第４回　実践的ソフトウェア開発環境に関する集中討論

Ｐａｎｅｌ－４

ＣＡＳＥ（Computer Aided Software Engineering）

―― いまのＣＡＳＥツールは、どの程度有用か？　どう使えばよいか？

テーマ：　Ｓ／Ｗ生産効率化に関する試み

三菱電機　　小野　康生

１．まえがき

我々のセクションでは、鉄鋼・紙パルプなどの工業プラントにおいて、プログラブルコントローラというμ－Ｐ応用製品を多数納入し、制御監視及びＭＭＩシステムの構築を手がけている。コントローラは汎用計算機と比較して

（１）オートノーマスI/O 等の適用により H/Wよりも S/Wの負荷を低下させている。

（２）制御専用高位言語(POL) によりプログラミングが容易に可能。

（３）プログラムは高速応答を要求される。各々の制御機能に対し数msから数百msまでのサンプリング時間で処理される。

（４）OS等の適用はない為、それらオーバーヘッドによる処理ロスがない。

（５）プロセスと直結できる入出力を有しており、入出力命令をメモリロード、ストアと同格にしている。

（６）モニタ（スーパバイザ）、ユーティリティなどのベーシックソフトが不要である。

（７）制御応答確保と故障によるシステムへの波及阻止のために、分散化とネットワーク化によりシステムを構成する。

などの違いがある。

2．コントローラ S/W生産の実情

工業プラントと一言でいっても、ライン構成・規模・制御内容・操業形態などプラント毎に様々であって、常にオーダメイド的にシステムは計画設計され、実現される。

あるプラントを受注すると、計画者は客先との打合せにより、電機品構成や操業オペレーション・制御項目を決定し、運転方案を作成する。運転方案はプラントの仕様を規定するもので、いわば S/Wの要求定義書（外部仕様）にあたる部分である。方案記述がすべての基準となるものであるが、ほとんど手作業に頼っている。計画担当者の個人的作業が多く、その進め方や方案そのものにその個人の考え方が色濃く反映される。仕様を共通化した標準方案もないわけではないが、すべてそれで OK とはいかず、実際のシステムを記述するには不足する部分が存在し、ＣＡＳＥツールの活用が遅れている。

計画部門のアウトプットである運転方案や電機品資料が図面の形で設計部門に送られると、S/W 設計者はそれらによりプラントの制御対象群及びそれらの相対関係、制御方法やシステム要求精度を十分理解する。時には方案作成者から直接説明を受けることもある。方案を理解し、要求精度や制御応答を検討した上で、ＣＰＵ毎の担当制御範囲とプログラム実行周期を決定する。メインループ実行周期で仕様を満たされない場合には割込処理やマイナループ処理なども講ずる。これが S/Wシステム設計業務で S/W基本設計の部分をなすものである。

プログラム設計者は、１ＣＰＵに与えられた全機能を制御機能毎に分割及びレベル分けし、プログラム実行順番を決定する。分割された機能毎の詳細処理フロー図を作成しその説明書も準備する。これらの S/W詳細設計をもとにしてコントローラ専用言語置換作業（ラフスケッチ）を行う。プログラム１ステップが「ＣＰＵは何をどうする」式の制御記述であり、「何を」にあたるソースのアサインやソースコメント（ラベル）の定義が必要となる。これら S/Wの設計製作は設計者の手作業に依るところが大きく、その進め方やプログラムそのものにその個人の考え方が色濃く反映される。標準プログラムのパッケージ化もなされているが、ロジックは同じでもソースが同じ訳でなく、工事オーダ毎に人為的に設計作業が行われている。

プログラム製作段階ではＣＡＤを活用し S/Wオブジェクトを生成する。ＣＡＤの機能として下記に示す。

（1）計算機端末との対話処理にてソースファイルの編集。
（2）設計者に記述された高位言語をコントローラの言語への変換（正アセンブル）、及び逆変換（逆アセンブル）。
（3）プログラム図面作図と図面シートの分割及びシートの取合い、ソースコンコーダンスの自動出力。

設計完了した S/Wは、コントローラにローデングされ、単体機能試験・シミュレーションテスト・システム総合試験を通じて社内で動作確認をしたあと、現地プラントにおいて調整試験を実施して完成となる。 S/Wのできばえをテストし製品を客先仕様に適合させるようにするもので、試験方法や評価は試験者の技量に任せられる。計画・設計の品質が人間のパーソナリティーに大きく依存しているため、自動試験もなかなか進まない。現地トラブルが発生すると、計画・設計・試験の各担当者が総力で対応する。

3．ＣＡＳＥツール活用上の問題点

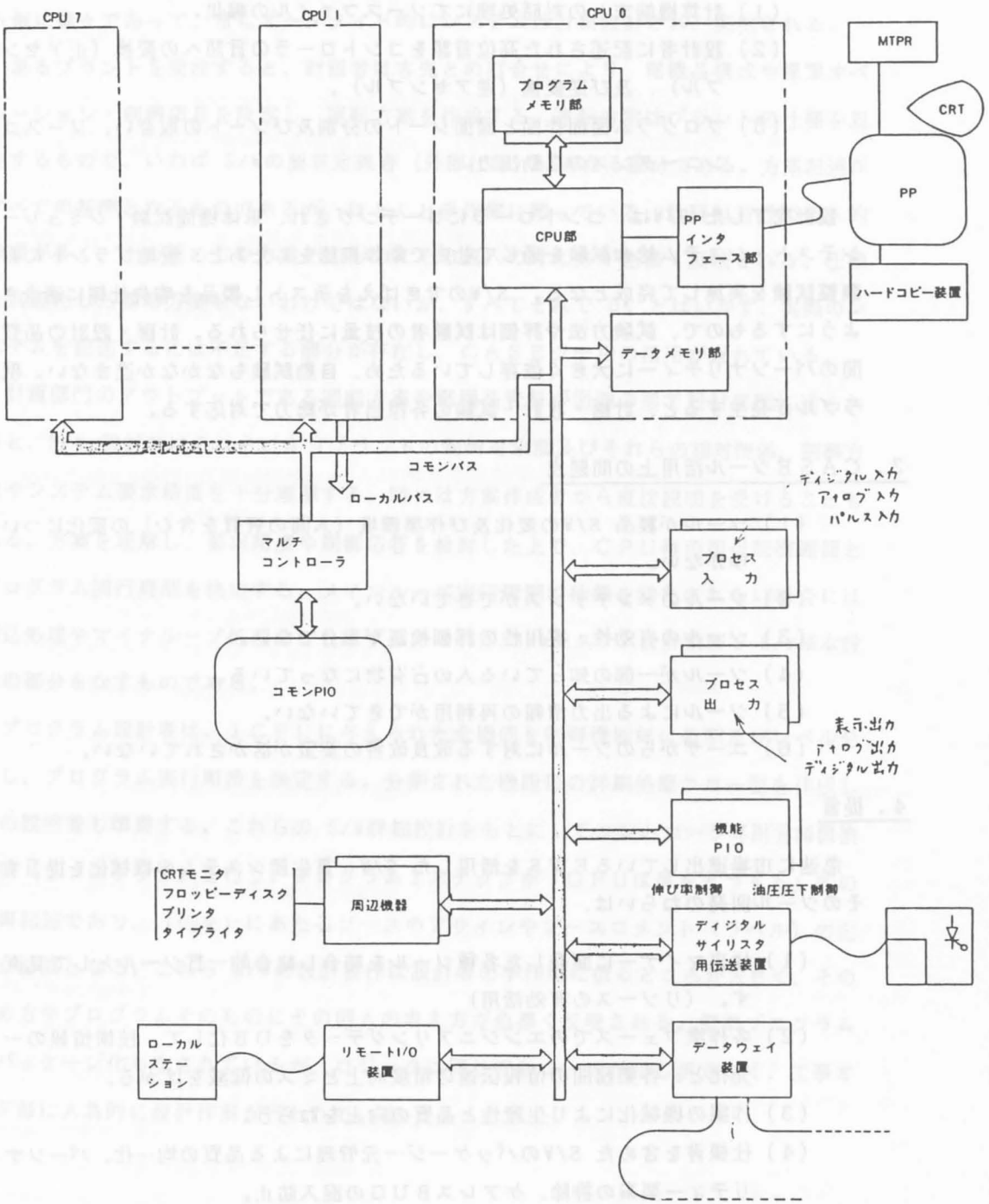
（1）ツールが製品 S/Wの変化及び作業環境（人間の資質を含む）の変化についてゆかない。
（2）ツールのメンテナンスができていない。
（3）ツールの有効性・実用性の評価検証が十分でない。
（4）ツールが一部の知っている人の占有物になっている。
（5）ツールによる出力情報の再利用ができていない。
（6）ユーザからのツールに対する改良改善の要望が活かされていない。

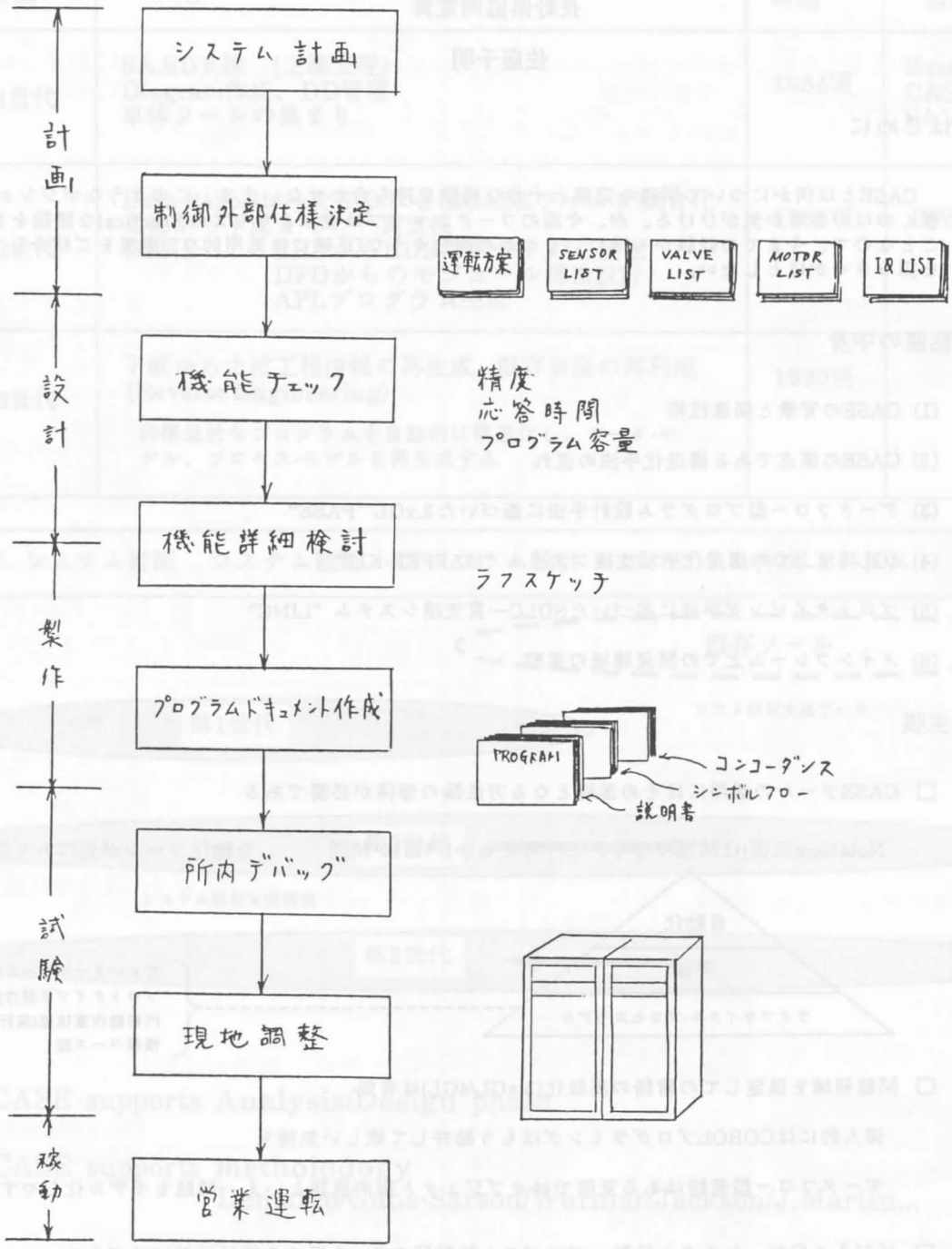
4．提言

急速に市場進出しているＥＷＳを活用した S/W一貫生産システムの機械化を提言する。そのツール開発のねらいは、

（1）従来マイナーに開発した各種ツールを結合し総合的一貫ツールとして見直す。（リソースの有効活用）
（2）各作業フェーズでのエンジニアリングデータをＤＢ化して、技術情報の一元化と、各業務間の情報伝達の精度向上とミスの低減をはかる。
（3）作業の機械化により生産性と品質の向上をねらう。
（4）仕様書を含めた S/Wのパッケージ一元管理による品質の均一化、パーソナリティー要素の排除、ケアレスＢＵＧの混入防止。

である。


CPU 7
CPU 1
CPU 0
MTPR
CRT
プログラム
メモリ部
PP
CPU部
PP
インタ
フェース部
ハードコピー装置
データメモリ部
コモンバス
ローカルバス
マルチ
コントローラ
コモンPIO
プロセス
入力
プロセス
出力
機能
PIO
伸び率制御
油圧圧下制御
CRTモニタ
フロッピーディスク
プリンタ
タイプライタ
周辺機器
ディジタル
サイリスタ
用伝送装置
ローカル
ステー
ション
リモートI/O
装置
データウェイ
装置


計画
設計
製作
試験
稼動
システム計画
制御外部仕様決定
機能チェック
機能詳細検討
プログラムドキュメント作成
所内デバッグ
現地調整
営業運転
運転方案
SENSOR LIST
VALVE LIST
MOTOR LIST
IO LIST
精度
応答時間
プログラム容量
ラフスケッチ
PROGRAM
コンコーダンス
シンボルフロー
説明書

メインフレーム環境のCASE

長野県協同電算

佐藤千明

0. はじめに

CASEとは何かについて明確な定義と十分な経験を持ち合わせないまま、このようなポジションペーパを書くのはいささか気がひける。が、今回のワークショップは従来よりもよりPracticalな議論をしようということなので、今までの経験から得たいくつかの面白そうな(正確には実用的な?)環境をご紹介し、その可能性を探るきっかけとしたい。

1. 話題の中身

(1) CASEの背景と関連技術

(2) CASEの原点である構造化手法の流れ

(3) データフロー型プログラム設計手法に基づいた3.xGL "PASE"

(4) 4GL環境上での構造化手法支援システム "MAPPER-KIT"

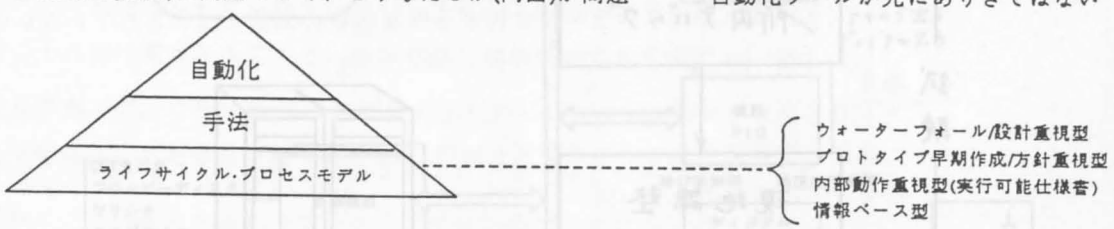
(5) プロトタイピング手法に基づいたSDLC一貫支援システム "LINC"

(6) メインフレーム上での開発環境の実態

2. 主張

□ CASEツールの活用にはその基礎となる方法論の修得が必要である

Notation(表面)は問題でなく、どう考えるか(内面)が問題　　自動化ツールが先にありきではない



□ 問題領域を限定しての言語の高級化(3.xGL/4GL)は有効

個人的にはCOBOLプログラミングはもう勘弁して欲しい気持ち

データフロー型言語はある意味ではオブジェクト指向言語といえ、問題をモデル化しやすい

□ ビジネス分析→システム設計→プログラム設計間のGapを埋める模範回答はまだない

現状の統合化ツールは単に各工程支援ツールの寄せ集めの域を出ていない

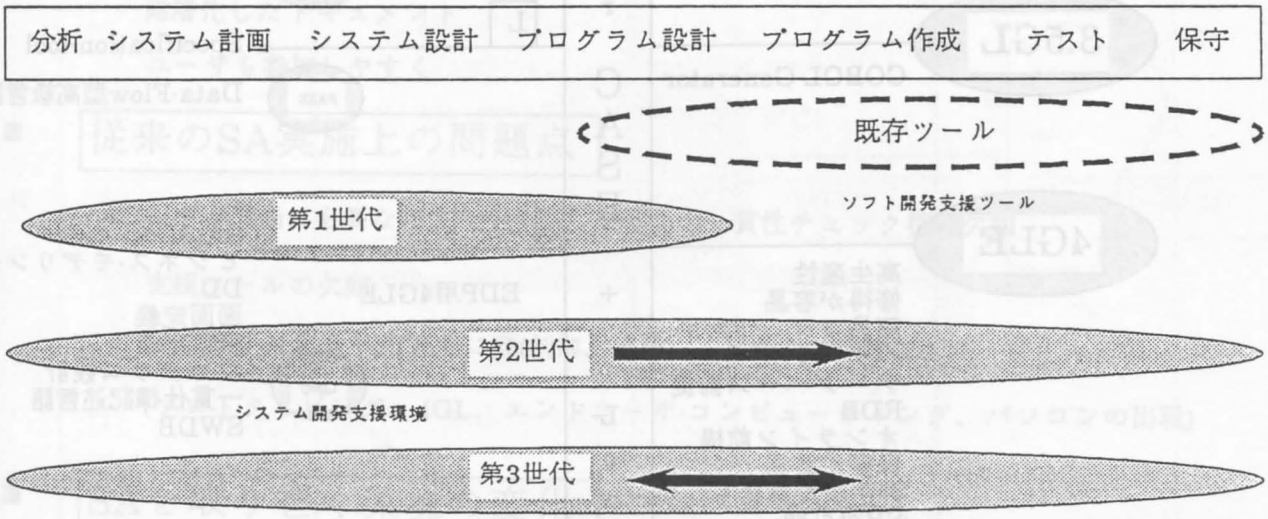
DFDの最下位プロセスとプログラムモジュールは1対1対応しない

鍵は、モデル化手法と、抽象度を上げた仕様記述言語の実用化にあるのではないか

□ プログラム設計支援よりもシステム分析/システム設計支援用の方法論/自動化ツールが欲しい

■ CASEの世代分類

分類	内容	時期	製品
第1世代	SA,SD支援 (上流支援) Diagram作成、DD管理 単体ツールの集まり	1984頃	Excelerator CASE2000 SAVER
第2世代	Data Repositoryを中心に各種CASEツールが統合化 上流から下流までの一貫支援 自動化… ERDからRDB用データの正規化 DFDからのモジュール構造設計 APLプログラム生成	1988頃	IEW
第3世代	下流から上流工程情報の再生成 既存資産の再利用 (Reverse Engineering) 非構造的なプログラムを自動的に構造化し、データ・モデル、プロセス・モデルを再生成する	1990頃	



■ CASE supports **Analysis/Design** phase

CASE supports **metholodogy**
Demarco/Gane·Sarson/Warnier/Jackson/J.Martin...

構造化手法: データモデリング、プロセスモデリング、プロトタイピング、データ中心設計
機能分解法、再使用可能モデル/コード、オブジェクト指向、ES、...

■ CASE = システム開発の全工程を対象に
構造化手法に基づいた各種支援ツールを組み合せて実現した
統合型開発支援環境

CASEを取り巻く技術
(SA→4GL,Prototyping,RDB→CASEE)
SA,SD
WSの発展
3GL
SW·CAD/CAM
部品化、再利用
DD
3.5GL
COBOL·Generator
4GLE
高生産性
修得が容易
簡易コマンド
パターン化
データベース前提
RDB
オンライン前提
保守が容易
非手続型言語も
SPが可能
プロトタイピング
迅速な処理結果
視覚化
DB技術
RDB
3層スキーマ
SQL
分散DB
Prototyping
Visual化
CASE‖
Upper
CASE
+
Lower
CASE
U
DFD,CFD,ERD,STD
DD,MS
Kit
L
データ項目定義
レコード定義
ファイル定義
データベース定義
画面定義
帳票定義
手続部以外の自動生成
手続部はスケルトン生成
Detail·Chart·Editor
EAGLE
YPS
L
Specification·tool
Data·Flow型高級言語
PASE
U L
SA,SD
ビジネス·モデリング
DD
画面定義
帳票定義
プログラム設計
一貫仕様記述言語
SWDB
EDP用4GLE
LINC
U L
Data·Modeling
Model·translation機構
MML
柔軟なデータ構造
データとプログラムの独立
U L
Diagram·Editor
Screen·Painting
4GL

構造化手法の環境

■ **SP** Dijkstra 1968 NATO

SD Stevens,Myers,Constantine 1974 IBM モジュール設計<データの流れ>

Warnier,Jackson プログラム設計 <データの構造>

SA TomDemarco 1978 Gane、Sarson 1977

Softec社 SADT

■ ## SAの狙い

システム範囲を明確化後に分析……Context·Diag.、Level 0·DFD

DFD、DD、プロセス仕様書(MS)、ERD、STDというドキュメントに体系化

図を使う 分割した記述

階層化したドキュメント

ユーザも参加しやすく

■ ## 従来のSA実施上の問題点

モデル開発に必要な膨大な手作業、文書間の一貫性チェック機能欠如

支援ツールの欠如

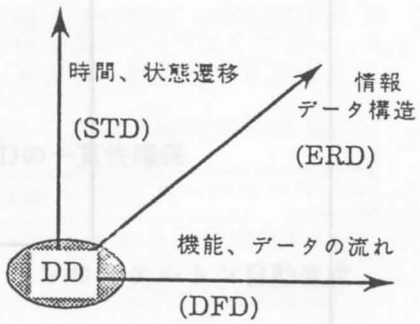
リアルシステムへの適用に難(制御、信号) バッチ処理向き

(プロトタイピング、4GL、エンドユーザ·コンピューティング、パソコンの出現)

■ ## SAを取り巻く環境の変化

モデル化技法の発展

- DFD(機能,機能間データ)
- CFD
- ERD(データ構造)
- STD(時間、動作)



CASEツールの出現

Diagram·Editor (Graphicsサポート)

Data·Dictionary

Consistency·Check(DFD↔DD,DFD↔DFD,各モデル間,,)

4GL、プロトタイピング、RDB等との補完

プロセス・ネットワーク for PASEGEN
TomDemarco流
データフロー型プログラム設計
OLD
マスタ
照合M
マッチング
照合TR
単チェック
チェック済TR
マスター
更新
NEW
マスタ
トランザ
クション
データリスト

MAPPER-KIT1100 概要

■ MAPPERは 統合情報処理環境(インタプリタ)

リレーショナル型DBが基本 ＋ 既存DBとのインタフェイスを持つ

利用部門向き会話型コマンド ＋ 簡易(?)プログラミング言語

オンライン処理が基本 グラフ機能に優れる

TPS、バッチ、FAX、光ファイル、TELIDON等とのインタフェイス

統合OA機能 (電子メール、文書管理、伝票処理、秘書業務等)

実行環境支援(機密保護,リカバリ,稼働情報) 集中管理

利用形態
① MAPPER単独利用
② 現行(基幹)システムとのオフライン接続(MT)利用
③ MAPPER↔バッチ結合
④ MAPPER↔リアルタイムトランザクション処理結合
⑤ MAPPER↔MAPPER(垂直接続)
⑥ MAPPER↔MAPPER(水平接続)

■ MAPPER-KITはMAPPER上のMAPPER･APL開発支援環境

4GL環境上での構造化手法支援システム

Kit･Build 開発技法SUMによる分析、設計、開発支援ツール

- 徹底的な分析
- 要求の明確化
- 利用者との協同作業
- 業務分析、データ分析、DDの一貫性確保
- 全工程の文書管理、標準化
- 画面生成、レポート生成、ランのスケルトン自動生成
- Diagram自動作成 (SA)

Kit･Aid KitBuildで作成したAPLの使用/保守支援

- DD使用による項目属性変更時の自動保守
- DD変更→DB変更→画面定義変更→ラン変更

標準開発モデルSUMの開発工程

SA用語

1 業務分析(SA) →目標アクティビティ·ダイアグラム

1.1 ターゲット(対象)エリアの明確化　Context·Diagram作成

1.2 問題点の明確化　業務システム、情報システムの調査、真の問題点指摘

1.3 現状システムの明確化　レベル0のDFDより階層的にダイアグラムを作成

1.4 代替案の作成　代替アクティビティ·ダイアグラム(AD)の作成

1.5 目的システムの決定　目的アクティビティ·ダイアグラムの作成

2 要求仕様 →インフォメーション·ダイアグラム

プロセス

2.1 目的システムの詳細AD作成　最下位レベルのアクティビティ·ポイントまで詳細化

2.2 仕事の分類　それをタスクに分割し手作業か機械化処理かに分類

2.3 インフォメーション·ダイアグラム(ID)の作成　手作業と機械化処理間の利用者I/F·SET定義

2.4 機械化部分のシステム化案の検討　例えばオンライン/バッチ、パッケージ/開発、DBMS

2.5 構造化ウォークスルー　機械化処理チェック、利用者I/Fチェック

2.6 費用対効果分析

2.7 バックアップ、リカバリ方法、セキュリティ考慮

3 システム分析(JSP) →SUMデータ·モデル

3.1 インフォメーション·ダイアグラムの各タスクを分析し、サブ·タスクに分割

3.2 各サブ·タスクの入力、出力、マスタファイルの決定と項目の洗い出し

3.3 各入力、出力、マスタファイルのデータ分析を行い、階層的データ構造図を作成

3.4 各サブ·タスクの処理内容(入力→出力変換方法)の記述

3.5 データ構造図を分割して基本データ項目を作成

3.6 基本データ項目を正規化

3.7 基本データ項目をもとにSUMデータモデルを作成

3.8 データ辞書の構築　(項目名、タイプ、桁、内容)

4 データベース設計 →MAPPERレポート関連図

4.1 SUMデータモデルから論理MAPPERレポート関連図を作成

4.2 論理MAPPERレポート関連図を調整し、最適化されたMAPPERレポート関連図の作成

5 ランの設計とコーディング

5.1 MAPPER会話機能による処理の実現と記述

5.2 会話機能で足りない場合はMAPPERランの作成

(画面作成→ラン·スケルトンの自動生成→ランの完成→ランのコンパイル)

MAPPER- KITの特徴

■ 全工程成果物の文書管理

◆ 開発過程で文書化を同時に行う

■ 図によるコミュニーケーション

◆ 今はコードから図への一方通行のみ

◇ 逆も欲しいが...今のインテリジェント端末には重過ぎる

■ データ辞書によるデータ項目の一括管理

◆ 単なる管理ではなく、設計/自動保守に生かされている

■ ディレクトリによるレポートの一括管理

◆ すべての情報がMAPPERレポート(RID)内にある

■ 自動生成による開発の生産性向上

◆ 画面入出力部分の自動生成は有効

◇ RUNのロジック部分の自動生成は今後の課題か?

◆ KITの思想は会話機能の命令語(コマンド)の組合せによる処理実現にあるため、ロジック部分の生成に対する支援が少ない(少なくて良かった)

■ 自動変更による保守の生産性向上

◆ 今までのMAPPERは本当の意味でのリレーショナル型ではなかっただからKITの自動保守は有効

◆ 最新バージョンのMAPPERはリレーショナル型になっている
(フィールド指定が項目名で可能) (従来はカラム数指定)

LINC

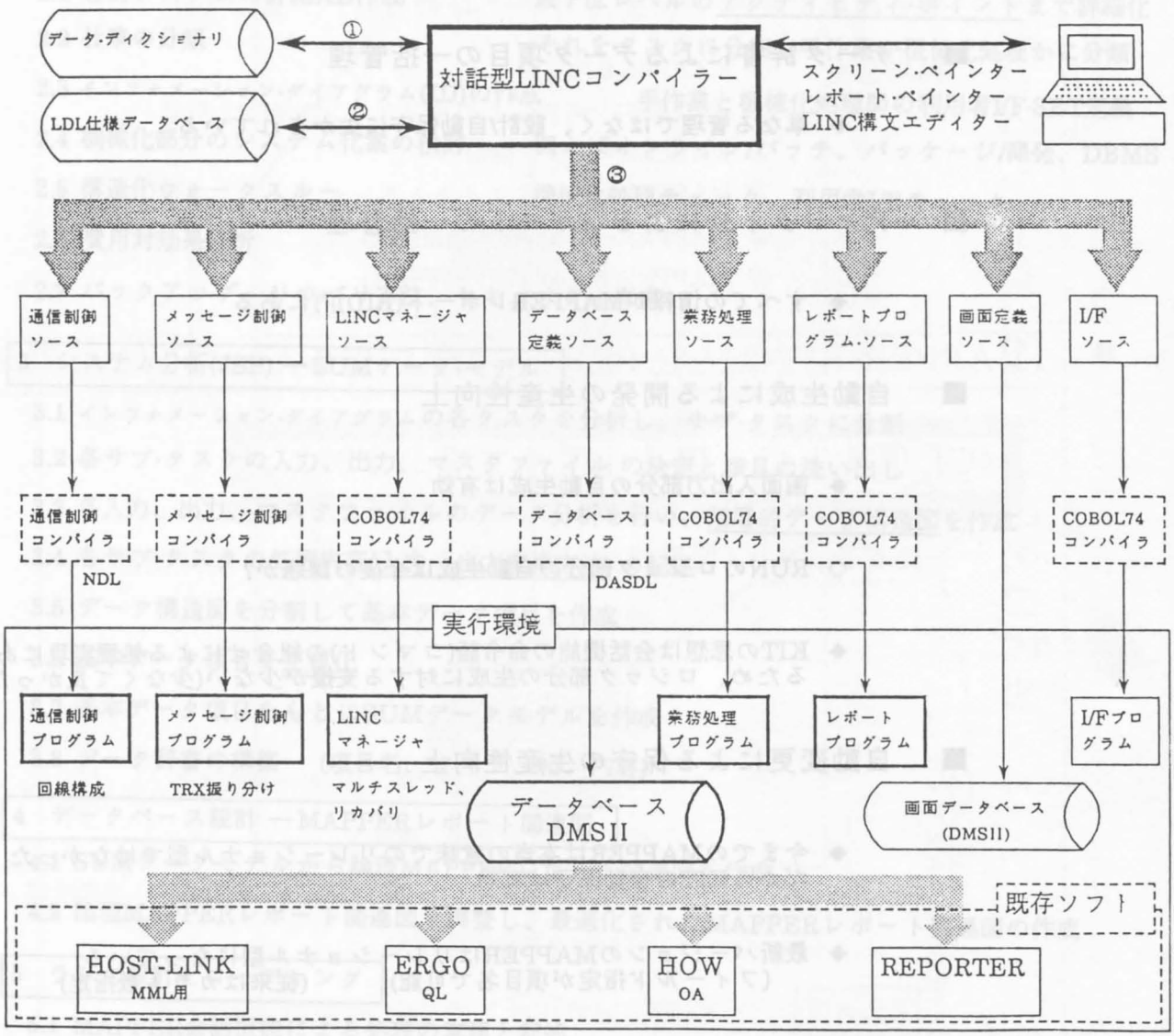
オンライン･データベース･システムを中核とした統合情報システムを開発･保守するための4GLE

Integrated･System、Prototyping、Business･model、Screen･Painting、DD、DOC･Less

(Profile)

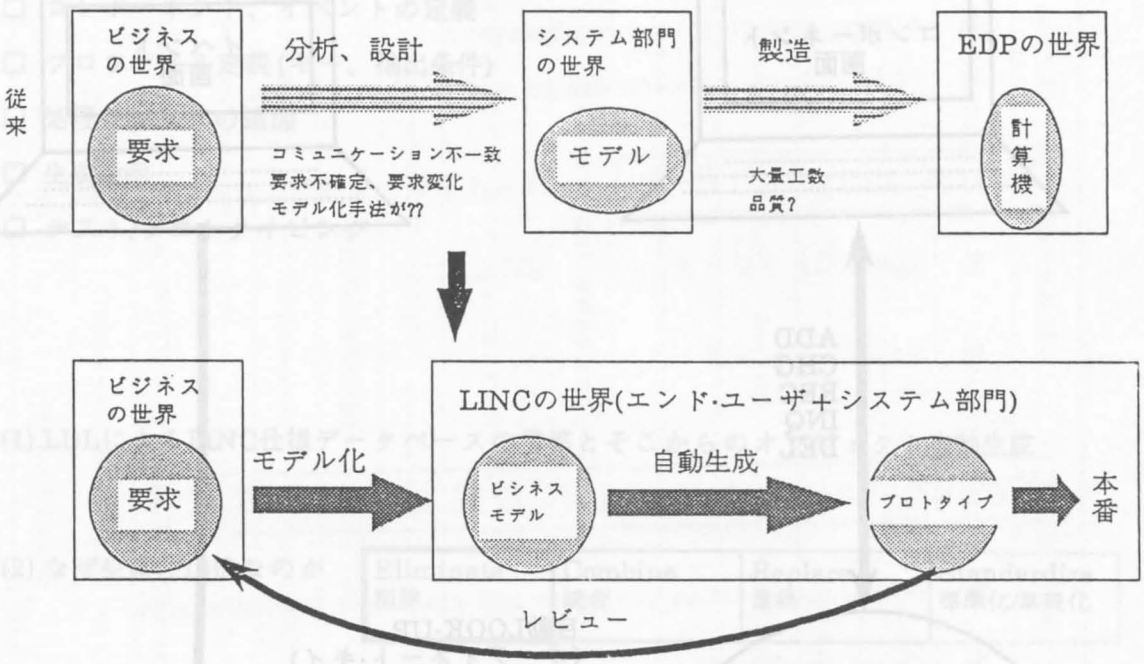
概要

アプリケーション･プログラム(Logic)、データベース(Information)、メッセージ制御･ネットワーク制御プログラム(Network)に展開されるシステム仕様を、メニュー方式による対話型操作と構文エディタにより高水準の単一言語(LDL)仕様データベースとして作成する。次に、それに基づいてLINCコンパイラーが各ソースプログラム、定義体、さらにはオブジェクト･プログラムまでを自動生成する。



開発方法

(1) ビシネス･モデルとプロトタイピングに基づいたシステム開発



(2) ビシネス･モデルとは

□ コンポーネント(資源)

◇ マスターファイル、顧客(台帳)、製品(台帳)、従業員(台帳)、

◇ スクリーン･ペインターで台帳の画面を作成

□ イベント(活動)

◇ トランザクション･ファイル、発注、受領、出納、集金、入庫、出庫

◇ スクリーン･ペインターで伝票の画面を作成

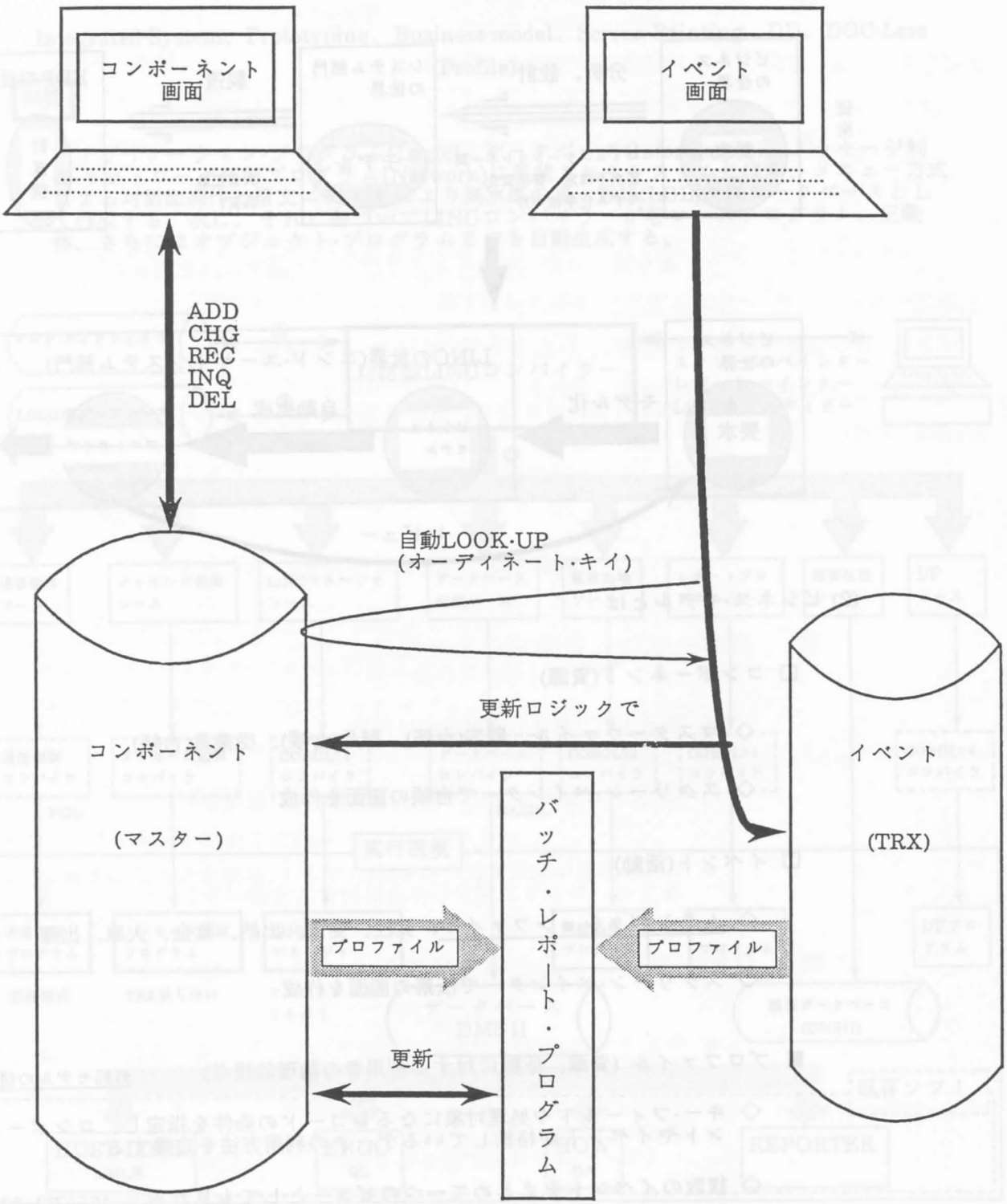
■ プロファイル (資源、活動に対する利用者の論理的視点) 外部モデルの提供

◇ キー･フィールドや処理対象になるレコードの条件を指定し、コンポーネントやイベントが格納しているデータの利用方法を定義する

◇ 複数のイベントをまとめて一つのビューとしても見れる View･Fileの提供

- 在庫＝<入庫、出庫>
- 売掛＝<販売、入金>

モデルの概念


コンポーネント
画面
イベント
画面
ADD
CHG
REC
INQ
DEL
自動LOOK-UP
(オーディネート・キイ)
更新ロジックで
コンポーネント
(マスター)
イベント
(TRX)
バッチ・レポート・プログラム
プロファイル
プロファイル
更新

開発手順

- □ ビジネス･モデルの分析
- □ データ･ディクショナリにデータ登録
- □ コンポーネント、イベントの定義
- □ プロファイル定義(キー、抽出条件)
- □ 処理ロジックの追加
- □ 生成指示
- □ テスト/プロトタイピング

考察

(1) LDLによるLINC仕様データベースの構築とそこからのオブジェクト自動生成

(2) なぜ生産性10倍なのか

Eliminate 削除	Combine 統合	Replace 置換	Standardize 標準化/単純化

LINC用のスケルトンを持ち、枠を固めていることが成功要因

(3) LINCの狙いは単なる生産性UPではなく、一つの仕様で情報システムに関するあらゆることがらを定義、構築可能にすること。従って今までの開発方法論の延長線上でのCAD/CAM指向(for 3GL)とはアプローチが違う。

(4) 従来の構造化手法はシステム分析(SA)⇒システム設計(SD)⇒プログラム製造(SP)への橋渡し部分が不明確であった。 LINCはビジネス･モデルからISPECへのスムーズな展開が可能であり、LDL仕様データベースを仲介としてプログラム製造工程を大幅に自動化した。

(5) プロトタイピングの成功はしっかりした分析手法を必要としないのか?
だとすれば今の構造化手法はどうなる?

ビジネス･モデル化の方法論は必要 (例えばMAPPER-KITのSUM)

絵からコードへの自動生成効果がポイント
頑張って絵を書いて確認後、3GLE上でまた力仕事するのは疲れる!

(6) LINCにより電算プロパーは技術者でなくなる時代がくる!?

確かに面倒くさいDB/DCインタフェイスのプログラミングはいらなくなる

事務処理システム用プログラマーは事務屋さんと同一視される

残された道はシステム･インテグレータ??

Water Fall Model の問題点

(わかっているけど他になかった)

- Communication gap と Documentation cost
- End user による検証が開発最終工程でないと難しい
- Always changing user requirements
- Technical gap between design and implementation

Prototypingへの期待

ユーザをいかに開発に参加させるかがポイント

- 使い捨てアプローチ 3GLE
- 進化型アプローチ

Evolutional·Prototypingに必要とされる機能

- Formal specification language and automatic implementaion
- Visual specification
- Software database Specifications,objects,decisions,memos
- Flexible software structure

LINCはどこまで Balzer's Paradigmに近づいたか

- Formal specification ----- LDL
- Prototyping standard
- Specification is the prototype
- Prototype validated against intent
- Prototype becomes implementaion
- Implementaion machime aided
- Testing eliminated
- Formal specification maintained
- Development automatically documented ----- for engineer、operator
- Maintenance by replay

アプリケーション・ソフトウェア開発環境技術分析　on Main-Frame

分析支援ツール		DFD,CFD,ERD,STD　SEWBのSDF業務設計(WS)　MAPPER-KITのSUMのアクティビティ・ダイアグラム(TSS)　IEW/PWS,AWS(WS)
システム設計支援ツール		SEWBのシステム・フロー設計(WS)　SEWBの画面遷移シミュレータ(WS)　MAPPER-KITのSUMのインフォーメーション・ダイアグラム(TSS)　IEW/DWS(WS)
COBOL生成系　3.?GL ■ CAD・CAM方式 標準化の延長 □ EDP部門向け □ 一貫支援 □ DD □ MML指向	Chart・Editor	PAD⇄ソース(WS)　YPSエディタ(WS)、YPS→ソース(ホスト)　SPD⇄ソース(WS)　H/W技術の成果
	Symbolic・Debugger	SEWBのPADテスタ(WS+ホスト)　SEWBのソース実行テスタ(WS)　YPSデバッガ(TSS,Load-M)　YPSテストカバレッジ(TSS,Load-M)　PADS1100(TSS)
	Screen・Painting	画面、帳票、ファイル設計ツールはどこも装備　WS上で設計しファイル転送でホストに送り、ホストの定義体に変換(IDLII,SOFPIA,SEWB,CASET)　TSS方式(TSX,LINC)
	Documentation	YPSドキュメンタ(ホスト)　SOFPIAのPMA/DOCUMENTER(ホスト)　SEA/IのPWB/DOC(WS)　ALICEのSIDB,波及分析　その他多数あり
	Data・Dictionary	どこも装備している　ALICEのSIDBはデータ・Dicとプロセス・Dic(段階別仕様記述 by SDL)
	Language	IDLII...アクションをデータフロー型でつなぐ　CAPE...ユニットをデータフロー型でつなぐ　ALICEのSDL...基本設計からの一貫記述、ジャクソン法でスケルトン生成
COBOL生成系　3.?GL □ 言語支援のみ		PASE、HYPER・COBOL.....データフロー型がきれい　YPS.....部品化によるインクリメンタルな抽象化が面白い　その他...JASPOL,CORAL,JSP-COBOL,
EDP部門用 4GL ■ オートメーション方式 プロトタイピング □ 一貫支援		LINC.....すべてをLDLにて形式的に記述した仕様データベースからすべてのオンライン環境プログラムを自動生成（大掛かりなシステム） (IDLII.....すべてをIDLIIにて形式的に記述した仕様データベースからプログラム/ファイル/画面オブジェクトを自動生成　(中規模なシステム))
エンド・ユーザ向 4GL ■ プログラムレス方式 □ イージー指向		非定例、非定型、特定部門用システム向き→DBの検索/加工/レポート作成→RDB＋QL　CSP＋DB2　TQFII＋RIQS　MAPPER　ETOILE/OP 容易なプログラム作成→視覚化(Visual・Programming)、メニュー・ガイダンス表示　CASET　IDLII
プロトタイピング		使い捨て:SEWB　半分完成:CASET(WS上でインタプリト後ホストで生成)　そのまま本番:LINC　早期視覚化:IDLII、MAPPER
MMLによる機能分散		チャート・エディタ、画面/帳票/ファイル定義はどこも装備　これからは上流支援ツールがでる?　SEWB、IEWが機能分散面で先行か?　まだセッションの切り替えが必要(ローカル/ファイル転送/TSS)
AIの活用		INTEGRAL　IEW　仕様記述言語が自然言語に近くなる　仕様記述の一貫性チェック　課題に最適な部品、パターンの自動選択

第4回SEA環境ワークショップ Panel-4 CASE

Software through Pictures[TM]の利用経験と評価

株式会社 ＳＲＡ 環境開発部

桜井麻里

mari@sra.junet

1. はじめに

筆者は、1988年2月末から現在までの約8ヶ月間、Software through Pictures (以下StPと略す)という構造化分析手法を支援するCASEツールを使用した。使用目的が日本語StPの整備と販売促進のためであり、本格的な開発に利用したわけではないから、生産性にどのくらい寄与したかということはあげられない。それでも、StPがどのくらい便利で楽しいツールであるか、StPを利用する上で何が問題だと感じたか、ということはあげられる。また、たび重なるStPのデモ経験から、はじめてStPを見た人がStPに対してどのような疑問と感想を持ったか、ということについて、若干ではあるが、まとめることはできる。

そこで、今回の発表では、これらの経験を通して得たことを簡単に紹介して、プログラム委員より指定された論点である、

・いまのCASEツールはどの程度有用か

・どう使えばいいのか

について、話題を提供したい。

一口にCASEツールといっても、さまざまで、議論しにくいと思うが、この発表を元に、会場の皆様と一緒に、CASEツールの利用法について、考えてみたい。(本当は、どういうCASEツールがあるとうれしいかについて、ざっくばらんに意見交換をしてみたいのだが.....)

2. StPの稼働環境

基本的には、ビットマップディスプレイを持ったUNIX[1]ワークステーション上で動作する。英語版のStPであれば、SUN, HP, Apollo, VAX stationなどの各種UNIXワークステーション上で動作するものが用意されているが、日本語版は、SONY-NEWS(NEWS-OS 2.x, X10R4)のものしか用意していない。

2.1. パンフレットでの「日本語版StP」の稼働環境

機種=SONY-NEWS;
OS=NEWS-OS Release2.x(日本語対応4.2BSD+NFS);
メモリ=4MB以上; ディスク(ユーザ・エリア)=20MB以上;

Software through Picturesは、米国IDE社(Interactive Development Environments, Inc.)の商標です。

[1] UNIXはAT&Tベル研究所で開発され、ライセンスしているＯＳです。

ウィンドウシステム=X-Window V10R4; かな漢字変換エンジン=sj2;
ビットマップディスプレイ;
その他=レーザビーム・プリンタなど。

2.2. 私の「日本語版 StP」利用環境

機種=SONY-NEWS(NWS-841);
メモリ=16MB; スワップ=32MB;
ディスプレイ=17インチ カラー・ビットマップ;
OS=NEWS-OS Release2.2;
ウィンドウシステム=X-Window V10R4; かな漢字変換エンジン=Wnn;
プリンタ=CANON LBP-8 II AJ-2 と SONY NWP-533 を場合により使い分け;
ドキュメント環境=SRA 社内用 DWB 2.0;
その他=UNIX ネットワークで流れてきたさまざまなソフトウェア;

3. 今の StP では、何ができるのか

StP でできることはいっぱいあって、いちいち説明していると、時間が足りない。くわしくは、デモンストレーションを見ていただきたい。とりあえず、できることのうち主なものを箇条書であげておく。

- 構造化分析／設計に必要なことは一通りできる。しかも、簡単。
  - データフロー・ダイヤグラム，データ構造ダイヤグラム，Ｅ－Ｒダイヤグラム，プロセス仕様(ミニスペック)が楽に描ける。バブル(プロセス意味する)を移動すれば、バブルにつながっている線(データフローを意味する)も一緒に移動するし、undo もできる(一回だけだけど)。関連のあるダイヤグラム間は、自由に行き来できる。モジュール設計時にあると便利な構造化チャートも描ける。
  - 従来の構造化分析では人手で書いていたデータ辞書は、ダイヤグラム編集中に、マウスで“データ辞書生成”をクリックするだけで、自動的に作ることができる。もちろん、ダイヤグラム編集後に、バッチ的にデータ辞書生成することもできる。
  - データフロー・ダイヤグラムの階層間で、入出力に矛盾がないかどうか、ダイヤグラム編集中に、マウスで“詳細化チェック”をクリックするだけで、自動的にチェックすることができる。チェックでのエラーメッセージは、別のウィンドウに表示されるが、ファイルにとっておくこともできる。エラーメッセージは、とてもわかりやすい。もちろん、ダイヤグラム編集後に、バッチ的にチェックすることもできる。
  - チェックできるのは、それだけではない。名前だけしか定義されていなくて、詳細が未定義のものを洗い出すこともできるし、変更日付のチェックもできるし、入力フローのないプロセスや、出力フローのないプロセスの洗いだし、他のダイヤグラムで定義された内容との整合性のチェックなどいろいろできる。
  - データ辞書の定義内容は、いつでも簡単に参照できる。どこで定義されていて、属性が何で、どこで使われているか(あるいはどこにも使われていないか)ということは、一目でわかる。
  - 各ダイヤグラム編集中に、ダイヤグラムを印刷したくなれば、マウスで“印刷”をクリックするだけで、レーザプリンタにきれいな図を出すことができる。もちろん、バッチ的に一括して印刷することもできる。
  - すでに作成したダイヤグラムを、読み込んできて編集することもできる。ダイヤグラムの複写、名称変更、削除なども簡単。

- イベントや信号、システムの状態遷移など、リアルタイムシステムを分析／設計する際に必要な情報も、ハトレーの手法による図式表現を用いて定義し、管理することができる。
  - 状態遷移図，制御仕様(control spec)の定義に必要な表(決定表，プロセス活性表，プロセス活性化マトリクス，状態遷移表，状態イベントマトリクス，イベント論理表，動作論理表）を扱える。
- 分析／設計情報や、作業の履歴情報は、プロジェクト・データベースとして、一元管理されているから、何かと便利。
- プロジェクト・データベースの情報をとりだして加工するための仕掛けにはかなり柔軟なものがあるから、自分たちに必要なものを自分たちで作ることができる。
- 版管理ができる。
- 異なるワークステーションから複数の人間で分析／設計作業しても大丈夫なロック機能がある。必要があれば、ロック機能を使うことができる。このロック機能は、かなり強力。
- いろんなわがままを許してくれる機構があるから、バイナリユーザでも、自分の好みの環境を作ることができる。

4. StP の魅力

4.1. ToolInfo と ToolSpec でかなり遊べる

やりたいことが、簡単にできるのがいい。ほとんどの操作は、ウィンドウ上で、マウスで１回か２回クリックするだけで、できる。

小さな親切、余計なお世話がないのがいい。個人の好みに左右される部分は、できる限り、個人の好きなようにさせてくれている。一度、自分なりの環境を整えれば、いつでもその環境を使えるし、人に渡したければ、渡してもいい。

もし、将来、構造化分析以外に、いい方法が見つかったら、その手法を支援するようなツールを自分でつくって、自分用の StP をつくることができるような仕掛けがすでに用意されているのが、うれしい。(もちろん、作るのは、それなりに大変であるが。)

4.2. オープンで強力な RDBMS 機能でいろいろ遊べる

プロジェクト・データベースは、TROLL という RDBMS で管理されている。この TROLL は、テキストでかかれたスクリプトを読んで実行できるインタープリタとしても、また、C 言語のプログラムからよべるライブラリとしても提供されている。しかも、TROLL の簡単な対話型エディタも提供されている。これらを使うと、バグレポートの管理や、顧客の名簿管理などが、簡単にできる。(実際、SRA における StP のバグ管理は、TROLL を使って行なっている。)

5. StP のデモの時、よくされる質問と、それに対する答え

StP のデモをしているといろいろなことを質問される。それぞれの人が描いているイメージによって、出てくる質問もさまざまであるが、その中でも、多いのが次のような三つの質問である。

Q1. 上流工程から下流工程までを全部サポートすることはしないのですか?

Q2. 分析から設計への流れはどのようになるのですか?

Q3. ドキュメントは、どのようなものがサポートされるのですか?

これらに対して、私は、次のように答えることにしている。この答えから、StP の考え方と現時点における限界を察して欲しい。

A1. 全部をサポートすることはしていません。下流工程については、すでにいろいろなツールが出回っておりますし、利用者それぞれに、使いなれたものがあると思います。ですから、StP では、そのようなツールを取り込める環境を提供することで、下流工程へのサポートとしたいと考えています。StP では、どういうシステムの開発を支援するかということは、特に限定していないので、下流工程への直接的サポートもしにくいのです。

A2. 分析から設計への自動的な変換などはしていません。分析結果である各ダイヤグラムやデータ辞書を参照しながら、設計するということになります。

A3. 各エディタで描いた図と、データ辞書の一覧が、そのまま、ドキュメントとなります。

5.1. 大体においていえることは;

- はじめての図なら効率は悪いかも知れないが、一度入れてしまえば、修正は楽。
- たった一人で、その場限りのソフトを作るなら、こんなものはいらない。しかし、組織の中で作成するようなシステムであれば、これらの機能は必要であるし、あって損にはならない。

6. 今の CASE ツール

構造化分析手法をサポート，上流工程中心。下流工程までサポートしているものもないわけではないが、単一のメーカー／ＯＳ／言語で、閉じた世界しかサポートされない。

今のところ、汎用的なプロセス仕様の記述方式にネックがあり、支援対象のアプリケーション分野や言語を特に絞らない場合は、ソースコード生成まではなかなかいかない。

7. CASE ツールを利用する上で何が問題だと感じたか

まず、ツールが支援している方法(例えば構造化分析という手法)論とその考え方を学ばなければならない。そして、そのやり方に熟練しないと、ただのお絵書きツールになってしまう。

どんなアプリケーションにでも、構造化分析が一番いいやり方だというわけではない。たいていの場合、アプリケーション分野を絞れば、その開発に、最適の方法というのは、別にある。構造化分析は、現時点で、もっとも平均的によいといわれている手法に過ぎない。だから、CASE ツールを導入する際には、支援対象のシステムがどのようなものなのかを考慮しないと、満足な結果は得られない。

上流から下流まで、本当に、全部、一つのツールで、サポートすることが必要だろうか? 現時点では、StP のように、オープン・アーキテクチャにして、カスタマイズに任せるのが最良と思うがどうだろうか? お仕着せのツールでは、どうしたって無理がある。

はじめて、構造化分析という手法を使ってみて感じたことは、上流から下流というよりも、論理モデル(Essential Modeling)から物理モデル(Implementation Modeling)の方が、問題のような気がするということである。物理モデルのデータフロー・ダイヤグラムなどからは、比較的簡単に implement できるが、論理モデルからいくつかの物理モデルを作成し、検討するというのは、相当な手間がかかるし、むずかしい。こればかりは、人の頭を最大限に活用しなければ、きちんとできない。この辺に関しては、現在の CASE ツールでは、ほとんどサポートされていなくて、少々不便である。

従来のシステム開発で、物理モデルと論理モデルをはっきり分けて、開発しているところは、わずかだと思う。必要なことはわかっていても、そう簡単には、頭が切り替わらない。練習が必要だと痛感している。

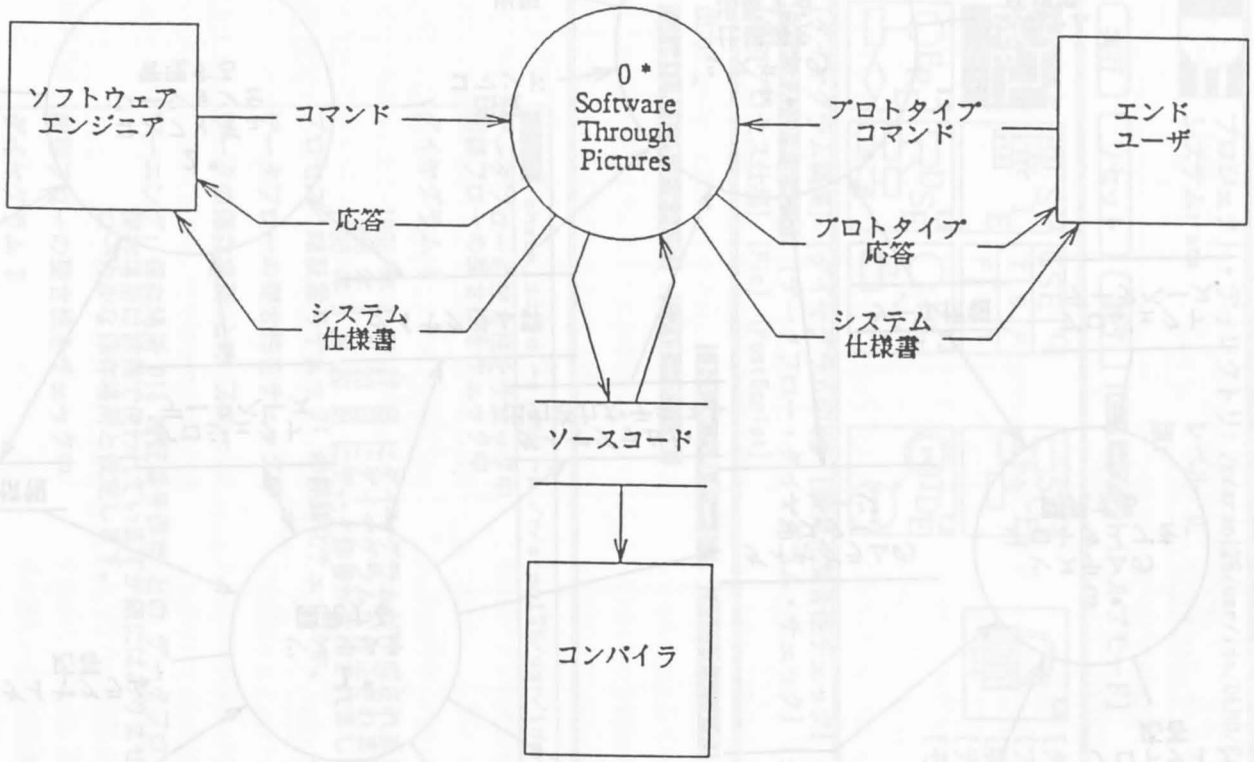
8. 私のまとめ

ツール提供者側から見ても、ツール利用者側からみても、一つの大きなツールですべてを行なおうとするより、細かいものを自由に組み合わせて一つの枠組の中で自分たちなりにまとめあげた方が、より有用であるというのが、私の拙い経験から得た見解である。そのために、枠組の提供とその利用技術が、いま、必要とされているのだと私は理解している。その意味では、StP は、かなり強力で柔軟な統合化のための枠組とカスタマイズのための機構を用意している。構造化分析に関していえば、それに必要な中身(個々のツール)も用意されている。私は、まだ、マニュアル全部をよみこなせていないが、読むたびに新しい発見がある。これからも、どんどん StP を使って、その利用技術を貯めていきたい。StP は、今後が楽しみな CASE ツールといえる。

9. ひとりごと

私は、構造化分析という手法も知らなかったし、ちゃんと設計したこともなかった。考えたことを表現するのも苦手である。でも、StP を使って、何かしようと考えるうちに、少しずつではあるけど、"考えたものを表現する" ということができるようになってきたような気がする。使ってよかったナ。

stp: レベル top

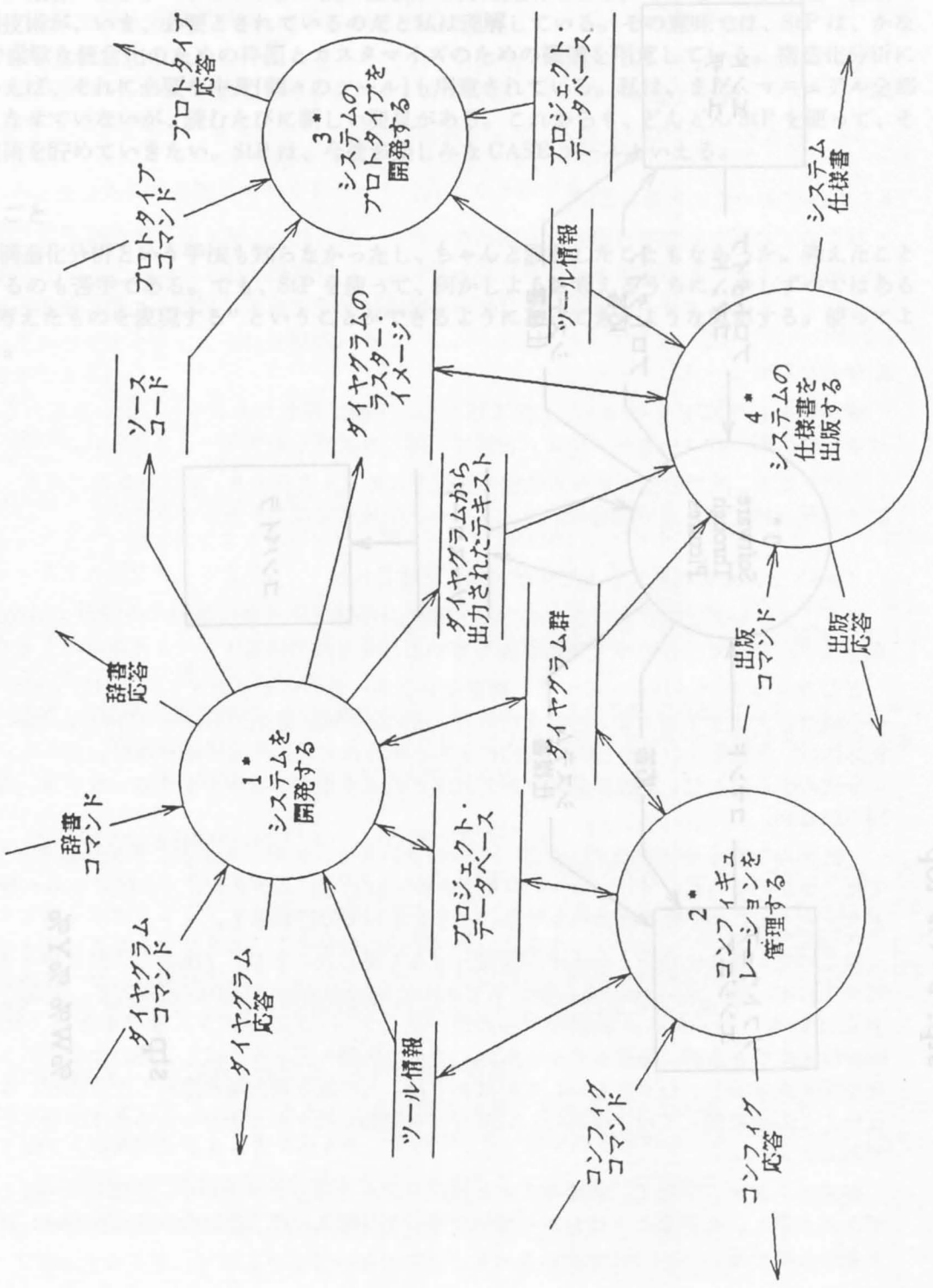


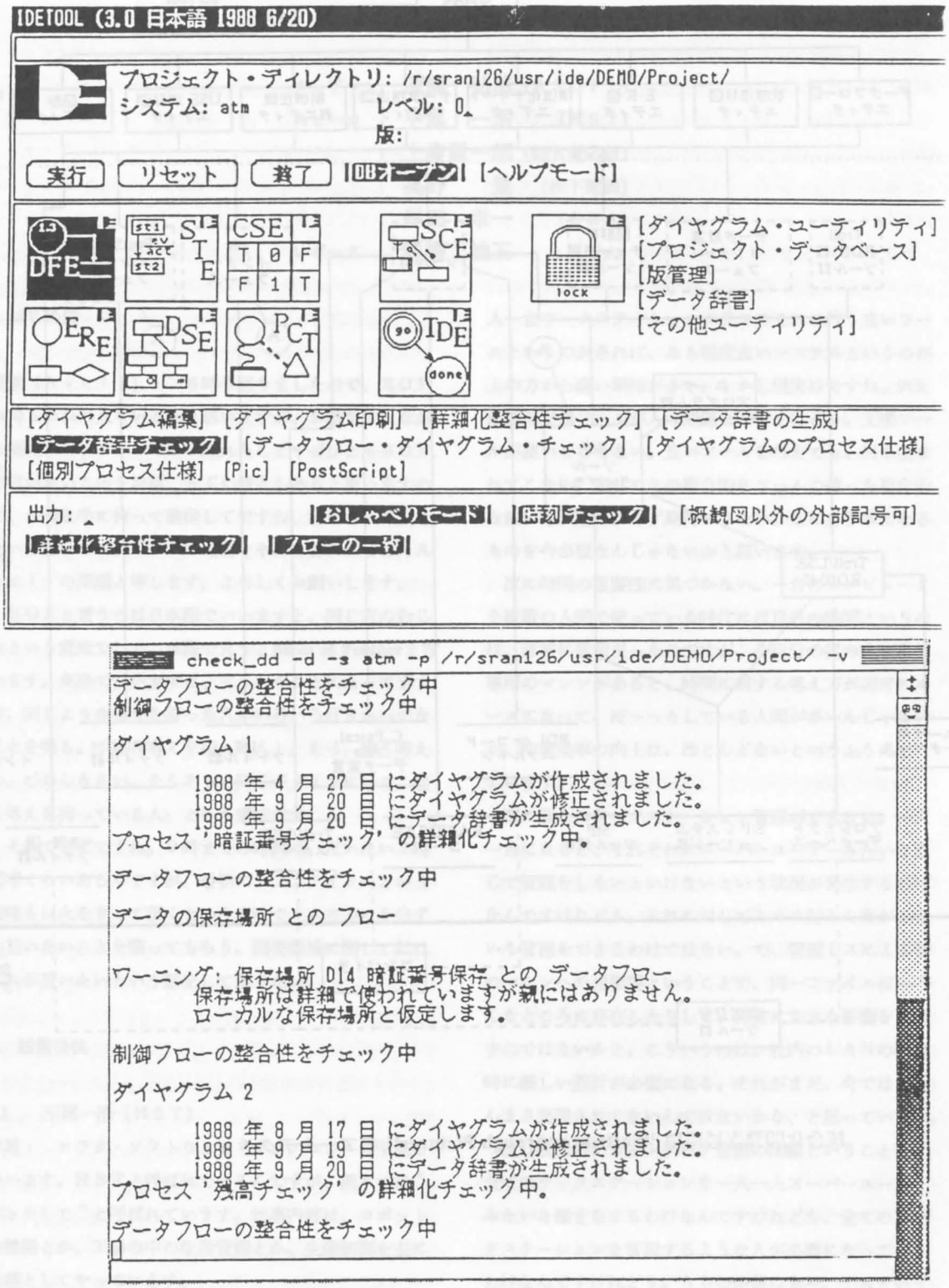
stp

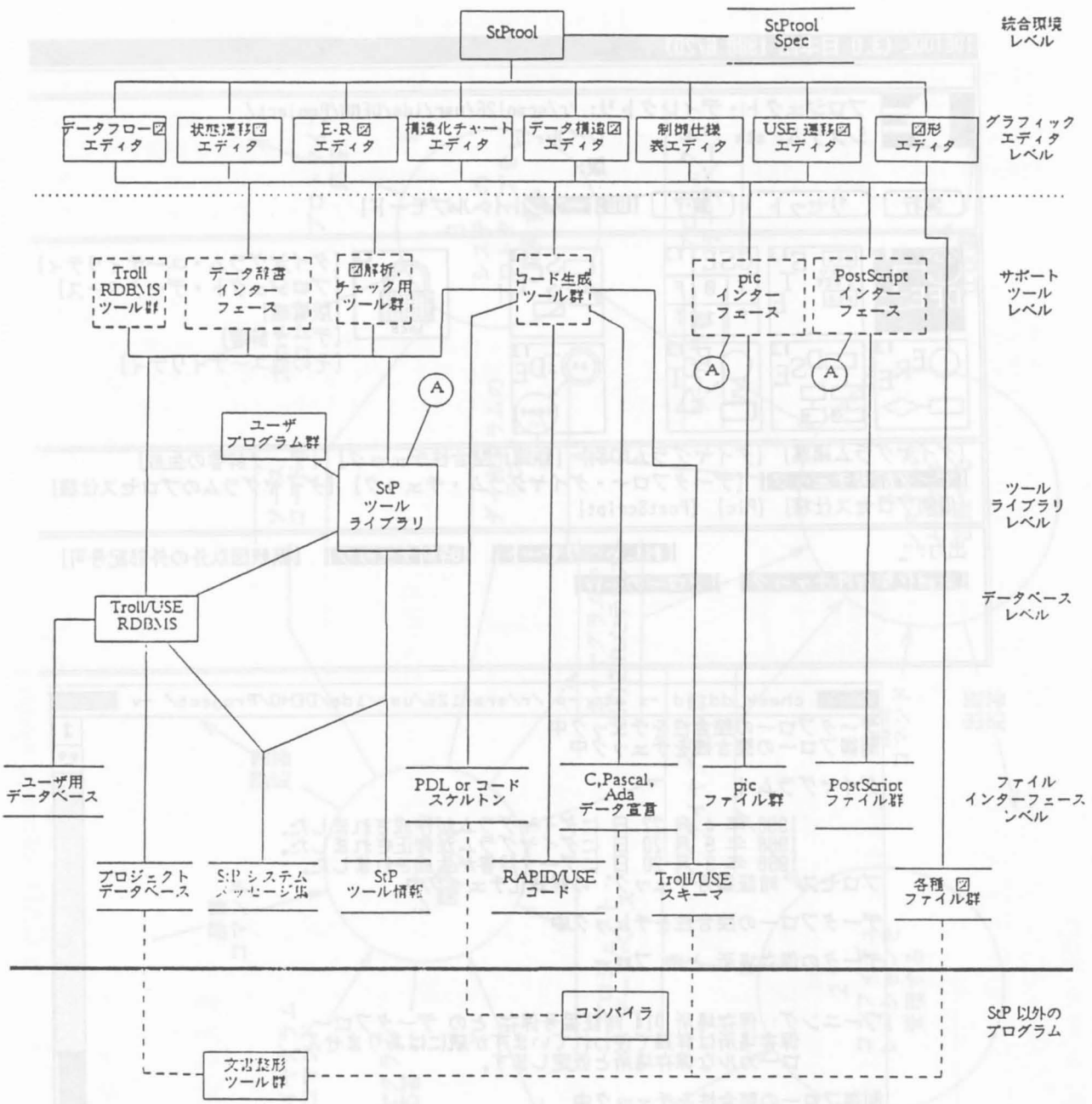
%W% %Y%

プロセス

Software_Through_Pictures (0)
a CAD software system


1*
システムを
開発する
2*
コンフィギュ
レーションを
管理する
3*
システムの
プロトタイプを
開発する
4*
システムの
仕様書を
出版する
辞書
コマンド
辞書
応答
ダイヤグラム
コマンド
ダイヤグラム
応答
ソース
コード
プロトタイア
コマンド
プロトタイア
応答
ダイヤグラムの
ラスター・
イメージ
ダイヤグラムから
出力されたテキスト
ダイヤグラム群
プロジェクト・
データベース
ツール情報
ツール情報
プロジェクト・
データベース
出版
コマンド
出版
応答
システム
仕様書
コンフィグ
コマンド
コンフィグ
応答


IDETOOL (3.0 日本語 1988 6/20)
プロジェクト・ディレクトリ: /r/sran126/usr/ide/DEMO/Project/
システム: atm
レベル: 0
版:
実行
リセット
終了
DBオープン
[ヘルプモード]
DFE
STE
CSE
T 0 F
F 1 T
SCE
lock
ERE
DSE
PCT
IDE
go
done
[ダイヤグラム・ユーティリティ]
[プロジェクト・データベース]
[版管理]
[データ辞書]
[その他ユーティリティ]
[ダイヤグラム編集] [ダイヤグラム印刷] [詳細化整合性チェック] [データ辞書の生成]
[データ辞書チェック] [データフロー・ダイヤグラム・チェック] [ダイヤグラムのプロセス仕様]
[個別プロセス仕様] [Pic] [PostScript]
出力:
おしゃべりモード
時刻チェック
[概観図以外の外部記号可]
詳細化整合性チェック
フローの一貫
check_dd -d -s atm -p /r/sran126/usr/ide/DEMO/Project/ -v
データフローの整合性をチェック中
制御フローの整合性をチェック中
ダイヤグラム 1
1988 年 4 月 27 日にダイヤグラムが作成されました。
1988 年 9 月 20 日にダイヤグラムが修正されました。
1988 年 9 月 20 日にデータ辞書が生成されました。
プロセス '暗証番号チェック' の詳細化チェック中。
データフローの整合性をチェック中
データの保存場所 との フロー
ワーニング: 保存場所 D14 暗証番号保存 との データフロー
保存場所は詳細で使われていますが親にはありません。
ローカルな保存場所と仮定します。
制御フローの整合性をチェック中
ダイヤグラム 2
1988 年 8 月 17 日にダイヤグラムが作成されました。
1988 年 9 月 19 日にダイヤグラムが修正されました。
1988 年 9 月 20 日にデータ辞書が生成されました。
プロセス '残高チェック' の詳細化チェック中。
データフローの整合性をチェック中



統合化開発支援環境 "Software through Pictures" のアーキテクチャ

第4回　ＳＥＡ環境ワークショップ

ＢＯＦセッション：　開発環境の理想と現実

チェアマン: 深瀬　弘恭（ＡＳＣＩＩ）
スピーカ　：平尾　一浩　（ＨＳＴ）
：上妻健一郎（熊本電応研）
：高野　豊　（松下電器）
：岸田　孝一　（ＳＲＡ）
レポータ　：馬場　始三　（阪大）

1. はじめに

深瀬（Ａｓｃｉｉ）：　6時半回りましたので、ＢＯＦお待ちかねの人もいるかと思いますが、お酒を飲みながら昼間のセッションでいい忘れたことやぜひともＢＯＦで言いたいというお話、たぶん皆さんあると思いますので、お酒を手に持って確保してですね、席について頂きたいと思います。ＢＯＦの司会をやれと言われましたＡｓｃｉｉの深瀬と申します。よろしくお願いします。

ＢＯＦと言うのは日本語でいいますと、同じ穴のむじなという意味でして、英語で言うと**Birds of Feather**と言います。英語で言いますと、そういうことになってますが、同じような考えを持った人がいろいろ好きかってなことを喋る。「違う考えを持った人」。えっ。違う考えね。ごめんなさい。そうそう、同じドメインにいて、違う考えを持っている人。という意味です。

それでまずですね。9時までの時間で、だいたい2時間半くらいあるんですが、最初にコメンテイターという意味も口火をきって頂くという意味で4人の方に5分ずつ言いたいことを喋ってもらう。開発環境に関して私はこれが言いたいという話をして頂きたい。

2. 話題提供

2.1　平尾一浩（ＨＳＴ）

平尾：　ヒラタ・ソフトウェア・テクノロジーの平尾といいます。ＨＳＴと呼ばれているんですが、熊本では”エッチして”と呼ばれています。仕事内容は、ロボットの整備とか、工場の中の生産管理とか、生産制御を主に仕事としてやっています。

開発環境の理想と現実ということで、まぁあの、一人一台ワークステーションの時代とよく言われてます。一人一台ワークステーションクラスのそれに伴う良いツールというのがあれば、ある程度良いシステムというのが上の方から凄い期待が多い。しかし現実はですね。例えば開発支援ツールが人の問題なんですけども。支援ツールが使いこなせない。良いツールを与えても、何も生まれてこない。ましてその複合的なツールの使った複合的な使い方が応用がまず期待できない。なにか手本になるものを今必要なんじゃないかと思います。

次に時間の重要性に気づかない。一台のコンピュータを複数の人間で使っている時代には自分の時間というのは、非常に重要だったのですが、今自分の机の上に自分専用のマシンがあると、時間に対する考え方が非常にルーズになって、ぼーっとしている人間が多いんじゃないか。作業効率の向上は、ほとんどないというふうに思っています。

次に個人のワークステーション管理ができない。一人一台になると、それぞれにスーパーユーザーみたいな感じで管理をしないといけないという状況が発生するわけなんですけれども、それに対して１００％みんながそういう管理をできるわけではない。で、管理ミスによる他のマシンへの悪影響ということで、同一ファイルはいろんなところに存在したりして、非常に大きな影響を及ぼすのではないかと。こういうのは、社内のＬＡＮの構築時に厳しい設計が必要になる。それがまだ、今では、あんまり実現されてないんではないかな、と思っています。

次は統括的にネットワーク管理の困難ということで、全てのワークステーションを一人一人スーパーユーザーみたいな働きをするわけなんですけれども、全てのワークステーションを管理するような人が必要になってくるわけなんですけれども。うちの企業はまー、十人というとこで、ちっちゃい会社なんで、まー、ある程度できるかなという気があるんですけど、大きくなると、そうい

う総合的な管理をできる人間はなかなか難しくなる。各人がどういうふうな環境を作っているかとか、なかなか把握できなくなるという問題がある。

次に、全開発フェーズを統一的にカバーするツールがない。ということで、要求分析から設計、開発、ドキュメント、テスト、メンテナンスまでを統一的な一定の基準に基づいたツールが今のところないのではないか。まー、熊本にはまったくないと、言えるのではないかと思います。

最後に基本となる開発モデルはないか。ウォーターフォールモデルとかが現実の開発行程からずれているということは、もうすごい指摘されているわけですけれども、新しい現実の開発に即したモデルというのが、まだ、出ていない。という問題があると思っています。会社の方からは社内の恥をさらしてくるな、と言われてきたのですが、ちょっと言わしてもらうと、ぶつぶつなんですが、

ぶつぶつ　プロジェクト
　　スケジュール管理欠如
　　マスタースケジュールの更新時期判断ミス
　　見積りミス
　　コミュニケーション不足

これ、プロジェクトでいうと、スケジュール管理が全くできていないというか、スケジュールミーティングとかで出すような資料ぐらいしか用意していないプロジェクトリーダーとか、まぁー、若干いますけれどね。「まぁー、どこでもあるよ。」　マスタースケジュールの更新時期に判断ミスということで、まぁいろんな要求の変更とかいろんな技術的な問題から発生する、スケジュールの変更をせざるを得ないという時期にそのマスタースケジュールの変更を時期を間違ってしまうというのがある。後は、見積りミス。というのは、凄い短い期間で難しい大きなシステムを上からむりやり作ってしまえと、言われる。これらは、見積りミスではないか。後はコミュニケーションの不足ということで、分散開発をやっていると、コミュニケーションが不足になってきて、そこらへんのコミュニケーションの不足から出てくる歪というか、そういうのが問題ではないかということです。

ぶつぶつぶつ　個人
　　観察（仕掛）
　　思い付き仕様
　　事後報告
　　同じミスの繰り返し
　　優先順序の判断
　　問題解決技術
　　相談と質問とおんぶ
　　自分の意見、考えへの固執
　　悩みの足踏み
　　変なプライド

後、これを出すと、小学校の学級会というか、そのレベルになるのではないかと思いますけども、まず、観察とか仕掛けということをまずやってる人間はほとんどいない。あれはどうするんですかという質問を受けたときにちょっと貸してと言ってワークステーションを使ってやっていると、それを見ないで外をぼぉーとして見ている新人類と。後はＵＮＩＸの基本的な使用法がまず出来ていない。ＲＣＳでバージョン管理をやりましょうねと言って、やってますかというと、はいと言うに関わらずやっていない。どうしてやってないのと聞くと、使い方を知りません。とんでもないわけです。後は思い付き仕様でありますが、これは、仕様はこうなってますけども、こうやるとお客さんはきっと喜びますよという、変な判断を持ってきて、仕様を勝手に下のレベルで変えてしまう。後は、事前事後報告とありますが、これ、報告を全くしない。同じミスの繰り返し。人のことを全然見てないんで、同じミスを繰り返す。後、ボンボンボンとありますが、こういうぶつぶつが今、実際の一人一台のワークステーション環境の中でも、こういう人に依存した部分の問題がかなりある。

現状の開発環境をとらえた場合に、今の開発環境は、どうしてもハード的なワークステーションがあって、それに伴う各種の支援ツールがある。その下に問題なのが我流の開発モデル。別に確立しているわけじゃないんですが、何年かやっていると、自分なりにこうやればうまくいくんだみたいな判断のもとに自分の知らず知らずに開発モデルを持ってしまう。ということで、これを無視して直にワークステーションで何かをやろうとしているのがあって、現在は、これは完全に無視されてこの部分が、ワークステーションとツールの部分が開発環境と呼ばれているのではないか。

人の問題を含めて、今後の人を中心に、いちばん問題を起こすのはマシンよりも人ですので、人を中心に現実に沿った開発モデル。昨日Ｈｙｐｅｒのところで出てま

したが、モデルは必要なのではないか。そのモデルを中心に人とのインタフェースは開発手順書を置く。下へはモデルに沿った統一的な支援ツールを置く。こうなると、ワークステーションは半分くらい入って一つの開発環境というのになるのではないか。というふうに思っています。

それで、いきなりまとめなのですけれども、今必要なのはさっきの絵に沿った現実に沿った開発モデル。Ｈｙｐｅｒのところでも出ていましたが、Ｈｙｐｅｒでいろんな開発自体のとらえ方が変わってくるのではないか。そういう時期にきてると思うので今、開発モデルを考える時期ではないかと思っています。それに開発モデルに沿った統一的な支援ツール、それに開発モデル支援ツールと開発手順書、これは人がするもの、というのを構築して行かなければならない。最終的に言うと、人をいれた開発環境システムの構築というのが、一番重要である。さっき、ぶつぶつぶつで出した問題を解決して行くためには、というか、うちの会社みたいに凄い少人数でやっている会社にはいろんな教育とかそういうのに使う金とか時間もない、というのが現実ですので、なんとか手順書とかモデルが構築できれば、あなたも明日から立派なメンバー、明日からすばらしいプロジェクト・リーダー。というようなものを構築して行けば、いいのではないか。とりあえず、開発手順書などというと、凄い批判を受けそう名のですけれども、ある程度の自由度のある開発モデルを作って行けば、いろんなチームとしての開発の効率を上げられるのではないか。最近思っています。ありがとうございました。

深瀬： ありがとうございました。十五分ちかく喋って頂いたのですが、せっかく用意して頂いたんで、まぁー、教育的指導はせずにですね（：－）。ありがとうございました。

次に、松下電器の高野さんにお願いしたいと思います。５分ですよ。５分。開発目標の部分、努力目標は５分で、リミット十分です。

2.2 高野　豊（松下電器）

高野：

こういうことだとは今までわからなかったものですから、急きょこの場に話を合わせようと、がんばっております。

まず第１に、私がなぜここに立っているかのかというのが未だにわからないのですけれども、まぁーこれは、一方的に中野先生が悪い、ということで片づけておきたいと思います。え、私の自己紹介ですか。そうですね、ちょっと活動の場が違うのでまず自己紹介をさせていただきますか。私、松下電器の高野と申します。日頃はＪＵＳの方の活動をしているので、ＳＥＡの方にあまり顔を見せないのですけれども、中野先生等々からいろいろ、教育的指導がありまして・・・・・・。で、これから話させて頂くという次第です。

それでですね、私どもの会社では、家電を作って稼がせていただいてるわけですけれども、最近家電といいましても、賢い家電が増えてまいりました。たぶん、皆さん方の家庭内で捜していただければ、１０個や２０個くらいのマイクロコンピュータは、すぐ見つけ出すとが出来ると思います。もし、家捜ししてみて一個も見つからなかったとしたら・・・・それは文化的な生活を営んでいないという証拠ですので（：－）・・・・・急きょこれから取り揃えようとお思いの方はぜひ当社のものをお買い上げ願いたい（：－）。え一、斯様に考えております（：－）。

それで、いろいろ問題はありまして、いまは単品ごとに賢くなっています。そのうちそれらがつながって行くのだろうということは予見できるかと思いますが、そうなってきますと分散型のＯＳが欲しい。で、使えそうなものはなにかなというふうに、いろいろ考えてみますと、まぁー、MACHぐらいかな。で、そろそろ新しいプロジェクトを起こそうかと考えているんですけれども、プロジェクトの命名をしなけりぁならん。とりあえずおかまも含むシステムだから、"おかまのMACH"（：－）（：－）・・・・・。

（　　）は、
目的を持つものが
それの使い方を知って
正しく使用するとき
効果を発揮するものである。

話題はこんなふうなものだろうなぁと思って、書いてみたＯＨＰがこれなんですけれども、括弧の中に何をいれて頂いても結構です。これ、いろんなものを当てはめてみますと、いろいろおもしろい解釈が出来るんですね。例えば、人生なんてものをいれてみますと、"人生は目的を持つものがそれの使い方を知って正しく使用すると

き、効果を発揮するものである。"というふうになります。あるいはここに、ツールとか環境をはめて頂きますと、それはそれなりに意味をもってくるのではないかと思います。

（　　）は、
目的を持つものが
それの使い方を知って
正しく使用するとき
効果を発揮するもので
なければならない。

で、これはちょっと表現を変えますと、別の見方が出来ます。"なんとかは、目的を持つものが、それの使い方を知って正しく使用するとき、効果を発揮するものでなければならない。"で、このことを頭にいれて置いて頂きたいと思います。

私の所でも最近、若いものがいろいろうろちょろやってますので、いろいろと言わなけれりゃならない立場にあるんですけれども、やっぱり仕事ってものは効率が第一です。しかしながら理想と現実はなかなか食い違いまして、どうしても、一言口をはさまなければならないことが、多々あります。で、その時におまえはアホかと、ただそれだけを言いますと、反撃を食うことになります。"それではアホな点を説明してみぃ、"と言われた時に、"お前はこの点においてアホである。"と明確に言うための、いろいろなチェックポイントというのがあります（：ー）。えー、そのために考えだしたのが、今の格言であると考えて下さい。

これをもう少し具体的に説明しますと、まぁー、こんなふうになってくるじゃないかと思います。さきに上げました3行、それを各々対応させていただければだいたいご理解頂えると思います。

1．目的　仕様
　　特性、性能、機能
2．知識
　　構造・特性・性能・機能
　　限界・歴史（必然性）
　　他との関連
　　方法論
　　物性
　　ＫｎｏｗＨｏｗ
3．能力
　　訓練・慣れ・カン
　　能力・センス
　　見通し能力

一つは、目的を持つ者が使うものである、ということから、使えるものはなんらかの目的を持って作られているはずだということが言えます。それと、仕様ですね。これは非常に重要な意味を持っています。この点で抜けがあるとツールとして、あるいは環境として問題外、という評価にもつながってくるわけです。ここで重要になりますのが、特性であるとか、性能であるとか、機能です。こう言った項目がキーワードとして上げられます。あんまり時間がないので解説は省略しまして次へ行きましょう。

2行目はユーザの知識というものに対応してくるものです。これはいろんな切口を持っていまして、一つは構造というものがあります。それから特性、性能、機能というものがあります。それで、これを知っているかいないかによって、随分、ツールとか環境というものの使い勝手が違ってくるわけですね。これらの知識無しに使おうとしても、おそらく不可能であろう、という様な局面は多々あります。それから、必ず限界というものがある。これを知ってないと、あらぬ間違った使い方をやりかねません。それから歴史というものがあります。歴史が何につながってくるかというと、必然性というようなものにつながってきます。それからもう一つ重要なのは、他との関連性ですね。そのほかに、方法論ですとか、物性ですとか、それからこのＫｎｏｗＨｏｗをどっちにいれるかで迷ったのですけれども、ＫｎｏｗＨｏｗと言ったようなものが関係してくる。従って私が見ておりまして、よく行き着くのはだいたいこの辺のところですね。"道具の使い方も知らんと、なにやっとるんや"、という言い方はかなり強く言うことが出来るという強みを持っています。

それから3番目は、いわば能力。これは非常に重要なことでして、すべからくツールと言うものは、目的を持って、かつその使い方を知って使われるべきものですから、その使う人間に依存する所というのは多々あるんですね。これが金物の世界ですと、こう言った制度と言ったものはかなり確立しております。なんらかのツールを使う人間は、必ず訓練をされている。例えば、私が工場でハンダ付けさせてくれ、といっても、絶対させてもら

えません。電器製品にとってハンダ付けというのは命ですから、"これをど素人にやらせられるか。"という面があります。ちゃんと訓練を受けた人間でなければさせてもらえないわけです。しかし訓練を受けて、それで出来るかというと、慣れという要素も必要なんです。それとカンですね。これがあるかないかでもって、道具の使い方というのは随分違って来てしまいます。これを言い出すと、どうしようもなくなっちゃうんですけれどもね。能力、センスの問題になりますから、ヘタに言いますと、最後通牒になってしまうので（：－）、まあできるだけ言わないようにしているんですけれど・・・・。しかしこの壁にぶち当たることはよくあることです。そのときはその時なりに、何か別の面でどうにかなるんではないか、ということでもって、いろいろ配慮して上げるわけなんですけれども、難しいですね。端的に、端的に言えたらどんなに楽か（：－）。そういったところにいろいろ気配りがあるものですから、最近だんだん薄くなってきました（：－）。それと、私としては、重要視しているのが、見通し能力です。これは全てに関係してくると思うんですが、"こいつ使いものになるかならないか"といったことは、この辺の能力を見ているとよくわかるんです。

まとめて言いますと、一つはツールという性質そのものが重要なんですけれども、もう一つ重要な要素として、人的な要素がありますよ、と言うことですね。今のお話と重なるところがありますけれども、そういうふうに私は考えております。

もう一つ、切口はいろいろあると思うんですけれども、こういう見方も出来るのではないか、ということで、別にまとめてみたのが、このＯＨＰです。括弧の中は、また勝手に単語を放り込んで頂いて結構です。たとえば環境とでもしてみましょうか。

（　　）は、
1．歴史の中で、分化・統合
2．適用範囲
3．適用範囲を無視
4．万能でなく
5．メインテナンス
6．広範な情報を持って
　　使用されるべきである。

するとこれらについて言えることの一つは、歴史の中でつながりを持っているものである、ということ。かつ、その中でもっていろいろ分化し、あるいは統合化され、で、枯れて行くものである、という性質があります。

それから２番目として、必ず適用範囲がある。しかしながらその適用範囲を無視したとき、おもしろい使い方が出来るものでもある（：－）。ただし、万能ではない。

そして常にメンテナンスして行かないと、そのうちに腐ってしまう。ソフト屋さんが使っているＵＮＩＸなんてツールは、目を離しますと、すぐ腐っていっちゃうんですね。死臭ふんぷんというＵＮＩＸがあちこちで動いている、というな現状がある（：－）。

それと、その単体の性質とかそう言った純粋なものではなくて、非常に広範囲な情報をともなって、使用されるものである。まぁー、こういうふうに解釈することが出来るのではないか、と思います。

ちょっと時間の方があまりないようですので、端折りましたけれども、キーワードをご紹介したということで、一応終わらして頂きます。

深瀬：　はい、ありがとうございました。次に、電応研の上妻さん、お願いします。

2.3　上妻健一郎（熊本電応研）

上妻：　電応研の上妻です。先ほどと違って、今度はユーザの立場からものを言わせてもらいます。今回のために特別に作ったＯＨＰというのが、ちょっとありませんで、申し訳ないんですけれども。

うちの電応研が何をやっているかというと、研究をやっているということはなんとなく、わかっていただけるかと思うんですけれども。だいたい、熊本県と一部の企業からの出資金によって出来ている第３セクターの財団法人の中の下部組織に当たるわけですが、電子応用機械技術研究所というもっともらしい名前がついてますんで、一応コンピュータを使ってなんとなくいろいろコントロールしてみようかということで。

私がやってますのは、ロボットの制御ですね。特に、うちの場合、要素技術として、３次元視覚センサーというものを作ってますんで、それを応用した形で、何かもっともらしいロボット言語を作れないかなということで、やってます。その中で、今のところまず、ロボットを動かす前に、シミュレータを作ろうよということで、立石さんのシグマワークステーションを使いまして、こういう非常に粗末なですね、スケルトンモデルを作ってみま

した（：－）。

これ、結局私がコーディングする時間がなかったものですから、言い訳をいっときますけど、研修生制度というものがありまして、高卒の人を２年間ほど預かって、その中で座学をやったりＯＪＴをやって、その中の成果物として卒業研究を残すということで、組んでもらったのがそれです。そこで、問題になってきますのが、結局これだけしか作らなかった研修生が今度の３月で卒業して行ってしまうわけです。どこでもそういう研究所とか、学校関係というのは、一緒だと思うんですけれども、だいたい非常に歩みが遅いわけですね。研究の。例えば、これ、シグマＯＳと言うことで、ＵＮＩＸなのですが、研究にＵＮＩＸを使いますというと、まず、頭にｃｓｈｅｌｌを勉強しましょう。ｖｉを勉強しましょう。ｍａｋｅを勉強しましょう。ということで、例えば大学の卒研なんかですと、もうそれで夏休みに入ってしまう。それからですね。おもむろにプログラムを始めて、で、卒論をまとめることになる。正味コーディングしている時間と言うのは２カ月か３カ月と言うことで、しかもこれ卒業して行ってしまうとなると、また新しい人が入ってくるわけですね。毎年毎年１年間何か一生懸命やっているはずなのに、そのぶんしか進まない。まぁー、これを恥ずかしかったのですが、お見せしたというところです。

ＵＮＩＸの開発環境においてもうちょっと、今かなりましになって来つつありますけれども、もっとＭａｃとは言いませんけれども、キーボードからのユーザインタフェースにしろ、なんらかの形でマニュアルレスに近い形で扱えるユーザインタフェースが必要なのではないかなと今、頭の中でぐるぐる回っているところです。

で、開発環境と言うことで、電応研の中のネットワークをご説明して置きますと、ＶＡＸの７８５とμＶＡＸ－２が３年前に入ってまして、その下にキャラクターミナルがこのぐらい、グラフィックターミナルが３台ですね。漢字の端末が１３台、キャラクタ端末が３台。後、これ全部花子で作ってあるんで、絵が全部ＰＣになってますけど、形はＰＣですけど、一応これはＭａｃ２ですね（：－）。すみません。私の同僚のものが、９８の花子で作ったものですので、こういう形になっておりますが、すべて違うんです。Ｍａｃ２、それから、シグマＯＳ、オムロンのシグマワークステーションＳＸ－９１００が２台、それから東芝のＳＵＮ、ＡＳ３０００が２台。後、ＤＥＣｎｅｔ－ＤＯＳパソコンが３台入っています。それからもう１台がＰＣ－ＮＦＳが１台入っております。

今までＰＣで作っていたものとかを、ワークステーション上へ乗せたいとみたいなことが起こったときに、非常にもう単純にですね、送れてしまう。しかし、漢字の変換などは、いろいろワークステーションとかを捜して、ユーティリティを捜してやるかというと、ＶＡＸのほうがＶＭＳで走ってますので、ＤＥＣｎｅｔ－ＤＯＳとやりとりするときは、ＤＥＣＮＥＴでやるわけなんですけれども。そのときにシフトＪＩＳをＤＥＣ漢字（≠ＥＵＣ）に変換しまして、さらにＶＭＳの上のＴＣＰ／ＩＰでＥＷＳへ転送するということをやっていまして、それでなんとなく事なきを得ています。転送程度だったらこの程度でいいんじゃないか、ということでやっています。

そこで、一つ問題があるのは、まだＮＦＳが導入されていないということで、ＰＣ－ＮＦＳが導入されているといいましたけれども、実際にはＶＡＸ／ＶＭＳ上の問題でして、というか、まだ、バージョンが４．Ｘ代ですので、ＮＦＳがサポートされておりませんので、今のところ、ＴＣＰ／ＩＰを市販のソフトを使ってやっている。ということで、転送程度しかできていないというのが現状です。ＮＦＳそのものをやりますと、今度、また学生に対してですね、ＮＦＳの使い方まで教えなければならない。というところがあって、結局そう言ったこと事態も勉強にはなるのですけれども、実際のユーザとして使わせる場合には、もっと管理までやらせるのではなくて、ほんとにユーザとして本質的なところを勉強してほしいなというところがあります。そういうところが問題になっているのですけれども、実際ですから、例えばこの２台を使って開発をしたわけですが、例えば、研修生がこちらでプログラムを作っていたとして、私がたまたまこちらへも入っていたとしたら、彼はそっちに行って、リモートログインをして、ディレクトリィをいじっているわけですけれども、また、ＣＰＵパワーが欲しいということで、こっちにファイルを持ってきてやるとかですね、そうゆうようなことをやるわけです。それで、もう少し、ビジブルに見えれば、教育そのものも短くできるし、そういったネットワークの良さも生かせるのじゃないかなというふうに考えているところです。

それとですね。メーカの人にちょっと言いたいのですけれども、どうも日本の学校とか研究所関係にですね、やっぱり、もっとどう言うか、応援して欲しいというかですね。例えば、私が卒研をやったときはですね、どう言う状況だったかというと、ＰＣ－９８のＮ８８ＢＡＳＩＣでＦＥＭによる熱応力解析のようなものをぐちゃぐ

ちゃと。よその大学の研究でＦＥＭのプログラムがあったので、それがＦＯＲＴＲＡＮだったのですけれども、ＦＡＣＯＭのＦＯＲＴＲＡＮにそれをのっけるのですけれども、その前にそのプログラムの改造を行なわなければならないということで、改造を行なうわけですが、そのときに使うメインフレームの使用料金が研究費としてないと。それじゃしょうがないからＢＡＳＩＣに書き換えてから必要な部分をつける。そして動作を確認してから配列の規模を大きくして、メインフレームのプログラムに書き換えて、メインフレーム上で短時間実行する。絶対エラーがないようなコードにして、実行するという無理なことを押しつけられまして。それで半年が過ぎまして、その次の半年は何をしていたかというと、その中の微分方程式を解きまして。いまだと、ＲＥＤＵＣＥとか数式処理パッケージみたいなものが揃ってるんで、まぁ、瞬時とは言わないまでも、１日あれば、出来てしまうわけですが。一応かなり２カ月３カ月ばかりかけて、本質的じゃないところをやっているわけですよね。私が知っている高専の先生の研究室なんかでも、ＳＵＮを購入したはいいんですけれども、それ以外ユーティリティを買うお金がなくて、立石に泣きついて・・・

深瀬：　科研費の話はこの場での話題ではないので、・・・

上妻：　すみません。

深瀬：　あと２分でまとめて頂きたい。

上妻：あのＯＳ／９で今まで開発していたのを、ターゲットだけをＯＳ／９にしてＵＮＩＸ上で開発したいということで、ＳＵＮを買ったにも関わらず、ＳＵＮ上のクロスツールコンパイラを買うことが出来ないで、立石に泣きついて立石のシグマワークステーションとその上のクロスコンパイラを買って、それを学生さんに一年掛けてＯＳ／９用にモディファイするというような、ことをやっているわけです。ですからあの、アメリカの大学にばかり、投資しないで、日本のそうゆう研究所とか大学とかにもっと、投資してもいいんじゃないかということですね。失礼しました。

深瀬：あ、終わりですか。・・・はい、どうもありがとうございました。えー、最後にＳＲＡの岸田さんから、十分でお願いします。

2.4　岸田孝一＋栗原正利（ＳＲＡ）

岸田（ＳＲＡ）：　私はＳＲＡの岸田です。現実から最近遊離しておりますので、現実はあそこでＯＨＰを書いておりますけれども、私は理想を話して、現実パート１、パート２どこへいったのかな。現実は他の人に任せて、理想だけ喋ります。

“Ｍｉｎｄ　Ｓｅｔ”の問題
- ＵｎｉｘＣｕｌｔｕｒｅは異常か？
- ＭａｉｎＦｒａｍｅはやっぱり病気だ！
- Ｈａｃｋｅｒたちはなぜ
  ＳｏｆｔｗａｒｅＥｎｇｉｎｅｅｒｉｎｇ
  が
  きらいなのか？
- すぐれたＳｏｆｔｗａｒｅはなぜ
  Ｆｒｅｅなのか？
- いいものは、やっぱりあちこち
  自分で探し回らないと
  見つからない。

それで、さっきのＭＭＩのセッションを聞いていて、ちょっといいたいなと思ったんですけれども、どうせ後で喋るんだからして、言い残したことを喋りますと。開発環境に関しましては、気の持ちよう、Ｍｉｎｄ　Ｓｅｔというのは気の持ちようのことですけれども。ということが非常に大きいのではないかという気がしました。ということは、ここに今、Ｕｎｉｘのトレーナーを着ておりますが、ＵｎｉｘのＣｕｌｔｕｒｅというのに過去、自分とこにＵｎｉｘが入ってから８年、その前からＵｎｉｘの世界へ入っていた人なんて１０年近くＵｎｉｘも何かと親しんでおりますけれども、そちらの方がやっぱり正常であるのかなと。異常ではなくて、正常じゃないかなという気がします。

で、私はＳＥＡとＪＵＳと両方入ってますけれども、やっぱりＳＥＡはちょっと環境の問題なんかを議論するとき、ちょっとおかしいかな（：－）。おかしい理由はやはりメインフレームは病気だからだというふうに思います。それで病気の原因はいろいろあると思うんですけれども、基本的には、メインフレームを作っている人も売っている人も使っている人もやっぱり、心が病んでるような気がします。だからといって、Ｕｎｉｘ　Ｃｕｌｔｕｒｅがそう、万々歳ではなくて、私はもともと、年が年ですから、Ｈａｃｋｅｒじゃなくて、Ｓｏｆｔｗａ

ｒｅ　Ｅｎｇｉｎｅｅｒなわけですけれども。私の会社でもそうですし、うちの会社以外のところのＵｎｉｘの環境でもそうですけれども、いわゆるＨａｃｋｅｒと呼ばれる人たちにＳｏｆｔｗａｒｅ工学の話をすると、フンと言って、どっかへ行ってしまう（：－）。これはやっぱりいけないと思いまして、そうゆう意味で、Ｕｎｉｘ　Ｃｕｌｔｕｒｅは異常だと私は思うわけであります。

それで、これ絶対関係ないんですが、値段つけて売っているソフトウェアというのは、絶対良くない。ただの奴がいちばんいいというのはどうして（拍手）。これはかなり本質的なことだと思うんですよね。そのことにやっぱり、Ｕｎｉｘ屋さんはだいたい気づいていると思いますが、メインフレームを使ったり売ったりしている人たちは、やっぱり、いいものは高い値段を出さないと買えないという頑固な迷信があって、それはおかしい。

それで、なぜ病気かという話の一つは、開発環境とかソフトウェアに関するテクノロジー全部そうですけれども、技術は決して凍結されているわけではなくて、いろいろ動いているわけで、必ずしも正しい方向に進歩しているとは限りませんけれども、新しいものはどんどん出てきている。そういうものがやっぱり自分で、世の中をきょろきょろ見回して、捜さないと手に入らないということは基本的な原則ではないかと。メインフレームのユーザの方々に特に言いたいのですけれども、メインフレームのユーザの方々の基本的なＭｉｎｄ　Ｓｅｔというのは、ハードウェアもソフトウェアもメインフレーマーの人がいろいろ世話をしてくれて全部こちらへ持ってきてくれるもんだという固定観念があるような気がするんですが、そんなことは決してなくて、メインフレームを作っている人たちもあんまり世の中のいい技術を知らないで、古い技術で、たぶん５年か１０年遅れている技術でハードウェアを作っているわけで、ソフトウェアもそうですけれども。そうゆう人たちに頼っちゃいけないんで、自分に一番いいものをすぐ見つけてくるという精神に切り替えないと、ＭＭＬなんてのは絶対うまくいかないというふうに、私は思います。

そこで、環境という話しにからんで、この絵は非常にありふれた絵で、ありふれた絵とゆうのは、Technology Transfer を議論する人たちにとっては、ありふれた絵という意味で、Technology Transferというのは世の中にこの世の中と言うのは必ずしも強制ではなくて、まぁあちこちに偏って分布していると。例えばソフトウェアの新しい技術みたいなものを、あと一分ね。世の中で何か新しい技術を作っているところを見つけてそこから持ってこなければいけない。それで、持ってくるためには何が必要かというと、そういう新しい技術を作っているところで、お節介な人がいて、自分が作ったものはいいからぜひ使えという人がいるわけです。まぁー、これは必ずいるわけです。しこしこ自分でまじめに研究だけしてればいいのに、研究の成果をやたらと人に押しつける人が必ずいるわけで、そういう人を見つけてですね。で、かつ自分の会社を取り上げると、自分の会社の中で、世の中にそういう面白いものはないかなときょろきょろ見回している人が必ずいるわけで、二人がペアになって、こういう橋を作って持ってこなければならない。そういうのを受ける側の機能から言うと、”Ｇａｔｅ　Ｋｅｅｐｅｒ”、”Ｇａｔｅ　Ｋｅｅｐｉｎｇ　Ｆｕｎｃｔｉｏｎ”というのですけれども、開発環境なんかを整備して社内で使わせる人たちというのは、てことは、ここに集まっている人たちですけれども。で、必要なことは、Ｇａｔｅ　Ｋｅｅｐｅｒとしての自分の能力をいかに高めるか。それは、世の中を見回してこういうのを他に面白いことをやっている島はどこにあるか。その島の中で、自分が研究したものを早く世の中に広めたいと思っているお節介な人間がどうゆう人がいるかということを、早く見つけて、その人とコンタクトを取って、両者の間にコミュニケーション・チャネルを結んで、なおかつ、自分の属している組織の中にそれを広めるということをすることが大事ではないかなというふうに思います。で、私の理想とする環境は、さっきいったことと矛盾しない話ですけれども、非常に矛盾した言葉ですが、Model Hacking とかSoftware Engineering Hacking とか、単にそのＣのコーディングをハックするのではなくて、開発プロセスモデルをハックする。データフローダイヤグラムをハックする。そうゆうそのもうちょっと抽象的なアブストラクトなハッカーのための環境というのが、私の理想の環境でして、そう結うものをぜひ作りたいと思っているのですが、現実には絶対そうゆうふうになっていないという話を現実に語る（：－）。

栗原（ＳＲＡ）：　後半、たぶんＳＲＡの・・・

深瀬：　自己紹介をお願いします。

栗原：　えー、ＳＲＡの栗原です。これ、「まさとし」と読みまして、間違っても「せい」と読まないように（：－）。でもぼくは整理整頓が得意なんです（：－）。

昨日も言いましたように、私たちは休みにきたんですけれども、すぐに何かを押しつける人がいて、ＳＥＡの岸田が急きょＳＲＡの岸田さんに変身したりして、こういうふうに出てきてしまうわけですね。

さて、そんなこと言ったって、やっぱり仕事抱えてたらできないよ、という話をしても全然つまらないから、私が私なりにまとめてみました。はっきり言って面白くない結論になりましたが。

精神論を述べて終わりにしたいと思います（：－）。

必要は発明の母という言葉がありましたが、問題は解決するために存在しているのであって、解決する。ただそうすればいいだけです。しかもそれは、人間にとって解決したということは喜びになるわけですね。これは人間が生きてて良かったなぁ、という喜びがありますから、ぜひともやればいい。問題はあればあるほどいいわけです（：－）。どうやってやるかというと、やっぱり自分で喜びを味わいたいわけです。自分でやらなくちゃいけない。で、岸田が言いましたが、欲しいものは自分で手にいれる。手にいれるには２つ、あ、３つあるかな。１つは、全部自分で作る。人はいっさいあてにしない。極端に全部人から買う。全て金で解決する。で、中間は今の岸田の話しで金をかけずにやるよと言っている人からもってきて、その努力だけを自分で払う。で、そのどれを選択するかは皆さんの判断であって、それによっていろいろこうやっていくというふうに思っています。以上です（：－）。

2.5　深瀬弘恭（ＡＳＣＩＩ）

深瀬：　はい、ありがとうございました。えー、ＢＯＦの口火を切って頂く意味で、４人プラスお一人方にお話をうかがったんですが。プログラム委員長の方から何か喋れと言われていますので、私も何か喋んなきゃいけないようなので、５分喋りたいと思います。

私、あの、Ｕｎｉｘのエンジニアというかマネージャというか、火付け役というか６年ぐらいやっているのですけれども。その前はメインフレームで仕事をしてまして、メインフレームも長いんで、メインフレームの方々とのお話は身に染みて良くわかるんですが。とにかく開発環境というお話になりますと、運用のシステムと非常に似かよった考え方、問題、建前があるんですね。ＵｎｉｘとかワークステーションとかＭａｃを使っている方々は、できあがっている環境自体がかなり使いやすい環境なので、ターゲットシステムとものを作る環境と言うのは非常に似てる環境でやっているんですね。先ほどのＭＭＬで理想的なＭＭＬの話をされた坂下さんの場合と非常に似かよった環境で仕事をしているわけですね。

ところがその、ある意味でお客さんの注文に応じてテイラーズソフトウェアを作っている方々、メインフレームのＴＰシステムがたぶん、典型的な例だと思うんですけれども、運用環境と開発環境、似かよって作ってしまうと使いづらいと言うような話があって、私は今までの経験からいって、その、メーカの側も非常に勉強不足。メインフレームの方々ね。それから使う側も勉強不足だ。最近非常に強く感じるようになってきています。

これはその、ある会合で、情報処理サービス産業協会という、会合の一つでお会いしたソフトウェア会社の社長さんの話を聞いていて、がく然としたのですけれども、肉体労働者に近いんですよね。ソフトウェアというのは知的労働であるはずなのに、ブルーワーカーのメンタリティで仕事をしている方が非常に多くて、その方がたが開発環境を考えたところでいい環境が出来るわけがない。で、ショックを受けたわけですね。日本のソフトウェアの開発環境はアメリカと比べてどうかという話はすぐ出ると思うんですけれども。で、あるいは知的所有権の話とか、いま問題になっている話はあるんですけども。日本のプロフェッショナルと自ら思っている方々の業界、ドメインですね。ドメインのメンタリティというのかな。かなり、その、なんていうのかな。知性とはほど遠いところで仕事をされているというのが。例えばＭＭＬの話が出たときにどうなるか。ソフトウェアの環境のモデルを考えているときにどうなるかとか。典型的なパターンとして現われているというのが、私は残念なんですよね。それで、それについてがく然としたのは僕だけなのか、あるいは、皆さんそう思ってらっしゃるのか、私は聞きたいと思っています。

えー、私の話はこういうことなんで、５分ということですので。会場から言いたいことがあれば、座っている方、どんどん出て頂いてですね。いまどう考えているかというのを話して頂きたいと思います。

不特定多数：　最後のことをもう一回言ってください。

深瀬：　私はねー。がく然とした印象がありまして、このドメインの人たち、どういうふうに仕事をしているかというのをね、情報処理サービス産業協会というドメインがあるでしょ、それから電子協ドメインがある。で、その人たちが・・・。

具体的にはですね。埋立地、東京の１３号埋立地にソフト村を作るという話があって、そこに集まってきた人たちがもちろん、自社の社長さんとか取締役の方々なんですね。当然その、ソフトウェア会社では自分では資本力がないから、開発環境を整えたいといった場合に、通産省が音頭をとって援助しますよ。そういった場合に、通産省の課長さんレベルのいうことにカウンターを通そうとしても出せないというような業界があるんだろうか。ようするに自分たちがやりたいことというのがはっきり決まっていない。その業界の人たちが開発環境を云々するというようなことはできるんだろうかという、非常にそのがく然とした覚えがあるんです。うちはもちろんその通産省のお世話にならなくとも仕事は出来ますし、そういうように人は育ってますけれども、でも、そこで出てきて、最初にプレゼンテーションされたあるソフトウェア会社の社長さんの話を聞いていて、私はがく然としたんですね。で、そういうことなんですけどね。皆さんはどうお考えかと。

3 討論

池谷（富士通静岡エンジニアリング）： いいですか。発言させてもらいます。先ほどのメインフレーム事情のところでメインフレームの話を一生懸命したのですけど、やはり岸田さんのおっしゃる通り我々は病気にかかっている、というふうな自意識持ってます。それはなぜかというと、やはり、開発環境という面からみるとＵＮＩＸは全くだめだし、現実に例えばあのー、富士通のユーザでありますコダック社。ここはひとまずＵＴＳもしくはＵＮＩＸでもってシステムを統一していこう。開発環境、ＶＰ、そういうところをひっくるめてとにかくＵＮＩＸでいきましょう。という会社やユーザがたくさんいるわけですね。

その面からみると、我々富士通いわゆる現場の人間は、非常にブアーな開発環境で仕事しています。それも、１０年来古典的な手法で、いわゆるウォーターフォールの方法でもってやっている。で、岸田さんにちょっと聞きたいんですけれども、ハイパーのときに、マネージメントの話をしましたね。どうゆう尺度でマネージメントをしたら良いか。という話を申されたんですけど、かなり私どもとしては、開発環境が変わって新しい環境が出来たときに、そういったマネージメントの尺度はどうあるべきかというのを岸田さんの考え方でどういった尺度でもって考えておられるか、ちょっと聞いてみたいな。ま、少なくともメインフレームの人間というのは病気だということは絶対確かにぜひ通りのあることで、岸田さん宜しく。

岸田： えーと、マネージメントという話はですね。私の、なんていったらいいんだろな。普通の意味でのマネージメントというふうに考えると。例えば、開発者と管理者というのが別々にいて、管理者が仕事を開発者に発注して、開発者がどれくらいいったかを管理者がみてるというのは、まぁー、普通の意味でのマネージメントであろうと思いますけれども。私が思うマネージメントと言うのはそうじゃなくて、開発者サイドが自分自身で一体自分はどこまで仕事をしたんだろうかなということをやっぱり、とらえられないとやはり、本当に開発していることにならないと思う。趣味で何かやっているんだったら適当にやっていていいわけだけれども、仕事としてやっている以上は。

例えば、私は昔小説を書いたりしていたことがあるんですけれども、雑文を書く場合にも例えば原稿の依頼がくるとですね。４００字詰め原稿２０枚であれば、締切がきてから書き始めて翌日には渡せるというふうな、自分のメジャーではかって仕事をしないと、むやみに仕事を受けられないですね。そういう意味で、自分自身のパフォーマンスに対するメジャーというのはプロフェッショナルは必ず持っているんです。野球の選手だって、自分はまぁー１０本に３本ヒット打てるという感じで打つわけですよね。あのピッチャーだったら絶対１試合に１本ホームランが打てると。そういう自己管理というか、自分自身のことをわかったものの見方というのが必ず必要で、そうすると、ハイパーだとか、なんでもいいですよ、何か新しいあるパラダイムがあったときに、開発者が自分自身のパフォーマンスをどうやって自分で変えるかということが多分まず大事で、それがないといけないんじゃないかな。管理者の側からしたらば、管理者というのはやはり、プロジェクトってのは自分自身とくっついているから、プロジェクトは一体どういった格好で進行しているかということを管理者自身は把握する必要があるわけで、単にボーナスの査定とか、エンジニアの査定という意味ではなくて、自分自身の仕事の下をプロジェクトの進行状況というのを自分なりに把握するメジャーというのがなければ、管理はできない。

池谷： まぁー、ウォーターフォールの場合には、何人

月というような尺度ではかっていると。それで、ハイパーの場合には、ハイパーカードの場合には、一つの目安として、例えばオブジェクトの数だとか、あるいはレイヤの深さだとかそういったものの表面があって、それで新しい開発のメジャーになると、いうふうな話を岸田さんから聞けると思ったのですけれども、まだそこまでは断定的に・・・。

岸田： 私はそういうふうにべったりしたメジャーじゃない。例えばハイパーだとすると、例えばツリー書いてネットワークのノードができますよね。多分、このシステムだとすると非常に重要なノードというのは何個だと。そのノードに関する情報が全部集まったかどうかということは、僕がもしマネージメントするとしたら一番のメジャーです。それ以外のろくでもないノードはどうでもよい、というふうに考える。

池谷： ということは、基本的なノードの数と最終的なオブジェクトの数、そういう見積によって、実際のかかるコスト、かかる規模、といったものをメジャーできるんじゃないかというご意見ですか。

岸田： 個数というのはあまり関心がなくて、納期もあまり関心がない。納期なんてのは、お客とネゴシエーションできる数ですから、やっぱり出来上がったものはどのくらい良いかという話へいって、良いか悪いかということはやっぱり、定性的にしか判断できないわけで、それ以外に定性的に判断を下す前のまずラフなメジャーデータとしての定常的なメジャーをまずセットするわけです。自分でね。このことは終わった、このことは終わったんだということをものがいくつかあって、後は終わったとしたら、その中身を自分なりにいれてみる。

池谷： わかりました。どうもあっちゃこっちゃ言って申し訳ありませんでした。

新田（ＳＲＡ）： 今のことに関連してお答えしますというか、岸田さんの答えが間違っていたので（：－）、私が指摘します（：－）。

あの、よく聞かれるんですけれども、ハイパーカードとかよく聞かれんですけれど、よく聞かれるというのは、私、プロトタイピングとか割と早い時期から言い出したので良く聞かれるのですけれども。プロトタイピングとかハイパー系のマネージメントというか、もっと言えば、どういうことで今何パーセントまできてるかということを把握すれば良いのということを時々聞かれることがあるのですけれども。それに対して今正確にお答えしておきたいと思います。

よーするに、何が何個出来たからという、バイト数とかノード数、リンクの個数で物の出来加減を判断するというのは全く間違っています。作り方によって全く違うからです。それで、物を作って欲しいというからには何かこういう機能が欲しいとか、あるいはこの検査に合格したからこれは出来ているよ、というのは必ずあるはずですね。そのソフトウェアを発注した側にも受取検査というものがあるはずだし、それから、作る側にもこれが出来たから完成したのですというCount Trans Arguments?ですね。この項目の個数によって、どれぐらい出来ているかという判断をする場合ですね。百個検査項目があれば、何個の検査項目に対してタッチしているかというので、今何パーセントで来ているかというのを判断するればいいと思います。

もっと細かくいうと、例えばこのソフトウェアに対して何個検査項目があるかという最初の個数は、えいやっというか、これは何ステップだよというのを決めてたのと同じように、経験的に例えば百個なら百個というふうに決めれば良いわけです。で、最初の数は決めます。それから具体的に、こういう検査項目を検査しようと決った時点で既に３分の１は出来ていると考えてください。それから、その検査項目が実際に検査できるレベルに達していると考えたところで３分の２が出来ている。それから、その検査に合格したというところでその項目は百パーセント出来ていると考えてください。後はその検査項目に関して重み付けをして、これが出来たら１だとか、これが出来たら１０だとか。後、全体の総和で割れば現在のステータスというのが定性的に、じゃなくて、定量的に把握できると思います。これで、ハイパー系のメソッドを使うプロトタイピングで仕事しようとしている人はそういうふうにして、自分が今どこまで来ているかというのを判断してもらえれば良いと思います。

杉田（ＳＲＡ）： 新田君に反論するのですけれども、別に肩を持っているわけじゃないのですけれども。

一言言いますと、私はやっぱり要求、検査という立場でみると、内部検査と外部検査というふたつの面があると思います。で、今新田君が言ったことは、主に外部検査の方じゃないかと思うんです。ようするにユーザと出来上がった製品を評価する場合にどれだけのものが出来上がっているか、というそれはスペックレベルの問題だ

と。その場合、我々というのはあくまでもソフトウェアを開発するという立場で考えていくと、・・・（テープ区切れ）・・・と僕は思う。

篠田（東工大）：　明るい公務員の東工大の篠田です（：－）。

深瀬：　ＯＨＰ使う？

篠田：　いや、使いません。あまり人の背中に向かって話をするのに慣れてないんで（：－）。

結構たくさんの方、特に背広、ネクタイ組の方が帰られたのでそれが残念に思うことを一言申し上げておきます。こことこういう機会があった時には、自分の上司に、これはぶっちゃけた話が聞けるところだからぜひ参加した方がいいということを一言お伝え願います。

問題の理想の開発環境のことですけれども。新しい方法論というのはどうやってメジャーしたらいいのかわからないというのは本当で、それは使っているうちにわかっていく。で、アカデミックなウェアから言えばステップ数の場合は、考えて欲しくないというのが本当なんですね。既存のリニアなメジャーで。例えば、うちの学部の学生とマスターの学生が同時に同じ会社に就職して、ほんの少しの給料の差。２年間よけいに勉強して１万６千円くらいしか違わない。それなのにこう、同じプログラムを書いて、マスターの学生はいろんなアルゴリズムを勉強しますね。方法論を勉強します。すごい、こう早いアルゴリズムを作った。それで、インプリメントした。あるモジュールを。それをですね。学部の学生の方は、まぁ、適当にやって動けばいいや、デバッグしてポンポンポンポンと。こんなシステムってプログラムをいったときにその二人の評価は同じになってしまうわけですね。ステップ数で計っちゃうと。だから、リニアなメジャーというのは避けて欲しい。で、それをやるのがたぶんプロジェクトリーダーさんみたいな管理職と言われる人のやることだと思うんですよ。

僕もそうですけど、うちの先生があの、ちょっと関西弁になってしまいましたけれど、その先生が言われるにですね、そうだと思うのは、ソフトウェア技術者の地位が低すぎるということです。低すぎる。全然もう、低い。もう本当、さっきあの深瀬さんが言われたように肉体労働者、ブルーワーカー的労働者。もっと先ほど言われた、えっと、ちょっとＯＨＰ見せてくれますか。えっと。ＨＳＴの方の。

あの、いろいろな環境を揃えればプログラムができるかというとそうではなくて、皆さん指摘されましたけども。そうですね。結局ＯＨＰ使うことになってしまいましたね。

えっと。まずですね。開発支援ツールが使いこなせない。ようするに、メソッドを知らない。方法論を知らない。時間の重要性に気づかないというのはこれはもう、人間としては最低ですね。これはあの、採用の時に間違えたとしか言えようがない。少なくとも面白そうな物が目の前にあって、キーボードを叩いてない人間というのは、まぁ、女の子が側にいれば別ですけれども、ちょっと問題があると。個人のワークステーション管理が出来ないというのは多分、個人の管理もできないですね。自分の仕事。それから管理ミスというのもまずいですね。ようするに、おっちょこちょいということですよね。ようするにこれはもう完全に教育の欠落としか言えようがない。それはですね。商学部でのプログラマーを軽々しく雇ったりする（：－）ことに。

まったく違う人が使った違うＯＨＰですけども、全く同じ事を言ってますよね。目的を持つような問題意識。それから、使い方、方法論ですね。正しく使用する。注意深く使う。みんな同じです。かれもそうですし、環境、環境という前に、人的な資源の問題が非常に深いと思います。まぁ、アカデミックな立場から言うと、もうちょっとね。高等な教育を受けた者をそれなりの扱いをして欲しい。あの、見せていただけますか。岸田さんのＯＨＰ。

岸田：　あの、著作権というのは？（：－）。

篠田：　あの、人的資源と言うのは、こういうのも含まれるのであって、ここで架け橋になるのは、アカデミック。まぁ、ここアカデミックだとしましょう。ようするに役に立たないことをやっていると。ここは役に立たないというか、ごめんなさいね。役に立つことをやっている所。うちは役に立たないことをやっているので。役に立たないことをやっている。でも面白そう。将来役に立つかも知れない。ここは毎日おかまを作って暮らしている。ここの架け橋になるこの部分と言うのは、こっちから、というふうに働きかけない限り就職する学生なんですよね。で、その学生も例えば学部レベルのもう、うちの学部もたいしたことないですよ。東工大の学部だってね。学生の時はカードパンチ。ぼくの学部の時は計算機演習はカードでしたから。８０カラムのカードで打ちました。いまだったらＲＪＢでやってますけれども。たい

したことないです。それでマスター、ドクタークラスの人間がここへ来るとですね。ここの問題をとらえてこっちへフィードバックする力を持ってますよね。

環境の前に人的資源をもっと深く考えて欲しいですよね。そう思います。事例もたくさんあります。ここでは説明しきれないくらいあります。で、ここのこっちに向かって働きかけようとしても、許可が出ないためにその会社を辞めていった学生も、主張して辞めていった学生もたくさんいます。気をつけてください。マスター出の学生は恐いですよ。潜在的な力があるにも関わらず、発揮しきれないために辞めていきます。それだけです。

深瀬： はい、ありがとうございました。ちょっとその、私も一点だけ喋りたいと思いますので。

日経コンピュータに載って、動かないコンピュータという第3次オンライン、一千万ステップつき合っていた某F社のプロジェクトマネージャというのは、私の昔の同僚だったのですが。これはあの、外注さんを５００人使ってプロジェクトやってたんですね。あの、クラッシュして、プロジェクトが。それで、山本琢磨さんというのが謝って言ったんですが。その人たちのコード見積りが上のほうだったんですね。ウオーターフォールモデルでＯｒｄｉｎａｒｙＣＯＢＯＬプログラマーを使ってだいたい１５００ステップくらい書くんですね。まぁ、一般的な人は商学部卒とか。

会場： 普通のプログラム？

深瀬： ええっ。そのプロジェクトマネージャー自身はですね。実はあの、ＣＯＢＯＬジェネレーターを使って１００００ステップくらい書くんです。で、１００００ステップ書く奴が何人かいると済んでしまうプロジェクトというのがあるんですよ。で、ウォーターフォールでコード見積をして、何ステップだから何人でしますよ、という見積を書くんであれば、下のアプローチをとるんですよね。で、浮いた個数でお勉強をしてそれで自分たちの環境を整えるということを考えた方がいいと。

まぁ、日本の産業というのはそのメインフレームを使って同じＣＯＢＯＬプログラムを使って支えているという話をされてましたが、だったらですね。それを逆手にとってですね。ステップ数で換算するのだったら、ＣＯＢＯＬジェネレーターを使って、6倍の生産性にして、残りの5倍の生産性を浮かしてですね。で、その開発環境を整えて営業すればいいじゃないかというのが私の考え方です。その、世の中がそうであれば、ですね。そういうふうにやって逆手をとって自分たちの時間を作る。というのが僕はいいと思っているんですよね。

実際にこの人はですね。メーカーをスピンアウトして月に個人的な収入として３００万円あります、今。１ステップ３００円で受注をするというのは、簡単なんですよね。安いところで１００円でしょ。たぶんね。ＣＯＢＯＬだと。やっぱり逆手にとってね。その、こういうふうにやればいいと。見積はこれで工数見積りしてお客さんから取ればいい。それで浮いたお金で設備投資して。私は持論なんですが、ソフトウェア産業は設備産業なんですよ。それで、もちろん人もいるんですけれども、設備産業であるという認識をすればですね。こういういわゆる英語で言うとスティングといいますが（：－）、こういうことをやれば、どんどんやればいいと思うんですよ。みんなで。お客さんがステップ換算でいくらというのなら、そうだったらこうやればいいと。逆にそのノウハウに対していくら払うというお客さんに対して初めてまともに対応してあげると。いうことをやればいいんだと、僕は思うんですね。

佐原（ＳＲＡ）： 今、深瀬さんのですね。ステップ数は大きなシステムでやると間違っていて、とても１５００も出ない。３００で見積って優秀な人を投入するから５００にしようかと。言うからもっと儲るということですね（：－）。

田中（ＳＲＡ）： 佐原さん、違ってます（：－）。えっと、現在レベルだと１３００です。まぁ、それはおいといてですね。

佐原： ＳＲＡはですね。苦しいテストをしないからです。テストは発注から優秀なマスター卒業した人が一生懸命やって、あの、何だっけ、商学部の人は１３００まで一生懸命つき合うけど、トータルすると月３００になる。

田中： それも違います。テストはですね。ＣＯＢＯＬのテストベットがやってくれますから、大丈夫です。単価が高いのは工数が高いからです。ま、それはおいといてですね。先ほどの篠田さんの言ってることと、言われたことは確かに正しいと思います。それでですね。その逆の人をまず、マスターに成れなかった人。学部出身の人をですね。どういう道を辿ったら良いか。たぶん、高野さんの、言われたように、すなおに拒否の言葉、おま

えは辞めたほうがいいと言えない人に、の救い道というのが一つあると思うんです。それは、僕はその人間だと思うんですけど、理屈じゃない、体で覚える（：－）。こういうふうに言ってもらえると、その人間は救われます。そういうわけです（：－）。

高野： 確かにそれもありましたね。あの、まぁ一難しい事ですけども。おまえはだめだと最後通牒出す前にならやってみなさいと、そういうチャンスを与えてやるべきだとそういうふうに私は考えています。そういうふうにやってきているわけですね。けれど、何というのかな。その意志っていいますかね。やろうとする意志。それはこちらからみてもよくわかるんで、その辺を読まれてしまうと、やっぱり上に負けるなぁと気がしますよ。

田中： えっとですね。負けても良いんです。適材適所という言葉がありますから、あの、その人間なりに使える所って確かにあると思います。それを見つけてあげるということもマネージメントとして必要だと思うし、それをやるためにはですね、最低一人は理屈じゃない、体でやってる人間を養っていかなきゃならないんです。それがＳＲＡの僕ですから（：－）。

高野： えっと、どういって言ったらいいかわからなかったのですけれども。確かにいわれる通りだと思います。

古閑（ＨＳＴ）： 話題変えてもいいんですよね。

深瀬： はい。

古閑： えー、うちの平尾からいろいろなんか言われていづらくなっているのですけれども。私まだ若干２３才で多感な年頃ですので、まぁ、いろいろ思うんですけれども。ここにＳＥＡに初めて参加したんですけれど、名詞交換の時とか、役職聞いたら、課長だとか、管理職レベル、マネージメントレベルの方が多いですよね。ここに来ておられる方は違うと思うんですけれども、そういうふうにさっき深瀬さんのお話でステップ数で能力を計るお客さんに対しては、それなりの対応。で、アカデミックに対応するようなお客さんに対しては、それなりの対応をすれば良いという話をされましたよね。で、私はそれは開発者とマネージメンターの関係にも当てはまると思うんです。開発者がステップ数でしか、プロジェクトの進行状況を計れないようなマネージメンターには、ステップ数でも教えておけば良い。アカデミックな部分まで要求するようなマネージメンターにはアカデミックなそれなりに対応してゆくと、というのが感想のひとつです。

会場： ステップ数でしか開発行程のわからない人だったら、辞めた方がいいよ（：－）。

古閑： そしてですね。それともう一つあるのは、理想と現実という、うちの平尾から理想と現実のギャップとかいろいろ言われたのですけど、まぁー、私から言わせてもらえれば、基準のない理想はいくら言われても仕方がないなぁと思っているのですけれども。例えば、環境が整えばいいだとか、という話はもういいんですけども。理想、理想と凄く言われるんですけども、例えばここまでできたら理想なんだよとか、ここから先の理想は今はまだ良いんだよとかみたいなことを、ただもうちょっと基準を持った理想というものを持った理想を挙げて頂きたいなというのが。基準でしか動けないのが新人類の悪いところです（：－）。基準を持たない管理職というのは、管理能力がないんじゃないかと。

平尾： 私は別に管理者をやっているわけじゃなくて、そういう人たちをちょっとだけ見ろと言われているんで見ているだけで、べつに給料も高く貰っていません（：－）。

深瀬： ここは人事管理のセッションじゃなくて（：－）、開発環境のセッションなので、話題をもとに戻して欲しいんですが（：－）。

佐原： 開発環境ってやっぱり人事と非常に絡むんですけど、ＳＲＡにおいては好き嫌いだけで人事が行なわれてますんで（：－）、だから自由に出来る（：－）。

北野（ＫＣＳ）： 実はですね。ステップ数で計っているマネージャーをどうのこうの、悪いとか言って、そこで大声で笑っているうちのボスは、実はステップ数で多くのプロジェクトを計っておるのですけれども。だからですね。だいぶ前言われてた、特に人材教育ですね。ソフトウェアの開発環境を云々する前に、大前提としてそれは凄く大事な問題だと思います。

正直な点。ある、ＪＳＴＡＲのユーザ会でですね。月に一回づつ、数時間づつですね。いろんな企業の方とお会いしました。その企業の方と言うのは、日本経済をしょってたってきたような鉄鋼ですとかそういう大企業の

方で、やっぱりその、根本的にはその、最初はオフィスの生産性をどうやって挙げるかという話をしているとたんに結局人材教育に行ったと。

実際に我々の職場ではそんなに教育の時間はほんの一部で、せっかくこうして神戸で開催しているにも関わらず、うちのメンバーでここに来ているのは数えるほどで、ＳＲＡからわざわざ東京から来られている方よりも少ない。というのが、現状なんですね。それでまぁ、私は好運にもここに来られている、加わっているのはうれしいですし。盛田は来られてどんどん引っ張ってくれるんですけども、本当にマネージャーの多くの方がこうやって教育に出してくれるようなそういう雰囲気というのは、もっともっと本当に社会的にソフトウェアの産業というのが、お金が儲けられるとか、知的環境である。で、実際今、開発でやっている仕事というのは、全然ブルーカラーの仕事に近いような事をやっているという認識を社会的に広めないと、まだ２５才なんですけども、将来はすごく悲観してます。以上です。

古閑：　さっき、ステップ数で計る管理者がいると言われて、そういう人たちにプロジェクトはどれくらい進行したかと報告するわけですね。１０００ステップ見積っていて８００ステップ終わりました、と言ったら、プロジェクト終わりだと感じたら管理者は喜ぶわけですね。おお、お前はよくやっているなと。お前の査定はいいぞ。今度のボーナスはいいぞと。ラッキーとか思っていたら、プロジェクトが終わって、出来た物がバグだらけで失敗だったと。そうした場合、１８００万で受注した物が実は２３００万くらいかかって５００万の赤字である。そこで、やっぱり管理者から開発者に、何だ、このプログラムはっ、と言われるわけです。上の管理者と言うのはステップ数でしか基準を持ってないわけだから私たちはもう言われるとおりやったような気がして、どうしたらいいのかなという。

深瀬：　えー、管理者の盛田さん（：－）。
盛田（ＫＣＳ）：　管理者の盛田です。

会場：　ケーシーエスでは一人月、何ステップですか？
盛田：　いや、あの、皆さんはマネージメントというのを正しく理解されていない。あの、マネージャというのはもちろんですね。ステップで話をするときもありますが、ステップだけで見ているわけではないんです。だけど、お客さんがステップで言うからステップで話をするときもあるし、あるいは社内でステップで話をすると納得しちゃう人がいるからステップで話をする。

だけどステップだけで簡単に片づくものであれば、僕も世の中ハッピーだし簡単なんです。そうじゃないから、マネージャーというのは苦しんでいるわけだと。そういうステップがどうのこうのという単純な話じゃないんです（：－）。そういう事がわからないとやはり、マネージャーにはなれない。

深瀬：　どこがどういうふうに単純じゃないんでしょう。盛田さん。
盛田：　あのね、ステップで話をするときもあれば先ほど言ってたように感性の問題だとか能力の問題だとかその辺をうまく表現はできないですけども、でもそれなりにみんな自分なりのメジャーで持っている。と僕は思います。

深瀬：　それはオートワシューという話じゃないんですか。
会場：　日本語で！！
深瀬：　つまりアメリカでは、お客さんが何って聞いてきたらこう答えるというのがあるんですよね。レジュメ。マニュアルがあるんですよ。
会場：　でもできると答える！？
深瀬：　いや、それは感性とは別の問題だと思うんですよね。
篠田：　だってそこら辺て営業がやるんでしょ。で、営業のマニュアルにはさ、何でも出来ると答えろとかいてあるんだもん。

臼井（ＪＩＰ）：　いやいや、何でも出来ると答えてもよいというマニュアルに沿って答えてもよいわけですけども。あの、話がとにかくマネージャーとそれから、実際に物を作っている立場の問題じゃなくて、ようするに、何をしているかということでしょ。例えば、本を出版する場合にですね。内容を書いている人とこの本の訂装を考えている人とか、実際本を製造している人とか、そういう他人の立場があって今何をしているかということだと思うんですよね。だから頁数が多いと１頁あたり紙のコストはこれだけかかるからこれだけ必要だという仕事をしている人は頁あたりの単価が必要だし、その中身の小説を書いている人はその小説の頁数じゃなくてその内

容でしょ。だから厚い小説書いていれば高く売れるわけじゃなくて、そこを同じレベルでわいわい言ってても駄目なんで、ようするに自分は今小説を書けといわれているのか、その本を、何頁の本を作れと言われているのかということ。単価もベースも全部違うわけでしょ。だから、マネージャーも今どちら側を管理しているかということじゃないですか。と思うんですけど。

篠田： それじゃ、困るんじゃないかと。進歩がないということで。

北野： えー、多少誤解されていたら困るので一言だけ言わせて頂きます。要は、マネージャーがステップで換算したがっているという話じゃなくて、社会的に、例えば、メインフレームで事務処理の大きなシステムを開発してますけども、そのユーザーさんが要はステップ数で計りたがっているから、お金貰うためには仕方なくマネージャーの方も痛いところを言われているのだろうし。で、それがそのまま下のほうまでどんどんどんどん降りて来ると。だから僕の言いたいのは、計り方が違うなということを社会的にもっと認識が上がらないとまだまだ難しいなぁと。まぁ、もちろんここで反論はいっぱいありますよね。

桜井（ＳＲＡ）： 先ほどの佐原の下で仕事をしているSRAの桜井と申します。先ほど話題に出た社会的認識のおかげで助かっている、という話を一つさせて頂きます。今年の春に某大企業から委託を受けてあるシステムを開発しました。当初１３人月という見積りでスタートしたんですが、実際には二人で２週間くらいでほぼできあがってしてしまいまして（：－）・・・といっても、これは見積りが間違っていたわけではありませんで、その背景には、アイデアとか非常な企業努力が存在したわけでして（：－）・・・もちろん、細かい修正を入れるともう少し時間がかかったわけですけど、それにしてもずいぶん余裕ができてしまいました。

で、その間にいっぱい遊べて（：－）社内システムまで作れてしまいました。遊んだといっても、実際には、その時使う言語のマニュアルをよんでたり、それを使って趣味のプログラムを作ったりして練習して（：－）いたんですけどね。要するに、その分、社内で使ういろんな道具とかマニュアル類とか、個人レベルのものも多少あるかもしれないけれども、日頃やろうやろうと思ってもなかなかできない仕事もできてしまったんですね。

この例から考えますと、契約で仕事をスタートした以上、こちらのアイデアと努力で工数を削減しても別に責められることではありませんし、またそうしたからといってユーザーは価格の値下げを要求してくるわけでもありませんよね。そういう認識ですから・・・　だから、そういった状況を積極的に利用して社内でいろんな物を貯めて、それをまたその後の仕事に生かすという形でやって行けばいいんじゃないかと。ようするにユーザーの方がいろいろ理解できてないと言うのだったら、それはそれでそれなりにこちらも要領良くやって行けばいいような気もするんですけど・・・。

浜野（ＳＲＡ）： あれは恵まれた環境開発という部隊だから許されることで。非常にお客さんとメーカの間にたって胃がきりきり痛む立場にいまして、あれは幸せな例であって、ＳＲＡ全体ではないので誤解をしないでください。２週間の仕事を１３人月、そんなこと許されるのだったら、やりたくてしかたないです。

佐原（ＳＲＡ）： 開発環境といっても、要は言語もあれはＡｓｃｉｉが売っているｉｎｆｏｒｍｉｘでやってたから２週間で出来たので。それでＣＯＢＯＬでやれば１３人月かかる。ＣＯＢＯＬというのは、Ｃだったのですけど、Ｃだと１３人月かかるわけですよ、やっぱり。それはお客さんの方がもっとかかると思ったらちゃんと割り引きしてそれでするんだと。だけど別に、１３人月を２週間でやっているばかりじゃなくてですね。そのほぼ同時期に４００万円しか貰えないのに明らかにひょっとすると４億かかるかもしれない仕事をですね（：－）。たぶん結果、４０００万円くらいかかっていると思う仕事も同時に受けてますから。お金は実際には某大企業から某ＥＴＬという所に移っただけの気がしますが。そういう慈善事業もやっています。

篠田： そこで値上げしてもいいですかね（：－）。

問題は結局、下のレベルのダウンストリームのプログラマーにとっても、管理職にとってもね。たとえで問題意識だと思うんですよね。プログラマーは例えば、自分のやっていること、何かおかしい、と思ってもいえないような・・・。

佐原： この際、言葉の通じない管理職。ここに来てないでしょうたぶん。環境と言ってもわからないマネージャって多いので、そういう人たちは除いて、言葉のわかる人たちでどう環境を整えていったらいいか話を・・・。

篠田： 今まで薄々気がついてるけどね。はっきりわかってないこともあるでしょ。薄々気がついてるけども、はっきり何が悪いのかわかってないということもあるでしょ。まぁ、誰と言わないけど。だからね。環境というか、問題意識のないマネージャー。

それで、問題意識のあるマネージャの野村さん、マネージャとして。

野村（ＪＩＰ）： あの、今、佐原さんの意見に関してなんですけれども。僕なんか今感じているのは、今議論がどこにいっているのかわからないでのですけれども。とにかくあの、環境がわからないマネージャーを除くというのは、僕らにとったらすごいこう、駄目なんですよ。結局、マネージャはそういう人たちなのですよ。で、私ども今、今日珍しくこの環境ワークショップにうちから７人。これ、画期的なことなのですよ。で、それになるべく人を増やして、そういう人たちをなるべく増やそうと今しているわけなんですよ。そうやって世を改革していこうとしているわけなんですよね。それで、そういう意味ではその人たちを除こうとかそういう意見てのは、駄目なんですよね。

佐原： ＢＯＦでは除こうといっているわけです。

野村： ああ、わかりました。思うんですけれども、やっぱりあの、管理職者というのはね。確かにその、ステップいくらと深瀬さんの言われた数字とはちょっと違っているのですけれどね。でも、そういったことじゃなくて、じゃ、生産されないとどうしようかということを実際に会社的にこう盛り上げる。あるいは深瀬さんが先ほどの前のセッションでやられましたように、技術者をいかに育てるか。そういった話を真剣にやろうじゃないかと。というのは、ここに集まっている深瀬さんは非常に情産協、ジョウサンキョウでいいのかな。それでショックを受けられたのと同じように、多分うちの管理職者がここに出てきたらものすごいショックを受けると思う。なんだ、この連中は！！っと思う。

篠田： 組合だと思うでしょうね。

野村： 何だ、ここ、と思う。深瀬さんがショックを受けた以上のショックを受けると思う。世の中そういう人たちが両方いるんです。だからそれをいかにね、僕はやっていくかということを、与えていただければ幸いだと思うんですけれども。

深瀬： 私は、駄目なマネージャーを黙らせる手法というのを私はたくさん勉強してきてますので。黙っていてくれれば良いというだけで駄目なんですよね。若い人にとっては。それからやる気のある人たちの邪魔をしないと。その、行く手を阻むなという方法論をミドルマネージャが体得するというのが仕事じゃないかと。それで、だいたいボードメンバーっていうのがわかんない人たちで、そういう人たちを黙らせて納得させて行く手法をね。あの、文字通りのメソッドですけどね。それを若い人たちに早いとこ体得して欲しいというのが、それが私の願いです。

岸田： ようするに、深瀬さんが話されて言いたいことは要するに、ちょっと野村さんの真意がよくわからないのですけどね。世の中にはわけのわかんない人たちが大勢いるわけですよね。だからこそ、我々飯が食えると思うんです。ざっとそういうことを深瀬さんは言っておられる。世の中のわかんない人はほっといた方が我々にとっては良いわけですよ。例えばメインフレーマーは病気だと私はさっき言いましたけども、病気だからこそ我々は飯を食えるわけで、メインフレームの病気は私は直さない方がいいと思う（：－）。

深瀬： なんかそれも切ない話になりそうな気がします。

盛田： 深入りしたくないんですけども、コメント、１コメントだけにします。確かによくわからないマネージャをはずして議論しようと、それはそれでいいんですが。ただ１つだけ肝心なことがある。それは、世の中というのは、これを言うとまた関西だといわれるのですが、経済原則でやっぱり動いてまして。ですから、世の中でわからないマネージャをはずして議論するのも良いが、多少はお金のことも心配しながら環境の事を議論して欲しい。というのが私の意見です。

深瀬： いや、経済原則はこれなんですよね。

盛田： いや、それはやっぱり、それもありますが、それ以外にもいろいろある。例えば、こうやってここへ来てますが、これただなんじゃないんですね。それなりに経済原則が働いてますということだけ、一言いって私は去る。

篠田： ここにいるマネージャーの間だけでも１ステッ

ブの見積りを今この場から２０倍に見積るというのはどうでしょう。あのね。情報工学科の学生はみんなね、自分の事、アーティストだと思っているんですよ。ところがね。建築科の学生はみんなアーティストだと・・・。

深瀬：　アーキテクト。

篠田：　アーティストなんですよ。

深瀬：　アーキテクトでしょ、建築家は。

篠田：　学部のね、建築の学部を卒業した学生がね。例えばなんとかの設計事務所に入って１枚図面描いていくら貰ったと思いますか？３日か４日で１枚の図で。

会場：　Ａ０？

篠田：　Ａ０までいきません。こんなちっちゃいの。・・・６０万。バイトでもそれくらいになるんです。ちょっとなんかおかしいですね。まぁ、こういうことは別としてですね、さっきから僕は話したいんですけど、いいですか。

うちで学生に何をやらせているかということをちょっとお話したいんです。学生にですね。あのー、計算機教育ですけども。研究室に入ってきて、何も言いません。ただですね、僕が普段欲しいと思っている、というかこうあったらなぁと思っていることを彼に伝える。それで、こういうことが出来るようにして欲しいと言う。まず、それが最初です。まぁ、時間かかりますけども、なんとか出来て来る。その過程でですね。こう１週間くらい経ったとき、出来たかと聞くんです。すると、どこが引っかかってます、何かこの辺がよくわかりません、と言ってくる。そこでちょっとアドバイスを与える。だんだんサイクルが短くなって加速してきますね。しまいに向こうから聞きにきますね。僕はもう３０才過ぎたおじさんで、向こうは２０ちょっと過ぎたくらいの若い子ですから。なかなか最初は聞きにくい。そのうち、サイクルが短くなるとどんどん来てくれる。適切な方向修正を与えてやると、だんだんサイクルが短くなってついになんかこう、定常波みたいなのが出来て、出来てきますよね、僕が最初欲しかった物が。

その次はですね。何をやるかというと、彼が何か言うのを待つんです。彼もその、一番最初にやったプログラムの作る過程で一応の環境というのを覚えますから、何でこういうことが出来ないのかと疑問を口にした瞬間、それは疑問を口にした人が自分でやることになっているという。まぁ、もし解決法があれば、教えますけどね。常々僕が思っていることだと、それは、口にした人がやる。向こうから言うのを待って、そういう形でやるんです。だから、やっぱり問題意識ですよね。問題意識を持たなくちゃいけない。マネージャーの人どうぞ。

山口（電力計算センター）：

先ほどMMLではたいぶ恥をかきまして、じゃ、恥のかきついでだということで。ちょっと話題がそれますけど、問題意識と言うのはかなりあると思います。先ほど岸田さんがおっしゃったように、確かにメインフレーマは腐ってます（：－）。私は、メインフレームのＨＩＴＡＣのM－６６０Ｄで、ＦＯＲＴＲＡＮプログラムを作ってます。それで、どうしてこうレスポンスが遅いんだろうと、いつも感じてます。それにモジュール化にしても非常に面倒くさい。ちょっと何かやろうとすると、ともかくＪＣＬを作るなり、コマンドを打つなりしなきゃならない。そういう環境なんですが、私は、今の立場からすれば、どうしてもそれを使わざるを得ない。もし、ハイパーメディアが使えれば、確かに生産性は上がるだろうと思います。

で、今、痛切に感じるのだけれどそれがメインフレーム上で使えたらいいだろうなと思うんですよね。それで、なぜこれがメインフレーム上にのっからないのか。仕方が無いから、プログラム開発はワークステーションなり、パソコンなりでやって、結果をメインフレームに持っていって動かすだけにしよう。言い替えれば開発環境は、小型機に任せて、実行だけは大型機にしようと。まあ、こんなことを考えているわけです。そうしたら、当然ＭＭＬは....

会場：　必須なわけです！

山口：　そこで、問題意識ということになるわけですが....何でメインフレームマシンあるいは、メーカーというのは、MMLには、冷たいのだろうということがある。つまり、先ほどご指摘があったようにＰＣ－９８０１には、確かにメインフレーム上のデータセットに画像ファイルを持っていくことができるわけです。ところが、日立の端末である２０２０とか、あるいは富士通の端末の場合、例えば富士通でいえば、リンクサーブを使うことになるのですが、そういったMML用のソフトウェアを使うとできることが極端に限られているんです。テキストファイルやある種のデータファイルは、できるけれど、それ以外は全くできない。ユーザがそのためのプログラムを作ろうとしても、情報が少ないため、非常に苦労する。だから、開発元に「こうしてくれ、ああしてくれ」

と頼むことになる。でも、バージョンアップされてもなかなか要求通りのものにはならないことが多いわけです。使う側としては、メーカの提供したものを使わざるを得ない状況です。それが、メインフレームが腐っていると言われる所以でもあるでしょう。それで、メインフレームメーカに頼らずにできるもの、ということでワークステーション、ＵＮＩＸに対してある種の期待があるわけです。

野中（日本電気）：　うちの職場に今年の６月からマイクロバックス入れて４．３ＢＳＤ動かしてますけれども、腐っているのはメインフレームだけじゃなくてＵＮＩＸだって相当腐っているし（：－）。あれ始めると何にもできないし、だいたいまぁ、入れて半年経ってますけれども、仕事には恐くて使えないですね、まだ。安定してないから。そんな感じだから。そんなに悲観することないです。

深瀬：　今の方のお話に答えたいと思います。だいたいメインフレームと言うのは、ＡＣＯＳの使い方は・・・ＶＯＳ３だとか、まぁ、どうってことないですよ。バイトマルチプレクサのチャネルにあの弁当箱をつなげればですね。ＴＣＰ／ＩＰみたいにアクセスできるんですね。で、メインフレーマーは作れなかった、こういう物を。で、とりもなおさずね、勉強不足なのですよ。先ほど富士通の久保さんもおっしゃられてましたが、ＬＵ６．２という概念はなかった。発想できなかっただけなんです。メインフレーマーというのはブルーワーカーですから、そういうふうに使えばいいという。それで私は１０年ぐらいそういう仕事をしてましたから、良くわかるんですよね。それでこういうふうに使えばなんてことはないんですよね。ＴＥＬＮＥＴ、ＦＴＰは全部動く。３２７０のエミュレータは下で動きますから。それで何で２０５０を使うのかというのがよくわからない。まぁ言えば、＊＊＊ＣＯＭのＧシリーズです。なんであんなものを使うのか、全然わからないですよね。まぁうちらもここの部分から信頼性、リライアビリティとかＲＡＳ機能ともいいますけどね。こわれない。ハード屋さんですから、あっこは。ソフト屋さんじゃないんで。そういうふうに使えばいいんです。で、先ほどお話したように、ステップ換算でやれば、こういうことをやればいい。スティングですよ、スティング。まだ、ネットワークで何か仕事したいんだったら、こういうふうにすればいい。それでなんかその、ソフトウェアとかそういうことを聞かれる方が無理でね。それはその、使う側が自分の選択できるというそういうその、明確なポリシーをもって、対応すればいいだけの話でね。で既にたくさんの材料をもっているわけです。現実的な既に動いている材料を。で、それを私は知って貰いたい。

新田：　私はメインフレーマーの方を弁護する立場にあると思います。今まであの、メインフレームは病気だといろいろひどいことを言われましたけども、メインフレームが全部病気だとしてもそれは、体の病気にかかっているに過ぎなくて、ＵＮＩＸ側はほとんど心の病気なんで（：－）。それで、メインフレームの方に私が希望しているのは、４．３＋自社機能というのを出すのをやめて、多分５年間くらい収益が悪化するのをあきらめてですね。もっと何というか、メインフレームのコンソールというのは一人で使っていると非常に使いやすいんです。だと思います。コンソールを一人で使えば。そういうところをですね、もっとなんか。個人で使えるような。つまりＵＮＩＸとかの世界に浸って来るんじゃなくてですね。今まで蓄えてきた知識をもっと生かして欲しいと思う。どうしてかというと、もうすぐ多分、個人で使えるワークステーションが十数ミップスから数十ミップスの時代がくると思うんですよね。今そういうマシンを使いこなせるものを持っているのは、メインフレームしかないんですよね。今ＵＮＩＸのワークステーションでは数ミップスのマシンをどうしようかということで頭がいっぱいなんです。でも、今うろうろしているともうすぐＵＮＩＸ勢がですね、そういうスーパーコンピュータに近いことを使いこなせる力を蓄えて来ると思うんですけれども。それより前に、今まで持っている力を生かせて後５年間くらいは収益が悪化するのを覚悟して６年先、７年先を見ると、今のメインフレームの生来の良さが生かせる時代が来るんじゃないかと思います。ですから今ＵＮＩＸどうのこうのと振り回さないで例えば、ＶＯＳが机の上にくればどうなるかということを考えた方がいいんじゃないか（：－）。

会場：　ＶＯＳが机の上に来たら、近寄らないよ（：－）。

佐原：　メインフレームを実は私は１０年近くさわっていたんですが、どうしてメインフレームで何かやることがあるか、と考えてみるとですね。唯一、あ、唯一じゃ

ない、ふたつだけあります。ＶＥＭのところと同じ？？？？？のシステム。それをですね。高速カードソータという意味では使えます。それ意外でメインフレームを使うアプリケーションは考えたら何もないんですよ。使うために非常に苦労をする。で、メインフレームのメンテを会社で一番できる私が一生懸命２週間やってＶＯＦというこれＵＮＩＸのフォートランで多少ツールを真似して作ったやつですけれでも、ハードディスクの中におさまるのに２週間もかかるシステムをなんで使う必要があるのかなとそういうふうに思う。

渡辺（電力計算センタ）：　佐原さんの言うことに半分くらい心からその通りだと思います。で、ただ実際うちの所で、あっ、これ録音とっているんですか。これオフレコにしておいてもらえます？去年うちは、山口君がお守りしているのはハイタックなのですが、それまでファコム１本だったのですが、うちは４カ所研究所がありまして、３カ所マシンが入っていたのですが、１カ所どうしても、何だか知らないけど、いれなきゃいけないことになって。それまでファコムのＮＳＰでネットワークを張っていたのが、某ハイタックが入ってしまって、ＲＪＥもつながった、つながったと上のほうは喜んでいるんですよね。要はそれだけつながったと思いたくなります。それから後、電通研の情報システムの方でＳＵＮ３の１６０が１台ありまして、それでフォートランのベンチマークをやっていた。私のお守りしているのがファコムのＮ３４０Ｓという中型になるんですが、ＮＳＰもついてて、８７実装しているですが。ＳＵＮ３でやったフォートランのあるベンチマーク用のプログラムを作って、私のマシン、ＮＳＴのせてるマシンの４倍（：－）。ところが、うちのコマイという所にいきますとＭ３８０があるんですね。３８０にいくとＳＵＮよりもやっぱり、たぶん５０倍くらい早い。ところが、ＮＳＰの３８０とハード的にはだいたい実装と形態は同じなんですが、パイプライン処理ができるＶＰ５０というチンケなベクタプロセッサがあるんですが、これだと、ベクタライズがものすごくって、むちゃくちゃ早い。だいたい１００倍は軽くでる。そうするとうちではだいたい、マークスのＣＰＵ時間は２時間なんですが、一番長い奴ですが、それで１週間だいたいジョブを２時間刻みで１週間かけてリストを３回くらいだして、あー、結果がでたなといって、捨てるという原子炉の炉特性なんかをやっていると、まぁちょっと今ワークステーションでは辛いなと。で、私が思うことは、結論なんですけども、今はやっぱりＭＳＴなんかを使って無理してやればあーゆーふうな繋ぎかたができるんだけども、僕のところのばかでかいマシンがそのＳＵＮぐらいの性能のマシンに負けるのだったならば。で、ブルーワーカーの人たちはそれで良いと思っているんだったならば、机の脇にワークステーション吊してＮＳＴのせてそこでしっかり一人でやらせといて、つなげてやって。そして他にもっと新しい世代を目指す人はそっちを使えば良いんじゃないかな。私はそう思うんです。

吉本:　ちょっと、ミップスばっかりでどうもお話するのはおかしいんじゃないですか。ミップス、ミップスとおっしゃるけど、ミップスくらい非科学的な基準はないんでね。それこそプログラムをステップ数で勘定するのと全く同じになってしまうんでね。こういう方がミップスでものを言われたら、非常に困ります。例えば同じＩＢＭでも同じミップスの機械でね。例えば４３００というものがあります。４３００というのは実は、もう一つ３０３１という古いのがありましたけどね。どっちもミップス１です。これでかたや同じ仕事をやるのに、３倍ほどかかってしまう。結局はノイマンボトルネックの問題なんですよね。御存知だと思いますけれどね。そういうことを計算にいれてもらわんとね。ミップスだけで物を考えてもらうのはプログラムをステップ数で考えるのと全く同じくらい非科学的だ。たしかにメインフレームは病気だけどね。そこんとこでやっぱり、どうもならん、腹立つけど使わなならんということが実はあるということなんですわね。本当を言やぁ、僕も使いたくはありませんけども、まぁそういうことです。

山口：　メインフレームは病気だと先ほど私も言いましたけども、利用できるものは、あとひとつ、ふたつはあると思います。つまり、メインフレームで腐っているのは、ユーザインターフェースの部分だけです。馬力という面では非常にあると思うんですよね。私は、メインフレームはあんまり好きじゃないんですけど、結局弁護する形になります。例えば有限要素法とか、メッシュ切ってそのメッシュ間で計算するようなシステム、アルゴリズムとかいうのがありまして、そういったものに関しては、大型計算機でないと困ってしまう。同じことをＳＵＮ３でやったら、どのくらいかかるか。１週間かかって、たとえ結果が出ても、そんなにかかっちゃもう１回試行

するわけにはいかない。つまり、大型計算機で必要なのは馬力であって、ユーザインタフェースがどうあれ、少なくとも馬力があるところを私はかいたい。だから、ユーザインタフェースに関しては、別のものがサポートしてもかまわない。メインフレームが腐っているのは、ユーザインタフェースの部分だから、それを切り落としてしまう。馬力だけを使う。それがメインフレームのいい使い方なのではないか、と私は思うのですが。

佐原： 具体的な所は深瀬さんが言ってくれると思いますが、メインフレームはユーザインタフェースが腐っているんじゃなくって、基本アーキテクチャーから既にもう３０年前から腐っているわけでして。つまりあれより早いワークステーション作ろうと思えば作れる。スーパーコンピュータみたいにどんどんやればいい。で、あれは一生懸命減速してそれよりちょっと早いと言ってますが、それは早いんですが、人間がそのプログラムを作るのが遅いから結局答えがでるのが遅い。

高野： えー、すいません。あのぉ、金物屋から一言。老婆心ながら、解説を申し上げたいと思います。確かにメインフレームは腐っているというそういう意見はわが異あらず。でも、全部が腐っているわけじゃなくてね。見直すべき所もある、というところをご紹介したいんですよ。

僕ら、コンピュータ使って仕事をしてるわけですよね。それでそのコンピュータに高いパフォーマンスを発揮して欲しい。で、今パソコンどうか、全然役に立たない。ワークステーション役に立つか？くその役にも立たへん。逆だった、ごめんなさい。パソコンはくその役にも立たなくて、ワークステーションは役に立たない。それはなぜかというとね。ハードが遅れてるんですよね。あのぉ、先ほどノイマンボトルネックと言ってましたけども、そのあたりの問題でもってずいぶん遅れている。えー、その辺がんばっているのにメインフレームがあるんですよね。その辺は見習うべきところがある。で、我々はこれから何をやらなければならないかというと、実はパフォーマンスを上げることじゃなくて、まぁソフト屋さんが考える仕事を十分なパフォーマンスでやっていこうと思うと、今のワークステーションのおそらく１００倍とか１０００倍とかのパフォーマンスが必要になってくると思います。これは誰も異論を差しはさまないと思います。で、そのパフォーマンスを実現するために何が必要かというと、それはＣＰＵではなくてメモリでもなくて、バスなんです。バスのアーキテクチャなんです。だからアーキストじゃなくてバスのアーキテクチャが必要なんです。その辺を日本は随分遅れている。アメリカに比べると。この中でバスの設計やったことがある人いますか？多分いないでしょうね。

深瀬： イーダスを使っているのはいます。

高野： ない、ない。僕は自分でバスを設計して使うから。

篠田： それで標準性のないやつができて困ると（：－）。

高野： それはいいんです。僕が使っているだけだから。

篠田： でも、ユーザは困りますよ。

高野： だけどユーザには提供しないから。

篠田： それも困るんだよね。

高野： 自分の所はね、肝心要めのところを設計できる人間を育成すると。

深瀬： レクシーのバスはいいですよ。レクシーのバス

篠田： レクシーのバスは速いね。あれ、いくつあるのＭＢ／ｓｅｃ１２０ぐらいあるの？

深瀬： ３００ＭＢ／ｓｅｃ

高野： メインフレーマーじゃないけれどね。僕もＣＰＵは設計してきてるし、コンピュータシステムは設計してきてる。だからその勘所というのはわかるわけですよ。何が重要かというのが。ＣＰＵなんか全然重要じゃない。何でもかまわない。３８６でも４８６でも５８６でもかまわない。３０でもかまわない。６８０００の３０でもかまわない。

篠田： いやぁ、そんな・・・。

深瀬： 国産のメインフレーマーの一番弱い所は、バスネックなんですよね。で、ＩＢＭより速いＣＰＵをまぁ、国産のメインフレームは供給しているわけですよ。どうしてか。それは、ＩＢＭはバスアーキテクチャの設計がうまいんですよ。トータルパフォーマンスでは同じになっちゃう。ＣＰＵのパフォーマンスが遅くともね。で、私の会社ではエレクシーというマシンを入れているんですが、これは国防システムで使っているコンピュータなんですけども。それのバスの転送スピードが３００Ｍバイト／ｓｅｃなんですよね。で、これくらいのメインフレームを作ってくれないと困る。

篠田： 今のメインフレームの転送スピードがどれくらいなの？

深瀬： １００Mバイト／ｓｅｃ。

で、そのＥＣＭの製造技術とかクロックのスピード上げてね。バスのアーキテクチャの遅れをそれでカバーしているわけですよね。で、それはもう、高野さんのおっしゃる通りなんですよ。失礼しました。

高野： ですから、どこがキーポイントかということを押さえて、それをできる人間を育成するというのが篠田さんのご意見ですな。

篠田： そうです。

高野： ＣＰＵはどうでもいいんですよ。さっきは極論をいいましたけども・・・。

篠田： ＣＰＵはどうでもいいから、かならず。日本の企業というのはポイント押さえるのがすごくうまいんだと思うんです。ただ、全部をマクロで見たときに、システムを組み上げる能力はないですね。ただ、バスの装置、つなぎの部分でしょ。ＣＰＵ速いの作ると論文になるけど、バス速いのを作ってもなかなか論文にならない。

高野： それで今大学で何やっているかというと一生懸命、一生懸命ＣＰＵを束ねようとしている（：－）。

海尻： （テープの切れ目） そのときは本当はちゃんと理由があって言ったので、そのときは簡単に言ってしまいましたけれど、あのぉ、そういえば大学の価値はどうのこうのという話があったんで、僕のほうはもう、かびがはえている大学の価値なんで皆さんの方とはちょっと違いますけれども、その辺の話を少し。

大学の方では、センサーはどこでもメインフレームでして、ちょっと迷いましたが全然出てこなかったのでどうしてかと思ったんですけれども。一つはメインフレームでないとできないのは、ＵＮＩＸでは納得できない、うちでは使っているのですけれども、どうしようもなくて困る場合があるんですけれども。スケジューリングが全くできない。コンベックスのような大きな機械、実際にはうちの情報工学科というのは非常に特異的といえば特異的でさっき申し上げたコンベックスというミニスーパーというかスーパーミニというか、を入れてまして、それにトータルで６０台か７０台端末を繋いでやってるわけなんですけども。回線的に考えてＵＮＩＸを繋げれないところがあるんですけども、実際にはそれで繋いで、それで５０人くらい繋げても使えるということでそれは問題ない。

ただし何が問題があるかというと、スケジューリングが全くないんで、要するにジョブのスケジュールというのが全くないので、一つの大きいサイズがばっと動くと全体での優先度もなにもない。優先度を少々つけたってそれが消えちゃって、どうしようもないと。他の奴が全く動かないということが起きると。もう一つはセキュリティがないということで、これはどこでやっているかということをいわなければ大丈夫なんでしょうけども。大学の場合は入試の処理という非常に恐ろしい処理をやるわけでして、合格不合格を一発で判定するという。で、そういうデータをＵＮＩＸで処理できるかというと、どこかにハッカーがいると恐くてできない。いうことで、セキュリティとスケジューリングの点から大学全体の線からとてもＵＮＩＸは使えない。という点があります。

それから、大学の教育の問題で、やはり東工大と信州大ではかなり違いまして、うちの場合はその前に大学の方でいろいろ研究しなければならないという意見もありましたけども、学生に対しては主に教育でして、教育がメインになっておりますので。それで、ちょっと教育に関して思ったんですけども、今の情報科というのは、この４、５年くらい前からほとんどの所がＵＮＩＸマシンをいれてまして、情報工学科としてはメインフレームの使い方というのをほとんど教えてないと。ということで、ふと考えた。それでいいのかなと。後数年するとほんとに全く情報工学科でＵＮＩＸしか教えてない。メインフレームの使い方はいっさい教えてなくて、メインフレームの使い方を知っているのは情報工学科以外の人間だと。そういうことになるんですけれども。どうなんでしょうか。いう気がします。

深瀬： えーとね。ちょっと岸田さんの前にお答えしておきますと、セキュリティのガイドラインというのは決まってまして。米国標準局、ＮＢＳ、名前は変わりましたけども、今はＮＣＳＣという組織があるんですけれども。そこはオレンジブックという本を出しているんですよね。オレンジブックというのはセキュリティレベル、国防的な観点からみたセキュリティレベルというのがあるんですが。それはＡ、Ｂ、Ｃとランクが決まってまして、それぞれ３つの階層なんです。で、その中で今ＵＮＩＸはグードっていう会社が出しているＵＮＩＸは、Ｂ

２というのをクリアしているんです。で、ＭＶＳとかＶＯＳ３とかはクラスＣしかクリアできないんですね。ＵＮＩＸがセキュリティレベルが低いと言うのは、それは間違いで、そういうシステムはあるんですよ、現に。ただベンダーがそれをサポートできてないだけです。例えばＩＢＭで言うとＲＡＦＦというシステムがあるんですよね。セキュリティレベルで。ＲＡＦＦという。で、あれはクラスＣしかクリアできないんです。で、クラスＡが一番上でマイティックスというのは唯一クラスＡの一番下の階層をクリアしていたんですね。で、それは誤解だと思います。

岸田： ここは開発環境の理想と現実を話して、メインフレームがなぜ必要かという時に実行環境としてメインフレームのＣＰＵパワーが必要だという意見がさっきありましたが、それは開発環境とは絶対関係ない話で、今のセキュリティの話も入試の処理っていう奴は、実行環境の話で入試処理のプログラムをどう作るかというあたりで本当にメインフレームはいらないの？、ていう話だと思うんですよね。

で、ハードウェアバスの話も出ましたけれども、僕は開発環境で大事なものはハードウェアバスよりソフトウェアバスで。それでそういう意味でＵＮＩＸは今、存在するソフトウェアバスとしては最良のソフトウェアバスだと。ということじゃないかというふうに気がします。それだけです。

中野（大阪大学）： 高野さんに反論、じゃなくてコメントだけです。あのぉ、ＣＰＵを束ねてどうのこうのとあるんですけども。アカデミックな立場から言うと、今いくらシーケンシャルなアルゴリズム提案してもね。論文通らないんですよ（：－）。現実の問題として。で、僕たちはペーパーを年に１つ２つ通さないと首は切られないですけどね。ＳＥＡにでてこられないわけ。それをちゃんと考えといてください。それともう一つ。一つはね、そのパラレル、ＣＰＵを束ねるんじゃなくて、バッファアーキテクチャで勝負しなさい、というのをね。もっと声を高くして言ってください。僕今日初めて聞いた。それだけ。

高野： すみません。中野先生にそれを申し上げてなかったのは申し訳ありません。で、ＣＰＵを束ねてもどうしようもないんですよ。それをなにかブレイクスルーをしなきゃいけないと。そのアイデアがね。日本から出てこないと、貿易摩擦も解消しませんよ。だからそのアイデアが中野先生の肩にかかっているわけです。そうですよ。アイデアの勝負です。

深瀬： そう！チューリング賞ですよ。論文集よりも。

岡本（ＫＣＳ）： 今回は設備下請けをやっています。実は，会費を払わないでここに来ているので発言しにくかったのですが，３０～４０分前にあった東工大の篠田先生の「航空母艦」の話に共感したので一言コメントします。

「航空母艦」の中にも色々な内情があると思います。その中で航空母艦を操るために色々な分業があるわけです。そこには航空母艦を営むための文化が営まれていて，その文化がそこに醸成されていることが非常に重要な事です。この考え方は，私自身の環境に対するひとつの問題意識でもあります。良い環境をあるひとつの世界に広めようとした時に大切になるものです。

良い環境というのは「人に勧められる環境」とか「自分が良いと信じている環境」等，判断基準の曖昧なものも指します。

ＳＲＡの岸田さんのＯＨＰに，大きな「矢印」を挟んであちら側とこちら側に「丸印」がありました。この「矢印」の部分にあたる「橋渡し」をどうするか....，と言う「カルチャーの架橋」の問題もありますが，この「丸印」の中で，さらにどの様にやって行くか

今回のＭＭＬセッションでも，この「丸印」の中の雑多な何かが問題になっていた様に思います。この「雑多な何か」は私自身の日常の悩みでもあります。これらの事はテクノロジトランスファーの問題として捉えるべき事なのかも知れませんが，環境を広める事を話する時には無視出来ない問題ではないかと思います。

この会場には，様々な環境を使って様々に仕事をしている方々がいらっしゃいます。お金儲けのために仕事をしている方々もいらっしゃいます。研究活動のために仕事をしている方々もいらっしゃいます。その中で行う環境を広める話は，ある意味で「運動」だと思います。やはり根強く，根気良くやっていく事も大切な事ではないかと思うのです。私からのコメントです。

深瀬： あのぉ、ちょっと補足しておきますけれど。米軍の第７艦隊の戦備配置のワークステーションというのは、ＲＱという今はつぶれてしまったワークステーションです。

田中： 今のケーシーエスの方に賛成で、すごくそっから先というのが重要で、もしかしたら岸田さんも考えてらっしゃらないのかもしんないですけども。考えてるんでしたら、後でご意見聞きたいですが。すごくそこの部分が。実は私今は営業セクションで営業をやってないんですけど、セクションにいます。そこの部分ですごく苦労していたというのは、本当で。ＳＲＡ今回一割近くきているんですけど。ベンダーレベルで来ているのは多分僕だけじゃないかと思うんですけど。本当にＳＲＡって理想的な会社に受けとめられている方が多分多いと思うんですけど。実は違っていてそんなのはごくひとにぎりの人間だけでＰＦＵのイランコセンターと同じように、ＳＲＡのこういうとこへ来てるのは、遊園地の人間がほとんどなんですけれど。

深瀬： サファリパークとも言いますね。

田中： はい。そこのサファリパークだか遊園地だか知らない、僕はわからないけど、確かにそういう人たちがいて、学術的なことをすごく引き出してきてくれるんですけど。使い先というのが結構難しくて、ほとんどベンダーレベルでは生かされていない。で、そういう人間というのは絶対必要だと思うんです。で、それをやる人たちというのがいなくて。で、さっきも僕言ったんですけど、理屈じゃない。体で覚えろ。というのは全て持論で、そういうわけのわかんないなんかコンセプトみたいなものがあったとして、それをわけがわかっていいものだなと評価するのも重要かもんないですけど、そういう評価というのはイランコセンターとかサファリパークの人たちにしてもらって、とにかく持ってきたというものを本当にベンダーレベルに落とすということをやっていく。その辺がすごく重要じゃないか。と考えているんです。もし、ご意見があったら。

古閑： 確かにですね。ここで話し合われている話、イランコセンターとかサファリパーク、まぁそういう話をそれをベンダーレベルに落とす、落とさないという話があるんですけれども、でも私は思うんですけれども、ここで話し合われたことをここで具体的に方法論が見つかったことというのはですね、具体的におこされたとき、使われまいが、使われようが、良いものは落ちて行くし、悪いものは落ちてもそのうち、消えてゆくんじゃないかなというのが、感想なんですけれども。

浜野： 話を流れを全然変えていいですか？今話を聞いて栗原と一緒にＯＨＰを描いていたんですけど。えっと、話がどんどん外れてしまってＯＨＰ描いたから一応出しとこうと思って出します。

私は教育とか管理とかいうものは後でついて来るものだと思ってますし、ぜひそういうチャンスがあれば、自分を教育したいと思っています。で、開発にパラダイムとして今一番頭で反趨していることは、抽象化を進めるという一つのアイデアと具体化でどんどんいくというアイデアとＮＳＴモデルがあると思うんですが。ＳＲＡの岸田さんがモデルハッキングというんですけど、下位をだいたいこうやってこうやってこうやったらわかるなぁという人たちにとっては抽象化というのは非常にまどろっこしいことだと思う。で、逆にこういうのをカルチャーとしてもってないと、わからんやろうし。ようするにこのやりかた、ＶＳなのかあるいは全部コンポーネントとしてバコッとまとめて環境があるのかよくわからない。まぁ、これはぜひ、まとまった方の意見を聞きたいなと思っているわけです。

あともう一つは、仕様がプロセスで厳密にいかないから非常にものすごいカゴが発生するという話がありますが、そのへんきちんとやるといってオスターワイルドさんという人はその辺からものすごいフォーマリズムでバシバシやっていこうとする人なんです。それと実際、あいまいであってぼんやりした中でどんどん一つの物を作ってゆくというその、ある意味でですかくて？？？？？というようなアプローチがあって、言いたいことはどうゆうことなんやろと。その辺はわからへんということですねぇ。それで私のイメージの中にあるのはエンジニアリング、ノットイコール、プログラミングだと思ってて。すぐにどうのこうのゆうけど、エンジニアリングといったら、ある程度年数がかかって、うちの人間と言ったらある程度これは大丈夫ですよとか、そういうなんかフィールドワークとか一つの体系で今出来て即使えますようなエンジニアリングじゃない。ただ、要するに、環境というのがよくわからんのですけど、ただ環境というのを考える場合にですね、なんか、このへんのこと嘘らしいとこを、いやそうじゃない、こうなんだと言ってくれる人がいると、今後どうゆう環境を作っていったらいいかというのがわかる気がして、要するに誰か教えていただけませんでしょうか。というのが最初冒頭でこのへんのセッションの方がやられた後で、感じたことです。誰か教えてください。

篠田： これはね。裏腹だと思うんですよ。人間は向上心ありますから、だれだったけ、オスターワイルドがね、ジェネラライジェーションを行ないたいという、常にジェネラライジェーションを行なう人間だということを聞いてます。ようするになるべく仕事したくないから、おんなじような仕事はまとめてやりたくなるんだけれども、というような抽象化ですよね。ところがですね、そういうことばっかりやってると食えなくなるので、しょうがないから具体的な仕事をしなくちゃならない。うちの大学院の学生はそうですね。むちゃくちゃあの、ブルーカラーのプログラミング、学校終わってから夜中の１２時までガーとやりますよね。次の朝学校来てなんか記号ばっかりの紙を見つめているという。

会場： 今のは反対じゃないんですか。

篠田： ごめんなさい。すみません。そうですね。で、これもそうですね。厳密にやると気持ちがいいんだけどもそれじゃ食えないんで、まぁやっつけ仕事してしまう。実際はですね。エンジニアリングは？？？？？ですね。。プログラミングは手仕事ですか？

会場： サンプ。

篠田： サンプですか。うちのねぇ。学部というか学科なんですけども、ドクターの最終試験に必ず聞かれる質問があります。技術と芸術はどうちがうか。これ、工学になっていますけれど、これ技術になりますでしょ。技術と芸術みたいな感じで。本当はこれイコールにならなくちゃいけないと思うんです。それをイコールのレベルに持っていくのが大学の先生の勤めであり、社会的にね。それをちゃんと受け入れていくのが管理職の仕事であって。今頭の、固いと言っては失礼なんですけれども、失礼なことはＢＯＦでは言っていいことになっているので。ここにはほとんどいらっしゃらないと思いますけれども、まぁ残っているような人は頭が固くないということで、頭が固い管理者がいる間は、黙って耐えている。その間に自分の勉強をしておく。少々欲しいと思っても黙っていると。で、我慢してればそのうち、自分も偉くなる。まぁ自分はさっきの深瀬さんのＯＨＰにあったように、適当なツールをですね、ほんのちょっとツールを作ればですね。２倍か３倍のプロダクトアクティビティがでるわけですからそれを使って、普段の仕事は上司の機嫌を損なわないようにこなしながら自分の勉強をしておきますね。自分がこう、役職についたときにじわっとせめていけばいいんですよ。自分の仕事も楽になるし、後輩の受けもよくなる。あいつ話のわからん奴だなと言われないようにしないといけないというのが、環境、環境という大それたものをアプローチする第一歩ですね。ユースフルな情報を仕入れる。それを使うという。

深瀬： えー、クローシング３分前です。プログラム委員長から一言。

中野（大阪大学）： 今回ものすごくうれしかったのは、今日の昼ごろのセッションのMMLでですね。今まで２回、第２回、第３回と環境ワークショップに出て、活動している人たちと話をしてね。全然話がわからなかったんですよ。でも今回MMLをやってみて、少なくとも同じ土俵にはならなかったけれども、少なくとも話は出来た。それはものすごくうれしかった。だから、今度の熊本では２日目の昼からはMMLです。その後はＢＯＦです。だから熊本だといわずにホットな人たち出てきてください。そうしないと話ができないから。お願いします。

深瀬： 熊本からはMMLをなくそうかという話が。あれはくだらんと個人的には思っているんですけどね（：－）。

中野： もう一回やりましょう、もう一回。

深瀬： やりますか？うーん、ラストチャンスでやりますか。えーと、後１分しかありませんので、最後に一言言いたい人ラストチャンスです。１分で。えーと、ここは９時には全部退席してきれいにしなくちゃなりませんで、後３０秒で終わりにしたいと思います。あのぉ、今日の口火を切って頂いたパネラーの方、４人プラス１人に盛大な拍手をお願いします。えー、ありがとうございました。

4 アンケート

野中 哲（ＮＥＣ）

人間の管理に関する話題はあまり面白くないので、やりたくないのだが、どうしてもみんな話題がそちらの方にいきがちになるのは、みなさんのうっぷんがたまっているということなんでしょうね。

濱田 勉（ＮＴＴ）

高野さんの話が聞けてしあわせです。

棚田眞司（ＳＲＡ）
問題がどういったことなのかしばれません。ただ、何か新しいこと（今までと違ったこと）を行う場合、メンバー全員の意志統一という事が、始めからなされていなければならないと思いました。（まとがはずれた感想ですいません）

杉田義明（ＳＲＡ）
環境をテーマにしたＢＯＦであったが、前セッションのＭＭＬで未消化だった議論をそのまま持ち越した状態で全体が推移したようである。しかしながら、本音のレベルで意見発表やコメントが続いたので、大変盛りあがったセッションになった。ＢＯＦとしては大成功だったと思う。

藤岡　卓（三菱電機）
発言しなかったけど残っていた価値はあったようです。

林　　香（ＳＲＡ）
今回の参加者の興味は、開発環境そのものよりも、管理、特にモチベーション・コントロールのあたりにあるのだろうか。若い人たちが、わりとフランクに話せるようになってきたことがうれしい。

渡邉雄一（電力計算せんたー）
　前日の情報交換パーティーのような盛り上がりを期待していたのだが．．．。どうもイマイチであったと私は思います。
　メインフレームの環境を，本当に開発に持込みたいと考えている人の多くは，メインフレーム上で実現されてべきターゲットシステムに必要な，データベース（特にそこでのスキーマの定義環境やエラー処理の検証・シミュレーション），言語処理系（もちろんデータベース検索言語の呼出し機能と日本語処理付き！），運用上のシミュレーション（機密保護機能の設定やファイルの排他処理の正否のチェック）を期待しているのであるから，極端に言えば，１人に１台ワークステーション上にＭＶＳなりＯＳＩＶ／Ｆ４を乗せれは良いのでは！
ＭＭＬのセッションでも指摘されていたが，「かなり完全なソフトウェアの等価性」が保証されなければ，分散開発というものは，生産性に真の意味で寄与しないということを私は言いたい。

山口郁生（電力計算センター）
メインフレームの話では、だいぶ、お騒がせしました。勉強不足と言われれば、返す言葉はありません。しかし、逆にＷＳを利用しているユーザもメインフレームを知らないのではないのでしょうか。メインフレームで１日かかる計算を、どうやってＷＳで実行するのでしょう？（開発環境とは、あまり関係がなかったかもしれません）

開発環境の理想と現実

— Position Paper —

昭和６３年１１月

ヒラタ・ソフトウェア・テクノロジー(株)
Hirata Software Technology Co.,Ltd.
平尾　一浩

1.　開発環境の理想と現実

現在の開発環境は1人1台ワークステーションがあり、開発を一貫して支援する使い勝手のよいツール類が揃っていれば良い環境であり、その環境から作られるシステムは良いものができると期待されている。ところが・・・

1.1.　1人1台ワークステーションの現実

そのような期待の中でいいシステムは本当に生まれてきているのか疑問を感じる。私がいる環境は既に1人1台のワークステーション環境である。しかし、実際に1人1台ワークステーションの環境の中で起きている問題は予想以上に多く— 携わる人間の問題も含めて —、その解決策を見い出していない部分も多く含まれている。以下に私がいる環境の中で私が感じる問題点を挙げる。

□ 開発支援ツールが使いこなせない

よいツールを与えただけでは何も生まれない。ましてや、複合的な使い方や応用は期待できない？　なにか手本になるようなものが必要？

□ 時間の重要性に気づかない

1台のコンピュータを複数の人間が使っていた時代は自分の時間が非常に重要で、有効に使うことを必死で考えていた。しかし、自分の机の上に自分専用のマシーンがあるといつでも使えるという安心感から時間を無駄にしてしまう傾向がある。結局、1人1台になってもその作業効率はあまり変わらないのではないか？

□ 個人のワークステーション管理ができない

1人1台ワークステーションになった場合、各自が管理者にならざるおえないが、全ての人が管理者になれるわけでもないし、なる必要もない。

□ 管理ミスによる他のマシーンへの悪影響

社内LANなどの構築時にファイル類の分散などによりクラッシュ時の影響は少なくすることができるが、同一ファイルが複数存在することになる。このあたりは社内LAN構築時の設計でうまいバランスを考えなくてはならない。

□ 統括的ネットワーク管理の困難

全てのワークステーションを管理する人が必要だが、台数が非常に多くなるので全てのワークステーションの状態やデータ類を完全に把握するのは困難になってきている。これでは各マシーンの力を統合、フル活用するのは難しい。

□ 全開発フェーズを統一的にカバーするツールがない

開発フェーズの部分部分を支援するツール類は有っても要求分析から設計、開発、ドキュメント、テスト、メンテナンスまでを統一的に支援するツールが無い。

□ 基本となる開発モデルがない

ウォーター・フォール・モデルが現実の開発工程からずれていることの指摘はすでにされているが、新しい現実に添ったモデルは出ていない。

以上、人的問題も含めて思い付くまま挙げてみた。

上記問題を解決するための開発環境を考えた場合、今までの環境の構築に加えて、さらに人を加えた基本開発モデルを構築してはじめて現実的開発環境ができると考える。これは1つのシステムと呼ぶに相応しい。

1.2. 現実的開発環境システム

前章で述べたように、今の開発環境には個人ベースの環境が前面にでてきている。しかし、ワークステーション1人1台の時代にこそ、そのパワーを引き出すための総合的な環境を考えなければならない。

図1 に現在の開発環境と呼ばれている部分を示す。

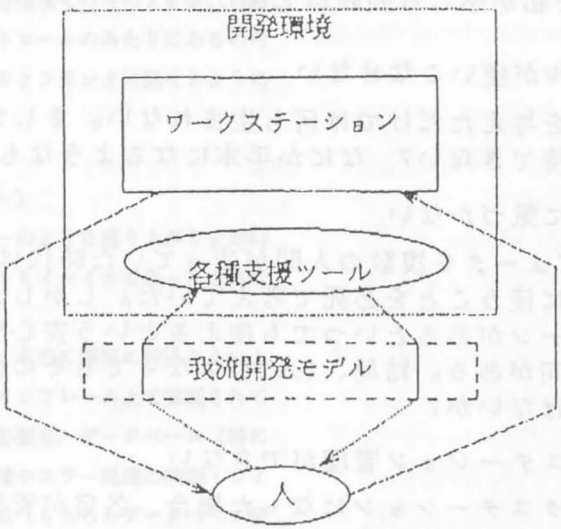


図1 現状の開発環境

現在、開発環境はワークステーションと開発支援ツール類に動きをおいて捉えられ、その構築と充実が重要視されている。加えて我流の開発モデルを各人が持ち、少しばかりのツール類を表面的にしか使わず、ワークステーションの機能を使うとはいえない状況である。よって実際の開発と我流開発モデルとの間に歪みが発生し、できてきたシステムは部分部分に歪みを持っているものになる。

そこで、これからの開発環境と呼ばれるのもは、今までの開発環境に人を加え、さらに基本的開発モデルを中心にした統括的なものを考えなくてはならない。

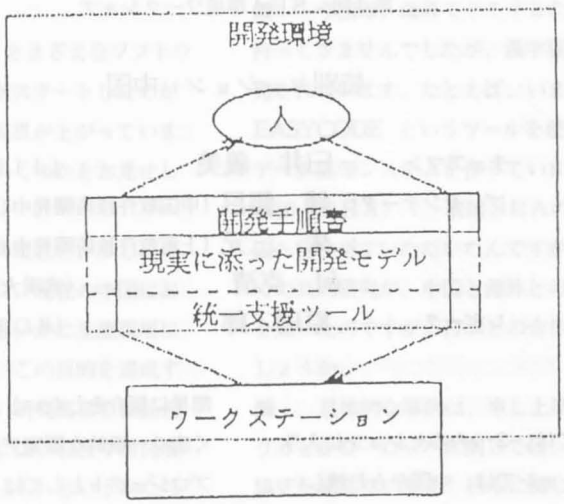


図2 統一的開発環境

図2 のように基本開発モデルを中心に、ワークステーション側には要求分析からドキュメンテーション、テストに至る統一的支援ツールを置き、人側には開発モデルに従った手順書を置くことにより基本開発モデルへのインターフェースをはっきりさせる。分散された開発環境下でチームの生産性を上げるためには、ある程度の自由度がある束縛を行ない、工程のスムーズさを重要視しなければならない。この点に関しては人的問題も含め、基本的手順書が必要と考えられる。

1.3. まとめ

現実的開発環境を考えた場合、必要なものは、

- □ 現実に添った開発モデル
- □ 開発モデルに添った統一的支援ツール
- □ 開発モデル、統一支援ツールに添った開発手順書

上記のものに加えてそれを環境と捉えることが現実的開発環境の構築につながることだと考える。これは既に単なる開発環境ではなく開発環境システムとして捉えることができると言えるだろう。

環境のワークショップで今後捉えなくてはならない問題は人を含めた開発環境であると考える。

第４回　ＳＥＡ環境ワークショップ

特別セッション　中国

チェアマン　：臼井　義美　（ＪＩＰ）
プレゼンテータ：鐘　錫昌（中国軟件技術開発中心）
：朱　三元（上海軟件技術開発中心）
：何　克清　（武漢大学）
レポータ　：原田二郎　（ＫＣＳ）

1. はじめに

臼井（ＪＩＰ）：　それでは，２つめのセションに入りたいと思います．このセションでは，中国からお越しいただいた３人の先生方に，あちらの現状をいろいろお話いただき，そのあとで質疑を行いたいと思います．

私は，日本電子計算の臼井です．チェアマンをやらしていただきます，鐘錫昌先生からお話いただきたいと思います．スライドをお使いになりますので，会場を少し暗くしてください．ご紹介が遅れましたけど，通訳はＳＲＡの楊（ヤン）さんにお願いいたします．じゃよろしくお願いします．

2. プレゼンテーション

2.1　鐘錫昌先生

鐘：　まず，自己紹介から始めます．私は，北京にある国家科学技術委員会のソフトウェア技術開発センターに所属しております．今日は２つのことに関して，簡単に紹介したいと思います．第１は，いま中国が進めている国家プロジェクト全般に関して，第２は，数年前から実施しているソフトウェア関連のプロジェクトについてです．

まず，私が所属している国家科学技術委員会ですが，これは，中国政府に所属している機関であり，その主な目的は，さまざまな機関や委員会の活動の調整をすることです．ですから，いろいろな国家プロジェクトの方針を決めたりとか，企画を立てたりとか，そういういった仕事です．この委員会の下部組織として，中国ソフトウェア技術開発センターがあり，ソフトウェア産業の発展促進を目的として，技術開発のための国家プロジェクトを運営しています．それ以外に，国際交流とか，合作に関連する仕事も，かなりのウエイトを占めています．

いま中国で進めれている国家プロジェクトについて，簡単に紹介をしたいと思います．ソフトウェアだけでなく広く一般的な国家プロジェクトについて紹介します．プロジェクトとしては，国の５ヶ年計画のなかに，高度技術発展計画，たいまつ計画，かがり火計画，国家自然科学基金などさまざまなものがあります．

かがり火計画というのは，中小企業を対象にしたプロジェクトでありまして，短期計画です．たいまつ計画の主な目的は，開発した技術を生産に結びつけることです．期間としては，そんなに長くなく中期ぐらいの計画です．高度技術発展計画は，いま中国が持っている科学技術を集中して，国際交流をスムーズに行くようにするのが目的です．この計画はかなり長期です．十数年先の事を考えています．計画は，７つのグループに分かれており，情報処理，オートメーション技術，バイオテクノロジー，宇宙技術，レーダー技術，エネルギー技術，材料工学があります．国家自然科学基金は，その中にソフトウェア開発を含んでおります．第７次５ヶ年計画は，かなり幅広い計画でありまして，その中にソフトウェア開発に関する計画がいくつか含まれています．

ソフトウェアに関しては，アプリケーション，企業ソフト，人工知能，ソフトウェア工学などの分野があり，ソフトウェア工学分野では，開発支援環境の構築が大きな目的になっています．その環境は，もちろん，デザイン，テスト，データベース，インターフェイスを含んでいます．それ以外に新しい技術として，再利用，自動生成，新言語などが注目されています．具体的なプロジェクトとしては，北（北京）には北斗，南（上海）には啓明という２つの大プロジェクトが走っています．それから８０年には，ISEE（統合ソフトウェア工学環境）という計画がスタートしました．これらのプロジェクトは，相互に関連を持ち，それらの成果を最後にかがり火計画やたいまつ計画に結びつけて，一般の企業にトランスフ

ァしようとしています．

いま述べた ISEE 計画の中では，さまざまなソフトウェアツールを開発しています．計画がスタートしたのが８０年でしたから，すでにずいぶん成果が上がっています．あとでスライドで，そのうちのいくつかをお見せします．

これから，ソフトウェア開発技術の現状を，少し紹介したいと思います．先に述べたように，現在の中国における技術発展は，主として，新技術をいかに生産現場に結びつけるかがポイントであります．この目的を達成するために，国は，教授とか研究者に，研究成果を製品化することを奨励しています．具体的には，教授や研究者たちが自ら会社を設立し商売をする（！）というかたちです．大学教授とか研究員が，現職のまま，そうした会社の役職を兼務し，経営の仕事をするわけです．

これ以外に，国際合作というか国際交流にも，どんどん少し力を入れていくつもりです．この場をかりて，在席の皆さんと，合作の企画が作れればいいと思っています．私が知っている限りでは，中国では，かなり多くの組織単位や機関あるいは企業が，日本の企業と交流したいと願っています．私のいるセンターでも，そうした合作活動の支援をかなり広くやっていくつもりです．

それと，さっきのセッションでが討論されていたハイパーテキストやハイパーメディアですが，中国でも，これらの技術についてては，かなり興味を持っています．まだスタートしたばかりなので，たいした成果は上がっていませんが，さっき紹介した国家プロジェクトの中でも，音声処理とか，イメージ処理とかに関しては，かなり力を入れています．これらの分野でも今後，日本の皆さんと協力していきたいと思っています．

座長：　どうもありがとうございました．それじゃここでスライドで説明していただきます．

［スライド説明］

座長：　それでは，鐘先生のお話についての質問を，先に受けつけましょう．国家的なレベルからお話をいただきましたので，大きな話についての御質問がありましたらお願いします．まず私から．いま見せていただいた画面には，ぜんぜん漢字は使われていませんでしたけど，実際あまりまだ漢字は使われていないのですか．

鐘：　今回は，海外でデモするために英語版のものしか持ってきませんでしたが，漢字版のツールももちろん開発されています．たとえば，いまお見せした EASYCODE というツールを使って，アジア大会の競技データ処理システムを作っています．

杉田（ＳＲＡ）：　前回ＳＥＡの日中シンポジウムで中国へ行かせていただいたんですが，その時，かなり数多くの共同研究が，中国と海外との間で行われていることを知ったのですが，日本との合作研究はどの程度なのでしょうか．

鐘：　具体的な事例は，申し上げにくいのですが，アメリカとかヨーロッパに関しては，国家レベルでそういう協定を結んでいます．日本に関しては，そういう条約がまだないありません．一日も早く，そういう条約を結びたいと思っています．

座長：　東欧諸国との共同研究なんかはあるのですか．ソ連とか，ポーランドとか，そういう共産圏との共同研究ですか．

鐘：　ソ連とか東ヨーロッパに関しては，資料交換程度しかしておりません．

熊谷（ＰＦＵ）：　いろんな計画の名前が出ましたが，その中で一番重要というか，国家が力を入れている計画は，どれですか．また，その中で，ソフトウェア産業の占めてる割合はどの程度なのでしょうか．つまり，ソフトウェア産業を重点的に育てていこうとしているのか，それとも，もうちょっと様子を見てやろうとしているのか，そこらへんの微妙なニュアンスを知りたいのですが．

鐘：　皆さんご存じように，中国では，ハードウェア技術がかなり遅れておりまして，経済力に関しても限界があります．しかし，ソフトウェア開発に関しては，かなり潜在的な能力が高いと考えられるので，政府としても，ソフトウェア開発に，かなり高いプライオリティを与えて，その計画を進めております．

野中（日本電気）：　国産のマイクロコンピュータを開発するという計画はありますか．

鐘：　あります．現在は，すでにＩＢＭ／ＰＣと互換性を持つ長城という機種を生産しております．

野中：　ＰＣではなく，チップ自体の製造はどうですか．

鐘：　今は，まだ海外から購入しております．

2.2　朱三元先生

座長：　それでは次に，上海電子計算機ソフトウェア技

術開発センターの，朱三元先生からお話をお伺いいたしましょう．通訳は，静岡大学の方（ファン）さんにお願いします．

朱： 私の名前は，朱三元（シュサンゲン）といいます．簡単に履歴を言います．１９５５～１９６１年までは，中国の清華大学の学生として，コンピュータのアーキテクチャ設計を勉強していました．ソビエトの研究者たちと一緒に，数学的な計算の仕事をやりました．最後は，インタプリタシステムを開発しました．

１９６１～１９７２年までは，中国科学院の計算技術研究所に勤めていました．主に言語（ＡＬＧＯＬ６０）の研究をしました．中国製のコンピュータでコンパイラを作りました．また，ＯＳの研究も始めました．

１９７２年から上海計算技術研究所に勤めました．ここでは，主に計算機のアーキテクチャーの設計とか，ＣＯＢＯＬコンパイラの開発を手がけました．また，上海計算技術研究所の自家製のコンピュータのＯＳも作りました．７０年代後半からは，データベースの研究を始めました．１９８２年には，中国の全国のデータベースの技術交流会を開催しました．今も，ソフトウェア工学についてのテーマをやっています．

１９８４年からは，上海計算機ソフトウェア技術開発センターに勤めています．今は，主に技術の標準化とか，技術交流に関する仕事をやっています．

今日は，主に中国におけるソフトウェア工学関連の標準化について御紹介したいと思います．こうした国家標準を設定する時は，もちろん外国の標準化動向を参考にしています．

中国には，ソフトウェアの標準委員会があります．１９８０年ごろから活動を開始し，１９８６年には，ソフトウェア技術に関する３つの標準を設定しました．今年７月には，国家標準局から計算機ソフトウェア製品開発ドキュメントの標準が公布されました．ＧＰ８５６６８８と呼ばれるものです．他の標準はまだ現場にはおりてきておりません．

標準を設定する時は，いろいろの方針がありますが，われわれは，なるべく国際標準に近い方がいいと思っています．したがって，もしすでに国際標準が存在すればそれをそのまま採用します．なければ，アメリカの標準（ＡＮＳＩやＩＥＥＥ）を利用します．それもなければ，大きな会社の標準を参考にします．これが基本方針です．もちろん，まだ研究開発の途上にある技術に関しては，そうした標準化は行ないません．

たとえば，今，ソフトウェア工学の分野では，ライフサイクルだけを標準化しています．他のものはまだ標準化されていません．国家標準の意義や役割については，いろいろの意見がありますが，結果として，それはある種の指導原理みたいなものだと思います．国家標準以外に，それぞれの地方での標準も許されています．また，各企業や組織の中では独自の技術標準があります．それらは，国家標準と矛盾がない限り特に問題にはなりません．

座長： ありがとうございました．質問を受けます．

池谷（富士通静岡エンジニアリング）： 私どもの会社では，現在，上海のある大学と，中国語プログラミング言語というテーマで共同開発をしています．共同開発というよりも，まず調査依頼ですね．ですが，日本では，日本語プログラミングは非常に生産性向上に寄与すると考えられています．また，保守性等も非常によい．中国ではどうでしょうか．また，そういった中国語プログラミングは可能かどうか．その辺のお考えをお聞かせください．

朱： 中国の国家標準局には，いろいろの分野の標準分科会があります．ソフトウェア工学の標準化はその１つです．言語の標準化委員会もあります．そこでは，ＡＬＧＯＬ，ＦＯＲＴＲＡＮ，ＣＯＢＯＬ，ＰＡＳＣＡＬなどの標準化が進められています．中国語プログラミングについては，まだ，それほど国内の中での研究は進んでいないように思います．

池谷： 続いてもう１つ．中国語の入力方式は，いろいろ研究されていると思うのですが，何かいいアイデアはありますか．

朱： 入力方式は，およそ４００種類あります（笑い）．これらの各方式の優劣についての評価はなかなか難しいようです．

海尻（信州大学）： ＡＬＧＯＬの話が出たのですが，中国でＡＬＧＯＬかなり使っているのですか．というのは，今から１０数年前に，大学でＡＬＧＯＬを使おうとしたら，センターができてから数年たっていたにもかかわらず，それまで誰も使った人がいなくて，私が初めてだったのです．日本ではその当時でもほとんど使われていなかったのではないか，という気がするのでちょっとその辺おお聞きしたいのですが．

朱： 中国でも，現在はＡＬＧＯＬを使っている人は少なくなっています．いまは，主にＦＯＲＴＲＡＮとＰＡＳＣＡＬがよく使われています．

座長： 野村さん，うちの会社はまだＡＬＧＯＬを使っていますね．

野村（ＪＩＰ）： もちろん（笑い）．

座長： 先程，ソフトウェアの設計に関する標準化がなされているようなお話があったと思うんですが，実際ドキュメントとしては，例えばフローチャートとか，ＮＳチャートとか，さっき，何先生に見せていただいたＰＡＤとか，いろいろな種類がありますが，主に，どういう技法が推奨されているのですか．

朱先生： 基本原則はさっき言った通りです．何らかの選択が必要な時は，なるべく国際標準に近いものを選ぶということです．また，どこかの国または企業が採用しており，効率がよいということがわかっていれば，それを採用します．フローチャート記号などは，まったく国際標準と同じものを採用しています．

座長： いま，国際標準はどうなっているのかな？ フローチャート以外に，何種類かが申請されていますが，あれは全部標準化されたのかしら．だれかご存じの方．はい．

久保（富士通）： いや，質問だけ，今のとは関係ない．

座長： ご存じの方．

久保： 何？

座長： ドキュメントの記述法で，国際標準では，何と何が決まっているのですか．

久保： ドキュメントって何？

座長： フローチャートとか....

久保： ＨＣＰとか.....

座長： 日電のＳＰＤなんか....

久保： そういったものは，標準じゃなくって参考になっているんですか？

座長： そういう風にきいています．国際標準として決まっているのは，いまのところフローチャートだけだと思います．したがって，中国ではフローチャートが中心になってると考えてよろしいのでしょうか．

朱： 一応，国際標準の中では，ＩＳＯ５８０７の標準を採用しています．これは，情報処理についての標準なんです．その中には，プログラムフローチャートとか，システムフローチャートとか，ネットチャート，試験チャートなど，いろいろあります．

久保（富士通）： 標準化の徹底のさせ方について，一般的な質問をしたいと思います．標準化の大事な目的の１つに，産業界の生産現場の技術水準を引き上げる，ということがあります．日本の場合，ＪＩＳというものが，多分，終戦直後に作られて，ＪＩＳマークを取ることの競争が，特に，製造業の分野で行われて，現在の日本の製造技術の発展，それから製品の品質の向上が確保されたと思うのです．その意味で，日本の場合には，標準をうまく作って成功した．そこで，中国に関して気になる事があるのでお聞きしたいのです．

この間中国へ行った時に，上海の計算機工場を見学させてもらいました．見学しのは一部だけなので，それから全体を推測するのは，失礼だとはと思うのですが，正直にいって，それほど高品質の製品が作られているという感じがしないかった．現場の水準を引き上げるには，とにかく標準を決めたら，それをうまく徹底させるメカニズムがないと，いくらいいマニュアルをを作っても意味がないんで，その辺について，一般的にどういう仕組みをお持ちなのか．ある意味で中国の場合には，トップダウンのアプローチがかなり有効な国ですから，やりやすい面もが結構あるんじゃないかと思うんですけども....

朱： 中国の標準は，分野によっていくらか違いますが，半分はトップダウンで現場におろされて行きます．あとの半分は，生産工場とかの現場が，それぞれの必要に応じて，ボトムアップで標準を決めています．

ソフトウェアの分野でも，標準化はやはりトップダウンが原則です．中国のソフトウェア産業は，まだまだ未熟ですから，いろいろな標準をトップダウン方式で制定しなければなりません．そうした標準は，主にある種のガイドラインみたいなもので，具体的な細部は現場の自由裁量にまかされています．たとえば，ソフトウェア開発のライフサイクルに関しては，きちんと標準化がされていますが，その各段階に対応して，ドキュメントが存在します．そうしたドキュメントの書き方は，多分，ケースバイケースで違うでしょうが，ライフサイクル標準は各段階において，何らかのドキュメントが必要なことだけを規定しているのです．

標準化の普及について，上海ソフトウェア技術開発センターでは，セミナーを開催して，そのＰＲに努めています．今年は，３回ぐらい実施する予定です．

久保： いま僕自身は，ソフトウェアの限定せずに，もう少し一般的な質問をしたつもりだったんですが，ソフ

トウェアのことが主だったんで，ちょっとだけ意見を言わせて下さい．ＪＩＳは，日本の産業の発展に非常に寄与したのですが，ソフトウェアに関しては，ほとんど機能しなかったと思います．開発の現場の技術水準を引き上げるという意味では，今のところ効果が出ていない．多分，ソフトが難しいからだと思います．むしろ，ユーザ保護の立場のための標準という観点で考えた方が効果があるんじゃないかというのが，私の意見です．

座長： はい，ありがとうございました．ちょっと，時間もせまっていますので，この辺で，質問を打ち切って，最後のゲストである何克清先生からお話していただきます．何先生は，日本語でお話になりますので通訳はつきません．

2.3 何克清先生

何： 私の名前は，何克清（カコクセイ）と申します．７年前，日本に留学したことがあります．所属は，武漢大学ソフトウェア工学国家実験室の主任およびソフトウェア工学研究所助教授です．

武漢市は，中国のちょうど真ん中です．交通は便利ですね，もちろん，日本と比べると，あまり便利じゃありませんが．１９８５年，中国政府の教育委員会の指示で，武漢大学にソフトウェア工学研究実験室が創立されました．この実験室の目的は４つあります．まず，ソフトウェア工学の基礎および応用技術の研究開発．２番目は，ソフトウェア工学に関する実験システムの開発．３番目は人員養成．４番目は実験室の設備を共同利用することです．

現在全国で各種の科学技術の実験室は５０か所以上あります．そのうち，ソフトウェア分野の実験室は２つです．１つは，ソフトウェア新技術実験室であり，．もうひとつが，うちの大学の開放型ソフトウェア工学実験室です．

開放型と名付けた意味は，単に学内だけでなく，広く国中の研究者，学者，学生たちがこの実験室を利用できるということです．自分の研究計画を提出し，それが実験室の学術委員会のレビューでＯＫされればよいのです．この実験室の学術委員会は，国際的であり，アメリカや日本の先生方もメンバーにはいっています．組織的には，この実験室は独立機構になっていますが，いろいろ事務処理は武漢大学に頼んで管理してもらっています．現在，研究者の数はあわせて５５人です．

今までの研究・開発プロジェクトについて，簡単に紹介させていただきます．まず，ソフトウェア工学基礎研究についてですが，並行処理アルゴリズム，プロトタイピング開発のモデル，オペレーショナルモデル，データモデル，ユーザモデル，オブジェクト指向開発方法論，部分計算法，ソフトウェア信頼性の評価，定量的評価などです．

応用技術研究についても，いろいろやっていました．例えば，先程朱先生が，お話された標準化の研究にも参加しました．開発試作の分野では，統合化ソフトウェア開発環境ＭＥＬＯＮやその他の実用的ソフトウェア工学ツールの開発を，ＩＢＭ／ＰＣ，あるいは「長城」を用いて行ってきました．ソフトウェアの理解・文書化ツール，テスト・ツール，詳細化設計ツールなど，いろいろなものを開発しました．また，ＰＯＰＥと呼ばれるプロトタイピング支援環境も開発しました．

次は，応用ソフトの開発について簡単に説明します．まず，工場の企業経営管理システムを，ＩＢＭ／ＰＣとＬＡＮを用いて作ってきました．例えば，中国第二自動車工場の経営管理システム，これはいま稼動しています．この他，洪水予想システム，短期天気予報システムなども手がけました．

ソフトウェア工学分野の新技術についても，いろいろ研究をしています．例えば，ソフトウェア開発向けの知識ベースを研究してます．しかし，この実験室の目的は研究だけではありません．実際に商品を開発し，それを実用化・普及することが大事です．ですから，最近，武漢大学付属の情報システム株式会社を設立ました．いま，この会社を窓口にして，国内の企業と一緒に，いろいろなアプリケーション・ソフトを開発しています．これからも日本あるいは，アメリカの企業と一緒に，協同開発プロジェクトをやってみたいとかんがえています．うちの実験室には，日本語ができる人が７人ぐらいいますので，日本の会社，研究者と一緒に何かやりたいですね．

さきほど話した統合化環境ＭＥＬＯＮは，うちの実験室の中での開発を支援する環境ですが，標準化や部品化，モジュール化の概念を充分に取り入れています．データモデルは，基本的にオブジェクト指向の概念にもとづいています．マシンは，ＶＡＸ／８６００システムを使い，ＯＳはＶＭＳを利用しています．構成要素ツールはまだ全部が完成しているわけではありません．これらのツー

ルは，もともと，みなＩＢＭ／ＰＣで開発したものを，ＶＡＸに移植したものです．

もう１つのプロジェクトＰＯＰＥは，プロトタイピング支援環境です．このプロジェクトは，最初の設計段階で１年間，アメリカ人の研究者と一緒にやりました．マシンは，ＶＡＸステーション，Ｘ－ＷＩＮＤＯＷシステム．ＯＳはＵＬＴＲＩＸ（ＵＮＩＸ）です．詳細については，お手元にお配りした武漢大学の研究紀要をお読みください．

この実験室では，研究者を募集するため，研究プロジェクトのガイドを毎年公開しています．国内の大学の先生，学生，大学院生，産業界の研究者，だれでもこのガイドにしたがって申込みをすることができます．そして学術委員会が，審査をしてＯＫになれば，本実験室の研究者として迎えられます．その他には，実験室が外部から共同研究の研究者を招待することもできます．国内で国外でもかまいません．

最後に，実験室の設備について説明します．ＶＡＸ８６００が１台，ワークステーション２台，イーサネットで全部，ＩＢＭ／ＰＣとも一緒につながっています．

座長： ありがとうございました．それでは，時間もなくなってきましたが，質問をお受けいたしたいと思います．

三浦（ＳＲＡ）： さき程，鐘先生の方から，企業と大学の人が一緒に研究するとか，先生が会社を経営するというようなお話がありました．また，いま，何先生が実際に大学の中に会社を作ったという話をされました．具体的に，そこではどんな研究や開発をされるのでしょうか？

何： 研究については，中国政府のいろいろ計画があります．これらの計画にしたがってプロジェクトをたてます．会社との協同開発は，相手側のニーズに応じて，たとえば，第二自動車工場の場合，これはトラックを生産している大きな向上ですが，そこの情報管理システム（営業・サービス業務を支援するシステム）を開発しました．そうしたアプリケーションの開発を将来は日本の会社と一緒にやりたいと思っています．

座長： 今のお話は研究ですね．商売じゃなくて研究ですね．

何： 研究です．商売じゃないです．

久保： ＰＯＰＥのことを教えてください．何先生は，世界各国の研究の最前線をよくご存じだと思うんですけど，先生から見て，ＰＯＰＥの水準が，世界的に見てどんな感じなのか．僕は，あまり，その領域知りませんので，どんなレベルの研究なのでしょうか？ また，応用領域としては，どういう所にお使いになろうとしているのか？ できるだけ，具体的にお願いします．何先生は，日本の事を，よくご存じなんで，日本の計算機の使われ方の中で，どういう所からみたいな感じの事を，教えていただければ理解に役立つと思います．

何： そうですね．１つ目の質問ですが，世界的はよくわかりませんが，アメリカの研究者と一緒に最初の１年間やりましたので，ある程度のところまでは到達していると思います．このシステムの特徴は，オペレーショナルモデルを採用していることだと思います．

また，応用領域としては，やはり情報管理システム，例えば，会社の或るいは，工場のデータ処理システムのプロトタイピングを狙っています．リアルタイム・システムかもいずれは手がけようと考えてはいますが.... **座長：** はい．ありがとうございました残念ながら時間がきましたので，この辺でセッションを終わりたいと思います（拍手）．

3.「中国セッション」に関する感想

藤岡　卓（三菱電機）
逐次通訳ではどうしても時間がかかりますね．スライドの説明がもう少しあれば良かったと思います．

杉田義明（ＳＲＡ）
Ｑ＆Ａ時間が少なかったし，資料がなかったので少し盛り上がりに欠けた．もっと共同研究の可能性などのテーマで深くつっこみたかったが，どうせ噛み合わないと思ったのでやめた．

飯塚正樹（ＹＨＰ）
中国のソフトウェア開発の現状についてＵＮＩＸやＸｗｉｎｄｏｗといった言葉が多く出て，より身近かなものとして感じられるようになった．

野見山和則（三井銀ソフトウェアサービス）
インターフェイスの難しさから，内容がつかみづらかった．

黒坂靖子（ＪＩＰ）
話としてテーマが大きすぎて実践的なものが見えてきませんでした．中国語はまったく分かりませんので，少なくとも英語ぐらいでやって頂きたかった．お互い漢字を使うという独特の世界を持つのだから，何かそういう方面での問題点とかが分かれば良かった．

広瀬隆夫（ＪＩＰ）
ソフトウェアの標準化が中国の国家プロジェクトとして進められていることを知った．日米に比べて，まだプログラム資産の少ない時に，そのような標準化が行われることは大いに意味があるのではないかと思った．

久保宏志（富士通）
Ｐｒｏｄｕｃｔｉｖｅセッションではなかった．火曜日に東京でやったＦｏｒｕｍの方がはるかにＰｒｏｄｕｃｔｉｖｅであった．

中野秀男（阪大）
久しぶりに中国語を聞きました．たった２週間でも中国のいろいろを見たのでそのバックグラウンドをもって聞くのと，何も知らずにこのセッションを聞くのでは全然違うでしょうね．

古閑幸一（ＨＳＴ）
中国と日本のソフトウェア事情がどの位ギャップがあるのかはあまりつかめなかった．日本以上の漢字文化圏であるので教わる点は多いのでは？

上妻健一郎（電応研）
標準化の話は，中国においては日本ほど容易でなく，むしろ国際標準を作るような難しさを持つだろうと思った．

濱田　勉（ＮＴＴ）
ＩＢＭコンパチ機を作られているとのこと，中国の文字入力について興味がある．１６ｘ１６で間に合うのか．アジア圏の文字入力にあった独自のアーキテクチャを，日本と一緒に考えられないのか．

熊谷　章（ＰＦＵ）
鐘先生：中国の政策が分かりなかなか面白かった．少し時間が足りず残念．日中の間の契約が国の間で結ばれていないのはおかしい．直ちに結ぶべきだ．
朱先生：標準化の話がよく分かった．具体例があればもっと分かりやすい．
何先生：解放型実験室の試みは非常に面白い．１９ものプロジェクトが稼働しているのには驚いた．うちの会社もやってみようかな．そのためには，マシン設備を何とかしないと！

海尻賢二（信州大）
資料の説明という感じでわかりにくかった．

中田修二（シグマ）
実際にどのような状況にあるのかが，プレゼンテーションでは余り知り得なかったので，この点が残念であった．武漢大学の何先生の話は分かりやすかった．

坂本　治（シグマ）
意外に幅広く研究されていることが分かったが，もう少し範囲を絞って説明して欲しかった．

臼井義美（ＪＩＰ）
同じ漢字文化圏として，アルファベット文化圏と異なる独自のソフトウェア開発環境が提示できるようになりたいですね．

野村行憲（ＩＣＳ）
・自分の思っていたもの（かなり遅れている）と実際（実はかなり進んでいる）のＧａｐを感じた．
・ただし，それがごく１部ではないのか？　一般への広がり具合いはどうか？

野中　哲（ＮＥＣ）
？　？　？

北野義明（ＫＣＳ）
民間よりも国家が先導しているようで，日本とどういう違いがあるのかもう少し詳しく知りたくなった．

安間文彦（富士通エフアイピー）
次回は，中国のソフトウェアハウスまたはＣｏｍｐｕｔｅｒ関連の産業の現状についての話を聞きたい．

今別府芳暢（ＮＴＴ）
入力方法が４００通りもあって大変だと思う．漢字を使用している日本の協力できる事があるのではないか．

高橋哲也（神戸製鋼）
概要や大略の説明が多かったが，中国における具体的技術内容の現状をもっと紹介して欲しかった．

4. セッション・サマリ

臼井義美（日本電子計算）

私は３日目のＣＡＳＥセッションのチェアマンを担当する予定だったが，ボストンで行なわれるＳＤＥ３に参加するため，急遽日中セッションに予定を変更していただいた．私は，１０月に中国で行なわれた日中ソフトウェアシンポジウムに都合で参加できなくなり残念に思っていたが，今回中国から参加していただいた先生がたから，中国におけるソフトウェア工学の現状をお聞きでき，いろいろな意味で大変興味を覚えた．

このセッションの発表を聞いて，西欧と日本における情報社会への移行と中国における情報化の動きについて気がついたことを書いてみたい．

今回は主に中国の大学におけるソフトウェア工学の研究成果についてお話しいただき，少なくとも研究レベルでは日本と同じようなテーマが対象になっているように感じた．ソフトウェアに関しては世界の共通言語としての性質や，情報化社会への発展段階としての同時性を持っているので驚くには当たらないが，少なくとも文化，生活様式の欧米化が極端に進行している日本と，多くの点で独自性の残っている中国との，ソフトウェアに対する取り組み方がどのように異なるのかをもっと知りたかった．

ところで，社会制度や経済状況が日本のそれと大きく異なる国において，情報化というのはどのような役割を果たすのだろうか．特に日本に比べて人件費が安く労働力が豊富であること，共産主義という計画経済を基盤とした国家指導による産業や企業活動，さらに個人の社会や産業に対する考え方の相違などが，情報化社会の構築に向けてどういう影響を与えるのだろうか．

他の多くの国と同じように，中国でも国家の近代化を目指すための手段としてコンピュータを中心とした情報化社会への移行を必要としており，それに対して積極的に大学を始めとした研究機関で研究に取り組んでいこうとしているのは理解できる．しかし，欧米や日本では工業化を完了した後での情報化移行であるが，中国では工業化と情報化を同時に進めようとしているかに感じられる．

ソフトウェア開発の技術が突出してもそれを利用し役立てる社会の受入体制が整わなければ，本当の意味での情報化につながらないと思うが，中国ではそれに対してどのように対処しようとしているのだろうか．既に，中国では資本主義を経ずに社会主義を実現しようとして多くの実践を行なった経験を持っているので，その経験を有効に活かして独自の情報化社会を実現できるかも知れない．

しかしながら中国では，社会主義実現の過程で様々な問題が生起し，現在も未解決の問題を数多くかかえているのと同じように，工業化を経ずに情報化社会を築くことは他の国々が経験しなかった多くの問題を解決しなければならないだろう．

もちろん，日本が情報化社会を目指してすべて成功したと言えるかどうかは疑わしい．経済的には成功したと言えるかも知れないが，多くのソフトウェア技術者にとって地価の高騰で我家さえ手が出なくなり，残業が日常茶飯事となっており，テクノストレスに悩まされている．一言で言えば集団犠牲の上に繁栄が築かれていると言っても過言ではない．これからは個人が求めているものを，いかに実現していくかが日本の情報化社会の課題でもあると思う．

もともと日本人は数千年の昔から非常に好奇心が強く，あらゆる外国の文化や技術をブラックホールのごとく吸収した．つまり，情報の獲得，加工は得意なのだから情報化社会の構築はお手のものなのだ．だが，情報の発信という面では経験がないので，今後の情報社会におけるリーダ役としてはあまり期待できないかも知れない．

一方，中国は太古の昔から中華思想が身に付いており，やすやすと外国の文化や技術に迎合してこなかったので，ひょっとしたら，それが災いして情報の収集がスムーズにいかなかったり人々が受け付けなかったりして，情報化社会への移行の阻害要因となるかも知れない．

いずれにしても，私にとって中国の情報化は非常に興味深いものである．

ところで，現在のコンピュータおよびソフトウェアは他の多くの科学や工業と同じく西欧合理主義の成果であり，その思想の論理性，合理性の上に開花したものであ

ることは周知のとおりである．しかし情報社会におけるコンピュータおよびソフトウェアの役割は単に計算機としての高速演算や大量のデータ処理だけでなく，さらに多様な知識を取り扱うとともに，もっと複雑で曖昧な対象に対しても積極的に取り組んでいかなければならない．

そのためには，最近の物理学などに見られるように，西洋思想のもとで発展した科学のうえに，東洋的思想をベースとした独自な視点から新しい形の科学を創造する必要があるのではなかろうか．西洋と東洋のものの考え方の融合は，情報社会の在り方について特に有用であるような気がしている．

その一つに文字がある．つまり，文字は単なる意味の記号化の手段というだけでなく，利用している民族の文化や思想に大きな影響を与える．というより，文化や思想の凝縮されたものが言葉や文字に現われたと言ってもいいだろう．そして情報のほとんどの部分が文字で取り扱われる．

さて，中国や日本は世界的にも非常に特徴のある表意文字である漢字を利用している．ある人によると，西欧のアルファベット文字をベースとした表音文字を「ラジオ型文字」，漢字などの表意文字を「テレビ型文字」と呼んでおり，漢字は文字自身が持つ透明性，つまり文字の組み合わせからその意味を類推させる機能に優れているというのである．単に音の記録としての文字というだけでなく，文字の持つイメージを積極的に利用することによって，ソフトウェアの世界をもっと魅力的なものにできるのではないだろうか．

ただ，日本語は仮名漢字変換を行なうことによって比較的容易にキーボードによる文字入力が可能であるが，中国語では入力方式が数百もあると聞いて呆然とした．もしろん，現在のキーボードが日本語の入力にどのくらい適切であるかには異論もあろうが，現在のソフトウェアの開発に英文字の利用が不可欠であることを考えると，ローマ字仮名漢字変換は日本語と英語を比較的自然に入力できる方法として便利なものであろう．

中国語でも，現状の入力機器ではやはりこれに近い方法が便利であろうと思われるが，近い将来やはり自国語での優れた入出力のために何らかの画期的な方法を発明しなければならないと思う．いずれにしても他人事ながら，早急に入力方法を統一化しないと前に進めないのではなかろうか．幸い漢字の出力方法は既に確立されているのだから．

いずれにしても，東洋の代表的な国として日本との文化的共通性も多く，西洋文化には見られない様々な独自性を持つ国の研究者同士が，これまでの情報処理に適用されてこなかった科学または文化を取り入れ，より多様性のある情報社会に応用できれば意義があるのではないだろうか．そのためには，今後ともさらに多くの機会をとらえ，日中の技術交流，文化交流を行なっていかなくてはならいと思う．次の再会を楽しみにしている．

第4回　ＳＥＡ環境ワークショップ

ツールの説明とデモの感想

ツールの説明に関する感想

中野秀男（阪大）
時間がないので、せわしなかったのはしかたがない。

今別府芳暢（ＮＴＴ）
感覚を広く配置し説明を聞き易くして欲しい（隣の話が聞こえる）

野見山利則（三井銀ソフトウェアサービス）
ＳｔＰが面白そうだ。ＡｌｇｏｒｉｔｈｍＡｎｉｍａｔｉｏｎが教育に使えそうだ。早く、ＰＤＳになることを期待しています。

藤岡　卓（三菱電機）
時間が短くて全然わからない。

高橋哲也（神戸製鋼）
デモだけでよい。
または、デモを行いながらのＰｒｅｓｅｎｔａｔｉｏｎという形をとったほうがいい。（プレゼンテーションだけでは何だかわからない）

海尻賢二（信州大）
ＯＨＰ画面に直接出してもらえば分かりやすかった。

熊谷　章（ＰＦＵ）
発表は簡明で良かった。

辻本恵一（シグマ本部）
・短すぎて内容が理解できない。
・デモする対象の説明をしたほうがよい。

坂本　治（シグマ本部）
紹介時間が短く、内容がよく分からない。

中田修二（シグマ本部）
デモ紹介が短く、要領良く行われて良かった。

平尾一浩（ＨＳＴ）
ＰＷＢ、ＳｔＰに興味あり使ってみたい。（実際のスケールで）

古閑幸一（ＨＳＴ）
ビジュアルＭＭＩ作成ツールに興味あり

濱田　勉（ＮＴＴ）
ＶｉｓｕａｌＭＭＩに興味があった。

上妻健一郎（電応研）
ＶｉｓｕａｌＭＭＩ、ＰＷＢに興味があり。
プレゼンテーションの時間をもっと取るべき。

山口郁生（電力計算センター）
さすがに５分（１分？）だと内容がよくわからない。ツールの名前、使用機種、デモの場所、一言で言うとどんなツールなのか、を発表すればよいと思う。

ツールのデモに関する感想

北野義明（ＫＣＳ）
ｔｏｏｌについては今すぐ身の回りで用いることの出来るものを中心に見て回った。ＤＦＤを上手に表現したものが、個人的には気にいった。

田中正則（ＳＲＡ）
ＬＩＮＣでユーザ主導型という考え方があるようだが、本当にエンドユーザに要求仕様を任せてよいのか？

藤岡　卓（三菱電機）
・みんなそれぞれのマシン（ウィンドウ、メニュー、マウス、キーボード、・・）の上で自分のツールのデモを見せてくれます。自分では、気に入っているようではＭＭＩは難しいですね。
・各ツールはとてもビジュアルなのに、裏で（コンソール上で）ｃｄとかｌｓとかをキーボードからやっている・。

上妻健一郎（電応研）
ＦＸＩＳのビジュアルＭＭＩには感銘を受けた。これならリアルタイム系のプロトタイピング化にも十分使えると思う。
電脳机のリンクが実はＵＮＩＸファイルやＣシェルコマンドの指定であるところは面白い。ＵＮＩＸ上のオーサリングツールというステータスを十分生かした仕様の切り方で、ＹＨＰの方の思い切りの良さには感服した。
ＳｔＰはあまり時間をかけて見られなかったので残念だが、データディクショナリの生成ができると聞いて驚

いた。

岩井田浩章（電力計算センター）
電脳机のイメージ処理は、キャラクタでものを考えていたものにとっては情報整理をする上で大変参考になった。しかし、はっと思ったと同時に「もっとリンクが自動的に出なければ面倒くさそう」（勉強不足で申し訳ありませんが、私はできないと思っているのです）なんてわがままなおねだりをしたくなった。とにもかくにもマックがほしくなってしまいました。

古閑幸一（ＨＳＴ）
ＳｍａｌｌＴａｌｋを使用したＶｉｓｕａｌＭＭＩが良かった。しかし現場にどうやって導入するのか？どう利用していくのか？電脳机：欲しかったツールの１つです。ペーパレスではなく、ペーパを使うことによって発生する余計な作業をなくす、という点に注目。

関口純一（ＪＩＰ）
ＨｙｐｅｒＭｅｄｉａ（電脳机）のコンセプトにはひかれるものがある。

坂本　治（シグマ本部）
各デモのサマリを記述したペーパあるいはパネル等を用意して欲しい。

中田修二（シグマ本部）
時間、展示数、各展示における説明など適切であったのと思います。各デモの名札や１、２枚のｈａｎｄ－ｏｕｔを用意して欲しいという希望があります。

鐘　友良（ＦＸＩＳ）
ＰＷＢ、ＳｔＰのほうは面白いし、良く説明してくれるし良かったと思います。ＳｔＰにはプロトタイピングをどのように処理しているのか？まだ分かりません。

野中　哲（ＮＥＣ）
Ｍａｃのふくらむゴミ箱が面白かった。ＳｔＰを使ってみたい。本当に便利なのでしょうか？

何　克清（武漢大）
・ツールはだんだん実用的になり、ソフトウェア開発に役に立ちます。感心しました。
・ＰＷＢの多ビューから支援するソフトツール研究の発想は、ソフト開発の上流段階にとても適用するアプローチであると思います。
・ソフト開発プロセスについての管理し、評価するツールシステムは大切だと思います。

海尻賢二（信州大）
ＯＨＰｉｎｔｅｒｆａｃｅを用いての全体としてのデモを軽く一度説明してもらえれば良かった。

藤野晃延（ＦＸＩＳ）
・なかなか種類が多く、見栄えのするツールプレゼンテーションになってきたと思う。
・ＷｏｒｋＳｔａｔｉｏｎＢａｓｅになってきたこともあり、移動可能な開発環境になってきているのだろう。新しいＳＤＥとしての方向が見えだしているように思う。
・技術的な説明、可能性としての説明、この２通りが明確になっていないように感じた。この辺を明確にできれば、ＴｏｏｌＰｒｅｓｅｎｔａｔｉｏｎとして、独立できるのではと思う。

杉田義明（ＳＲＡ）
討論会場の後方にデモコーナをおいたことで、つまらない発表のあいだや気分転換をはかる上で、リラックスして見学できた。デモ内容の発表は、もう少し内容を詳しく話して欲しかった。

林　　香（ＳＲＡ）
ハードが本当にやすくなったなー。
基本ソフトも本当にやすくなったなー。
デモソフトのレベルも高くなってきたなあ。
でも、早くラップトップＵＮＩＸワークステーション
　　ラップトップＭａｃが出て欲しいなあ

渡邊　雄一（電力計算センター）
電脳机はすごく便利！。感心した。でもドキュメント・サポート・プロセッサで終わってしまう危険もあるような気がする。何でもイメージで取り込めば良いわけではないから，きちんとテキストとして支援を受けた方が良いものに対して，いかに柔軟できめ細かい枠組みを（もしくは利用者が用意出来る仕掛け）を用意出来るかが鍵でしょう。でもスゴイ！。頑張って他のＷＳにもインプリメントとしてねェ。

山口郁生（電力計算センター）
ＷＳの画面を見れたことで、それぞれのシステムの「感触」を得られたことが良かった。自分で操作できればもっと良かった。（Demonstratorからマウスを取り上げるほど、図々しくできなかった。）

第4回　ＳＥＡ環境ワークショップ

全体の感想

小野康生（三菱電機）

初めての参加で、Ｓ／Ｗといっても思っているイメージがまるで違っていることに気づきました。思ったことをズバズバ言える雰囲気がとても良かったです。

岸田孝一（ＳＲＡ）

そろそろワークショップのフォーマットを変えた方がいいのかな。たとえば、第１回目にやったようなグループ討論方式をもう一度試みるとか。

野村行憲（岩手電子計算センタ）

ＢＯＦの盛り上がりとＴｏｏｌ展示が面白かった。会場も良かった。

北野義明（ＫＣＳ）

今回はメインフレームの話題が予想以上に出て、かつＵＮＩＸやＷＳの方面の進歩も相変わらずであり、大変気持ち良く過ごすことが出来ました。ワークショップ後、各分野で情報交換をし合って、互いにレベルアップできることを期待しています。

桜井麻里（ＳＲＡ）

環境ワークショップも広まってきたなあと感じました。参加者の層が広がってきた。これはとてもよいことである。これからも広がることを期待しています。

田中慎一郎（ＳＲＡ）

正式参加者じゃなかったのと、以下の理由で個々のセッションに対する感想は書いていない。すいません。

初日はＳＩＧＥＮＶ作業をしていたので、全く聞いていないのでよく分からないが、ＭＭＩ、ＭＭＬ、ＣＡＳＥとも話題提供者の話もしくはその後の討論の方向がテーマとずれていたことがままあったような感じがした。（「全体として」です）

ワークショップの場合、討論の方向がテーマとずれること自体はなんら問題にならないと考えているが、どうもあまり建設的というか生産的というか、そう言った内容になっていかなかった部分を少々感じた。それでも個人の発言単位でみると結構良い話が聞けたような気がしている。

ＭＭＬのときの船と飛行機のアナロジーなどは批判的な人もいたようだが、私にはなかなかいいと思った。（ただＭＭＬという名前づけはもうあまり適当ではないのでしょうね）

ＭＭＩのサブタイトルが、プロセス／プロダクトの可視化ということだったが、ＭＭＩには他にも扱うべき内容がたくさんあるように思う。今回もその辺が変な風に混じりあってしまったのではないだろうか。なまじＭＭＩのセッションとは呼ばず、「－可視化－」のセッションとはっきりさせた方が面白かったのではないすだろうか。

あとやっぱり人数が多すぎるような気がする。前のように（知らないけど）小グループの並行セッションの方がワークショップらしいと思う。

田中正則（ＳＲＡ）

大変面白かった。ただ汎用大型機の人とＵＮＩＸワールドの人との意見食い違いがあまりにもむなしかった。双方の歩み寄りを期待したい。

藤岡　卓（三菱電機）

初参加でホットなテーマばかりで期待してきたのですが閉じた世界というか身内、同じ会社内の人たちの間での愚痴のこぼし合いという感じで、がっかりしました。

話題提供者や座長の方々の中に自分がどうしてここにいるのか分からないとか、ピンとのずれた内容とかがあって運営面にも問題があると思います。

有益な話題もたくさんありました。

上妻健一郎（電応研）

問題意識を持つメインフレーマの話が聞けて参考になった。ＦＸＩＳのビジュアルＭＭＩとＹＨＰの電脳机の

デモを見ることが出来たのは収穫だった。

柳瀬健一（ＫＣＳ）

面白いワークショップだったが、@神戸であったため仕事も忘れることが出来ず半分以上はＷＳをぬけだしてしまった。（残念）

ＷＳ初参加にして発表を仰せつかり、不安がいっぱいでＷＳにのぞんだが皆さん優しくフォローして下さったので安心しました。（発表に関しては低レベルで申し訳ございませんでした）

今回はＭＭＬ、ＭＭＩ・・と広範囲であったため、的が絞れず、もう少し分野を絞った集中討論をすれば良かったのではないかと思います。

黒坂靖子（ＪＩＰ）

結論的には良かったです。今後に少しでも＋になれば・・・と期待しています。

平尾一浩（ＨＳＴ）

全体を通してまだまだつっこんだ話をする必要がある。もっとつっこむために、来年に引き継いでいきたいと思う。しかし、年々、ネクタイの方の参加が増えて言っているみたいで、本音の部分が聞けなるようで心配です。

パーティも必要だけど、ネクタイをはずすためのＢＯＦを、もっとやった方がよいと思います。

今別府芳暢（ＮＴＴ九州）

開発に関する本音の話が聞けて有意義であった。自分の環境のレベルがまだ低いのと、初めての参加で意見を述べるまでにいたらなかったが、次会は何かを持って来たい。ホテルと会場がもう少し近い方が良かった。

濱田　勉（ＮＴＴ九州）

初めて参加しましたが、アカデミック、マネージメント、肉体労働各方面の人が立場（地位）を離れて（少なくとも表向きそう見える）それぞれの問題意識を論じるスタイルは大変新鮮で面白かった。そのなかでもＢＯＦは、かなり危険で癖になりそうです。次のワークショップでは課長や部長も引きずり込んでうりうりしたい。

古閑幸一（ＨＳＴ）

３日間のＷＳでいちばん楽しく（？）議論できたのはＢＯＦでした。ただ、理想と現実のギャップが余りにも離れていて、それがたくさんの面を持っているので話が発散し易いと思いました。

ＷＳではある程度ソフトぎょうかいのトレンドを感じ取る事が出来有意義でした。

次回は九州（熊本）で行われるとの事で、地元でもあることだし是非参加したいと思っています。

関口純一（ＪＩＰ）

初めて参加していろいろ勉強になりました。現場の仕事の改善の糸口になればいいなと思います。（実際にはまだまだ遠い感じ）

坂本　治（シグマ本部）

非常に刺激的で勉強になりました。

中田修二（シグマ本部）

ワークショップの雰囲気などがよく分かりました。意見交流が本ワークショップの参加者において十分されることが大きな目的ですが、この点では大変よいワークショップと思います。

プレゼンテーションについては、プレゼンテーションを担当された方により内容充実度合が違っているようです。具体的で整理したプレゼンテーションが増えるようになれば、更に良くなるとの印象があります。（ＭＭＩのセッションはプレゼンテーションの内容、討議ともにポイントを得ずに流れたとの印象があります）

鐘　友良（ＦＸＩＳ）

２回目の参加なので、比べると、第３回（長野）より良い（自分にとってわかりやすい）と思います。各セッションの設定、セッションの発表のレベル、座長の組織能力、方法など。（実は、たしか昨年と同じだったかも知れませんが、自分の日本語が進歩してきたか。）

今度も新しい発表・デモを期待しています。ＢＯＦは、１つだけで集中し過ぎじゃないかという気がします。２つぐらい作ればどうでしょうか。

セッション・アンケートは会議のフィードバックとしては良いが、各セッションに対して「発表」と「討論」を合わせて１つのアンケートでも十分だ（自分の日本語ではまだまだなので２つのアンケートを書くとなかなか・・）と思います。

久保宏志（富士通）

全体的には、かなり生産的な議論を持つことが出来たといって良いであろう。共通の関心事を、より本質的かつ困難な領域に誘導する効果ももったのでないか。

野見山和則（三井銀ソフトウェアサービス）

ＨｙｐｅｒＭｅｄｉａとMMLがおもしろかった。

ＨｙｐｅｒＭｅｄｉａについては、とりあえずＰＤＳも頂いたことだし

ＨｙｐｅｒＣａｒｄで遊んでみて理解を深めていきたい。

メインフレームユーザとして、日頃から問題だと思っていた事がMMLのパネル、ＢＯＦで活発に討論され有難く思います。はやくＮＥＷＳやＭａｃＩＩを買って、同僚をメインフレーム病から守らなくては・・・。

ＭＭＩ（ＨＭＩ？）のパネルでは、「人間にとって使いやすい」という本質的なことについて議論したかったですが、体調を崩してしまったために、問題提起が出来ず、少し後悔している。

いろいろ書きましたが、３日間とても楽しく過ごせたので次回（熊本？）もぜひ参加したく思っています。

何　克清（武漢大）

１．ツールｏｒ環境の実用のため、企業ｏｒ企業的開発機構とともにワークショップするのは、本ワークショップの特点の１つであると思います。実用化の問題はツールｏｒソフト・開発環境の研究に関する現在の大切なポイントです。本ワークショップは、これについて大変成功しました。

２．ＳＥＡの研究・開発方針と目的は我々のソフトウェア工学実験室の目的と合いますが、これからの交流と協力をお願いします。

この気持ちをＳＥＡのメンバーみなさんにお伝え下さい。

どうもありがとうございます。

海尻賢二（信州大）

少なくとも発表中の後ろの雑談（議論？）はどうかな？

大学の場とは全く違った側面からの話で興味深く聞きました。

全般に発表時間が長く、討論時間が短いのではないか。

発表時間の厳守（討論時間を十分取るため）。

学会式にｂｅｌｌを用意しては。

棚田眞司（ＳＲＡ）

新しい（開発に有効な）環境が出来たときに、すなおにそれを受け入れられるか？と聞かれたら、私は今のままでもできるのだから新しいことを覚えるのはいやだというでしょう。こんな事を言っている間は私の進歩はありません。

小前俊哉（三菱電機コントロールソフトウェア）

環境を構築するに当たりもやもやしていたものをはっきりさせるために出席しましたが、その結果以下の感想を持ちました。

１．私の考えはそんなに間違っていない。（いつか発表したい）

２．他人（特にＳＥＡ）に頼ってられないな。

２に関しては、かわいそうな私のような環境スタッフのやすらぎの場であるとおもっていたＳＥＡが、そうではない、ただいじめられ、悲しくなるだけだなと感じてしまったのが本音です。発表・質問される方々は、それなりに思ったことを実行できる恵まれた環境におられ、どちらかというと（若くても）管理組織的権利があり、いつもＨａｐｐｙ！

そのうち日本のＳ／W産業は、同じ能力センスを持っていたとしても、Ｈａｐｐｙな人とそうでない人の２人種に分かれてしまうのかな。ど一考えてもＨａｐｐｙになりたいけど成れなくて悲しい毎日を送っている人の方が多いと思うんだけれど・・・。（Ｓ／W産業の中心となるべき人は少ないだろうけど、支えている人たちはアホみたいに多い。その人たちはほとんどは、たいしたことのない無免許エンジニア。また今のマネージャもＳ／Wの世界では無免許が多いんじゃないかな）そういう人たちの技術／情報交流とやすらぎの場がＳＥＡであってほしいと思います。

今、ＳＥＡはＨａｐｐｙな人だけのサロンのような感想を受けました。ただし、私もＨａｐｐｙになりたいから、いかなる圧力にも負けるつもりはありませんが。

メインフレームは何故必要か？←こんなバカな議論はやめてほしい。Ｈａｐｐｙなひとがいかに何も考えていないかが分かったから良かったかな。世の中、その時（瞬間に）に必要なものだけ存在するわけはないと思う。技術やセンスだけでＳ／Wが発達するなら、苦労しないでしょうね。そこに存在するもの（材料）がいかに“カス”でも、なんとかするのがその時の最新技術でなんとかするのが（するのも）大切だと思っています。（ただ、

諦めの気持ちで絶対なくて、やるべき事はやってから）

林　香（ＳＲＡ）

強く興味を引かれるセッションと、全く逆のセッションのメリハリがついてきた。以前はわりと全てのセッションに普通程度の興味を持っていたような気がする。年とともにコンセントレーションが続かなくなったのだとしたら悲しい。がしかし、若い人の参加・発言が多くなってきているので期待している。いろいろな興味を持った人が参加し始めてきた（参加者がバラエティに富んできた）とすれば良いこと。

にしても少し大人しくなってきたような気がする。以前はもう少し、ワイワイガヤガヤの感じがあったのでは。今回、非常に素敵な場所（神戸）で開催されて素晴らしいが、やはり雪の中というのが、何か目に見えない刺激を与えていたのかな。それとも、都会でやるより山の中の効果なのかな。次に期待

藤野晃延（ＦＸＩＳ）

人数としてはちょうど良いくらいかな。ただし、本会議に並列してＢＯＦが走るように出来ると良いのではないかと思う。ＢＯＦの取りまとめなどはＤｙｎａｍｉｃになるので、事前に会場や人数のセッティングは難しい。それにＢＯＦの人数は３～５人程度となるだろう。深い議論をするためにも、このように小さい（？）ＢＯＦの活動が要求されるのではないか。

杉田義明（ＳＲＡ）

充実した設備、きれいなコンファレンスルーム、新しいホテル、天候、すべてが揃ったワークショップであった。

良い環境を提供することで、内容が豊かになることが実感できた。今回は特に中国からの３名の方が来日され、そのお世話も含めて、ローカルアレンジメントは大変だったと思う。

プログラムの運営においても、今までの経験を積み上げた上での改善がみられた。各セッションごとの面倒なアンケートの配布、回収や、討論・発表をじっくり集中できたことの意味は大きい。

内容については、特にＭＭＬの発表と討論が盛り上がったように感じた。従来、参加者がその討論内容とのミスギャップが問題になったことがある。それはＵＮＩＸに代表されるような環境構築のプラットホームが認識されていない状態で、いきなり討論が始まったものだから、メインフレームしか知らない人にとって理解に苦しむ事だったからだと思う。

その意味で、今回のＭＭＬは良かったと思う。メインフレーム上での開発環境は、１０年前の技術とほぼ同じだと常日頃考えていたが、そのことがはっきり実感できた。

野中哲（日本電気）

＜ハイパーメディア＞

この技術は、やっと黎明期を迎えた段階ものであると理解した。今後、これらの基礎技術のうえにどのようなアプリケーションが出てくるのか期待したい。

＜マイクロメインフレームリンク＞

１－２－３がたくさん売れて、それにたくさんデータが蓄積され、そのデータを何とか活用出来ないかというのが、マイクロメインフレーム出現の動機であったという話には、非常に説得力があった。メインフレームは、データーベースサーバーに徹すべしという主張は大正解である。

＜Ｍａｃの塵箱＞

ワークショップの楽しみの一つに、ツール展示のＭａｃに触ることがある。Ｍａｃは、雑誌の広告でしか知らない私でも、ファイルを削除するのは、アイコンを塵箱の上に持って行けばいいのを知っていた。そしてそれをやった瞬間何が起きたか！！！なんと、塵箱がブッと膨れたのである。塵箱を漁ったら、いかにも、今削除したファイルがでてきそうだ。こんな凄いMMIを、私はかつて見たことがない。

＜ボートピアアイランド＞

まずボートライナーに、運転手がいないのにビックリ。それにしても、不思議な街だった。整然として美しいが、とても寂しい。人の顔が無表情に見える。「華氏４５１」の世界だ。でも、私はこの街を嫌いではない。この街に住んでみて、自分の精神状態がどのように変化するか実験してみたい気がする。

渡邊雄一（電力計算センター）

ボートライナーに乗って行った会議上は，立派なところだった。（昔の大阪万博を懐かしく思いだしていたのは私だけなのだろうか？）　でも風が強くで寒いのは勘弁してもらいたい。

あっという間に３日間が過ぎてしまった。考えていた

ことの１割位しか，発言出来なかったように思い，余計にストレスがたまってしまった。

問題は整理されてきているのだが，実際にそれらに立ち向かう理論武装なり，経験不足というところか？。普段からブアーなマシンにしか接していない私としては，ますます世の中の進歩から取り残されていくようで気が気でない。このような距離を「計る」というよりは「感じる」だけで終わってしまっている現状を打破するためにも，多くの刺激を受けたことだけは，明らかである。

山口郁生（電力計算センター）

一言で言えば、「やっぱり、ＵＮＩＸユーザ」。

ＭＭＩのセッションを聞いていて、ＵＮＩＸのユーザ会に来たのか、と思ってしまいました。現在、ＵＮＩＸを使っているわけではないので、ＵＮＩＸの話をされてもピンとこないのです。興味はあって本を見たりしても、実際に仕事に使ってみないことには．．．（でも、システムダウンが恐い）

少なくとも「何でそんなツールを作成したのか」の理由がわからなかった。ＵＮＩＸの走るＷＳなら、良い開発環境があって当り前なんじゃないのか、と思ってしまう。（まぁ、誤解でしょうが）

もっと何か、新しい考え方、発想、アイディアがほしい。別に、学術論文が聞きたいわけではなく、こうしたら、便利になりますよ、といった発想の転換が聞きたい。（と思っていたけど、主旨とは違っています？）

そう言った意味では、「電脳机」は、おもしろかったです。イメージデータにボタンのリンクをつける、といった考え方は「なるほど」と思わせるものがありました。例えば「時刻表」は記載されている数字をキーインすると大変な労力がいるが、イメージにすれば、それほどでもない（はず）。

重要なのは、人間が見てその数字の意味がわかれば良いのであって、計算機のわかる情報（ＪＩＳコード等）にする必要はない、ということだと思います。そのイメージデータを簡単に画面に呼び出す方法は、表示画面を直接クリックすればよいわけですね。応用範囲がかなりありそうだ、と感じました。（できれば個人的にも使ってみたいです）

それにしても、メインフレームの「悪口」には、まいった。私自身は、メインフレームが好きではないし、まして使いやすいとは思いません。が、「腐っている」といわれれば「いや！そうではない」と反論してしまう。いったい、この心理は何なんでしょう？

どうも、「うーーー」とうなるだけで、あまり人とお話ができなかった。言いたいことや聞きたいことが、頭をめぐるだけで、言葉にならなかった。結局、何を言いたいのかわからん、といった発言を、唐突にしてしまってすみません。次回、参加できれば、そこの所を反省して、もっと積極的に発言したいと思っています。

第4回　ＳＥＡ環境ワークショップ

Selected Position Papers

Selected Position Papers とあるが、手元に残っているファイルののみを載せた。Emailで届いたもののうち、プログラム委員長を通過したものだけである。手書きのものや、直接ＫＣＳに届けられたものは再入力の元気がなく掲載出来ない。参加者がファイルを保存しているはずなので、興味のあるかたは一番近い参加者から見せてもらって下さい。

氏　名:鐘_友良__(ふりがな)_ショウ_ユウリョウ
年齢 40(歳)　　男
会社名:_富士ゼロックス情報システム(株)
部　門:_技術推進室_企画課
住　所:(〒160)_新宿区西新宿3-16-6_西新宿水野ビル
TEL:　(03)-(378)-(8010)　内線(6789)

簡単な自己紹介:

◆1985年12月に中国より日本に来る。最初の1年間は、群馬大学工学部の情報工学科で並列処理言語の研究を行った。その後、富士ゼロックス情報システムに入社し、今日まで、オブジェクト指向プログラミングに関する研究開発を行っている。

◆日本に来る前は、中国の浙江工学院という工業大学の情報工学科の講師をしていた。

最も興味あるセッション: CASE

◆CASEの中でも、仕様記述からのラピッドプロトタイピング、あるいはプロトタイプの自動生成ということに一番の関心を持っている。

◆個人的には、1つのポイントは仕様記述言語であり、もう1つがプロトタイプの生成アルゴリズムであると考えている。仕様記述言語に関しては、オブジェクト指向のフレームワークを用いた仕様記述法に興味をもっている。

◆今回のワークショップでは、この点についての意見や経験を是非、聞かせて頂きたく思います。

◆また、中国からゲストの方々がいらっしやるそうなので、中国のソフトウェア事情に関するセッションにも当然ながら関心があります。

氏　名:_藤野_晃延__(ふりがな)_フジノ_テルノブ
年齢:_34(歳)　男　　血液型:_B
会社名:_富士ゼロックス情報システム(株)
部　門:_技術推進室_企画課
住　所:(〒160)_新宿区西新宿3-16-6_西新宿水野ビル
TEL:　(03)-(378)-(8010)　内線(6791)

簡単な自己紹介:

◆オブジェクト指向プログラミングに関する技術動向の調査、および研究開発(のマネージメント?)を行っている。同様にソフトウェア工学的アプローチの社内における実践を推進している。

◆もとはlisperだった、CLの巨大さ(?)に嫌気がさし、今はMLとかMirandaとかに

最も興味あるセッション: Hypermedia

◆HyperText/Mediaシステムは、今迄にない「道具」を我々人類にもたらす。多分、電話の発明、普及以上に強烈なインパクトを我々の生活に及ぼすと予測される。それに対し、今、我々は何をしなければならないのか?

◆Mediaとして様々なモノが扱えるようになると、1件当たりの情報量が非常に大きくなる。例えば音声やimages、あるいはvideo等のデータを通常の text-fileと同様に扱うことが必要になるが、今迄のcomputationは、件数は多いがデータ量はすくないので、これからは1件あたりのデータ量が大きい場合の computationを考えなければならない。

◆今回のワークショップでは、hypertext/mediaのソフトウェア開発に与える影響、または効果的な利用法といったものに就いて討論してみたい。就中、CSCW(コンピュータによる協調作業支援)といった観点から、幾つかの論点は提供できる予定。

氏　名：浜野　剛至（ふりがな）はまの　たけし
年齢：28（歳）　男　　血液型：A
会社名：（株）SRA
部　門：関西支社・開発部　役　職：SE
住　所：（〒530）大阪市北区堂島　1 - 5 - 17
ＴＥＬ：(06) － (344) － (2611)

簡単な自己紹介：
大阪で仕事をしています。
現在は、プラント設計支援ツールを EWS 上にインプリメントしようとしています。（ CASE TOOL みたいです。）

最も興味あるセッション: CASE
CASE はイメージとしてはとても良いものを持っていると思います。しかし、実際の業務に適応しようとすれば、色々と問題があるんじゃないかと思っています。（モデル、方法論、DB の持たせ方など）そういう意味で何でも OK の汎用 CASE と、アプリケーション向き専用 CASE と分けて考えるべきだと思っています。その専用 CASE をお膳立てするのが、プログラマの仕事のような気がします。で、 CASE て何でっしゃろ？という事を皆さんと考えたいと思ってます。
また、いま流行の分散環境開発と CASE の関わりみたいなものも考えてみたいと思います。（ CSCW ）

氏　名：井川裕基　　（ふりがな）いかわひろき
年齢：31（歳）　男　　血液型：?
会社名：日本電子計算株式会社
部　門：大阪支店 情報機器営業課
住　所：（〒550）大阪市西区江戸堀 1-9-1
肥後橋センタービル
TEL:　　06 － (448) － 6022 （グループイン）

簡単な自己紹介：
microVAX(4.3BSD) を頭にして Sun、NEWS 等の UNIX ワークステーションとパソコンを組み合わせた作業環境を構築／運営しています。
ネットワークはもっぱら Ethernet を使ってきたのですが、この秋から大阪に移ったため一気に環境が変わって(なくなって)しまい、 modem (2400BPS) 経由で仕事をしなくてはいけなくなり、途方にくれています。
ネットが無事に動くこと以外に興味があるのは、異質なネット（?）との接続で、オープンなネットワーク機能を持たないホストや、XNS との接続をいろいろやろうとしています。

最も興味あるセッション: MML
(まぁ、一応 PC ですから..)
一口に開発環境と言っても、「端末機」、「支援ツール」「開発自体の概念」まで幅広いのですが、MML なんていうテーマはそれらの全てを含む分野になり得ます。
とりあえずは、実際の MML ユーザの状況をレポートしてもらったり、メーカやターゲットホストの違いによる MML のバリエーション(?) を眺めてみたいと思います。
それらに対して、それぞれの参加者の立場からの意見が聞ければそれで良いのですが、各 MML システムによって「なにをするか」の狙いが違うでしょうし、ユーザの受け止め方はさらに幅広いでしょうから、論点がうまく集中するかどうかとても不安です。

...発散してしまったらどうしよう... X-(

氏　名：海尻賢二　（ふりがな）かいじりけんじ
年齢：３９（歳）　男　血液型：Ａ
会社名：信州大学　工学部
部　門：情報工学科
住　所：（〒３８０）長野市若里５００
TEL:　(0262) － (26) － (4101)　内線 (3711)

簡単な自己紹介：情報工学科で１０数年教官をやっております。大学をでてすぐ教師になったので、ソフトウェア工学の実際的な話はいまいち実感できません。７、８年前からプログラミング環境をテーマに研究を行なっていますが、いまや環境といえばソフトウェア環境であって、プログラミング環境はあまり話題にされないようです。現在はやはり知的＆ビジュアルかなと考え、プログラミングチュータ等を手がけようとやっています。

自前の環境でいうと１年前ようやく世間並の計算機環境がととのい、つかいなれたといったところです。しかしいずこも同様でしょうが管理者不在に悩んでいます。

最も興味あるセッション：Hypermedia

現在HyperMediaに関連したものとして、ＮｅＷＳを利用した
HyperText的なプログラムデータベース（プログラミングデータベースではない）と、
HyperCard上でのプログラミングチュータを作っています。そこで感じることはHyperCardは本であって、動的なデータベースではないということです。HyperCardそれ自体は非常にすぐれたものですが、一種のだい４世代言語であって、応用は限定されます。

このまえのワークショップでも話題になっていましたが、プログラム（ミング）データベースとしてのHyper-Mediaの機能要件を検討する必要がある。

最近「メディアラボ」という本を読みましたが、日本でもデモは必要ではないか。そのためにはとりあえずHyperCardをりようして何か作ってみるというのも意味があるかなと考えています。

氏　名：棚田真司＿＿（ふりがな）たなだしんじ
年齢：２７（歳）　男　血液型：ＡＢ
会社名：（株）ＳＲＡ
部　門：環境開発部
住　所：（〒１０２）東京都千代田区平河町１－１－１
TEL:　(03) － (234) － (2611)　内線 (574)

簡単な自己紹介：
現在ＳｍａｌｌＴａｌｋを使ってＰＷＢ（Ｐｒｏｔｏｔｙｐｅｒｓ　ＷｏｒｋＢｅｎｃｈ）の開発をおこなっています。

最も興味あるセッション：　MMI

氏　名：_高野　豊____（ふりがな）　たかの　ゆたか
年齢：_４０（歳）　男　　血液型：B
会社名：_松下電器産業（株）　特別プロジェクト室
部　門：_ソフトウエア設計チーム　主担当
住　所：（〒５７０）大阪府守口市八雲東町１丁目２６０
TEL:　(06) － (908) － (1001)　内線 (2075)

簡単な自己紹介：
本職：
本人にも不明な部分多し
元々は　ハード屋
コンピュータは大嫌いだったので　なぜ今の様な仕事をするようになったのか自分でも不思議な気がしている。
使用ツール：
ＵＮＩＸ使用暦　８年
（ただし　ＵＮＩＸを用いてハードを開発）

最も興味あるセッション：　Hypermedia
hypermedia は　果して何の役にたつのであろうか？
又それをささえるハードはどんなものになるか？
あるいは　ならねばならないか？

氏　名：塩谷和範（ふりがな）しおやかずのり
年齢：３７（歳）　男　　血液型：O
会社名：株式会社　ＳＲＡ
部　門：環境開発部　　課長
住　所：（〒102　）東京都千代田区平河町 1-1-1
TEL:　(03) － (234) － (2692)　内線 (　575　)

簡単な自己紹介：
昨年までハワイ大学で研究開発を行なってきた
Prototyper's Workbench
の商品化プロジェクトに取り組んでいる。
趣味：アマチュア無線 (JA2XBF/1) を十うん年ぶりで始め,会社ではＳＲＡハムクラブ (JK1ZEE) を作って活動を始めている。

最も興味あるセッション：　MMI
1．MMIの共通化・統一化は可能か?
AT&T, Xerox 等の OPEN-LOOK, SONY 等のWidgetsについて皆さんのご意見を聞きたい．(MacのMMIは本当に良いのか? NeXT の NeXT step は?) また,MMIの共通化・統一化をどう進めればユーザの利益となるかを話し合いたい．
2．究極のMMI・理想のMMI
どのようなMMIが望ましいか? (Knowledge Navigator のMMI?) 現在のMMIに不満な点・どう改善したらよいか? (マウスはあまり好きではない.-->他のポインティングデバイス?)　(PULL-DOWN　Menuもすきではない.-->Function-Keyではいけないのか?) など私は表示と入力操作が一体になっているタッチスクリーン方式が優れていると思う。
3．日本語入力のMMI
現在主流となっているローマ字かな変換方式はこのままで良いか? 他に有力な方式はないのか?　"風"の変換方式・ローマ字かな変換方式・三角偏号法.・他キーボードは?　OASYS方式・M式・TRON方式・他やはり究極は音声入力＋手書き文字認識なのでしょうが。。。それまでの繋ぎとしては "風"等で行なわれているように書き手が直接文字を指定する方式が意図しない変換を引き起こさないのでよいと思う。

氏　名：林香（ふりがな）はやしかおる
年齢：38（歳）　性　別：男　血液型：A
会社名：（株）SRA
部　門：環境開発部　部長
住　所：（〒102　）千代田区平河町 1-1-1
TEL:　(03) − (234) − (2611)　内線 (570)

簡単な自己紹介：
大規模銀行シシステムの開発/保守、鉄鋼制御システムの開発といったアプリケーション開発を経て、ソフトウェア開発環境/ツールの構築を行ない現在に至る
今の仕事
ソフトウェア環境/ツールにおける先端技術の研究、導入
社内環境の整備のためのセクション(環境開発部)のマネージャ

最も興味あるセッション：　Hypermedia
Hypermedia の応用全般について討議したい。
　ソフトウェア開発支援向けには?
　アプリケーション・ソフトウェアの可能性?
　etc 現在手に入れることができる程度のものでは、まだオモチャのようなものしか作れないような気がする。
どうなればいいのか？スピード・アップか？データ格納能力の増大か？はたまた機能の向上か？すべてが必要なのか？

氏　名：_三浦あさ子____（ふりがな）みうらあさこ
年齢：_28（歳）　女　血液型：_O
会社名：_株式会社_ＳＲＡ
部　門：_環境開発部
住　所：（〒530　）_大阪市北区堂島_1 - 5 - 17 堂島グランドビル
TEL:　(06) − (344) − (2611)

簡単な自己紹介：
２年半ぐらい X-window 上でのプログラム開発支援ツールの開発をしている。
今年の９月末に大阪に転勤になったが、引続きその仕事をしている。

最も興味あるセッション：　MMI
・いろいろな分野、レベルでのMMＩについて
・望ましいMMＩ(ＵＩ)というのは存在するかどうか
自分が現在手掛けているツールのＵＩについても「使いやすい」ということがどういうことなのかとても定義できない。
始めは、豊富な機能やヘルプを持ったＵＩが使いやすいかも知れない。しかしユーザの好みや仕事の内容によって、使い方に差が現れて来るようになると、ユーザ自身がアレンジしたり、カスタマイズできるほうがよいと思う。
そのようなツールはそれ自身が変化、進歩することが可能であるから。MMＩというのはカバーしている範囲が広いと思うので、討論がかみあわないことが多いのではないでしょうか？

氏　名：栗原正利_____（ふりがな）くりはらまさとし
年齢：３２（歳）　男　　血液型：B
会社名：(株) ＳＲＡ
部　門：環境開発部_____役　職：課長
住　所：(〒１０２) 東京都千代田区平河町１－１－１
電　話：(０３)－(２３４)－(２６１５) ダイヤルイン

簡単な自己紹介：
78 年に陸地の孤島で合った筑波を第一期生として抜けだし、当時メーカの下請けと考えられていたソフトウエア・ハウスに就職した。最初の 2 年間はいわゆる現場でアプリケーションの受託開発に携わり、80 年から UNIX のお守役を仰せつかり、以来 UNIX 環境全般の運用と、UNIX 上におけるアプリケーション開発支援環境構築に携わってきた。開発に関わった主たるツールには Testbed と Pygmalion がある。

最も興味あるセッション：　MML

これまでのコンピュータ・システムにおいては、利用者とコンピュータは文字を中心（というよりはそれのみの場合がほとんど）にデータのやり取りが行なわれている。しかし人間の間では、デジタル化されたデータよりはイメージを使ったほうがより容易に意志を伝達することができる場合が少なくない。実際人間には“文字の読み書き”の他に“図形の認識”や“会話”の能力がある訳であるから、当然コンピュータとのやり取りにもこれらをうまく組合わせてゆくべきであると考えている。

当然そのような状況を実現するためには、現在の文字コードの取り扱いしか考えていないコンピュータ・システム自身を、 Hypermedia を容易に取り扱えるように改善しなくてはならない。私としてはそのような状況下で、これまでのデジタル化された唯一無二のものしか意味しないデータの世界から、人間の曖昧さをデータとして認識／表現できるコンピュータ・システムとそのユーザ・インターフェイスについて考えてみたい。

氏　名：桜井　麻里　（ふりがな）さくらい　まり
年齢：27（歳）　女　　血液型：B
会社名：株式会社　ＳＲＡ
部　門：環境開発部　ＣＡＳＥ技術グループ
住　所：（〒102　）東京都千代田区平河町１－１－１
TEL:　(03) －(234) －(2611)　内線 (573)

簡単な自己紹介：

入社以来ずっと、ソフトウェア開発支援ツールの開発およびサポートをしています。

開発環境にはとても興味があるので、このワークショップに参加しました。でも、私は、新しいことが苦手なので、UNIX 4.x BSD と X-Window しか使っていません。

今は、NEWS-OS　2.2　＋　X-window　V10R4　＋ wterm(Wnn+xterm) ＋ jvi ＋ StP です。最近やっと Nemacsを使いはじめました。Junetのニュースとメイルは、特別忙しくない限り、毎日欠かさず読んでいます。横浜在住でも在勤でもないのに、YDOC(SEA横浜支部)によく出席しています。

最も興味あるセッション：　MMI

ソフトウェア開発の問題は、ＭＭＩにあると思う。ＭＭＩの改善に、MMIは、どれだけ、どのように、貢献することができるでしょうか?

ソフトウェアプロセス／プロダクトの可視化をすることが、ＭＭＩの改善につながりそうな気がするので、期待しています。しかし、何を可視化するのか私にはまだまだよくわからない。

可視化されたものを見るだけならいいけど、材料を整理して、表現するのは大変です。結局、「スキルの向上」という一言ですまされてしまうのではないのでしょうか。スキルの低い人でも、どんどん使えて、使っているうちにスキルが向上していくというような仕掛けはないものでしょうか?

いずれにしても、使いこなすためには、高いスキルが要求される現状のＭＭＩでは、まだまだ広まらないと思う。どうすればいいでしょう。みなさんと一緒に考えたいですね。

氏　名：渡邉　雄一　（わたなべ　ゆういち）
年齢：２９歳　性別：男　血液型：A
会社名：㈱電力計算センター　大手町事業部　主任
住　所：１００　東京都千代田区大手町１～６～１
財電力中央研究所内
：０３－（２０１）－６６０１
E-mail: watanabe@denken.junet

簡単な自己紹介：

1.会社の概略：財電力中央研究所（電研）の大型計算機の運用・保守をしている．電研の計算機は，狛江にM－３８０，ＶＰ－５０（富士通），大手町にM－３４０Ｓ（富士通）我孫子にM－６８０Ｄ（日立），横須賀にM－６６０Dで構成されており，同じメーカ同士はＳＮＡ相当で接続され，また異機種間はＲＪＥ回線で接続されている．また各所には，経理処理のためのオフィースコンピュータも設置されている．

2.私の仕事：私はこのうちM－３４０Ｓの運用とそこでの幾つかの事務アプリケーションの開発に携わっている．主要な言語は，ＣＯＢＯＬ，ＳＱＬ，コマンドプロシジャ，ＪＣＬなどである．但し私の得意とする言語はＦｏｒｔｒａｎである．

最も興味あるセッション：　４．ＣＡＳＥ

現在の私の会社での環境を考えると「ＣＡＳＥツール」をすぐに導入出来る訳ではない。だが，そのような時がいずれ（極めて）近い将来に訪れるであろうことを期待して，細々と仕事を続けている。そのような立場では，

1)今までの考えられてきているＣＡＳＥツールとは何なのか？

2)ＣＡＳＥのための（ソフトウェア工学的な）理論背景はどうなっているのか？

さらに

3)ＣＡＳＥツールの出来不出来に影響するMMＩやHyper-mediaに関して，最新の成果を取り込むような準備が出来ているのだろうか？

といった点が大いに気になる．

私の現在の主な興味は『実務に密着した運用環境の改善をうながすような良質なソフトウェアの開発のための枠組みを探ること』にある．そのような観点から現実を見つめていて「（現実の世界に於いては数の点で）極めて一般的な事務アプリケーションすらなかなかモデル化出来ないことにあるのではないか？」と感ずるようになってきた．特にオブジェクト指向やデータ指向といわれるパラダイムの登場してからも，まだ実際のシステム開発に大きくそれらが貢献したという話に接したことがない．単に私の情報不足なのだろうか？．率先してそのような問題にアプローチ出来るような立場にあれば別であるが，まだ現実には何も出来ずにいる．

近年，「分散環境だ！」「ワークステーションだ！」と言われ続けている．ソフトウェアに関与している人々へのインパクトという点では，とりあえずＵＮＩＸに先導される形での衝撃があっただけで，ソフトウェア工学の成果を実質的に取り込む形になったとは思えない．そのような状況に対して，これからのインパクトは，今まで実現されていなかったグラフィックの類を駆使したより考えられたソフトウェアが，主導権をにぎって世界を引っ張ってゆくことで，なされるだろう．（またそうあってもらいたい！）　そのためにはＣＡＳＥには，十分な思想があってもらいたいと思う．先日ＳｔＰ（ｂｙＳＲＡ）の概要を知る機会を得たが，まだ私自身十分に理解をし得ていない．また他社（ソニーテクトロニクスなど）でも同様なＣＡＳＥツールがあるようだ．

以上のような程度の知識しか持ち合わせていないが，なんとか今日の業務に押し流されることなく活発に参加者各位と意見交換をしたい．

氏　名：　山口郁生　ふりがな：　やまぐち　いくお
年齢：　２９才　　男　　血液型：　O
会社名：　電力計算センター
部　門：横須賀事業部主任
住　所：　240-01　神奈川県横須賀市長坂２－６－１
（財）電力中央研究所内
ＴＥＬ：　０４６８－５６－７５９５　（直通）

簡単な自己紹介：

私は、（財）電力中央研究所　横須賀事業所における計算機運用全般を担当しています。業務の一貫として、運用・管理の合理化、自動化を目指しています。また、そのほかに図形処理プログラムの開発、パソコンを利用したＤＢの開発を行っています。

主に、使用する言語は、ＦＯＲＴＲＡＮです。

興味あるセッション：　Ｈｙｐｅｒｍｅｄｉａ

私が担当しているプログラムの開発には仕様変更が頻繁にあります。場合によってはプログラムを大幅に変更しなければなりません。そういった、仕様変更があっても用意に対処できる開発環境を作るには、どうしたら良いか。それが私の最近の悩みです。

ＦＯＲＴＲＡＮによる開発でも、環境さえ良ければなんとかなるのではないかと思うのですが．．．それには、Hyper cardに代表される Hyper mediaの手段が使えるのではないかと考えます。

数値によって形状が変わるような図形を印刷するプログラムを作るとします。ＦＯＲＴＲＡＮでゴリゴリ作っても仕様変更が来れば、ソースを修正するしかありません。これは、非常に面倒なことで、できればそんなことはしたくない。なにかいい方法がありませんか？

氏　名：臼井義美
年齢：41（歳）　　男　　血液型：A
会社名：日本電子計算株式会社
部　門：情報システム営業部システム２課　課長代理
住　所：（〒550）大阪市西区江戸堀 1-9-1
肥後橋センタービル
TEL:　（06）－（448）－（6022）

簡単な自己紹介：

長年、ソフトウェアに係わってきましたが、どうもソフトと呼ばれる商品の製造業（Hard of Soft）にひたりきっていたようです。最近、知恵、知識、感性の面からみたソフトな世界（Soft of Soft）に関心を持っています。

最も興味あるセッション：　CASE

どのセッションにも興味がありますが、現在、開発環境にも利用環境にも感性が感じられません。技術的な面からいろいろ論じられていますが、あまりにも技術者のマスターベーションに終始しているのではないかと思います。ソフトウェアはもはや一部の人間のものでなく、全人類が係わるものとなってきた以上、技術は大衆に歓迎されるものとして実現されなくてはいけません。それには、ソフトウェアを民族のオリジナリティに基づく文化の領域まで高めなくてはなりません。それが我々が世界人類に貢献できる道であります。日本ソフトウェア教（狂？）の信者の皆さんと、ソフトウェア技術をベースにした、大衆のための理想的環境について討論したいと思います。（本当は機械オンチの私を救って欲しいのです。）

SEA 1989 - 1990 年の主要イベント（実績と予定：1989.7.31 現在）

	***** これまで（1989年 1月～ 7月） *****	
1/12-14	第 1 回 テクニカル マネジメント ワークショップ	北海道：函館ロイヤル ホテル，参加者 16名
1/20	SEA Forum「最近の中国におけるソフトウェア工学研究の動向」	東京：青年会議所会館，参加者 22名
1/24-25	第 5 回 ソフトウェア技術交流会	神奈川：横浜国立大学，参加者 44名
2/2-4	第 1 回ソフトウェア プロセス ワークショップ	静岡：機械振興協会伊豆研修所，参加者 23名
2/28	SEA Forum「Unix, Mac, NeXT, etc をめぐるホットな話題」	東京：青年会議所会館，参加者 63名
3/4	環境分科会（SIGENV）特別 Forum	岩手：盛岡市，参加者 21名
3/22	SEA Forum「ソフトウェア プロトタイピングの効用」	東京：青年会議所会館，参加者 25名
3/22-24	SEA 春のセミナー ウィーク '89	東京：青年会議所会館，参加者 277名
4/18	SEA Forum「ソフトウェア技術者とテクノストレス」	東京：青年会議所会館，参加者 31名
4/19	SEA 関西支部・各種イベント通算 100 回記念ミーティング	大阪：日本能率協会，参加者 55名
4/20-21	第 9 回 ソフトウェア信頼性シンポジウム	大阪：ホテル ガーデンパレス，参加者 96名
4/22	SEA 九州支部・設立 1 周年記念講演会	熊本：県立劇場小ホール，参加者 44名
5/9-21	第 11 回 ICSE 研修ツアー	Los Angels - Ann Arbor - Pittsburgh，参加者 19名
5/30	SEA Forum「ソフトウェア工学の最新動向（ICSE 報告）」	東京：機械振興会館，参加者 50名
6/13-15	ソフトウェア シンポジウム '89	東京：虎ノ門パストラル，参加者 273名
6/28	SEA Forum「ソフトウェアに関する標準化の動向」	東京：機械振興会館，参加者 36名
7/12-13	第 1 回 SEA 関西セミナー	大阪：大阪大学工業会館，参加者 78名
7/28	SEA Forum「コンピュータ ネットワークの現状と展望」	東京：青年会議所会館，参加者 39名
	***** これから（1989年 7月～ 1990年 6月） *****	
8/31-9/2	第 3 回 教育ワークショップ「ソフトウェア設計教育」	熊本：ホテル角万，参加予定者 30名
9/1-2	管理分科会（SIGMAN）特別セッション「標準化」	東京：代々木オリンピック センター（参加者募集中）
9/5-8	第 7 回 夏のプログラミング ワークショップ（若手の会） 「ソフトウェア品質の定量的把握を目指して」	岩手：盛岡市，参加予定者 46名
9/19-21	SEA 秋のセミナー ウィーク '89	東京：青年会議所会館（8月下旬募集開始）
9/26	SEA Forum「ソフトウェア品質定量化の枠組」	東京：機械振興会館（8月下旬に案内）
10/6	環境分科会（SIGENV）特別 Forum（プログラム未定）	兵庫：神戸市（9月上旬に案内）
10/21	SEA 名古屋支部・特別イベント（プログラム未定）	愛知：名古屋市（9月上旬に案内）
10/24-25	第 10 回 ソフトウェア信頼性シンポジウム	大阪：ホテル ガーデンパレス（現在発表者公募中）
11/13-14	International Symposium on Future Software Environment	京都：京都リサーチパーク（9月上旬にプログラム決定）
11/29-12/1	第 5 回 実践的開発環境ワークショップ	北海道：釧路市（現在参加者公募中）
1/11-13	第 2 回テクニカル マネジメント ワークショップ	北海道：函館市
期日未定	第 2 回 ソフトウェア プロセス ワークショップ	開催地未定
期日未定	第 3 回 日中ソフトウェア シンポジウム	開催地未定
期日未定	第 6 回 ソフトウェア技術交流会	開催地未定
3 月中旬	SEA 春のセミナー ウィーク '90	東京：青年会議所会館
3 月～ 4 月	第 12 回 ICSE 研修ツアー	Nice（France）
6 月	ソフトウェア シンポジウム '90	京都：京都リサーチパーク

ＳＥＡ会員状況（平成元年8月21日現在）

正会員	1670名
賛助会員	58社

正会員の勤務地および居住地域分布

	勤務地域	居住地域
北海道 =	10	9
青森 =	0	0
山形 =	1	1
宮城 =	6	5
岩手 =	10	10
福島 =	1	1
秋田 =	1	1
新潟 =	7	7
栃木 =	6	6
群馬 =	5	4
茨城 =	12	13
埼玉 =	21	125
千葉 =	20	99
東京 =	884	528
神奈川 =	133	298
山梨 =	1	1
長野 =	36	37
富山 =	3	3
福井 =	4	3
石川 =	3	3
静岡 =	19	19
岐阜 =	2	8
愛知 =	56	45
和歌山 =	2	3
三重 =	1	4
滋賀 =	11	12
京都 =	21	32
大阪 =	226	167
奈良 =	4	14
兵庫 =	38	68
愛媛 =	2	2
徳島 =	6	6
高知 =	0	0
香川 =	1	0
鳥取 =	0	1
島根 =	1	1
岡山 =	2	2
広島 =	5	5
山口 =	0	0
福岡 =	18	20
大分 =	6	6
熊本 =	34	33
長崎 =	0	0
宮崎 =	1	1
佐賀 =	1	1
鹿児島 =	5	5
沖縄 =	1	1
韓国 =	1	1
米国 =	4	4
豪州 =	1	1

＜男女分布＞

男	=	1584
女	=	86

＜年齢分布＞

20以下	=	0
20_24	=	37
25_29	=	324
30_34	=	453
35_39	=	384
40_44	=	287
45_49	=	116
50_54	=	32
55_59	=	19
60以上	=	16

＜血液型分布＞

A型	=	649
O型	=	481
B型	=	364
AB型	=	176

賛助会員会社名

ジェーエムエーシステムズ，日本システム，リコーシステム開発，ＳＲＡ，辻システム計画事務所，ＰＦＵ，近畿日本ツーリスト，東電ソフトウェア，セントラル・コンピュータ・サービス，日本能率コンサルタント，構造計画研究所，ケーシーエス，サンビルド印刷，日本システムサイエンス，富士ゼロックス情報システム，ＩＮ情報センター，富士通，エムテイシー，クレオ，情報システムサービス，インターナショナル・データ・リサーチ，日立ビジネス機器，アトラス情報サービス，三菱電機セミコンダクトソフトウェア，伊藤忠エレクトロニクス，三菱電機，富士通ＢＳＣ，千代田製作所，村田技研，ニコンシステム，日本ビジネスデータプロセシングセンター，阪神計算センター，昭和電工コンピュータサービス，エヌ・ティ・ティ・システム，日本情報システムサービス，ヒラタ・ソフトウェア・テクノロジー，日本エム・アイ・シー，日本電気ソフトウェア，ソニー，コマス，三菱電機コントロールソフトウェア，インターフィールド・システムズ，ジャストシステム，マイクロリサーチアソシエイツ，松下ソフトリサーチ，シャープ，東洋エンジニアリング，日本ユニシス・ソフトウェア，清鳳オフィスメーション，新日鉄情報通信システム，テックシステムズ，熊本電子計装工業，ＮＴＴ九州技術開発センタ，立石ソフトウェア，三菱電機関西コンピュータシステム，日立エンジニアリング，スインク，テスク

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