「ソフトウェア」は今日ではさらに意味を広げてあらゆる場で使われ文脈を無視してはまったく意味不明な用語になっています。著者による用語の説明はじつに丁寧で、プログラムとソフトウェアの用法の差や英語では software は抽象名詞で softwares とはならないことなども注意しています。用語を出典を明にしたがらの解説はこの著者ならではです。そういえば、まず、「まえがき」が衝撃的でした。まえがきの冒頭で著者はディケンズが彼の週刊誌「家庭の言葉」に書いた一文を引用します。そこに soft-ware と hard-ware とを見つけるのです。つぎは「ソフトウェア工学」です。

「本書の目的は、このようなソフトウェアという対象のもつ独自の性質に注目しながら、それを作るプロセスや手法を探求することにある。とくにソフトウェアが人工的な構造物である点を重視し、これを構築するためのエンジニアリングとしてソフトウェアを捉える立場をソフトウェア工学という。」
(3頁)

そしてこの工学の誕生からその発展と現状を概観します。また他の既存工学と対比しソフトウェア工学の「成熟度」にも触れます。

第2章は「ソフトウェアプロセス」です。ソフトウェア工学の主題はソフトウェア作成でした。現実のソフトウェア作成においては目的コード以外にも計画書、費用見積書、仕様書、設計書、利用手引書など多くの文書を副次的に、あるいは、主要な目標としてつくります。つくったも

のをプロダクトといい、つくる過程をプロセスといいます。このプロダクトとプロセスについて、いわば、ソフトウェア工学の管理的側面を議論します。ここに現れる鍵語には、ライフサイクル、極端プログラミング、CMM、プロセスプログラミング、ISO9000、SPICE があります。ソフトウェア工学研究の流行について述べられたこの章の最後の文は印象に残りました。

「... プロセスに関心が集まる時期とプロダクトに関心が集まる時期とが交互にくる大きなサイクルがある ...」(30 頁)

第3章は「開発計画と要求分析」です。ソフトウェア開発の開始は「なにを」「なぜ」つくるのかを明らかにすることです。そこで要求を分析し開発計画を策定することになります。開発計画書に記すべき内容に触れ、その一項目である費用見積もりに関連して、機能点法を紹介します。その核心である開発工数 E とコード行数 S との関係 $E = aS^k$ について k が 1.05 から 1.2 の範囲にある (B. Boehm) などが述べられます。また、工数 E と人数 P と期間 T の間に一見成立しそうな関係 $E = PT$ は P が一定以上に大きくなると成立しないという Brooks 則についても注意します。

要求分析一般に関連して最近の要求工学を紹介し、分析の考え方である要求抽出型、目標指向型、領域モデル型などを解説します。さらに、要求には機能要求と性能などの非機能要求があることにも注意します。

この章ではまた年配者には懐かしい名前、PSL/PSA, SREM, SADT が発見できます。

この章の最重要概念は仕様です。仕様は要求の厳密な記述で、この意味で要求と仕様は同じものですが、詳細は本書を見ていただかねばなりません。要求、仕様、環境、システム、プログラムなどの間の関係についての M. Jackson による指摘を著者は式によって表現しています。示唆に富むそれはつぎです。

仕様、環境 ⊃ 要求
プログラム、機械 ⊃ 仕様

これは仕様以下を含む形式体系をつくる時、第1の式は仕様と環境から要求が証明できることを示し、第2の式はプログラムと機械から仕様が証明できることを主張するものです。なお、ここでは、システムを取り巻く環境があり、要求は環境の現象についての陳述、プログラムはシステムの現象についての陳述、仕様はシステムと環境の境界にあって両者に共有される現象の陳述であると考えています。

本論

さてこれからが本論です。要求分析を行って**要求モデル**の作成を考えます。

第4章は「**モデル化技法と UML**」です。用語**モデル**はソフトウェア技術者にとってオブジェクト指向分析設計や UML 以来なじみの用語です。それは実世界の課題を抽象しなんらかの記述体系で表したものです。著者はまずモデルの用法についての M. Jackson の指摘を注意します。すなわち、このモデルは論理学でいうモデルとは違い、その逆であると。つまり、論理学では理論対モデルの関係はクラス対その具体例であるオブジェクトの関係です。ところがソフトウェア開発ではこの関係がモデル対現実の対象になっているわけです。

理論：モデル = クラス：オブジェクト
= モデル：実対象

本論で考察するモデルはモデル対実対象の意味のそれですが、あとで解説されるモデル検査のモデルは論理学でいうモデルのことです。

以下で議論するモデルは皆なんらかの図式表現を伴っており、その図はみなグラフと見做せます。実体関連図、クラス図、意味ネットワークなどの静的モデルはグラフと見做せ、また、状態遷移、制御の流れなどの動的モデルも同様です。そこでグラフについて解説します。グラフとみる利点はつぎのようです。

- グラフは直観的である
- 対象世界の「もの」とその関係を自然に頂点と辺に対応付けできる
- 上の関係がもつ推移関係は辺がつくる経路に見出せる

グラフ構造は各種の有用な構造に潜在していて、よく研究されているのがなにより強みですが、反面グラフの直観的な解りやすさが過誤の種になるとも注意しています。

UMLに現れる諸図はグラフです。そこで UML の登場です。著者は、まず、UML に対する批判を述べた上で、

「... UML をこれまでの分析や設計のノウハウを図を中心に集大成したものと捉える。その立場からすれば、UML の各種の図は取捨選択して適宜用いればよく、また 1 つの図式に関

しても、UMLで定義されているすべての記法を用いる必要もない」(59頁)

として、UMLがもつ繁文縟礼を一蹴、UML図を素直に利用します。それがUMLについての凡百の入門書、解説書、たとえば、アディソンウェズレイのオブジェクト技術シリーズを10冊読むのに比べ、第4章から第9章の94頁を読むほうがはるかにUMLによる「モデル作成」の早解りになるという結果をもたらしています。それも実際に読んで確かめていただかねばなりません。

そこでここで二つの例題—倉庫管理と自動販売機—を説明します。以降の各章でこの例題を利用して解説が進行します。

第5章は「データの流れモデル」です。モデルは世界をどう見たかを表しています。このモデルでは世界に分岐したり合流したりするデータが流れていると見ます。グラフの頂点はプロセス、辺はデータの流れ、部分グラフは下位プロセスが対応します。ここでその見方の一例としてDeMarcoによる構造化分析が紹介されます。驚くべきはその紹介が僅か2頁で終わっていることです。そしてその2頁で構造化分析が完全に理解できるようになっているのです。例題を解き、作成したモデルが意図どおりであるか、プロセスの分解が正しいかの検証方法までも述べています。構造化分析は構造化設計の前段階としての役割を担うものですが、事務システムの仕様を如何に考えどのように書けばいいかという点で大いに啓発された思い出があります。

第6章は「制御の流れモデル」です。世界をまとまった処理が順次実行されるどころと見ます。このモデルではグラフの頂点に処理や判断が、辺に制御の流れが、部分グラフに下部手続きが対応します。その代表例はフローチャートや活動図です。

第7章は「協調モデル」です。世界は相互に交信しながら動いている一群の動作主体によって構成されていると見ます。交信をまた事象とか動作ともいいます。このモデルではグラフの頂点は交信の時点や主体、辺は交信や主体間の結合、部分グラフは下部主体の協調構造が対応します。系列図、協調図が代表例で、それらがじつに解りやすく解説されています。HoareのCSPやMilnerのCCSなど、プロセス代数もこの例ですが、著者はM. Jacksonのシステム開発法JSDをとりあげて短い完璧な解説を行います。DeMarcoやJacksonは商売柄、原理的かつ系統的な説明を避けて直喩や隠喩また多くの例示さらには笑話を交え話術で判らせようとします。そのお喋りを楽しむのもいいですが、著者の手になる直裁的

な早解りは自分の理解を確かめるために絶好でありがたいものです。当然、例題の JSD による解も与えられています。むかし、雑誌「情報処理」に掲載された同様の解に対しても、厳しくかつ優しいそしてなによりも正当な批判を加えています。そこに方法の正しい普及のために努力される著者の情熱を感じ、頭が下がります。

第8章は「状態遷移モデル」です。このモデルにおいて、世界はひとつの機械です。機械は内部状態をもち、環境と交信（事象）しつつ状態を変化させます。グラフとしては古来より馴染みの状態遷移図そのもので、頂点は状態に、辺は遷移に、部分グラフは下部状態図に対応します。伝統的な遷移図はさまざまに拡張でき、著者は UML でも採用している D. Harel の Statecharts を解説します。全貌の把握が決して易しくはない Statecharts ですが、その解説は10頁で易々となされています。現存の UML の解説書からは得られない貴重な早分かりです。

第9章は「オブジェクト指向モデル」です。すでに耳に馴染みのオブジェクト指向ですが、ここは虚心に耳を傾けなければなりません。まずその歴史を振り返ります。ここではシミュレーション言語、抽象データ型、知識表現、意味データモデル、動的プロセスモデル、マルチメディアそれぞれからの由来が説かれます。このモデルは UML のクラス図の基になります。したがって、この章の主題の一つはクラス図の解説になります。グラフとしては頂点はクラス、辺は汎化、集約、関連、依存を表します。

ユースケースを説明し、例題のクラス図を書き、ついでいままでに見てきた系列図、状態図などを与えます。

本論の最終章である第10章は「形式手法」です。これまでのモデルは図を用いていましたが、精確なモデル（仕様）を抽象的・形式的に記述しようという考えもあります。この考えを「形式手法」といいます。UML 図を補って完全な仕様記述にしようという Object Constraint Language (OCL) なども UML の一部になっています。OCL は Syntropy + Z から誕生したといわれています。ここでは、まず、この形式手法を概観し、その一つである Z を解説します。Z は型付集合論を基礎に、名前をつけた対象とそれが満たすべき公理を与えた図式を単位とし、それらの上の演算によって複雑な仕様が段階的に書けるように工夫した言語です。とくに欧州で実績のある形式手法として著名です。この解説の手際の鮮やかさと教育的配慮に感銘を受けます。鮮やかさは取捨選択の妙以外にも、なにを、いつ、どこで、どう説明するか、によって決まります。配慮は初心者がどこで間違えるかなどがよく解っていてはじめて可能です。敬服

するのみです。

特論

話題は3つ、設計、テストと検証、保守・発展です。それぞれに本論が必要なソフトウェア工学上の重要課題ですが、その概略が解説されます。

第11章は「設計技法」です。話題の一つはアーキテクチャであり、もう一つは算法の設計です。用語アーキテクチャほどソフトウェア分野において使われるとき困惑するものはありませんでした。単に大域プログラミングを指すもの、コンポーネントの利用を指すものなどさまざまでしたが、Shaw + Garlan の仕事以来その意味もはっきりしてきました。アーキテクチャ記述言語なども雨後の筍のように輩出しています。著者はアーキテクチャの例—パイプ-フィルター、事象駆動、黑板システム、層別、ルールベースなど—を明快に解説されます。

算法の設計はデータ構造と算法のことで、ここに書かれた解説は、一見、局所プログラミングのそれのようですが、仕様の正しい実現を図るという意味で疎かにできません。いわゆる洗練計算による実行可能なプログラムの導出についての解説になっています。そうはいってもプログラム作成の実際はじつに難しいもので、2分探索プログラムを講義していて正しいプログラムがこのようにしてできると説明したスライドに虫がいたという著者ご自身の挿話がそれを示しています。洗練計算によるプログラミングは日常言語に対する学校文法に相当するといえますが、同時に、見えていない問題の構造を見抜く眼力が備わっていなければならないことを強く語っています。

第12章は「検証技術」です。ここで著者は仕様の検証とプログラムの検証とに分けて議論を進めます。プログラムについて実行結果に基づいて正否を判定するテストとプログラムが仕様の正しい実現になっていることを厳密に証明する「検証」を概観します。

仕様の検証としては、見直し、査問、徒歩検査のほか、近年流行のモデル検査が解説されます。モデル検査は、原理的には、有限な構造 S と命題時間論理式 p を与えたとき S は p のモデルになるかを問うことです。この判定問題は決定可能で、つまり、計算機プログラムによって自動的に判ります。著者と中島先生による Enterprise JavaBeans のアーキテクチャの妥当性をモデル検査系 SPIN によって分析し、いくつかの不備を発見した事例によってモデル検査の原理と実際を簡潔に説明しています。

これだけでモデル検査の実践は難しいかも知れませんが、SPINの利用についてのいい手引きも出版されていますから開発現場での普及が期待されます。

プログラムのテストについてはここでもきわめて要領のいい包括的な解説がなされ、極端プログラミングにも触られています。なお、参考文献には出ていないようですが、テスト技術については著者たちの著書がありました。内容は覚えていませんがヴァレリーやジョイスからの引用が鮮烈だったことを記憶しています。

プログラムの検証について著者は大きな業績（事務計算問題）がありますが、検証一般についての解説はわずかです。この主題は荒木・張両先生の著書（「プログラム仕様記述論」オーム社）によって学べます。

最終章の第13章は「ソフトウェアの保守・発展」です。私はこの章から大変多くを学びました。ソフトウェアが使われ始めるにつれて使われる環境の変化や利用者の要望によってソフトウェアがどんどん変わって行きます。それをソフトウェアの発展と捉えます。保守と聞くとなにか不具合の修正のようで暗い印象をもっていますが、発展というと確かにもっと明るい積極的な印象を受けます。第1節は「発展」の特徴を述べます。第2節は発展のモデルについて述べます。そこではOS/360の発展についてのBelady + Lehmanによる観察結果や、システムの発展に対応してそこに含まれるオブジェクトの変化を観察した結果などの興味深い統計的知見が得られます。また、S. Blackmoreのミームの紹介など興味は尽きません。第3節は発展と保守です。ここで保守の体制、技術とツール、保守の戦略（いわゆる保守を続けるか、作り直すかなど）などが解説されます。技術問題として、余波分析、回帰テスト、プログラムスライシングが挙げられ、さらに逆構築が説明されています。第4節は再利用です。部品化、コンポーネント技術、再構築が議論されます。再構築ツールのRefineが紹介されています。これは元来トップダウンの仕様記述とプログラム開発の自動化が目的であったが、ボトムアップの再構築分野での実用的なツールとして利用されたといいます。具体的にどんなものかは本を読んでいただかねばなりません。しばしば耳にしてもその具体的な意味が私には解らなかつた多くの用語—部品化、コンポーネント技術、逆構築、再構築—がこの章でよく解りました。

お薦めの理由

内容は紹介の通りソフトウェア工学の本格的な教科書ですが、専門学徒や実務家以外にも読んでほしいのはつぎのように思うからです。

洗濯機、電子レンジ、テレビ、DVDレコーダー、携帯電話器などの家庭用品をはじめほとんどすべての工業製品に計算機が組み込まれ、そのソフトウェアが本質的な役割を演じています。そしてこれらの機器の使い方が難しくなっています。それらの使用解説書が割く頁数は大変なものです。さらにはテレビで携帯電話の使い方講座が放送されたりする始末です。組織的な説明を避けるために却ってわかりにくくなっています。ほんの少々のソフトウェアやその仕様についての知識を前提とすれば利用手引書が劇的に改善されるのではないのでしょうか？

失業保険や健康保険、年金また銀行などの公共システムに不備があれば、世の安寧秩序に深刻な影響を及ぼします。さらに、医療機器、自動車、航空機、船舶、発電所、ロボットなどは直接人の生命を危険に曝す可能性があります。この種の事故が起こった場合、その原因究明のためには、システムの仕様、実現、利用中の諸問題（データ、金物の磨耗や損失、利用前の準備、具体的な操作、システム全体に係わる保全システム）を峻別して、それぞれを明確に論議しなければなりません。その実施のためには専門分野の専門家であると同時にソフトウェアが議論できなければなりません。つまり、このような議論が健全であるためにはそれぞれの専門家間に共通する明晰な言語が必要です。このような言語はそれが必要な場でいまだに共通の認識とはなっていませんが、ソフトウェア開発で考えた仕様記述言語はそのもっとも近い言語のひとつです。というのは論理学であつかうようなそれ自身が考察の対象である裸の論理言語ではなく、実務のためのつくられた実績のある言語であるからです。

前世紀に生まれたソフトウェアは人類にとってもっとも知的で便利なものであると同時に、極めて危険でかついたるところに存在しているものです。そうであるにも拘わらず、質料を伴わないために見え難く、したがって、知覚し難いものです。新世紀の一般市民として、あるいは、ある分野の専門家としてもっとも大切な「読み書き算盤」は、マイクロソフトが販売するソフトウェアの利用能力ではなく、上の意味での「共通言語」を理解する力であると思います。その力を身につける第一歩としてこの「ソフトウェア工学の基礎」を皆様に読んでいただきたいと思います。

季節外れの札幌便り

熊谷 章

(TELST)

12月3日 2003年 札幌万年筆談義

愛用の Cross ボールペンのインキが切れた。最初に大丸8階に買いに行ったがなかったので、隣の「たづむら」で豚カツを喰った。自分でゴマを摺る、おひつに入った麦飯、キャベツと味噌汁お替り自由、プーアル茶のサービスなど気に入った。食べたのは当然ロースカツ。仕方がないので、調べて丸善を行ってみた。三越の隣で3階にあった。Cross ボールペンがあったので流石文房具の丸善だと安心した。「Cross のボールペンの替え芯を二本、中太の青と黒のを下さい」。「生憎、在庫がありません。細字ならありますが、、、」。俺は中太が欲しいのだ。一分間考えた。このまま引き下がったのでは男が廢る。大丸と丸善でだめだなんて許せない。なんという街だ札幌は、と感じながら打開策を思いついた。「じゃ、代わりに万年筆を下さい」、と俺は言った。売り娘は一瞬戸惑った。「万年筆ですか?」。Waterman、Pilot、Sailor、Parker、Mont Blanc などあった。その中から Pilot を選んだ。使ったことがないからだ。一万円と二万円のものがあった。迷わずに二万円のを二本試筆し、中太を買った。これで所期の目的が果たせて気分が晴れた。野原一杯の花を貰ったら、手紙を書けるぞと喜んだ。インキはカートリッジではなくインキ壺を買った。純正ということでインキも Pilot にした。

東京で使っている内にインキが切れた。家では Mont Blanc しか使わないから、Pilot の万年筆に Mont Blanc の Blue Black を入れた。インキは Blue Black しか使わないことにしている。札幌に戻って万年筆を使っていたらインキが切れた。札幌には、Pilot の Blue Black しかない。しかも、インキ壺は家に置いているので会社にはない。仕方ないので、昼食後 Pilot の Blue Black を買いに行った。大丸にあるかも知れないが、前の嫌な思い出があるから東急に行った。8階に申し訳程度の文具売り場があった。

Pilot 製品を置いていたので、Pilot の Blue Black を下さい、と言った。「カートリッジですか」、「いいえ、インキ壺です」、「申し訳ありませんが、Pilot の Blue Black はありません」、「何だとあるのですか」、「Sailor と Mont Blanc ならあり

ます」。しばし、考えた。何を買うのが良いのだろうか。俺は Mont Blanc の愛好者なのだから、Mont Blanc に決まっている。それなら、なんで Cross ボールペンの替え芯の代わりに、Pilot の万年筆を買ったのか。それで、その万年筆になぜ Mont Blanc のインキを入れて使うのか。試しに、Cross の替え芯で青の中太がありますか、と訊ねた。調べてから、生憎その芯は在庫がありません。それを聞いて俺は内心でほっとしていた。結局、Mont Blanc の Blue Black のインキ壺と Cross の黒の替え芯を買って来た。それらを机上で使っている。結構楽しい街だよ、札幌は。

1月14日 2004年 豪雪猛吹雪からの脱出

1月13日深夜、しんしんと雪が降る気配をベッドで感じていた。目覚めてそれを確かめるためにカーテンを引いたら、目の前に白濁した光景が広がった。雪が煙のように下から上に舞い上がり、雪の粒は粉雪状になっている。いよいよ本格的な冬到来かと喜んだ。実は昨夜までの3月のような気候には辟易していた。1月の札幌で雪解け水をかき分けながら歩くことは心外そのものだった。俺は犬だと思った。新雪、ドカ雪、ポタン雪、粉雪、時雨雪、吹雪など雪が降れば心身が喜びを隠せない。俺の原風景に雪があるのだろう。このときには、雪を楽しみ悠長にしていた。まさか、今夕の東京行きの飛行機が飛ばないとは夢にも思っていなかった。

自慢のフード付ダッフルコートを着て出社した。吹雪が凄いが手袋を嵌めるほどではない。1分で地下鉄に乗り10分後に札幌駅に降り立つ。いつもと変わりはない。駅の北口を出ると風が冷たい。大通りと札幌駅北口では空気が違うといつも感じていたが今日は特に風が冷たい。途中でKに出会う。雪道のせいで今日転んだという。でもそれを嬉しそうに話してくれた。窓際の俺の席から外の様子が良く分かる。20メートル先にある紅白の煙突が吹雪で見えなくなる。白濁した風景は、まるで海中にいるような錯覚を与える。空中動物から雪中動物になったような妙な気分になる。悪くはない。10階の窓から街行く人々を眺めるのは楽しい。救急車が忙しく通る。

朝の便で東京に行ったはずのMが空港で待機させられている、という話が舞い込んだ。その内、東京行きを諦めてオフィスに戻って来るといふ。ここで夕方の便も飛ばないかも知れない、とはじめて思う。猛吹雪も直に止み飛行機が飛び始めるだろうと高を括っていた。しかし、ニュースは全便飛ばないだろうという情報を流し始めた。今日東京に飛ばないと明日の午後に熊本に着けない

ことが分かっていたので、善後策を考え始めた。北斗で仙台まで行き、新幹線で東京に行く方法。これだと羽田午前9時に間に合う。しかし、肝心のJRが運行していない。次は旭川に行く案。なぜか旭川からは全便飛んでいるらしい。札幌から特急で2時間弱らしいので可能性が大だ。三番目の案は、明日の早朝の便で大阪か東京に飛ぶ方法。これが楽でよさそうだ。Tさんに明早朝の便の予約をお願いしたが満席だという。考えてみれば当然だ。今日全便欠航ならばその客は明日に変更するはずだ。残った道は旭川経由しかない。問題は、旭川までどうやって行くかだ。JR、タクシー、バスの三手段がある。Mが素早く行動しバス便が運行していることが判明。

午後三時二十分札幌発のバスで旭川に向かうことを決めた。バスの遭難に備え、ビールロング缶二本と、1リットルのOld Parrを買ってバスに乗り込んだ。乗客は俺たちを入れて10名ほど。こんな天候に札幌から旭川に移動する酔狂者がいることに驚く。荒れすさぶ札幌の冬景色を堪能しつつバスはノロノロ走った。この調子では旭川の最終便午後8時までに着かない可能性が大きい。俺たちのプロジェクトは運が良いので間に合うはずだと楽観する。四人は遠足に出たような表情で生き生きしている。会話が弾む。前の車のテールランプ以外何も見えない。バスの運ちゃんはこんな中を日常茶飯事のことのように自由自在にバスを操る。たいしたもんだ。

道央高速道路の入口が閉鎖されているので下を走る。このままでは間に合わないかもしれない。間に合わなかったら笑い者になる、いやその時は帰りをどうする、など話題が絶えない。確か、岩見沢から高速に入った。高速道路は空いていた。バスは順調に走り始めた。吹雪も段々と勢いを弱めた。バスは雪原の中を走っている。時折街の明りが闇の中で輝いている。こんな風景を望んでいたのだろう。北海道にずっと住んでもよいと思う。途中、深川で美しい夜景に出会う。旭川は大きい街でとても美しい。まるで札幌にいるようだ。午後6時半に着いた。間に合ったのだ。Eを話題にしながら、駅ビルにある旭川ラーメン「蜂屋」に入る。脂のせラーメン二杯、かけラーメン大盛一杯、ラーメン一杯、ビール二本。鯡の出しが決め手らしいと壁新聞で知る。醤油ラーメンだが味噌のような少し白っぽい色だ。NがEは旭川ラーメンはおいしくないと評価していることを教えてくれた。しかし、全員美味しいと評価した。旭川駅の佇まいは昔風で美しく雪に映えていた。タクシーで空港に行き無事東京行きに乗った。空港で旭川名物の「旭豆」、「-41度」クッキー、「旭川ラーメン」を求めた。機中達成感を身体で味わっていた。

2月22日2004年 またしても飛行機が飛ばない

2月19日二日連続飛行機が飛んでくれない。2月22日午後3時近く横浜のYCATで自動チェックインを試みたら、「現在チェックイン不可能なので、受付カウンターまでお越し下さい」といメッセージが出た。二度やってみたが同じだった。YCATのJALカウンターでどうしてチェックインできないのか、と言ったら発券してくれたがチェックインはここではできない、ので羽田のカウンターでやって欲しいと言う。感じ悪い、と思いながらバスで羽田に着いた。出発カウンターは人で溢れていた。チェックインカウンターに並び順番を待った。午後四時発の便なので、余裕がある。JALの電光掲示板は故障かなんかで点滅を繰り返しているばかり。15分待ちで、順番が来た。俺の予約便を見て地上係員がすかさず言った。「札幌便は現在様子見でチェックインを中断しています。アナウンスがあるまでお待ち下さい」。

絶句だ。今日も飛ばないのか。ここ一週間で、3回羽田発札幌行きをトライして、一回しか飛んでいないのだ。17日は、飛行機に乗り新千歳空港上空まで来て一時間以上旋回し、羽田に引き返した。悪天候と雪のために新千歳空港は閉鎖中で再開の見込みは無いようだ。即座に、チケット発行の長蛇の列に並んだ。明日の便に予約を切り替えるのだ、と咄嗟に反応したのだ。列には100名位がいた。フロアは行き場のない人々でごった返している。20分位並んでいたら、アナウンスがあった。「午後4自発札幌便を欠航します。航空券の手続きをします。27番カウンターにお並び下さい」。チケット発行のキューに並んでいる俺はどうすればよいのだ。ひょっとしたら、27番カウンターに行けば、待ち時間なしで明日の便の予約が取れるかも知れない、と考え、移動することにした。なんとこれが間違いだった。27番カウンターには、200名以上の行列が出来ていた。大失敗だ。手許にある搭乗券引換証をよく読んでみた。90日間有効と書いてある。なんだ、この券があればいつでも予約して使えるのだと理解した。

次のアナウンスで、明日以降の便の予約は予約センターに電話して下さい、と案内している。出口の人の少ない場所を見付けて、携帯電話から予約センターに掛けたのだが、何回電話しても話中である。万策尽きた。27番カウンターの200名を越える列には並ぶ気はない、かといって先程まで並んでいたチケット発行の列に再加入する気もない、レストランで一杯引っかけ天候回復を待つ手もあるが、羽田には美味しい飲み屋がない。四階の寿司、蕎麦、イタリア料理など魅力がない。午後五時半まで待ったが千歳空港閉鎖に変化がないので諦めて鎌倉に戻ることにした。

癩だから大船の市場で、本マグロ、カンパチ、牛肉を買って帰った。家族は、俺が鎌倉に帰ったのを訝しげみ食べ物だけを歓迎した。その夜に、インターネットで新千歳の発着便を調べたら、全滅であった。翌日の便は軒並み満席となっていた。翌朝、予約センターに電話した。予約できた便が午後四時527便だった。悪い予感がした。ずっと飛ばなかった便がすべて527便だったのだ。電話の向こうで、今日も空港閉鎖が続いているので出発前に運行状況を確認して下さい、という。そんなことは分かっているが、いつ運行を再開するか分からないから空港で待つしかないのだ。この日は、最初の便が飛んだだけで、後は全部欠航だ。期待を込めて羽田に行った。家族からはもう帰ってこないでと再三再四言われていた。羽田に着いたら、午後2時ころで既に俺が予約した527便は欠航が決まっていた。それ以降の便を調べたら、全部満席だ。仕方がないので、会社に連絡して飛行機の予約を頼むことにした。

しかし、VPNが旨く繋がらない。attglobalからメールを出せるTさんにメールを出し、Tさんに転送してもらう事にした。TさんとHさんから返事が来た。なんと24日の午後6時半の便しか予約できないのだそうだ。その結果にガックリして、鎌倉の家路に着いた。家に帰ったら、家族からなぜ帰ってきたのかとヤンヤンヤンヤ言われた。なんと午後四時発のJAL527便に4回トライし、3回NGだったのだ。一体、新千歳空港近くに何が起きたのだ。空港閉鎖は10年に一度くらいの安定した空港だと評判が高いのが売りなのに、この冬だけで4回も空港閉鎖が起こったのだ。24日の便は午後6時半だから飛ぶに違いない、という確信があった。午後4時発のJAL527便でさえなければよいのだという思い込みが出来上がっていた。

3月2日2004年 インキ談義再び

Architecture、Design、Implementationの違いを明確にし、そう区別することの効用を書いていたら、途中でインキが切れた。インキの色が変わるのが嫌なので、インキを買いに出た。今日は、弥生の二日目であるが、外気温度は氷点下で道路はアイス盤で覆われている。テカテカ光っていて、ある意味では美しく見える。昨夜帰り道で札幌駅近くの電光掲示板が氷点下6度を告げていた。氷点下の空気は乾いていて気持ちが良い。頬を撫でる風が心地よい。しかし、指先が寒さで痛くなる。特に、二週間前に出刃包丁で切った指先が寒さで疼くような痛さを示す。その痛さが伝わって頭に来る。その疼きが快感に変わる。癖になるといけない、と考え左手をポケットに入れる。道行く人々は皆外套を

身に着けている。俺だけが、セーター一枚で歩いている。見知らぬ街の異邦人になった気分だ。

目指すは、東急デパートだ。以前、Mont Blanc のインキを買った実績があるからだ。8階の文房具売り場に着く。「あの一、Mont Blanc の Blue Black のインキを下さい」「わかりました」、とカートリッジを探している。「あの一、インキ壺が欲しいのですが、、、」。別の引き出しを開けて探し、「生憎、Bont Blanc の Blue Black は在庫がありません。RoyalBlue ならあるのですが、、、」。「Blue Black でないと駄目です。他のメーカーだとあるのですか」。「、、、、、、、、」と返事がない。仕方がないので、「では、その Royal blue と Pilot の Blue Black を下さい」。どうして、札幌では俺の買おうとするものがないのか、ともうヤケ気味だ。万年筆を水で洗って Royal blue にして使おう。もし、気に入らなければ Pilot の Blue Black にするか、家にある Mont Blanc の Blue Black にしようと決めていた。

地下一階の六花亭で春の和菓子をプロジェクトのみんなのために買う。鶯餅、桜餅、草餅、など春の雰囲気伝えるものだ。オフィスに戻ったら、K さんがやって来てハワイのアラモアナ World Shopping Center で買って来た Mont Blanc の万年筆を見せてくれた。書く角度で筆跡の幅が変わるものだ。俺も何年前か前その店で同じものを買った記憶がある。偶然の一致だと感慨深い。そこで、K さんが衝撃の一言。「大丸に Mont Blanc の専門店があったよ。そこでインキも買えるはずだよ」。俺は、どこに目をつけていたのだ。大丸に行って、文房具がどこかを調べ、8階だと確認しそこに Mont Blanc がなかったの、大丸にはないと思い込んでいた。それなら、最初から大丸の Mont Blanc ショップに行けば何の苦労もなかったのだ。人生は、ほとんど無限空間を探索して歩いているようなものだと実感した。それにしても、六花亭の鶯餅は旨い。外気は氷点下だが、札幌の空は既に春色だ。

3月4日 2004年 朋あり遠方より来る

昨日は前の会社の旧友が札幌にやって来たので夕飯を一緒に食べた。札幌らしい店を希望されたので一押しのお店、「ヤマダモンゴル」に行った。生ジンギスカンではこの店は札幌一と評判が高い。店構えは、掘っ立て小屋風で何の構えもない。炭火、ジンギスカン鍋は焼く部分にスリットがある、羊肉は二種類しかない。熟成生ラムと極上生ラムだ。野菜は、タマネギがデフォルトで、ジャガイモ、キノコ、ピーマンがある。季節によってはアスパラガスもある。白

菜の浅漬けとキムチも旨い。

ジンギスカン鍋の頂上に羊のラードの塊を載せ、周りにタマネギを敷く。隙間にラム肉を置いて焼く。生でも食べられるそうなので火が通ったと思ったら喰い時だ。特性のタレをつけて食べる。肉が甘くて味があり旨い。韓国式焼肉も旨いが、この生ジンギスカンはいままで食べたことのない肉の旨さがある。タマネギ、ジャガイモと一緒に食べる。ラードと肉からでた脂が鍋の淵に溜まっている。それに野菜が触れると熱い脂が飛んで来て、顔や手を直撃する。その痛さと熱さが快感だ。「アチッチ」と言いながら肉をほおばむ豪快さはたまらない。生ビールを飲む。店に飾ってある、Cordin Blue の焼酎を飲んでみたいが色が妖しいのでまだ挑戦していない。

3人で7人前を平らげる。ビールをジョッキで三杯飲んだ。ここの締めくくりは、ラーメンとお茶漬けだ。この食べ方が面白い。ジンギスカン鍋を引き上げ、普通のアルマイト鍋にお湯が入ったのが炭火七輪の上に乗つけられる。別の皿に生ラーメンが盛られてくる。番茶がマホービンで持ってこられる。鍋で生ラーメンを茹で、ジンギスカンを食べた汁を番茶で割り、茹で上がったラーメンをその汁につけて食べるのだ。これがまた旨い。一人で二人前は食べられる。番茶で割ることを考え付いた人は偉い。

約二時間で食事が終わった。三人で締めて 12000 円。

その後はタクシーで我が家に行った。Bombay 帰りに買ってきた「Bombay Sapphire」を振舞うためだ。外気はマイナス 6 度前後。道路はコチコチに凍っており凍てつく寒さだ。ほろ酔いの顔にはこの寒さが気持ちよい。レモンを輪切りにし、オンザロックで Bombay Sapphire を飲む。部屋には、シングルモルトウィスキーがあるのでそれらを順番に出す。ラガプーリン、Scapa、Ardbeg、Bowmore だ。名古屋観光ホテルでみんなと飲んだことや松江で山崎さんと三人で飲んだことを思い出す。肴には、インド土産のカシュナッツ、手作りの松前漬け、北海道のスルメ、京都の山椒の味噌漬けを出す。締めくくりに、「常きげん」のひやおろしを飲む。ジン、ウィスキー、日本酒それぞれ違った味でそれぞれがとても旨く感じる。それらの差異を楽しむ。

話題は、俺が買ったダイニングテーブルだ。椅子が六脚と 1メートル×2メートルのベージュ色のテーブルを買った。メーカーは Morishige で一目見て気に入った。その隣には赤と白のソファを備えた。N さんが俺んちに来たときそのソファを見て連れ込みホテルのソファみたいだ、と言った。「なんで、そん

なことを知っているの？」と訊いたら黙った。

午前1時で我が家での宴会をお開きにした。皆が帰った後ゆっくりと風呂に入った。怪我した指をお湯の中に浸したら疼いた。その疼きが色々なことを想起させる。Calling you、北の宿から、知床旅情など口ずさんでみる。ひんやりとした札幌の夜更けにあなたを想う。

3月8日2004年 「魚御殿 MASARU」は凄い！

先週の金曜日はプロジェクトメンバーの若手と一緒に遅い夕飯を食べた。午
前零時半を廻っていたので深夜営業する店、「魚御殿」にした。薄野の外れにあ
るので氷点下の中をタクシーで行った。ニュー桂和ビル二階にあった。8階建
でのビルである。看板をみれば飲み屋とバーのアパートらしい。その二階に例
外として魚御殿と羊御殿があった。外見は「村さ来」、「つぼ八」などと同じだ。
これは外れかも知れないと話した。店に入り、メニューを見たがとにかく安い。
毛蟹イッパイが1980円だ。タラバ蟹でもイッパイ2400円。仲居さんに
接客の下見に来たのもっと高くてもいいが大きい蟹がないかと訊ねた。そした
ら店のマスターがやって来た。「蟹は大きければ旨いわけではない。当店では一
番美味しくて値段もいいところを出しているのだ。これくらいのサイズが適当
なのだ。でも、事前に要望があればもっと大きいのを仕入れてもよい。今日は、
毛蟹の2400円ほどのものがあるので食べて下さい」。とても真面目で誠意の
ある店長だ。魚の卸が本業なので良い品を安く店に出すことができる、とは Web
Home Page の宣伝文句。

取り敢えず、毛蟹、タラバ蟹、アンキモ、ボタン海老などを頼んだ。ビール、
メ張鶴、焼酎を頼んだ。パイガイのお通しが旨い。これは期待できるかもしれ
ない。蟹が来た。確かに大きくはない。早速、食べる。肉が硬くて新鮮な味が
する。タレがない。何もつけずに食べさせる。蟹に自信があるからだと思った。
タラバも旨い。タラの白子の天婦羅も美味しい。メジャーな「蟹將軍」や「蟹
本家」と比べれば、値段は半分以下だが味は10倍くらい良い。モノゴトは見
掛けだけで判断してはいけない、と思い知らされた。ここは午前5時まで営業
しているとメニューに書いてあるのだが、午前3時半に食べ物のラストオーダー
と言って来た。午前5時まで営業なのに早すぎるじゃない、と言ったら実はい
まは午前四時で閉店しているという。それから、雲丹丼とイクラ丼を食べて午
前四時過ぎに店を出た。家に着いたのは、午前四時半頃だった。

目が醒めたのは午前10時頃。洗濯と部屋の掃除をしたら昼になった。NHK BSで囲碁将棋ジャーナルを観る。その間に雑炊を作る。少し飲みすぎた翌日は雑炊が美味しい。携帯電話の留守電に東急からのメッセージがあった。先週作ったメガネが出来上がっている、というもの。午後四時、地下鉄で東急に行った。五階にある水野メガネは混んでいた。仕方がないので昼飯をと考えレストラン階に行った。東京の店が多く入っていた。最近食べていないものにしようと考え、中華料理「天府」に入った。かなり、混んでいた。若い人はいない。老齢の男女しかいない。綺麗な身なりに着飾った人が多い。「ニラソバ」と「チャーハン」、それに紹興酒を頼んだ。紹興酒の肴にザーサイも頼んだ。ウェイトレスがザーサイは300円するのですが、それでもいいのですか、と訊いた。この紹興酒は台湾産に違いない、と思いながら、山盛りの一皿300円のザーサイを食べた。残ったら、持ち帰りにしようと考えた。ニラソバとチャーハンが届いた。見たらザーサイの小皿が付いている。先ほど訊かれたときに、ザーサイは注文したものに付いていると言ってくれば頼まなかったものと思った。マーマーフーフー（馬馬虎虎）だった。

水野メガネに寄りメガネを受け取る。今度は、遠近両用を作って下さい、と薦められた。意地でも作らないと覚悟を決める。帰り道に、三越に寄って毛がに二杯とポタン海老3皿を求める。今夕、前の会社の飲み仲間だった輩が俺んちに来て酒盛りをするのだ。携帯電話が壊れかけて動作したりしなかったり安定しない。近くのDoCoMoショップを見つけて入ろうとしたが午後五時までらしくシャッターが降りていた。どうやって彼らと連絡すれば良いのだろうかとはんやりしながら、蟹とポタン海老を手にして街を歩いた。

3月9日 2004年 春の雪は風情がある

流石に北国も3月に入り寒さも緩んできた。昨日の昼には道路の雪も融けている場所があった。しかし、夜には氷点下になりフワフワした雪が舞っていた。舞い落ちる量が少なく軽やかに飛ぶ雪に儂い風情を感じる。昨夜は、午前零時半に夕食に行った。札幌駅の南口の向かいにある「ととと」という大きな釜がおいてある店だ。店内にある大きな釜で北海道産のジャガイモを蒸かしているときもある。かまどという懐かしいものが置かれている。午前2時まで営業しているのが売りだ。八海山を飲みナメタカレイ、タコの刺身、焼筍、筍ご飯、タラちり鍋などを菜とした。焼いたナメタカレイは海の味がして身が厚く絶品。東京近辺では煮て食べるのが普通なので新鮮味がある。ここの、ミズタコの刺身も甘くて美味しい。日本酒にピッタリ。筍は熊本産らしい。北国の人々が京

都料理に憧れ、沖縄や九州に多く旅する気持ちが最近は分かって来た。異質を欲するのだ。多くの有名な温泉地の、うたい文句は京都料理だ。旅行では沖縄に人気がある。

今朝は寒くはない。多分零度前後だろう。藻岩山がきれいに見え朝陽が部屋に差し込んでいる。大倉山シャンチェもよく見えた。オフィスに着くと最初に窓を開け放す。空気を入れ替える。このビルのエアコンディションは最悪で一日中ここにいると乾燥と空気の悪さで喉の調子が悪くなってしまう。だから、一時間に一回以上窓を開けて換気する。今朝の眺望はよくない。春霞のような靄が出ていて、千歳方面の山々も丘珠空港近くの山も見えない。薄日が差しているが、雲が多く典型的な春の天候かもしれない。

いま、俺のプロジェクトは佳境を迎えている。今月の末に最大のチェックポイントがあるのだ。昨年9月」から開発してきたプロダクトが製品の一部として認可されるか否かが決まるのだ。総勢30名のプロジェクトだ。まさか、こんなプロジェクトの責任者を務めるとは思っていなかった。このプロジェクトでいま面白い現象が起きている。俺が Project Leader で、その下に Group Leader 一人と、Sub-Group Leader が二人いる。若手から、3人の GL に対する反乱が起きたのだ。つまりは、俺のやり方が悪いといわれているのだろうが、この 3GL を外して欲しいという要望が出た。その理由を聞けば、物事を分かっていなくて、判断と指示がイカレテイルのだ、と主張する。確かに、あるサブシステムの設計を間違えて、半分くらいを設計し直しにならざるを得なかった事件が起きた。そのサブシステムのアーキテクトが 3GL の一人だった。そんなことを書かれても、詰まらないと思うだろう。ここで、表現したいことは、このような30名ほどの小さなプロジェクトで、なぜこのような事件が起きるかが問題なのだ。そんな他人事みたいなことを言っている Project Leader だからだ、と言われればにべもないが、なぜ彼らの間にコミュニケーションがなく、デシジョンメイキングがうまくいかなかったのか、と考えている。軍隊組織を連想させる企業文化がそうさせていると感じているが、PLとしては責任転嫁かも知れない。若手は 3GL には、熱い思いと愛がないという。一方 3GL の方は、若手が何を言っているか分からないと言う。あなたならどうしますか。

SEA年次総会報告

事務局

今年度の SEA 総会は Software Symposium 2004 の第 2 日 (2004 年 6 月 23 日) の夕刻, Symposium の会場である岡山国際会議場で開催された。

事務局から報告し承認された昨年度の収支および新年度の予算は次の通りである：

収支計算書 (2003 年 4 月 1 日～2004 年 3 月 31 日)

支出の部		収入の部	
人件費	12,500	新入会費	308,000
事務所費	2,555,421	更新会費	2,792,000
印刷費	432,499	賛助会費	2,200,000
通信費	1,783,720	雑収入	57,453
会議費	372,275		
支部支援費	231,360		
国際活動費	39,008		
消耗品費	75,032		
雑費	22,890		
当期収支差額	-167,252		
合計	5,357,453	合計	5,357,453

予算案 (2004 年 4 月 1 日～2005 年 3 月 31 日)

支出の部		収入の部	
人件費	20,000	新入会費	330,000
事務所費	2,400,000	更新会費	2,800,000
印刷費	1,000,000	賛助会費	2,200,000
通信費	2,000,000	EVENT収入	1,000,000
会議費	180,000	雑収入	170,000
支部支援費	240,000		
国際活動費	50,000		
消耗品費	80,000		
雑費	30,000		
当期収支差額	0		
合計	6,000,000	合計	6,000,000

新年度の幹事会メンバーは、次ページの原案通り承認された。

2003年度 SEA幹事

代表幹事	荒木 啓二郎	九州大学
常任幹事	熊谷 章	東京エレクトロンソフトウェアテクノロジーズ
	高橋 光裕	電力中央研究所
	田中 一夫	日本フィッツ
	玉井 哲雄	東京大学
	中野 秀男	大阪市立大学
幹事	石川 雅彦	SRA
	大場 充	広島市立大学
	落水 浩一郎	北陸先端科学技術大学院大学
	窪田 芳夫	東京電力
	小林 修	SRA
	小林 允	フリー
	桜井 麻里	ティップス
	酒匂 寛	Designers' Den
	塩谷 和範	SRA先端技術研究所
	篠崎 直二郎	NECソフト
	新谷 勝利	フリー
	新森 昭宏	インテック W&G インフォマティクス
	杉田 義明	SRA
	中来田 秀樹	ネクスト ファウンデーション
	野中 哲	フリー
	野村 行憲	アイシーエス
	野呂 昌満	南山大学
	端山 毅	NTTデータ
	平尾 一浩	III九州
	深瀬 弘恭	フリー
	藤野 誠治	富士通総研
	松原 友夫	松原コンサルティング
	渡邊 雄一	アスキーソリューション
会計監事	橋本 勝	朝日監査法人
	吉村 成弘	公認会計士
事務局長	岸田 孝一	SRA



ソフトウェア技術者協会

〒160 東京都新宿区四谷3-12 丸正ビル5F
TEL.03-3356-1077 FAX.03-3356-1072