9. グループ・コンサルタントの感想

1. PWB & Hyperbook グループ
菅野 卓矢 (岩手電子計算センター) 170

2. OOA/OOD グループ
佐原 伸 (SRA) 172

3. CASE Bench グループ
天池 学 (カシオ計算機) 175

4. StP グループ
荻原 剛志 (大阪大学) 178

5. Teamwork グループ
佐藤 千明 (長野県協同電算) 180

CASE Study of CASE Tool に参加して

PWB グループ　コンサルタント

菅野 卓矢

(岩手電子計算センター)

1. はじめに

今回のワークショップは,「Case Study of CASE Tool」というかなりセンセーショナルなテーマであり，自分がいま抱えている要求分析の限界という問題解決のヒントを１つでも見つけることができればという，淡い希望を持ちながらの参加であった．本来であれば一般参加者の１人であるはずのものが，どう間違ったかプログラム委員会の末席を汚すことになり，いささかとまどったのは事実である．構造化分析，オブジェクト指向分析など，付け焼き刃程度に学習したところで役に立とうはずはないことを自分にいい聞かせながら，議論の進行を見守ることにした．

以下は，ワークショップを通じての感想を述べたものである．

2. 雑感

2.1 牽制

第１日目は，構造化分析･設計，オブジェクト指向分析といった技法に関する講義であり，個々の参加者の性格が出てくる場面はなかった．しかし，グループに分かれてからの人間関係には興味深いものがあった．どんなケーススタディの場合にもあてはまるが，自分の技量と自分以外の人間の技量との力関係が計れるまでは，自分の力を十分に見せることはない．課題を読むのに時間がかかったのか，一向に議論が進まない．１時間ほどむだな時間が過ぎてしまった．

2.2 連帯感

討論の進行とともに，分業でやろうとの意見が出た．最初は９人のメンバで探りあいの駆け引きであったのが，１チーム３人ずつの構成になったところで連帯感を強め，これからは３人で解決して行くという自覚が，一気に話題を進展させる糸口となった．

2.3 ジレンマ

各チームでは，それぞれに共通のジレンマを持っていた．それは，手法に対する無理解に起因している．現場で作業するとき，ほとんどのプロジェクトチームは，手法として構造化分析･設計を採用することはない．それでも，いままで開発に支障がなかったわけで，今回このワークショップに参加した若手 SE 諸君の少なくとも 80% は，この手法を現場に生かしたことがないものと思われる．したがって CASE ツールを搭載したワークステーションを目の前にして，使いたいのに手を出せないというジレンマを感じていたに違いない．

2.4 完成

尺度がどうであれ，問題解決のために CASE ツールを使い，その結果として状態遷移図が書かれた．だが，この成果を見て満足感をおぼえたメンバは少なかったようだ．唯一満足できたのは，一応の形を整えて状態遷移図のアニメーションが見られたことであろうか．自分たちの思考を具現化したものを見ることによって，それが「当たらずといえどの遠からず」だったと感じ，自身を深めたことはたしかである．要求分析結果のプレゼンテーション法として，ラピッドプロトタイピングやアニメーションといったビジュアルな手法がいかに有効であるかがわかる．

3. まとめ

3.1 考察（その１）

今回のワークショップでのポイントは，手法の理解であった．参加者は，構造化分析など各種の手法をワークショップ以前にある程度習得しておく必要があったように思う．その意味で，各企業がこうした解析手法を日常いかにおろそかにしているかが伺える．従来の紙と鉛筆による机上での構造化分析など，考えただけでも逃げ出したくなるのは，私だけではないと思うが，そのための道具として CASE ツールが役立つのも，結局のところ，ワークステーションの普及とネットワーク化された開発環境なしには考えられない．

3.2 考察（その２）

個々の企業のなかで，だれが新しい手法の伝道者になるか？１つのプロジェクトが発足してから解散するまでのライフサイクルはさまざまであるが，１年から３年がふつうであろう．その間，プロジェクトはその内部で用いる手法を固定しなければならないことから，新しい手法への取り組みは難しくなる．ところが一般的に，プロジェクトリーダとなる人々は，次から次へと新しいプロジェクトをあてがわれるために，新しい手法の導入を思案するゆとりもなく，結局は伝統的手法での問題解決に走ってしまう．プロジェクトチームはそのリーダの技量に依存するところが大きいことを考えると，新しい手法の採用など到底おぼつかなくなる．

私の提案としては，役割としての手法伝道者をおくべきである．責任の思いリーダはさておき，次にリーダになるべき人達への新手法の教育が必要だということである．さまざまな基準はあろうが，リーダの立場になるのが３０代として，その後こなせるプロジェクトの数はせいぜい３～５というところであり，リーダになってからの手法の変更は本人にとってもかなりの重圧になる．このようなライフサイクル認識を悟った上で，社内の分析技術向上に力を注ぐのも１つの方策ではないだろうか？

3.3 CASE ツールについて

コンピュータが，時代のニーズに答えて，大型機，オフコン，パソコンと移り変わり，いまやワークステーションがステージの真中に立とうとしている．一方で，ソフトウェア技術者が今後着手するシステムはどんどん大規模になって行くに違いない．

この大規模化にともない，ユーザ要求の不確定性が強くなって行く中で，分析手法の確立，さらにその支援手段としての CASE ツールの存在は重要である．手法が構造化分析になるか，オブジェクト指向分析になるかはともかく，要求分析の時点で，そのシステムが内包するすべてのファクタを取り出すことができ，それらを組み立てて行く途中での見直しができるような環境は，すべてのシステム関係者が望むところである．

今回参加した若手技術者の一部から，「分析からコード化まで一連の作業ができなければ無意味である」といった意見が出されていたが，心情的にはわかるし，将来そうなるべきであると私も考える．しかし，まずは使ってみることが肝要である．このようなツールは，自らの手で使い育てる気持ちが必要であろう．

3.4 総じて

グループのメンバに苦言を呈する形になり，いささか恐縮している．プログラム委員として専門的な立場から助言できなかったことを反省している．しかし，客観的に見て，ケーススタディが必ずしも成功したとは思えないことは，それぞれのメンバが自覚しているところでもあると思う．この経験を一時的な研修として受け入れるのではなく，常に自分たちの問題としてとらえ，新技術の習得と普及に努められるよう期待する．

CASE ワークショップ @ 盛岡の報告

OOA/OOD グループ　コンサルタント

佐原 伸
(SRA)

1. はじめに

今回のワークショップは，構造化分析／設計／プログラミングとオブジェクト指向分析／設計／プログラミングの対決という視点から捉えられる．（個人的な）結論からいえば，CASE ツールの機能と可変性および方法論としての自然さと再利用性の高さからはオブジェクト指向分析／設計／プログラミングに軍配が上がり，修得のしやすさに関しては構造化分析／設計／プログラミングに軍配が上がった．

いずれにせよ，従来行われてきた機能中心の分析／設計／プログラミングよりはすぐれており，あとはいかにして技術移転を図るかという問題が残っているだけである．たとえば今回のワークショップに参加した若手メンバーの中の数人は，構造化分析に関するある程度の理解を得たと思われるが，オブジェクト指向パラダイムに関しては「これは凄そうだ」という直感以上に理解したメンバーはいなかったように感じられた．

筆者自身に関していえば，もともとオブジェクト指向プログラミングに関する確信はあったが，FXIS の青木氏の講演を聞き，同じグループに入って４日間過ごしたことにより，オブジェクト指向分析／設計に関して持っていた最後の疑問も解消したということがいえる．

2. オブジェクト指向分析／設計について

１日目の青木氏の講演の中では，以下の点が印象に残った．

(1) 仕様化技法には，モジュール構造図のように機能の構造に着目するもの，データーフロー図のように情報の流れに着目するもの，状態遷移図やペトリネットのように事象の推移に着目するもの，E-R 図やオブジェクト指向のように実体とその関係に着目するものなどがあり，それぞれ得意な局面があり，一般的にどれがよいとか悪いとかいえるものではない．したがって，オブジェクト指向分析／設計を支援するツールは，他の仕様化技法のツールを含んだ形で作ろうとしている．たとえば，データフロー図はオブジェクトの内部を記述するには有効であり，状態遷移図はオブジェクトの挙動の時間推移を表すには有効である．

(2) 従来のウォーターフォール型の開発モデルは，手戻りを嫌いそれをなるべく防ごうとしていたが，オブジェクト指向分析／設計／プログラミングでは，検証して間違いを発見しそれを直すことを必然と考える．いわば，究極のパッチシステムである．したがって，手戻りの手間を最小限にするプログラミング環境とプログラミング言語が必須となる．

(3) 心理学の研究結果では，階層構造の組織は最終的に上司の意見に収束する．このような組織では，上で述べた素早い手戻りと技術的決定を必要とするオブジェクト指向分析／設計／プログラミングを行うことはできない．オブジェクト指向にはフラットな組織が必要であり，私（青木）のところではこれを実践している（筆者注：フラットな組織ではリーダも技術的によい意見を出さなければ，技術的決定に参加できない．つまり責任はあるが，権力はないということ）．

２日目からは，青木氏がベンダーとして Smalltalk によるオブジェクト指向チーム８人を指導し，筆者がアドバイザとして参加した．他に静岡大学から記録係１人と中国人研究者（方氏）とフランス人研究者（マチュウ氏）が参加した．ここでは，共通例題の給湯システムをやり，図書館システムは青木氏の模範回答を見るという方針になった．ツールとしては，Coad/Yourdon のオブジェクト指向分析にもとづいた OOA Tool（Coad の会社で作った Smalltalk/V によるβ版と，青木氏の作成した Smalltalk-80 によるβ版）を採用し，Macintosh IIfx

と Macintosh Portable で動かした．マシンが２台しかないので，２グループに分けて分析／設計を進めた．

Coad/Yourdon のオブジェクト指向分析では，(1) オブジェクト名の決定，(2) オブジェクトの関係の決定，(3) オブジェクトのサービス（メソッド）の決定，という手順で進むのだが，２グループとも (2) のステップで暗礁に乗り上げた．オブジェクトの関係の決定では，まず (a) is-a 関係つまり継承関係を設定し，次に (b) has-a 関係つまり全体／部分の関係を決定する，ということを行うのだが，この継承関係の決定ができなくなったのである．結局，最後までこれ以上進まなかった．

うまくいかなかった原因は以下のようなものである．

(1) 進め方の問題: 再三，青木氏と筆者が「オブジェクト指向では最初の案があまりよくないのは当然なのだから，どんどん先に進んでプログラミング段階でうまくいかなかったら，分析に戻ってやり直してみることを経験してみよう」と忠告したのだが，青木氏の顔色を伺いながら，皆で合意を得ながら，要は石橋を叩いて渡らないという方式で進めようとするため，なかなか１つのことが決まらなかった．

(2) 抽象化能力: 機械の見た目にごまかされ，オブジェクトの抽象的な共通性を見抜くことができなかった．また，has-a 関係をなかなか見つけられなかった．

２日目と３日目の分析／設計演習の青木氏のコメントで参考になったのは以下の点である．

(1) 再利用性: 図書館システムの例題を青木氏がオブジェクト図にする過程で，「図書館の本」の親のオブジェクトとして「本」が，そのまた親のオブジェクトとして「著作物」が，「利用者」の親のオブジェクトとして「人」が出てきた．青木氏いわく「この人と著作物というオブジェクトが，再利用できる重要なオブジェクトなのです」

(2) オブジェクト指向をやるには: 「オブジェクトを抽出するのには非常な努力と訓練がいります．大きなシステムを作る場合に，私は最高７回オブジェクトの関係を作り直したことが有ります．Smalltalk などのオブジェクトの階層を覚えておくことも当然重要です．私は Smalltalk のクラスとメソッドはほとんど覚えています」

(3) Smalltalk について: 図書館システムのプロトタイプをあっという間に作り上げて，「オブジェクト指向の場合，再利用できるオブジェクトライブラリーがどれだけあるか，簡単にプロトタイプ／完成品を作るためのプログラミング環境がどれだけ充実しているか，プログラミング言語がメモリーの確保／解放など低レベルの仕事をどれだけやってくれるかが重要です．そういう意味で C++ などはオブジェクト指向言語としてはあまり使えないと思います」

(4) オブジェクト指向の効果: 「これから帰って，一晩徹夜で客に納めるシステムを作らなきゃ」(筆者注: えー，一晩でできちゃうの！？)

3. 他の方法論／ CASE ツールについて

PWB は Smalltalk-80 を使って作られているが，プロトタイピング支援ツールであり，今回の演習では状態遷移図の作成で苦戦していた．この原因は，(1) 何を状態にすべきかで迷い，フローチャートのような状態遷移図を書いてしまう，(2) 状態を段階的に詳細化することなく，システム全体をひとつの状態遷移図にしてしまった！ (3) 発表練習という，今回のワークショップの主眼ではないことに時間を費やしていた，といった点にあると思える．

StP は構造化分析／設計の支援ツールであるが，方法論を理解せずに演習をスタートしたため，データ中心でなく機能中心のデータフロー図を書き始めた．また，データの名前づけに苦戦していた．もっともこれは各グループ共通の問題で，データの名前をなるべく短く具体的にしかもある程度抽象化して考えるのは難しいらしい．抽象化能力を表すひとつの指標かも知れない．

Teamwork も構造化分析／設計支援ツールであるが，ここは２グループに分かれひとつは機能中心で分析を

進め，もう一つのチームはデータ中心で分析を進めた．機能中心グループは，あまり考えずに試行錯誤的にモデルを作成し，徐々にシステムの全容を明確化していったが，最終的なモデルは動くかどうか自信のないものになった．一方のデータ中心グループは，データモデルを苦しんで作成し，細かいところに気を配った分析を行ったため，完成度の高いモデルが得られた．データモデルがオブジェクト指向分析の場合のオブジェクト図と似た結果になっていたのが興味深い．

4. 参加者の感想

参加者の感想で主なものは以下のとおりである．

(1) やはり方法論を知らないと苦戦する．あるいはツールが方法論をガイドしてくれないのでわかりにくい．
(2) CASE ツールの操作は簡単だが，まだまだ改良してもらいたい点がある．
(3) これほど厳密な設計はやったことがない．
(4) CASE ツールなしでは構造化分析／設計をやる気はしない．
(5) 協調作業が支援される．
(6) 結局はコミュニケーションツールだったな！

5. おわりに

このレポートは，議事録をもとにワークショップ終了後約３カ月経って書いた．このため，オブジェクト指向に関する議論の一部は，このワークショップだけではなく，その後の CASE ワークショップや YDOC などで青木氏と議論したことも混入している．今回のワークショップでは，静岡大学の落水研究室が，演習を行った各グループに参加しその記録を取った．このため本レポート以上の興味ある事実および本レポートの誤りは今後次第にあきらかになって来ると思う．

筆者個人は今回のワークショップで多くのものを得た．特にオブジェクト指向に関する疑問点は，青木氏のアドバイスでほぼクリアーできた．同じような成果を若手参加者が得られたであろうか？

今回のワークショップの演習では，日本式の間違えると怒られる教育の弊害のためか，他人の顔色を伺いながら分析／設計を進めるので，作業がほとんど進まなかった．ワークショップの場合は，自分のアイデアを間違っていてもよいから早く考えだし，そのアイデアを他人のアイデアと比較することによって，自分のアイデアを補強したり修正していくことが必要なのだが，若手の参加者にこのような態度がほとんど見られなかったのは問題であろう．普段仕事の場でも，いわれたことをいかに手っとり早くやるかだけを求められているせいかも知れない．将棋でも囲碁でも「下手なうちは速く打たないと上達しない」のだが...

もっとも，我々のチームに参加した外国人研究者にも同じ問題があるように感じられたので，現場の若手技術者にここまで求めるのは酷なのだろうか？

しかし，従来のワークショップが机上の空論 (?) に留まっていたのと比べ，今回のワークショップは CASE ツールベンダ各社の協力もあり，多くのマシンとツールを持ち込んで実習を行うことができたのは非常によかった．各参加者は構造化分析／設計やオブジェクト指向分析／設計の方法論や抽象的思考に苦しんだとはいえ，最新の方法論と CASE ツールに触れ，その感触を掴むことができた．

我がチームでも，青木氏がその場で作った，ウィンドウがマウスで任意に引いた線の上を動く Smalltalk のデモプログラムに驚嘆の声が上がった．２～３年前までは遅いことで有名だった Smalltalk システムのデモプログラムが，速すぎて「見えなかった」ためである．他にもいくつかの例題で Smalltalk ワールドに触れた我がチームのメンバーは，ワークショップ参加前より確実に一歩前進したと思う．

CASE による問題解決プロセスと課題

CASE Bench グループ　コンサルタント

天池 学

(カシオ計算機)

1. はじめに

CASE Bench グループのコンサルタントとして，われわれのグループにおける問題解決のプロセス，その中で発見された作業上・運営上・方法論／ツール関連のさまざまな問題点，および今後の課題について簡単に報告する．

2. 問題解決プロセス

われわれのグループの問題解決プロセスを要約すれば，次の通りであった：

９月６日

(1) データの洗い出し
(2) システム化対象範囲の決定
(3) エクスターナルの決定

その結果として，コンテキストダイヤグラム Version-1 の完成

(4) 第 0 階層 DFD の作成
(5) データ・ヂクショナリの作成

９月７日

(6) 第 0 階層 DFD に不具合を発見し，コンテキストダイヤグラムを修正
(7) 第 0 階層 DFD の修正
(8) 第 0 階層より下位の DFD の作成
(8') ERD の作成 － (8) と並行作業
(9) 上記２つの作業の結果を調整
(10) 調整結果を踏まえて，DFD および ERD の修正
(11) DD の作成
(12) 構造図の展開

3. 問題点

3.1 作業上の問題点

今回実際に行った作業で発見された問題点は次の通りである：

(1) DFD の作成
　－ 機能分類については，バブル数が少なすぎた (最適値は 7 ± 2).
　－ ファイルの扱い方において上位階層の方に具体的なものがあった.
(2) DD の作成
　－ DFD の階層ごとに DD を作成すべきであった.
(3) 作業手順
　－ DFD → ERD の順序で作成したが，ERD の作成から始めたほうが効率がよかったようだ.

また，今回実施できなかった作業は次の通り：

(4) ミニスペックの作成
　－ 入出力データの関連，各ダイヤグラムの整合性が確認できる．
(5) 構造図の設計
　－ 機械的に作成された構造図（要求仕様）をもとに設計を行なう必要がある．
　－ モジュールスペックの作成が必要である．
(6) 構造図からプログラミングのためのスケルトンを作成

3.2 運営上の問題点

(1) ツール
　－ 操作は比較的簡単である．
　－ プリントアウトの操作が不便である．
(2) 運営
　－ 思ったより作業が進まなかった．
　－ チームの人数が多すぎた．
　－ ツールを十分に使いこなせなかった．
(3) その他
　－ CASE ツール以前の問題（手法の理解）があった．
　－ 今の仕事に生かせるかどうかが疑問である．

3.3 方法論／ツールに関する問題点

(1) 方法論

一見面倒そうに見え，また効果も短期間では見えにくいが，できあがったものの保守性や，変更に対する柔軟性が向上する点はよい．しかし，次のような点は問題であると思われる：

　－ データの整理が大変である．
　－ 時間がかかりすぎる．
　－ 個人の能力に依存する部分が多い．
　－ １つの事象は幾通りにも表現可能であり，他人の表現を受け入れるには抵抗がある．
　－ 正しく手法を学ぶのはむずかしい．
　－ これまでのやり方にとらわれ，手法を完全に生かすまでに至らない．
　－ 単なる日本語化ではなく，もっと日本の風土にあった手法は考えられないか？
　－ バブル名は「名詞 ＋ する」で OK になりがちになる．

(2) ツール

よい点はつぎの通り：

　－ 試行錯誤が簡単である．
　－ 修正が一瞬にできる．
　－ 視覚的に考えられる．
　－ 記述内容のチェックをしてくれる．
　－ モジュール構造図への自動変換など，てまが省ける．
　－ 手作業でおろそかになりがちなエラーチェックが厳密に行える．

その一方で，悪い点は：

　－ エディタの操作が面倒で思考が中断される．
　　・データフロー，バブルなどの名前を変えるさい再入力が大変である．
　　・フローの移動が面倒である．
　－ 多人数でワイワイ議論できない．

- レビューのためのメカニズムがない.
- データの整理機能がちょっと弱い.
- コマンド入力が面倒である（メニュー方式のほうがよい).
- より少ないコマンドで必要な処理ができるとよい.

4. 今後の課題

次のようなことが今後の課題であろう:

- 手法の理解･データの整理に要する労力をより少なくする.
- 原状の開発ラーフサイクルに対して，より広い範囲をカバーする.
- ツール導入の効果をもっと早く，よく見えるようにする.
- 機能面以外のユーザ要求の実現を容易にするツールが少ない.

事後報告風の雑感

StP グループ　コンサルタント

荻原剛志
(大阪大学)

調和宇宙や，そのすべての部分，またその諸相におけるどんな作用をも純粋な「数字」の象徴によって表すことは当然であり，適当なことなのである（イスラエル・リガルディ「柘榴の園」).

1. はじめに

今回は CASE ツールについてということで，私も CASE ツール自体への興味に加え，実際にソフトウェアの開発にあたられている方々の CASE に対する考えを知る上からも大変興味のあるワークショップであった．

私は初日に５０分ほど，SA/SD 法の説明を行ったが，参加者の方の多くは「"でまるこ" も "ちびまるこ" もわからない」（盛岡では「ちびまる子ちゃん」は放映していないのだそうです）
ということのようであり，これだけの時間に SA/SD さらにリアルタイム SA の説明を詰め込んで，どこまで理解してもらえたか？かというのが不安ではあった．

2. グループ作業について

２日目以降，５つの作業グループに分かれて，実際に CASE ツールを使いながら問題を解決して行くという作業に入った．私は SRA の StP のグループを担当していたので，以下，このグループの作業の進み具合いについて少し述べてみる．

演習問題としては事務系，制御系，それぞれ２題あったわけであるが，StP グループでは当初，まず簡単そうな事務系の問題（図書館の問題）をやってみて，次に制御系をやってみようか，というような思惑でいた．しかし，これは非常に甘い考えであったことが間もなく判明する．

StP が動く端末（ワークステーション）が２台用意されていたため，９人を２つのチームに分け，とにかく使ってみようということで，チームごとに相談しながら試行錯誤的にツールを使って図を描き始めた．ところが，構造化分析の最初の段階（コンテキストダイアグラム）から一苦労．ああでもないこうでもないと，図形を描いたり消したり動かしたりの繰り返しが続く．

中間報告では一応，各チームごとの DFD（データフローダイアグラム）を見せることはできたものの，翌日からは２つのチームが集まって一本化しようということになった．インストラクタの桜井さん (SRA) からのアドバイスを踏まえ，黒板等を使ってディスカッションを重ね，最終的に DFD とデータ定義の作成までを終えることができた．

作業がなかなかはかどらなかったのは，とにかく SA/SD の考え方が理解できていないところにあったのだと思う．桜井さんをして「DeMarco の本は奥が深い」といわしめる位だから，１時間にも満たないチュートリアルのみでは多くを望むことはできないであろう（チュートリアルが悪かったからだといわれると返す言葉はないのだが....).

しかし，CASE ツールの操作については特に難しい点もなく，皆がすぐある程度使えるようになったようである．StP グループでは，１つの端末に４，５人で作業を行うことができ，CASE ツールを実際に使ってみるという目的はかなり達成できたのではないかと思う．

問題解決の手順や，そのための CASE ツールの利用方法などについては，CASE ツールに直接触れられる折

角の機会だということもあり，かなりの部分をメンバーの自主性に任せる方針にしたつもりである．結果的にではあるが，作業の１日目は CASE ツールに触れてみること，２日目は CASE ツールを用いて問題を解決することが体験できたのではないかと考えている．時間が経つにつれてメンバーも打ち解け，積極的に問題解決のために協力しあうことができた．終始和やかな雰囲気の中で作業できたこともよかったと思う．

ところで，演習問題のうちの「ガス給湯システム」は，実は大学で私が作成したものであるが，演習問題としてはかなり複雑な部類ではないかと思う．この問題を５つの作業グループのうち３つまでが選んだのには驚いたが，各グループともかなり苦労していた模様であった．要求仕様が複雑であることと，システムを構成するさまざまな要素をグループ分けしにくいことが，この問題を難しくしているのではないかと思う．

それから，少し弁解させていただくと，私はソフトウェア開発の方法論の勉強はしてはいるが，大学内でのこと，それを実際のソフトウェア開発に適用したという経験はほとんど皆無である．開発法の CASE ツールとしての実現とか，実際の開発に適用する場合のポイントなど，やはり，ツールを持って来ていただいた企業の方々の話の迫力には勝てないのであった．

3. CASE について

参加者の皆さんはそれぞれ，CASE ツールとは一体どんなものなのだろうかという疑問と，自分の業務にも活かせるのではないかという期待を持ってこのワークショップに参加されたのではないかと思う．実際に使ってみて，「素晴らしい」という人もいれば，「たかがツール」であることを発見した人もいたようである．

私自身は，紙の上に DFD 等を書き散らすよりも，CASE ツール上で開発をした方がずっと楽であろうという感じが強くした．しかし，たとえばワープロで文書を作成する場合にも，まず紙に下書きしてからワープロに打ち込む人と，文章を考えながら打ち込んで行く人がいるように，CASE ツールの使い方も人それぞれ異なって当然であると思う．また，特定の CASE ツールが実際の開発作業に有効かどうかということは，ある程度その開発技法に慣れ，使い込んでからでないと実感できないのではないかとも感じた．

作業グループのメンバーの方とも話したのだが，ソフトウェア開発の現場では，今もって CASE どころか構造化分析すら持ち込めない状況とのことである．それはソフトウェア開発側よりも，多分にユーザ側の無理解から来ているらしい．したがって，CASE ツール自体の機能等も重要には違いないのだが，まず CASE ツールを使うことのできる基盤作りが当面の最優先課題ではないかという印象を受けた．

4. おわりに

このたび初めて SEA の若手の会に，しかもプログラム委員として参加させていただきました．初参加で勝手がわからず，プログラム委員としてはあまり役に立たなかったのではないかと恐縮しています．

最後になりましたが，このような好企画は，落水先生，事務局の方々，ツールを提供して下さった企業の方々，岩手電子計算，静岡大学の方々のご努力がなければ，到底実現しなかったことと思います．どうもありがとうございました．また，StP 担当の桜井さん，聖徳太子のような書記の中尾さん，どうもお疲れさまでした．

『第8回　夏のプログラミング･ワークショップ』　参加報告

Case Study of CASE Tool "Teamwork" を終えて

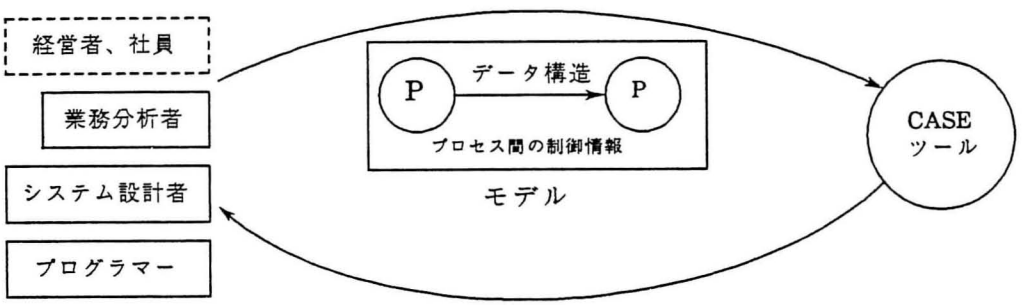
㈱長野県協同電算　　佐藤千明

1. はじめに

CASEツールを一つのシステムとして捉えると、そのターミネータには経営者･社員、業務分析者、システム設計者、プログラマー等が挙げられる。各担当者の作業内容は異るが、作業対象として与えらえたターゲット･エリアをモデル化するためにデータの流れ･プロセス仕様、データの構造･データ間の関係、事象･タイミング･時間、という3次元の尺度を用いてCASEツールに入力している点では同じである。

入力されたモデルを基にCASEツールがターミネータに出力するものは、モデルの正当性、モデル間の正当性、モデル間の変換機能である。現在のCASEツールは、決して新しいモデルは創出してはいない。

ゆえに、CASEツールを使いこなすにはターゲット･エリアをモデル化できる能力が必要であり、方法論をまずマスターしなさいと言われているのである。



モデルの正当性、モデル間の正当性、モデル間の変換機能

今回のワークショップの狙いは堅くいえば、①方法論、モデル化手法を学び②CASEツールへの入力操作を体験し③CASEツールの出力を評価し、CASEツールの可能性を探ることであり、方法論に沿った分析設計作業をCASEツールはどの程度支援してくれるかの実験であった。

2. 分析･設計作業を振り返って

Teamworkグループは、メンバーが基本ソフト･制御系が7名、事務系が2名、さらにツールがリアルタイム構造化手法を売り物にしているということもあって、ガス給湯システムのマイクロプロセッサ部の設計という難問に挑戦した。以下その過程を追いながら、実験結果を観てみよう。

2.1 コンテキスト･ダイヤグラムの作成

構造化分析の第一歩である。ツールもあることだし、この作業は机上でやるよりみんなでお絵書きツールを使ったほうがいいと判断し、いきなりツールのお世話になった。結果は上々。特に試行錯誤による書き直しには便利であった。ただ、リアルタイム構造化手法に慣れていないせいもあり、実線で示されるデータと破線で示されるシグナルの区別を単に値が離散的かどうかで判断してしまい、弁の開閉信号･切替信号、ランプ点灯信号等あとでシグナルからデータに訂正したものがいくつかあった。最終的に分析設計が完了したとき、ツールがこの間違いを指摘してくれるとありがたいが、今回はそこまで確認できなかった。

2.2 ダイヤグラム0の作成

いよいよ構造化分析に突入。ダイヤグラム0の作成は最も重要な作業であり、ここさえしっかり分析できれば後は2班に分かれてもそう問題は起きないだろう、ということでまた全員でツールに向った。

TeamworkはHatley手法のため、まずはコンテキスト･ダイヤグラムにあるデータをもとに適当にプロセスを決めてDFDを作り、後でプロセス間の制御を行う信号線を加えてコントロール･スペックを書こうとした。

結果は見事に失敗。その理由は①各人の考え方が異なり、一つに収束できなかったこと。②出力データを基準にして入力データからの変換プロセスを導出してゆくという"定石"を踏まなかったこと。③とにか

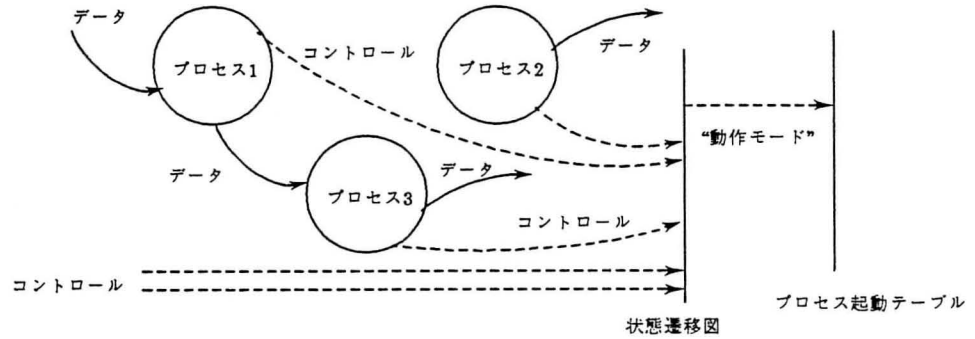
く書いて、後でどんどん追加･修正してゆけばいい、と考えず、はじめから正解を求めてしまったこと。①は本質的にはツールの支援は期待できず、②もツールは何も支援してくれず方法論マスターあるのみ。一方、③はツール使用経験を重ねればそれなりに解消できる内容であった。

そんなわけで作業初日が終了。このまま明日になっても進展は望めないので、ダイヤグラム0の作成を各自の宿題とした。せっかく盛岡まで来て酒も飲まずにホテルで宿題?!、と反対意見が続出するかと思いきや、さにあらず。とにかく自分の考え方で書いてくることにして、わんこそばやさんに向かった。大方の人はそのあとも街に繰り出したようだが、それでも翌朝には多くのレポートが提出され、少しづつ前に進み始めた。(深夜まで拘束し、申し訳ありませんでした。)

ダイヤグラム0の作り方は基本的に2通りに分かれた。①給湯する、お湯をはる、追い炊きするという機能面からプロセスを決定したAグループ。②湯釜の火力、バブルの出水量、弁の開閉･切替信号という出力データから湯釜の制御、弁の制御、センサー監視(これがポイントか!)というプロセスを決定したBグループ。とにかく大方針が決まったので、後は2班に分かれてまたツールに向い始めた。

構造化分析手法の最も肝心な作業であるダイヤグラム0の作成を、ツールを使わず紙と鉛筆と頭で行った(より正確には行えた)わけであり、これが現在のCASEツールを象徴しているといえる。

さて、ダイヤグラム0の各プロセスとその間のデータフローが決まり、次はそのプロセスがどのように起動/停止するかの制御仕様の記述となった。これはプロセス起動テーブルにて、コントロールフローに応じて起動されるプロセスを指定するのだが、ダイヤグラム0上にあるコントロールフローのみではうまく表現できない。そこで、各プロセスの起動/停止を指示する"動作モード"なる中間的なコントロールフローを導入し、ダイヤグラム0上にあるコントロールフローと"動作モード"との関係を状態遷移図を用いて記述した。ここがひとつのミソかもしれない。
これで、ダイヤグラム0のプロセス、プロセス間のデータフロー、プロセス間のコントロールフロー、各プロセスの起動/停止条件が一応記述できた。



2.3 下位層のダイヤグラム作成

A、Bグループともダイヤグラム0上の各プロセスについても同様に、データフロー、コントロールフロー、(プロセス起動テーブル)、(状態遷移図)を作成し、なんとか構造化分析の体裁を整えた。

2.4 ストラクチャ･チャートSCの自動生成

時間がなくなったので、空のミニ･スペックとデータ辞書を作成し、出来上がったダイヤグラムの内容チェックをツールで行ったあと、ツールにある自動変換機能を用いてSCを自動生成した。基本的にはDFDの各プロセスが一つのモジュールとなり、各プロセスへの入出力データとコントロール信号がそのままモジュールの入出力カップルに表現されている。状態遷移図やプロセス起動テーブルがどう表現されるか期待していたが、単なる一つのモジュールの箱になっておりチョットがっかり。リアルタイム手法でコントロールフローや状態遷移の概念を追加してあるのだから、SC上にもそれなりの動的なタイミング制御を表現するうまい仕掛けはできないものか。これはツールの問題というよりも手法の問題である。

SCは①論理モデルのシステム設計、②物理モデルのシステム設計、③論理モデルのプログラム設計、④物理モデルのプログラム設計、への活用が考えられる。①②③とも、基本的にはデータ構造を無視してデータが各モジュール間をどう流れるかに着目すればよく、各モジュールを静的、論理的に見ているのでSC記法でも内容は表現できる。ところが、④物理モデルのプログラム設計ではデータ構造を反映させて、各モジュールの動的振る舞いを考慮する必要が生じ、更に制御系ではモジュールの起動/停止という動的概念も入ってくる。

Ward、HatleyがDFDに制御プロセス、コントロール･スペック線を導入したように、モジュール間の静的関係しか表現しない今のSCにリアルタイムSC、物理モデルSC用拡張が必要と思う。

現場でSCがあまり使われなかったのは、SCはあくまでも論理モデルまでしか表現せず、物理的なプログラム･モジュールを導出する思考過程を支援してくれなかったからではないか。つまり、論理モデルSCと物理モデルSC間には単なるチューニングでは済まない大きなギャップがあり、時間をかけて論理モデルSCを作ってもその後工程があまり楽にならない。設計は論理的構造だけでなく、物理的なモジュール構造決定もその重要な作業なのである。

ならば、CASEツールはそのギャップ解消をどう支援してくれるか。鍵はソース･プログラムからの物理モデルSC自動生成にあると思われる。うまく構造化されたプログラムをCASEツールでわかりやすい物理モデルSCに逆変換し、SC上で思考する経験を積み重ねることにより、アナロジー効果を狙うわけである。TopDownばかり追求せず、BottomUpも経験してはじめて、TopDown思考をものにできる。物理モデルSC自動生成はドキュメント生成という直接的効果ばかりでなく、新たなSC作成を間接的に支援する。

ところで、**Teamwork**はDFDからSCへの自動変換にあたり、ダイヤグラムをチェックし、不完全であればその旨を表示してくれた。両グループともここをパスしたからSCが生成されたわけだが、このチェックはどこまで当てにしていいものだろうか。Aグループ例では給湯プロセス内の各サブ･プロセスがどう動くかのコントロール･スペックが無い。Bグループ例ではMCランプ信号、BCランプ信号がコンテキスト上ではデータ、DFD上では信号、又ダイヤグラム0のS1コントロール･スペックのSTDでは、次のプロセス起動仕様PATへ渡す"動作モード"の定義が抜けている。ツールは形式的なチェックしかできないだろうが、これらのチェックを期待するのは酷か。

2.5 全体を通して

最終日の発表にあったとおり、Aグループはこれらの作業をあまり机上で深刻に悩むことなく、ツールを用いて試行錯誤的にこなしていった。一方、Bグループは黒板を使って真剣に討論しながら、結果をツールに入力していった。その意味ではAグループの方がBグループよりもツールを有効に活用したといえる。つまり、Aグループは一つの手順･方法論に沿ってツールを使っていくうちに、粗削りではあるが徐々にシステム全体が見えてきたわけであり、方法論やツールがある意味では役に立ったのである。

一方、Bグループは課題解決に本格的に取り組み、ある程度完成した結果をツールに入力した。方法論自体は役に立ったが、それらの思考過程をツールがどこまで支援したかという点では、Bグループにとってツールは不要だったことになる。

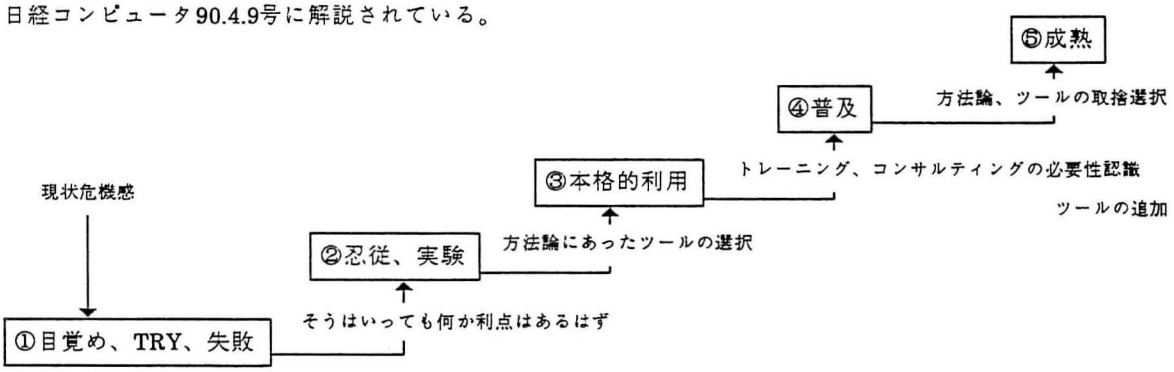
では、できあがった成果物はどうか。形式的には両グループともほぼ同レベルの分析内容であるが、これで目的の給湯システムが動くかはよくわからない。僅か二日間の作業のため結論を出すのは早計ではあるが、同一レベルの人が同一方法論で作業したとしたら、ツールをより長時間使ったAグループがより良い結果を示していればツールの有効性を証明できたことになる。しかし、問題はそれほど単純ではない。

3. CASEツールの有効性

方法論にせよツールにせよ役に立つ、立たないという議論をする際、①誰にとって②どの局面で、という前提を置かなければ総論で終始してしまう。

3.1 誰にとって有効か

ノーランの情報システムステージ理論と同様、CASEにも5段階モデルなるものがあると、以下のとおり日経コンピュータ90.4.9号に解説されている。



日本ではほとんどの人が最初の目覚めステージにあると思うが、その中は更に①何も知らない、②記法は知っている、③方法論を一応知っている、④方法論を使いこなせる、と分類できる。
①又は②の人にとって、CASEツールは一見面白く魅力的に映る。既に紙と鉛筆で③、④を実践し失敗している人にとっては、CASEツールは自らの思考を支援しない単なるお絵書きツールにしか映らない。

ツールへの期待度、到達点設定レベルが異なることにより、役に立つ/立たないとなるわけで、今回A、Bグループで方法論、ツールに対する評価に差が出たのはそのせいである。

3.2 どの局面で有効か

現状分析、システム分析という抽象化作業は人間の思考によるものであり、ツールはそれをきれいに表現するに過ぎない。思考そのものの支援を期待する人には物足りないが、ドキュメントツール程度に考えている人には有効である。DFD,CFD,STD,SC等を手書きで作れと言われたら断りたくなる仕事も、高級WSで操作性の良いツールが使えるならやってみようかな、と思わせる副次的効果は確かにある。

要求仕様、論理モデルを明確にして、ユーザー･アナリスト･プログラマー間でのコミュニケーションをはかるという目的で用いるのならば、ツールはあった方がいい。ただし、その方法論･記法がターゲット･エリアの分析にフィットし、かつ上流工程にそれだけの時間をかけられ、時間をかけないと下流にきて問題が起きそうな場合である。(このただし書きがくせ者なのである。)

裏返せば、ツールが支援する方法論がその領域分析に不十分で、時間がなく、かつ上流工程が不十分でも下流工程の頑張りで何とかなってしまっている状況ではツールは普及しない。“目覚め”へのトリガーとなる現状の危機感を体で感じない限りステージ1、2へと発展してゆかない。

4. CASEツールへの期待

現状の危機感をCASEツール導入に結び付けるためには、さらにツールへの“幻想”も必要である。結婚と同じく、過度の期待は後で裏切られるが、とにかく一度は騙さなくてはならない。でないとステージ2に進めない。今のCASEツールを魅力的に見せるには以下の機能くらいは欲しいと思うが、いかがなものか。

① 階層毎のDFDを一つにまとめて3次元で表現してみせる。これにより全体が見渡せる。

② 一定の作業手順シナリオに従って、ツールが次にすべき作業を指示してくれる。例えば、DFD作成後はCFD作成指示が出、次はPAT,STD...とその状況に応じてツールがガイドする。プログラミング･チューターならぬ『CASEチューター』である。整合性チェックも入力の都度行い、誤りは即座に指摘してくれるとありがたい。

③ リバース･エンジニアリングとして、ソースプログラムをインタプリトしながら、その実行状況を各種ダイヤグラムでアニメーションする。これで保守地獄はかなり救済できるだろう。

④ 分析･設計をデータフロー、データ変遷で行うのならば、プログラム構造へのマッピングもその延長で考えられる言語処理系が欲しい。データフロー型でマルチ･プロセスを支援する『N次元データフロー言語』である。私は数年前から事務処理プログラム設計にデマルコ流データフロー言語を使っているが、これは1階層しか支援していない。N階層まで支援してくれれば、まさに論理モデルがそのまま物理モデルに変換でき、より上流工程からDFDを使う気になるだろう。これにより、分析、設計、製造がスムースにつながり、プログラミング工程の省力化もはかれることになる。
つまりは、UpperCASEとLowerCASEの融合といったところか。

5. おわりに

今回のワークショップはプログラム委員というよりも、参加者と同じ立場で勉強させてもらいました。リアルタイム構造化手法は普段使ってなく、インストラクターの田中さんには大変お世話になりました。この場を借りてお礼申し上げます。また9人の参加者の方も、それぞれのバックグラウンドを棚上げして、進行にご協力頂きありがとうございました。静岡大学の皆さんには夜遅くまでお待たせし申し訳ありませんでした。
ある意味では競合する各社が技術啓蒙という目的で協力して頂いたわけで、大変意義深いワークショップであったと思います。CASEツールを完全にマスターするまではいきませんでしたが、ツールの現状、操作性、リアルタイム構造化手法、オブジェクト指向等多くの収穫がありました。
最後にこのような機会を与えて下さいました落水プログラム委員長に感謝いたします。

10. まとめ

SEA 若手の会 in 盛岡 報告

— SEA Forum, 1990.10.17 —

落水 浩一郎
(静岡大学)

1990 年 9 月 5 日から 8 日にわたって開催された CASE に関するワークショップの結果を報告する.

1. ワークショップの目的

演習問題を実際に解決する過程を通じて, 設計方法論, CASE ツールに対する理解を深める.

2. ワークショップの構造

(1) 参加者のプロフィル

・参加の動機

CASE ツールへの興味	24
仕事上の必要性	16
上司の指示	23

・日常の業務分野

事務系	20
制御系	14
その他	13

・その中でのあなたの役割

要求分析	24
基本設計	31
詳細設計	28
プロジェクト管理	13
その他	20

(2) 日程

9 月 5 日

・方法論講義 (SA/SD, OOA/OOD)	各 1 時間	計 2 時間
・CASE ツール機能紹介 (5 社)	各社 30 分	計 3 時間

9 月 6 日

・問題説明とグループ分け (事務系, 制御系各 2 題; 5 グループ)
(グループの構成) 参加者 1 名, コンサルタント 1 名, インストラクタ 1 名, 記録者 1 名
・グループごとのツール機能, 操作法詳細説明 2 時間
・問題解決 3 時間
・中間報告会 2 時間

9 月 7 日

終日問題解決

9 月 8 日

最終報告およびパネル討論 3 時間

3. ワークショップの評価 (参加者からのアンケート集計結果)

以下のようなアンケートを最終日，パネル討論中に実施した．

ワークショップを通じて CASE ツールに対する理解・認識がどのように深まったかについてお尋ねします．
御自由にお書き下さい．

1. ワークショップを通じて得られたもの (成果)
2. 欲っしたが，得られなかったもの (今後への期待)
3. お使いいただいた CASE ツールに対する要望
4. 全体的な感想

---

(1) 成果について

方法論について

・抽象化する時の発想について多くのヒントを得た (OOA)．
・オブジェクト指向が部品化，再利用に適していることがわかった．
・オブジェクト指向が入門程度理解できた．
・オブジェクト指向，分析の概観と理解の糸口がみえた．
・構造化設計，データ中心の設計の意義が見えてきた．
・SA に従った分析の実体験 (プロセス中心からデータ中心)．
・SA/SD に対する深い理解が得られた．
・SA/SD によるデータ中心の設計手法の理解．
・SA/SD の考え方を具体的に知ったこと．
・SA/SD を使った実際の体験 (誤解に気づいた)．
・SA/SD 手法を使うと従来試験時に発生する多くの問題をつぶせる．しかし理解に多大の労力を要する．
・方法論が多少なりとも理解できた．
・いかにすぐれたツールであってもその方法論を知らないと宝の持ちぐされである．
・CASE ツール使用にあたっての手法，方法論の必要性の認識．
・今後仕事を進めていく上で，なんらかの手法 (法則) があることを知ったこと．
・今回のように，分析に時間をかけてじっくり作業を進める方法を体験できてよかった．

ツールについて

・作図，編集が簡単にしかもきれいにできるのは感動した．
・ツールへの嫌悪感が取り去られた，と同時に夢も消えた．
・CASE ツールへの興味を持つことができた．
・現状の CASE ツールの機能の一部の理解．
・お絵書き機能の現状把握．
・上流の CASE ツールの一部に接し，その現状の一端をのぞきみたこと．
・CASE ツールで何ができるかということについて実感した．
・汎用性を考えるとまだまだ有用でない．

全般的に

・CASE は有効である．
・CASE ツール以前に人的な要素が成否を握っている．
・エンドユーザの立場からシステムを使うという貴重な体験をした．
・経験，馴れがいかに自分の頭を固くしているかがわかった．
・各ベンダーの考え方の違いがはっきりした (方法論，ツール)．

・色々なベンダーさんのアドバイスが聞けたこと．
・社風，技術等の問題で CASE ツールの導入は難しい．
・こういう有意義な会合はもっとどんどんやるべきだ．
・自分たちの保守性や目さきの利益を見てしまう傾向に対する再認識．
・他の会社での CASE ツールの使用状況．

交流について

・他社の方の話を聞け，自社の立場，自分のレベルが認識できた．
・同年代の他社技術者との交流．
・同年代の人の考え方．
・さまざまな人々との出合い (目ざすところは１つか)．
・ヒューマンリレーション．

(2) 欲したが得られなかったもの (今後への期待)

方法論関係

・OOA/OOD/OOP のすべてを経験しその上で，方法論やツールの有用性を確認したかった．
・OOA/OOD 全体の経験．
・OOA における抽象化手法の手がかり (トレーニングツール)．
・SA/SD と OOA/OOD の関係に対する将来展望．
・現実にそった方法論．

有効性

・事務系を本格的にサポートする CASE を体験したい．
・事務処理に対するツールの有用性．
・方法論が本当に実務に役に立つのか (従来の手法より金になるのか)？
・ノウハウについてまとめたものがあればもっと簡単にはいっていけるのではないか？
・大規模開発に CASE は有効か？
・リバースエンジニアリングの現状やワークステション以外の環境への普及状況等に対する解説．
・リバースエンジニアリングへの適用法．
・保守の支援．
・ライフサイクルとの対応を明確に．
・CASE ツールや方法論のすばらしさ．
・CASE が過去に積み上げてきたものを破壊するための勇気と力をあたえてくれるだけのパワーがあればよいと思います．
・ハードのアーキテクチャ，ソフト，コストのトレードオフで仕様がボンボンと変化するような場合，プロトタイピングを繰り返しながら仕様をまとめていく段階をサポートするものがなかった．

導入法

・導入にあたっての前提条件をしりたい．例えば，要求分析にあたっては純粋なユーザ，ユーザから SA になりつつある人，SE，プログラマに適するツールはどれか．方法論や記法の教育問題，前提とするコンピュータに関する知識等．
・実際に使用するためには，プロジェクトへの説明，理解が必要である，古い顔 (上司) に理解させることが難しい．効果について，開発工数や保守工数に関する値がほしい．
・いかにして現場に定着させるか？

(3) CASE ツールに対する要望

ツールのサービス

- リバースエンジニアリング的アプローチ.
- ツールオリジナルだけでは満足できる環境は作れない，CASE ツールはカスタマイズのお手伝いをして下さい.
- 方法論の支援が不十分.
- 思考のサポート.
- チームで分析設計する場合のネットワークのサポート.
- ソフトウェアライフサイクル全体の支援.
- ある手法にしばられないという姿勢を維持して欲しい.
- 多機種対応.

ツールの機能

- DFD 内に記述した名前 (データ名，プロセス名) を，キーボードからの入力フィールドにもってこれること.
- DFD 修正による DD の自動修正機能.
- 最適な形へのリフォーマッティング機能.
- HELP 機能の充実.
- 独学するには多くのドキュメント，マニュアルが必要だ.
- レビューをサポートする機能が欲しい.
- アニメーションやナビゲションの機能.
- 設計完了後，シミュレーションができること，とくに制御系のタイミング問題.

ツールの操作性

- エデイタの機能アップ (カット & ペースト，アンドウ).
- エディタを強化し，わずらわしさを感じさせないようにして欲しい.
- is-a, has-a の関係を示す線を引くときどうしても見づらくなる.
- 画面の縮少機能が欲しい.
- OS を意識させない操作性.
- Unix を意識せずに使用できるように.
- UI をもっとオブジェクト指向にして欲しい.
- UI がいまいち．操作の統一性があまりなく，モードが多用されすぎる．ただし，あまりツールがリコウだと人間がバカになるのでこの程度．でよいか?
- UIM をものすごく充実させて欲しい (ハードのアーキテクチャのシミュレーション等).
- UI がもうすこしよければよいのに.
- 性能，UI(特に日本語) の面での改善が必要.
- 日本語がいまいち.
- 日本語入力の簡易化.
- 文字入力がワープロに劣る.
- もうすこしマウスだけで操作できるようにして欲しい.

その他

- 数カ月試用したい.
- 保守業務に利用できるのか？
- 価格が高い.

(4) 全体的な感想

自分自身にとって

- 課題に対する成果はともかくとして，この 4 日間で CASE を導入して実作業にいかせる気がした．
- 貴重な経験，情報が得られた．
- 参加してよかったが，方法論を十分に理解しないまま参加したのはよくなかった．方法論を理解していなければ CASE ツールは使いこなせない．
- 環境がないものにとっては整えるのは大変だができればこれから使ってみたい．
- 自分も含めて参加者がまだまだ未熟である．
- この 4 日間で手法，ツールをマスターし会社にそのスキルを持ち帰ろうなどとは最初から期待していなかった．しかし今後の向学心，興味，理想追求のひとつの点火となればいいなと思っていた．今思えばこれも他力本願であった．やはり自分がまずやる気をおこさなければ何も良い結果はもたらされない．
- SA/SD,OOA/OOD など有用な考え方が提唱されているが自分自身に問題意識がなければ現状のまま推移していくだけである．
- 第一線で開発，保守に従事している他企業の若い人と接し，その考え方問題意識について話をする場をもててよかった．
- 方法論，ツール，記法の習得以上に社外技術者との意見交換ができたことが非常に有益な経験になった．
- CASE ツールを活用するとしたら，システムライフサイクルとしての有効性に期待したい．分析設計の部分はもっと'あいまいな部分を残したまま作業ができるような形にして欲しい．
- 要求分析や設計の支援だけでなく，現状システムの保守にも使えるように．
- ハード，ソフトの値段が高すぎるので自分には関係ない．
- 課題を分析する際の参考例が欲しかった (比較のため)．
- 手法 (含む CASE) と宗教の類似性を感じた．
- 構造化設計は従来通りの設計時間がかかる，しかし，テスト時間は短縮できると思う．

11. ポジションペーパとアンケートの回答

参加者名簿 190

ポジションステートメント & アンケート回答 191

After Workshop ポジションペーパ

富永 伸哉 228
平尾 一浩 230

参加者名簿（申込順）

No	氏　名	所　属	グループ	Page
1	田口 秀成	新日鉄情報通信システム	OOA/OOD	191
2	阿部 和弘	富士ゼロックス情報システム	CASE Bench	191
3	佐藤 哲太	富士ゼロックス情報システム	Teamwork	192
4	井上 忠則	シスプラン	OOA/OOD	193
5	平尾 一浩	ヒラタ ソフトウェア テクノロジー	OOA/OOD	194
6	富永 伸哉	ヒラタ ソフトウェア テクノロジー	Teamwork	194
7	木村 茂和	日商エレクトロニクス	PWB	196
8	満田 豊	シーイーシー	StP	197
9	刀根 利光	シーイーシー	Teamwork	197
10	池川 哲夫	アスキー	PWB	198
11	西垣 郁夫	東京電力	StP	199
12	清水 洋子	東芝	StP	200
13	高橋 秀行	SRA	OOA/OOD	201
14	飯塚 宣男	カシオ計算機	PWB	201
15	大篭 高之	日本システム	Teamwork	202
16	藤原 千尋	日本ラッド	CASE Bench	202
17	藤井 康範	MHI エアロスペース システムズ	PWB	203
18	遠藤 晋	アスキー	OOA/OOD	204
19	泉田 昭義	岩手電子計算センター	CASE Bench	205
20	似内 均	岩手電子計算センター	PWB	205
21	三上 修	岩手電子計算センター	StP	206
22	佐藤 琢美	岩手電子計算センター	OOA/OOD	206
23	松本 幸三	CSK	OOA/OOD	208
24	青柳 和久	東北 SRA	PWB	208
25	思坊田 和典	日本電気ソフトウェア	CASE Bench	209
26	松浦 光哉	中央システム	CASE Bench	210
27	本間 匡（代理: 前 雄介）	中央システム	OOA/OOD	210
28	山島 利彦	日本エムアイシー	StP	210
29	浜 恭士	SRA	StP	211
30	林 明賢（代理: 中谷 信子）	SRA	Teamwork	212
31	上田 賢一	SRA	CASE Bench	212
32	館山 純	SRA	PWB	212
33	阪本 善通	SRA	StP	213
34	奥村 吉彦	CSK	PWB	214
35	Dabin Matthieu	静岡大学	OOA/OOD	217
36	渡辺 瑞枝（代理: 小沢 英志）	SRA	Teamwork	217
37	吉田 一知	岩手電子計算センター	Teamwork	218
38	斎藤 基	富士通ビーエスシー	CASE Bench	219
39	森 彰三	富士通ビーエスシー	Teamwork	220
40	知念 徹	三菱電気コントロール ソフトウェア	Teamwork	220
41	柳瀬 健一	ケーシーエス	StP	221
42	築山 秀夫	ケーシーエス	StP	224
43	財田 雅史	ケーシーエス	CASE Bench	225
44	小林 透	NTT	PWB	226
45	大岡 俊之	東北コンピューターサービス	CASE Bench	227

ふりがな　たぐち　しゅうせい
氏名　田口 秀成

1. プロフィール

年齢:29 歳　　性別:男
会社名:新日鉄情報通信システム　　部門:研究開発部
これまでの主な開発経験
　大型汎用コンピュータの保守プログラム
　ATM の H/W の試験プログラム
　マルチメディア多重化装置の制御プログラム

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

　顧客とシステム開発の間に、共通の言葉がないため要求定義が明確にできない. いきおいシステム開発者が提供する要求定義は, あいまいなものとなってしまう. このことは, 開発工程を不鮮明なものとし, 後の工程での使用の変更を産み出す余地を作り出す. そこに要求定義が不鮮明であったことによることの, 使用への不満が入り込むのである. 以上に述べた要求定義の不明確さが, ソフトウェア開発の上流工程における現状の最大の問題点であると考える.

(2) それらの問題点解決に CASE はどう寄与するか?

　したがって, 先ず第一に要求定義を明確にするということが, 生産性向上の第一歩出あると考えるが,CASE は, 顧客とシステム開発者間の間に共通の言葉を提供することによって, この問題の解決に役立つものと思われる. そして, この確立された設計工程は CASE によってサポートされ信頼性, 生産性が向上すると期待される.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

　現在開発の対象となるシステムの, 大規模化と高度化につれてシステム開発者への負荷は, 増大している. 従って, この負荷を軽減するためにも,CASE の早急の導入が期待されるのであるが, 我が国に於いては,CASE の導入はあまり熱心でないのが現状である. これは, 我が国に於いては, ソフトウェアの生産性を職人芸と考える傾向があり, 工学的な手法に対して無関心であることが大きな原因のひとつではないかと思われる. つまり,CASE の導入以前にソフトウェアの生産性技術の導入がなされていないのが現状であり, ソフトウェアの生産技術にもとずいた,CASE に対する理解が不十分なのは当然である. 従って, システム開発者およびそのマネージャがソフトウェア生産技術を習熟し, CASE 導入の必要性, 有用性を確認することが第一歩である. もし, そうでなければ CASE を導入したとしても, 混乱を招くだけになる可能性もあろう.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ オブジェクト指向, 分析の概観
・ 同年代の人達の考え
・ 様々な情報
2. 欲したが, 得られなかったもの (今後への期待)
・ 時間が短かったので,tool, 方法論の理解が不充分であった.
・ SA/SD と OOA/OOD の将来的な展望 (合併, 吸収?)
3. お使いいただいた CASE ツールに対する要望
・ どこからどこまでが,St-80 で, どこまでが OOATools だかわからない.
・ グニャグニャしすぎている. キチッと説明してほしい.
4. 全体的な感想
・ 関係ないですが, ソフトウェアの値段が高すぎる.(ハードも)
・ たいへん勉強になりました.

ふりがな　あべ　かずひろ
氏名　阿部 和広

1. プロフィール

年齢:24 歳　　性別:男
会社名:富士ゼロックス情報システム株式会社　　部門:企画部付
これまでの主な開発経験
　Unitalk-80(視覚的 UNIX 環境) ポーティング
　GrapherGear(オブジェクト指向型プログラミング開発環境)

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

　要求分析の段階において, ユーザとの指向の疎通も難しく, 甚だしき場合には, 互いの持つドメインの違いによってコミュニケーションが成立しない場合すらある. このような状況では, 要求も明確な形で提示されなかったり, 無理なマッピングなどによって双方ともが誤った認識を持ってしまうことがある. この時点でかなりのノ

イズが発生する. さらに, 要求仕様がある程度明確化してきたとして, 次の設計段階では, 仕様を処理可能な形に落していかなければならない. このときは問題を適切に分割し, 関係を再構築するデザイナーの技量が要求される. ターゲットとして従来の手続き型の処理系を想定している場合は, 率直な表現が難しい為にひねりを加えなければならず, より困難である. さらには, 適用すべき要素技術の鑑定, ユーザとはもちろんのこと, メンバ間の意志の円滑な疎通などの問題もある. こうした困難を経て出来上がった設計にしても, 最初から最適会である可能性は極めて低い. 解が一意にしか決まらない問題であればよいが, 現実には試行錯誤的に様々な手法を実践する必要がある. しかし, 数をこなすことによってロードも必然的に大きくなり, 結果として妥協も多くなる.

(2) **それらの問題点解決に CASE はどう寄与するか?**

コミュニケーションの不足を CASE が直接に補うことはできない. しかし,CASE を用いることにより, 例えば, ユーザと開発者がお互いのドメインに共通した図法を用いて表示されたグラフを前に論議することによって,CASE は共通語としての意味を持つ. この際に動作のシミュレーションができればより有効であろう. 設計フェーズにおいても, 言語に特化されない図形表現は自由度が大きく扱いやすい. 現状で, メモを書きなぐっているものが, 画面上で楽に編集できるだけでもメリットはある. もちろんこの際にも動作シミュレーションができれば, 試行錯誤も楽に行なえる. 最終的に図形表現から各種言語へのスケルトンが得られれば良い. いずれにせよ, 今後は SA,SD の為のツールではなく, 問題を自然に扱える OOA,OOD の為のツールが主流になるであろう.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**

導入に反対する勢力に対しては, 圧倒的なアドバンテージを示せるだけの成果を上げて見せることが必要である. 当然世の中に完璧なものは無いわけだから, どうしても機能や性能に不足はあるだろう. しかし, それらのことを些細なものであると断言できるだけの生産性を上げることができればよい. 無論,CASE も機能や性能がチューンできるようなカスタマイズできる環境でなければならない.

3. アンケートの回答

1. **ワークショップを通じて得られたもの (成果)**

自分達の保守性や, 目さきの利益を見てしまう傾向を再認識させられました.

2. **欲したが, 得られなかったもの (今後への期待)**

CASE が, 過去や積み上げてきたものを破壊するための勇気と力を与えてくれるだけのパワーを持っていれば良いと思います.

3. **お使いいただいた CASE ツールに対する要望**

ファンクションは良くそろっていると思います. ツール間のつなぎ目, ユーザインターフェースの洗練などを改善していただきたいと思います.(CASE Bench)

4. **全体的な感想**

様々な人々の様々な考えを得ることが出来て有用でした. また, 手法 (CASE 含む) と宗教の類似性を感じました.

ふりがな さとう けいた
氏名 佐藤 啓太

1. プロフィール

年齢:26 歳 性別:男
会社名:富士ゼロックス情報システム (株) 部門:企画部 F21
これまでの主な開発経験
制御系マイコン開発
クロス C コンパイラおよびシミュレーション・デバッガ開発

2. ポジションステートメント

(1) **ソフトウェア開発の上流工程における現状の問題点は?**

現在私が経験してきたソフトウェアを開発する上でもっとも大きな問題点の一つは, ドキュメント [1] が定型化されていなかったことである. 今まで私は, ドキュメントを「書かなければないから || 」というように義務的に書いていた. ドキュメントに関してとくに教育を受けたわけでもなく, 読み手が必要としている情報を書き出すことは非常に困難であり, また苦痛であった. 真っ白な紙に「プログラムの説明をかけ!」 と言われた方が書きやすいし, 読む方にとっても後の方が読みやすいのは事実である. また, ドキュメントが不完全であるがために, 保守に耐えられず最初から作り直さなければならないと言うような無駄も生じた.

(2) **それらの問題点解決に CASE はどう寄与するか?**

ソフトウェア開発におけるドキュメンテーションをサポートするために, 分析・設計からテスト・保守に至るすべての段階のドキュメンテーションを定型化し, データベース管理を行なう. これにより資源の再利用が可能となる. この時に,WEB というプログラミング段階のドキュメンテーションをサポートする言語などを利用してみてはどうであろう. これは, プログラムにソースコードと解説を混在させて記述したものであり, ツールを用いてソースコードと解説を別々に出力するものである.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**

我々はドキュメンテーションをデータベース管理する研究を行なっている. 現在ここでいうドキュメントは, ソフトウェア開発にとらわれず幅広いものであるが, このようなソフトウェア開発のドキュメンテーションをサポートすることも興味深いテーマである.

[1] ここでドキュメントとは, 分析・設計からマニュアルやテキスト・保守までのソフトウェア開発に関するすべてのドキュメントである.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
   SA/SD がなんとなくではあるが理解できた.
2. 欲したが, 得られなかったもの (今後への期待)
   特になし.
3. お使いいただいた CASE ツールに対する要望
   tool の機能の一部にしか触れていないので今は特にコメントはありません.
4. 全体的な感想

☆ SA/SD & CASE tool について
- CASE tool というよりも, 思考支援の tool として興味を持った.
- 協調作業の機能を利用したかった.

☆ ワークショップについて
- スケジュール, 人数について問題があったのでは?
- 作業を行なうとき, 複数の人間で WS を使えるような机の配置をすべきであった.

☆ ありがとうございました. また Chance があれば参加したいと思います☆

ふりがな いのうえ ただのり
氏名 井上 忠則

1. プロフィール

年齢:25 歳 性別:男
会社名:株式会社 シスプラン 部門:技術部
これまでの主な開発経験
アプリケーション開発
通信. MS-DOS. MS-WINDOW.

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

私は, 仕事やアルバイト等でいくつかのソフトウェア開発を行なってきましたが, それらの仕事の内容のほとんどが設計された仕様書をもとにプログラムを作成するというものでした. ですからソフトウェア開発の上流工程における仕事を行なったことがないので現状の問題点の実例を上げる事はできません. しかし, それらの仕事で受け取った仕様書の中にはこれが仕様書というものもありました. どうしてこのような仕様書ができたのか?作成する過程でいくつかの問題点があったにちがいありません.

(2) それらの問題点解決に CASE はどう寄与するか?

先ず,CASE で何ができるのか, 何をさせたいのかという事が自分で意識できていなければ何もできないように思えます. また CASE が本当に必要なのか (他に簡単で安く行なえる手段がないのか?) など今ある環境を把握していてはじめて CASE をより有効に活用することも可能になってくると思います. 当たり前の事ではあるが, 今まで慣れてきた環境が変わることに対するデメリッとよりもメリットがなければ導入する価値がありません. それに所詮コンピュータを使うのは人間なのだから無理矢理習うより慣れろなどといってもダメなものはいくらやってもダメと思います. 自分自身に対していっているのですが今回の CASE でも他のシステムでも常日頃から興味を持ってアンテナを伸ばしておくと導入したときの障害などもすばやく対処できると思います.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

はっきり, 簡単にいうと, わたしは CASE についてよく知らないという事です. このポジション・ペーパを書くに当たって, 参考にするために"CASE ツール","構造化分析とシステム仕様"とかの読んではみましたがおもしろかったのは後者の中にあった構造化分析のケーススタディの例ぐらいでした. ちなみにそれは競馬と星占いをあわせたサービスを行なうというもので, ついそのくだらなさにひかれ一所懸命読んでしまいました. とにかく, 構造化分析などを行なうには豊富な知識と経験が必要なことがあらためて解りました. それと CASE の導入が今すぐにでも (将来的にも) 必要なのかと思いました. これは実際に上流工程での問題に直面していないし, 構造化分析が難しくめんどうに感じたせいなのかもしれません.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
   CASE は有効であるということ.
2. 欲したが, 得られなかったもの (今後への期待)
   具体例がもっと見たかった.

ノウハウに対して, まとめたものがあれば, もっと簡単にはいっていけたのではないかと思う.

3. お使いいただいた CASE ツールに対する要望

独学するには, 多くのドキュメントマニュアル等の準備が必要.

4. 全体的な感想

・ CASE ツールの一部分しか実際には使うことができなかったので全体を通して使ってみたかった.
・ 環境がないものにとっては整えるのは大変だが, できればこれから使ってみたい.

ふりがな ひらお かずひろ
氏名 平尾 一浩

1. プロフィール

年齢:29 歳 性別:九州男子
会社名:ヒラタ ソフトウェア テクノロジー (株) (略称 HST) 部門:システム開発部
これまでの主な開発経験
CRT 生産ライン 生産制御 生産管理システム
システムキッチン生産ライン 生産制御, 生産管理システム
DD ロボット
GKS (PC 版,NEWS-X 版)

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

要求分析や要求定義において, 方法論は複数あるが, それらの技術が現場で使われていないというところが大きな問題点であろう. 個人の勘と経験を基にした作業が多く, 方方論を用いた分析はあまり使われていない. これは, それらの方法論を簡単に使うことができないためにそうなっていると考えられる. また, たとえ方法論, 技法を用いたとしても, 下流工程への接続が困難である. これは開発プロセス全体を統一的にサポートする方法論もツールもほとんど無いためである. この問題が, 部分的な対応となる上流工程の方法論, 技法を使用しない原因の一つの要素になっていることも考えられる.

(2) それらの問題点解決に CASE はどう寄与するか?

方法論にしたがった一連の作業をマシーン上で実現でき, さらには, 分析などの作業工程で起こるミス, つまりデータの重複や, 矛盾など,CASE により, その過程毎にチェック, 修正が可能となる. このことにより, 中間生成物の信頼性を向上させるとともにその効率を高めることが期待される. ただ, 気を付けなければならない事は,CASE ツールがあってもそのサポートしているソフトウェア工学の方法論を知らなければ何の役にも立たないということである.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

CASE を導入することで, その基にある工学的方法論, 技法を導入したことにはならない. 導入の際には, その方法論をまず使用してみて (手作業で充分), 自分の開発分野に対しての効果を, まず検証してみる必要がある. その段階で効果か期待できれば, さらなる効果 (作業効率, 信頼性) を期待してその方法論をサポートしている CASE を導入すれば良い. この過程過程を抜きにすれば, 過度の期待を背負い, 聞いただけのデータで導入を検討しなければならなくなる. こうなればもう先は見えている.

CASE ツールも, 今後は単にツールと呼ばれるようにならなければ本当に使用され, 効果, 効率を出すことは期待できないんじゃないかな, と思っています.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ オブジェクト指向 (入門程度かな)
2. 欲したが, 得られなかったもの (今後への期待)
・ OOA から OOD などの流れを実際に行えなかったため, フェーズの切り方, 開発プロセスを得る事ができなかった.
3. お使いいただいた CASE ツールに対する要望
・ 思考のサポート
  ・ 方法論は本
  ・ 表現だけが CASE tool →これがつらい
4. 全体的な感想

CASE の大前提である方法論に対する理解やけいけんが少ないのはやはりちめい的だなと思った. やはり少しの思考サポートが必要ではないでしょうか.

ふりがな とみなが しんや
氏名 富永 伸哉

1. プロフィール

年齢:25 歳　　性別:男
会社名:ヒラタ ソフトウェア テクノロジー株式会社　　部門:システム開発部
これまでの主な開発経験
★ ロボット制御システム開発
★ 工業生産管理システム開発

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

ソフトウェア開発の上流工程における問題点としては以下のような事が挙げられる.–要求分析の甘さ要求分析の段階でユーザと開発者間の共通言語か欠如しているために, ユーザの要求を十分に聞き出す事ができない. また, 下の鵜システムを表現する諸気団会のモデルが欠如しているために, ユーザ自身のあいまいな部分を明確にする事ができない. そして, 設計, コーディング, デバッグと上流工程から下流工程へと影響し, 最終的にはメンテナンス地獄へと陥ってしまう.–各種開発手法の不浸透上流工程における開発手法として構造的手法等が随分前から提唱され実施されているが, 実際これらの開発手法はさほど浸透しておらず (私のまわりでは), 今だに開発者独自の設計等が行なわれている.

(2) それらの問題点解決に CASE はどう寄与するか?

–要求分析の甘さ要求分析の段階においては, ユーザと開発者間の間に共通の言語がないためにユーザの要求を十分には理解できない. そのために, 最終的システムは必要のない機能が実現されていたり, 重要な機能が抜けていたりする. 例えば, 構造的分析を行なうことによって言葉のあいまいな部分を排除することができるし,DFC や E-R 図などによって現行システムをモデル化でき,Visual 化することができる.–各種開発手法の不浸透各主開発手法があまり浸透しない原因の一つに, 手作業による作業進行が挙げられる. 実際に構造化手法を用いて上流工程の開発を行なったことがあるが,DFD やデータ・ディクショナリ,E-R 図などを手作業で紙の上に記述していくのは大変な作業である (消しゴムの一つや二つは覚悟しなければならない). また, 紙に書かれているために仕様の追加やバージョン管理, 整合性のチェックなど作業にも大きな労力とられてしまい, 分析途中でめいってしまう.CASE を用いてこれらの作業を行なった場合,DFD や E-R 図などの作図に対する労力は殆ど意識せずに済む. また, バージョン管理や整合性のチェックなども苦労せずに行なうことができるのではないかと考えられる.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

CASE の導入に関しては, 真っ先に考えられる障害として以下の二点がある.–各種開発技法の習得 CASE を導入し, 実際の開発に有効に利用するには各種の開発技法を理解することが大前提となる. しかし, 現在の CASE でサポートされている開発技法は随分前から提唱され, 実践されてきた技術であり, それなりの文献もかなりでそろっており, 技法の概念や文法を理解するのは問題ではないと思われる. さらに, これらの技法は実践で身につける部分が多く,CASE 導入によって短期間での習得が実現されると考える.– CASE の支援技法の古さ現在の CASE は大部分が構造化手法を支援していると思うが, これらは 1960 年代に提唱されたものであり, 既に多くの問題点も指摘されている. さらに, 90 年代はオブジェクト指向による開発手法 (OOA,OOD,OOP) が提唱され, 実践されてくると予想される. このギャップをどのように克服するかは, 今のところ私にはわからない. このようにソフトウェアの開発現場においては, 上流工程から下流工程まで様々な問題が発生している. これらの問題解決の手段として, 現在注目されているのが CASE テクノロジーである.

「各種 CASE ツールはどのようなものなのか!?」
「CASE ツールの問題点は何なのか!?」
「CASE ツールを導入し, 有効的に利用するには何をすればよいのか!?」
「最新のソフトウェア工学の技法と現在の CASE とのギャップは!?」

などを, 今回のワークショップに参加してプログラム委員各氏をはじめ, 参加者各氏の経験談と御意見をお聞きし, 共に論議できればと思っている.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
◎ SA/SD の方法論に関する深い理解が得られた.
◎ 現在世の中に出ている各種 CASE ツールを実際に触れる事ができた.
◎ 多くの人の意見を聞く事が出来たので大変良かった.
(やっぱりＳＥＡでないと聞けない!!)
2. 欲したが, 得られなかったもの (今後への期待)
3 と同じ
3. お使いいただいた CASE ツールに対する要望
☆ チームで分析・設計する場合のネットワーク機能の強化.
☆ ソフトウェアのライフサイクルを全体的に支援する機能の強化.
・ メンテナンスとバージョンアップを同時に作業できる
[全体 (仕様書等も含む) 版管理] 環境を支援してほしい.
4. 全体的な感想
◎ もう少し,CASE ツールを使いたかった.(9 人では,1～2 時間程度である)

◎ 方法論が中心になってしまって,CASE ツール自体の理解・認識を深める事が出来なかった.(CASE ツールに対する討論が出来なかった)
◎ 希望していた CASE ツールを使えなかったのが残念であった.
◎ ワークショップのプログラムがきつかったので, つっこんだ討論が出来なかった.

ふりがな　きむら　しげかず
氏名　木村 茂和

1. プロフィール

年齢:30 歳　　性別:男
会社名:日商エレクトロニクス (株)　　部門:ソフトウェア営業部
これまでの主な開発経験
　漢字 3270 エミュレータ (NET86(Multi16))
　外国証券システム・バッチプログラム (ibm3090 mvs)
　金融先物総合システム・バッチプログラム (ibm3090 mvs)

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
1-1 システム化要件がなかなかまとまらないこと.
1-2 Methdology が確立していない. 設計者によってドキュメントがまちまちになり第 3 者には判りずらい場合があります.
1-3 書き貯められた仕様書は下流工程で正確に反映されていない.
(2) それらの問題点解決に CASE はどう寄与するか?
1-1 に対して
　ツール (コンピュータ) 化により, 要件変更による下流での影響度が正確に計れる. ゆえに要件が確定しているところから次の段階の作業 (Makeing etc.) に入り途中で要件変更が発生した時でも柔軟に対応できるようになります. 従来は, 大規模なシステムになればなるほど修正による影響範囲が計れなかったり, プログラマレベルで混乱の元になったりで未確認要素を含んでの見切り作業は不可能であったが,CASE 導入によりある程度可能になるのではないでしょうか.
1-2 に対して
　上流工程での手法の統一化が計られる.(例 Data Flow Diagram/E.R Modeling) これにより個人に依存しないドキュメントが作られるため人のローテーションが可能となります.
1-3 に対して
　E.R Modeling からターゲットシステムで用いる Data-item を定義しそこから RDB の Table 設計にリンクしたり, プログラムが参照する Data dictionary が自動的に構築されれば上から下への工程は正確に反映されます. また D.F.D でプロセスを細かく記述していったとき最下位のプロセスがプログラム部品化の最小単位となります. あとは, ネスト状態の疑似コードかと部品からプログラミングの自動生成が可能となるでしょう.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
　現在, 世に出回っている CASE ツールで上流から下流までカバーし自分たちの現状にフィットするものはないと思います. そこで, ツールの手作りとなる訳ですができればありあわせのものを組み合わせたり, カスタマイズする程度で構築できるのが望ましいと考えます. 自分たちの開発の現状分析をし, 標準化案をまとめてみる. そこから次に CASE ツールを探すべきと考えます. その時そのツールが何ら化のカスタマイズを可能かどうかが重要でしょう.IBM ホストの適用業務プログラムの開発なら AD/Cycle はリポジトリィ(Data dictionary) と会話できる共通インターフェースを使って CASE ツールが作れると発表しています. これはひとつの自己構築ツールとして評価できるのではないでしょうか. 最後に私が考える CASE の基本機能を挙げます.
データ管理機能　→ ディクショナリィー構築を支援する.
プロセス表現　→ プログラミングの品質を保証する.
ツール管理機能　→ 部品 (モジュール) 管理
リソース管理機能 → プログラムソース・include ファイル等のバージョン管理
プロジェクト管理 → 案件・人・工数・進歩などの管理

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ 今回の各 CASE ツールは汎用性を考えるとまだまだ有用ではないことを認識しました.
・ CASE ツール以前に人的な要素が成否を握っていることが再認識しました.
2. 欲したが, 得られなかったもの (今後への期待)
・ 今回の参加者が各社に帰り成功した例を交換できればと思います.
3. お使いいただいた CASE ツールに対する要望
・ ツールオリジナルだけでは満足できる開発環境は作れないと思います.CASE ツールはカスタマイズのお手伝いをして下さい.

4. 全体的な感想
・ 自分も含め, 参加者がまだまだ未熟だと思いました.
・ パネルディスカッションにもっと時間が欲しかったです. 意見を言うタイミングがあったと思います.

ふりがな みつだ ゆたか
氏名 満田 豊

1. プロフィール

年齢:30 歳 性別:男
会社名:(株) シーイーシー 部門:第一情報システム部
これまでの主な開発経験
・ 銀行の経理システム 保守 開発 (バッチオンライン)
マシン iBM 3090 言語 PLi CoBol OS MVS/XA

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
ユーザーの希望する要件を分析しまとめる際に人によりまとめ方に個人差が生じる. それが, システムの漏れ等につながある恐れがある. また, ユーザーの希望する要件とシステム開発側のとらえた要件とのズレが起きる可能性がある. それは, ドキュメントがしっかりしていない場合, 特に注意しなくてはいけない.
(2) それらの問題点解決に CASE はどう寄与するか?
・ 標準化された開発技法をもとに, システム開発が進行でき, 仕様のもれを少なくすることができる.
・ しっかりしたドキュメントを残せるので第 3 者にも理解しやすく, ひいては生産性の向上にも寄与する. また仕様変更にす早く対応できる.
・ システム経験の浅い者でも, うまく活用すれば開発でき, 教育, 人材育成に寄与する.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
統一化, 標準化したシステム開発, 技法を研究し, 徹底させる.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ データ中心に考えていく構造化分析の手法に接することができ良かった.
・ オブジェクト指向も少しだけわかったような気がする.
・ こういう有意義な会はもっと力を入れてやるべきだ.
2. 欲したが, 得られなかったもの (今後への期待)
・ 事務系を本格的にサポートする CASE が体験できなかった.(IBM 系のもの)
・ 保守から見た CASE のサポートが各社とも乏しいと思う.(CASE は上流のものだけなのか?)
・ 各社とも事務系 CASE を軽視するな!
3. お使いいただいた CASE ツールに対する要望
・ 現在, 私は事務系の仕事をやっているが, それに比べたらはるかにすばらしい.
・ UNIX がわかっていれば, もっと良さを引出せたのかもしれない.(私は知らない為)
4. 全体的な感想
・ 色々の会社や分野の人と話ができ, 自分の刺激となった.(非常に良い)
・ CASE に触れられ, 新しい何かを見つけたような気がする.
・ 討論会の時, マイクが少ない.5 本位用意すべきだ.
(マイク移動の時間がもったいない!) 時間が少ない→不満!
・ SRA の桜井さん, 荻原先生, その他のスタッフ, ベンダーの方々, ご苦労様でした.
どうもありがとうございます.

ふりがな とね としみつ
氏名 刀根 利光

1. プロフィール

年齢:28 歳 性別:男
会社名:株式会社 シーイーシー 部門:千葉営業所
これまでの主な開発経験
ロボットのオフラインティーチングシステム (2 カ月)
3 次元測定機制御及びデータ評価システム (2 年 4 カ月)

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
(1) ユーザーの要求がうまく把握できない.
(2) (1) との関連も含めて, 仕様に抜けが生じる.

(3) 仕様がプログラマにいうまく伝わらない.
(4) プロジェクトによって表現方法がまちまちであるため理解に苦しむ. (一環した設計方針がない ドキュメントの基準がない)

(2) それらの問題点解決に CASE はどう寄与するか?

私の理解では, 上流工程の CASE ツールは大きく 2 つの側面を持っていると考えている. i) コミュニケーション・ツール … 他者との相互理解を促すための枠組みを提供する.
EX) 一環した設計方針の提示 (SA,SD 手法等) 標準的なドキュメントの自動作成 ii) 設計仕様 … 仕様の抜け, 不備, 矛盾等をプログラム以前にチェックできる.・設計方針が明瞭であるための手順を追うことによって設計所の不備を作成しながら把握し, 修正できる.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

(1) 教育, 普及の問題 … もっとも難しい問題であると考える. 導入するための人員の確保, 教育を行ない段階的に進めていくことが最も良いと思われる. ここで重要なのは,CASE ツールの使い方ではなく, 方法論の理解にあると思われる.

(2) その他・環境 … マシン,OS, ツールの整備・ツールの選択 … 自社の開発形態にそったものを選択する.・ツールの選択 … 同上・対ユーザー … ユーザの理解をどう求めるか. 逆に教育していく事が良いのではないかと思う.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
(1) 様々な方とのコミュニケーション
(2) SA/SD を使った実際の体験.(誤解に気づいた事)
(3) 上司ユーザの説得の方法 …(桜井さんありがとうございます).
2. 欲したが, 得られなかったもの (今後への期待)
(1) ツールの詳細な使用法.
(2) 現実にそった方法論.
(3) 全員の方との会話 (半数ぐらいはお話しましたが…).
3. お使いいただいた CASE ツールに対する要望
(1) 価格.(ちょっと他社に比べて高い).
(2) 多機種対応
(3) HELP, ナビゲーション, リバース
(4) 日本語化
4. 全体的な感想
(1) 有意義でした.
(2) 参加者より, 進行者 (プログラム委員等) の方が多く語っていたことは, 少々問題がある様に思われる.
(3) ツールというよりはやはり方法論なのだということを痛感した. なぜなら, ツールはまだ発展段階にあると思うし, 逆にもうこれ以上, 飛躍的に良くもならないと思う. そういう意味では, やはりマシン, ソフトより人が賢くならねばともいました. というギモンは残りますが.)
(4) 実行に携った方々に."ありがとうございました".

ふりがな いけがわ てつお
氏名 池川 哲夫

1. プロフィール

年齢:32 歳 性別:男
会社名:株式会社 アスキー 部門:システムソフトウェア事業部技術統轄部
これまでの主な開発経験
図形定義言語の開発
LSI 設計 CAD システムの開発
ドローイング・ソフトの開発 など

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

まずソフトウェア開発の上流工程, ソフトウェア設計ということを考えてみると, この作業自体が問題点を含むものであると思われる. つまり, 設計においてはメモリ効率/ スピード, 機能/コストなど様々なトレードオフが存在し, しかもどちらが良いかを示す尺度は曖昧であることが多く, 最適な解決法などは存在しないことが多い. このようなことは今のところ機械かではできがたく, 経験を積んだエキスパートにしか良いと思われる設計はできないだろう. ところが, 現状ではそのエキスパートは少数しか存在しない. けれども仕事の量が多いため, 経験を積んでない人とか経験があっても良い設計ができないような人が設計をして, システム全体に混乱を引き起こしているようなところが問題点であると思われる.

(2) それらの問題点解決に CASE はどう寄与するか?

上記のような問題点を解決するのにはCASEは2つの方向から寄与できると思われる. 一つは上流工程の機械的にできるところを,CASEによって強力に機械化し, エキスパートにもっと多くの仕事ができるような環境を作ることであるのだろう. もう一つは, エキスパートのやり方や知識を何らかの方法で蓄積し, 後で利用できるような環境をCASEによって作り, エキスパートでない人もエキスパートになれるよにすることであろう.

(3)CASE導入に際してさまざまな障害をどう克服すべきか?

本当に技術者の知性を増幅してくれ, しかも仕事が楽になるようなCASEなら導入にそれほど障害はないと思われる. それまでのやり方に固執する人がいるなら, こちらの方が良いという事実を説明し, 見せてやればいいのではないか. 障害があるのは問題があるCASEツールであるからではないだろうか.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ ヒューマンリレーション (知り合いがふえた)
・ CASE toolの表面的な理解 (種類, 名前, サポートする方法論など)
2. 欲したが, 得られなかったもの (今後への期待)
・ それぞれのCASE toolの詳細機能の理解
・ tool, 方法論の有用性の実感
3. お使いいただいたCASEツールに対する要望
・ マニュアルを充実させて (モデルの記法, 内容の詳細を含む)
・ ユーザインターフェースを洗練してほしい.
4. 全体的な感想
OOA/OODに興味がわいてきた

ふりがな にしがき いくお
氏名 西垣 郁夫

1. プロフィール

年齢:31歳 性別:男
会社名:東京電力株式会社 部門:情報システム部
これまでの主な開発経験
帝国データベースの「コスモス2」を購入し, 当社独自のデータベースを構築し, 社内に提供するシステムを開発した.

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

下流工程につながるような成果物 (内容) がなかなかできない.(例) 業務分析からシステムの検討をへて, 基本設計までの作業内容 (成果物) を詳細設計におとしていく時に, 手戻りがある. これは, 検討の不備やモレもあるが, 詳細設計以降の作業をすすめるには何をどこまでやっておかなければならないかがハッキリしていないからだと思います. その理由としては,(1)SEのスキル不足 (開発経験や業務側知識の不足も含め),(2) ツールの不足 (SEのスキル不足を補うもの, あるいは, 品質の向上そのものを目指すものなどの各種の方法論等が整備・充実していない) などがあります.

(2) それらの問題点解決にCASEはどう寄与するか?

(3)CASE導入に際してさまざまな障害をどう克服すべきか?

(2)(3) については現在検討中です.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ Upper CASEのツールの一部に接し, その現状の一端を「のぞき見」することができたことができたことは, 大変有意義でした.
・ 各ベンダーの考え方の違いがハッキリした (方法論やツールへのアプローチ etc.) 点では, 有意義でした. これによりケース・ツールに対し, ある程度サメた目で見れるようになりました. もっともっと勉強したいと思います.
2. 欲したが, 得られなかったもの (今後への期待)
(1) 全体の開発工程のなかで, 一体どの部分をどの程度サポートするものなのか.
(2) その導入にあたっての前提条件は何か ということについて, 例ば, 要求仕様の分析, 確定にあたって,(イ) 純粋なユーザー,(ロ) ユーザーからSAになりつつある人,(ハ)SEからSAになっている人,(ニ)SE,(ホ)PR etc. のうちどの人をターゲットにしたものはどのツールか.

それを導入するのに, 例えば, 構造化分析の手法,DFDの手法の教育問題, あるいは (イ)(ロ) の人々がコンピューターの知識のどの程度あった方がいいか, その教育の問題など (業務の分析とシステムの分析はやはり違うと思うので) もう少し明確になっていればいいと思います.

3. お使いいただいたCASEツールに対する要望

2 の回答と同じですが, その他に「stp」を私は使いましたが, もう少しマン・マシンインターフェースが良かったらと思いました.(インターフェースの部分については, 現在自分の会社で使っている某 W/S の某アプリケーションの方がいいと思われるので.)

4. **全体的な感想**

第一線で開発・保守に従事している他企業の若い人 (そろそろ中堅と呼ばれる人たち) と接し, その考え方, 問題意識について話をする場を設けてもらったことが, 上記 1 の回答のほかに, とてもありがたく感じた点です. 全体的には有意義な研修でした.

ふりがな　しみず ようこ
氏名　清水 洋子

1. **プロフィール**

年齢:23 歳　　性別:女
会社名:(株) 東芝　　部門:システム・ソフトウェア技術研究所
これまでの主な開発経験
・ プログラム変更支援ツール
・ CASE ツール評価項目の検討
・ OOA に関する研究

2. **ポジションステートメント**

現在「CASE ツール評価」「分析設計技法の研究」という仕事をしており,CASE に大変興味を持っています. 浅い経験の中から上記 3 点について私なりに考えることを別紙にまとめました.

(1) **ソフトウェア開発の上流工程における現状の問題点は?**

1.1 工程間の仕事の断絶

ドキュメントの最利用が各工程ごとに行なわれており, 上流工程の成果が下流工程に生かされていない.

1.2 後戻りと前工程のドキュメント保守

仕事変更などの後戻りに対して前工程のドキュメントを修正することがおろそかになりやすい. また, わざと上流におけるドキュメントで詳細を記述しないようにすることもある.

1.3 品質維持と納期キープの矛盾

人員不足や工数見積り誤りにより, 開発は納期に追われることがほとんどである. しかし, 品質面からは, ドキュメントをきちんと残すこと, 品質を維持するために工数をかけるようにする動きが強い.

(2) **それらの問題点解決に CASE はどう寄与するか?**

2.1 開発モデル・方法論の提供

分析・設計・コーディングの過程を方法論にのっとって支援し, そのなかで過去の開発ドキュメント類を有効に最利用することを支援する.

2.2 情報の一元管理

後工程での情報修正を前工程の情報に反映させていく. このためにも, 情報をどのように変換していくかという形式的なモデルを備えていることが重要となる.

2.3 品質維持

CASE が優れた開発手法を備えていれば, その手法にのっとって開発することで品質を維持できるという仮定に立つ.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**

3.1 一つのシステムの導入

一度にすべてのシステムに導入するのは困難であるが, 小さな一つのシステムに絞り, ライフサイクルを通して導入するようにする.

3.2 人間系の運用の確立

ライフサイクルを適して導入することが困難な場合, 導入部分と従来法とのつなぎめをどう運用するか, という人間系のシステムをよく考える.

3.3 長期的な視野

CASE 導入の効果は一回の開発ではなく保守や改良などを行なっていくうえで現れるものであり, 導入当初はかえって工数が多くかかることもある. この点を十分認識して導入するよう, 開発現場の意識を高める.

3. **アンケートの回答**

1. **ワークショップを通じて得られたもの (成果)**
・ いろいろな開発現場の方々の CASE に対する御意見が聞けたのがとても有益でした.
・ 方法論について, 具体的例題をもとに同じグループの方と議論でき勉強になりました.
2. **欲したが, 得られなかったもの (今後への期待)**
・ 5 つの CASE ツールが用意されていましたが, それらの思想の違いや特徴について, もっと具体的に知りたかったです.(ツール紹介や, 各グループの発表内容だけでは不十分でした).
・ 大規模開発に CASE ツールは有効であるのかを考えるきっかけが欲しかったです.

3. お使いいただいた CASE ツールに対する要望
・ 方法論の支援, 操作性が進歩していくことを期待します.
4. 全体的な感想
・ 方法論や CASE ツールを普及させていく活動をされている方々の経験や御意見が大変参考になりました.

ふりがな　たかはし ひでゆき
氏名　高橋 秀行

1. プロフィール

年齢:28 歳　性別:男
会社名:(株) SRA　部門:名古屋営業所
これまでの主な開発経験
・ ワープロに
・ 遠隔診断プログラム

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
矛盾を含んだ要求に, 頭でのあいまいな要求, 仕様案などを秩序だてて整理してゆくことの難しさ. またそのような過程でのコミュニケーションの問題は個人的な資質に左右される点.
(2) それらの問題点解決に CASE はどう寄与するか?
試行錯誤的なアプローチを支援し, あいまいさを指摘する機構があれば寄与できそう. が, コミュニケーション上の問題には (現状では) 寄与し得ないと思う.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
障害の具体的イメージがわからないためコメントを控えます.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
抽象化する時の発想について多くのヒントが得られた.
2. 欲したが, 得られなかったもの (今後への期待)
OOA から OOD,OOP まで経験することで方法論, ツールの有用性を確認するまで至りたかった.
3. お使いいただいた CASE ツールに対する要望
4. 全体的な感想
最初に方法論の説明があり, 適切なプログラム内容だった. 細いプログラム (わんこそば, パーティー) も楽しめて素晴らしい会でした. プログラム委員, コーディネータの人に感謝します.

ふりがな　いいづか のぶお
氏名　飯塚 宣男

1. プロフィール

年齢:27 歳　性別:男
会社名:カシオ計算機 (株)　部門:楽器第 3 開発部
これまでの主な開発経験
8 ビットマイコンのアセンブラプログラミング
UNIX ワークステーション上でのアルゴリズム開発

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
・ 仕様に沿った, もしくは, 仕様の変更に柔軟に対応できるコーディング (適切なブロック化やデータ構造など) を行なうにあたり, 定式化した方法論がなく, どちらかというとカンと経験に頼っているため, 開発効率やドキュメント作成に問題がある.
・ 複数の人間で共同開発を行なう場合も問題分析の方法について同様
(2) それらの問題点解決に CASE はどう寄与するか?
(CASE 自体を詳しく知りませんが)
・ ドキュメントが定式化されるので, 共有, 保守の効率が上がる.
・ 定式化した方法論によりソフトウェアの質を一定以上に保てる.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
何よりも CASE に導入による種々の利点について啓豪に努める.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ お絵かきツールとしてだけ見ても, 文法を持っているので使いやすい

・ WS 1 人 1 台ないとしょうがない
・ 船頭多くして, 船 陸に上がる (あの程度の問題を 9 人で分割する難しさは CASE どころではない)

2. 欲したが, 得られなかったもの (今後への期待)

ハードのアキテクチャ, ソフト, コストのトレードオフで仕様がポンポン変化するような場合 (コンシューマ製品などの開発現場) プロトタイピングをくり返しながら仕様をまとめて行く段階をサポートするものがあるかなと思ったがなかった.

PWB のアニメーション機能はほんのちょっと近かったが…

3. お使いいただいた CASE ツールに対する要望

細かなバグが多い. 早くつぶして欲しい.

UIM をもっとものすごく充実させて欲しい.

→ 具体的には, ハードのアキテクチャのアニメーション実行によるプロトタイピングしたい (CASE じゃなくなりますけど)

4. 全体的な感想

やはりグループの人数をしぼるべきだった. グループが 3〜4 人ならもっと先に行けた (= もっと CASE が見えた) だろうと思う. これがいちばんの問題だったと思う.

ふりがな おおごもり たかゆき
氏名 大篭 高之

1. プロフィール

年齢:28 歳 性別:男
会社名:日本システム株式会社 部門:基本ソフト開発第 2 部
これまでの主な開発経験

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

・ SE の主観的な判断により, 仕様の完成度が決まる.
・ 特定のリソース (メーカー・HW 等) に縛られる.
・ かこの遺産に引き掟られる.

(2) それらの問題点解決に CASE はどう寄与するか?

・ CASE ツールによりある水準の高い完成度が保証される.
・ 特定のリソースを 1 つのアトリビュートとして扱うことにより, リソースに補われない円滑な設計作業を行なうことが出来る.
・ CASE ツールにより, 過去の遺産の問題点を除去し, 逆に財産として蓄積していくことが出来る.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

・ 高価な HW,SW
→ 生産性の向上にともなう利益の向上により, 初期設定の回収, 黒字への転用の容易性を強調
・ 新たな教育
→ 全技術者が一定水準の方法論を習得出来, それにともない技術力だけではなく, 生産性も向上する事を強調

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)

・ 今まで SA/SD には全く無縁のいい加減な設計方法しか行った事がなかったので, 今回の様に分析に時間をかけじっくり作業を進める方法を体験出来た事は非常に良かった.

2. 欲したが, 得られなかったもの (今後への期待)

・ 自分が行っている実際の仕事への効果をはっきりとした形では得られなかった.(方法論が生かせるかどうか?)

3. お使いいただいた CASE ツールに対する要望

・ CASE ツールに触れた事がなかったが, 作図, 編集が簡単にしかも綺麗に出来るのは感動した.
・ ユーザー I/F に馴じめなかった.(もっとオブジェクト指向にして欲しい)

4. 全体的な感想

・ CASE に触れるという事では成果は大きかった.
・ 人数的, 時間的制約から実際に CASE ツールを使っての設計が最後まで出来なかったのが残念.
・ またこういう機会が有った時は SA/SD をもっと勉強してから参加したい.

以上

ふりがな ふじわら ちひろ
氏名 藤原 千尋

1. プロフィール

年齢:29 歳　　性別:女
会社名:日本ラッド株式会社　　部門:基本ソフトウェア部
これまでの主な開発経験

開発言語	OS	業務
C	UNIX	基本ソフト (コンパイラー)
ASM	OS-Z	(インタプリター)
BASIC	MS-DOS	制御ソフト
PL/I	I-TRON	
FORTRAN	IMS/VS	ビジネスアプリ

2. ポジションステートメント

私自身が日頃問題と感じているのは
1) 開発の初期段階でのユーザニーズ確定の困難さ
打ち合せ, レビューをさんざん繰り返した後に取り交わした要求を平気で変更してくるユーザ, しかも, そのころには, 既にコーディングさえ終りかけていたりする.
2) H/W 開発 この同時進行等で設計時点で不明な問題を抱えたまま, 下流工程へ渡さざるをえない.
等の, ソフト開発を上流下流工程に分けた考え, 例えば"上流工程の成果物が基本仕様であり, 下流工程の成果物はそれを具現化したプログラム自体である"という考え方自身に疑問を感じさせるもの. item3) 上流工程作業者間での誤解や, 作成された書類の表現の違い. 設計書の ふぉーむはもちろん, キーワードの表記法や, その他, 細々とした取り決めを行ないはするが, いつか, 何故か, 自分以外の人の書いた物は読みづらく, 次工程にアサインされた作業者は, 誰が書いた物かによって読み分けている事も.
といった, コミュニケーションの問題の 2 点です. 残念ながら,CASE をさわった事も, 見た事もない為, これらの問題解決に何らかの役にたつかどうかは, わかりませんが, 今回, 見せて (触らせて) いただいて, 考えてみようと思っています. また, 導入に際する障害の件は, 私の様に, ワープロさえ, できれば使いたくない人間もいますので, こういう人に, どうやって使わせるかが, 最大の問題となるでしょう. これをどう克服するかも, 出かけてから皆さんに教えていただくつもりです.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
ツールへの嫌悪感が取り去られたと同時に夢も消えた.
2. 欲したが, 得られなかったもの (今後への期待)
方法論が本当に実務に役にたつのか.(従来の手法より, 金になるのか?)
3. お使いいただいた CASE ツールに対する要望
・ レビューをサポートする機能がほしい.
・ エディターを強化し, わずらわしさを感じさせない様にしてほしい.
4. 全体的な感想
貴重な経験, 情報が得られました. ありがとうございました.

ふりがな　ふじい やすのり
氏名　藤井 康範

1. プロフィール

年齢:24 歳　　性別:男
会社名:MHI エアロスペースシステムズ (株)　　部門:制御システム二グループ
これまでの主な開発経験
対潜戦術訓練シミュレーションソフト
宇宙基地搭載ソフト (開発中)

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
ソフト開発の上流工程において, 要求定義を行なう際に以下に示すような要求仕様を行ないがちである.
(1) 上位仕様の反映し忘れによる不整合, 矛盾した要求
(2) 修正, 改訂に手間がかかり, 調整, 連絡, 周知に不手際が起こる.
(3) 客先/仕様設定者/設計者間の伝達情報不足
(4) 機能単位で分割されるため, ソフト構造とのアンマッチング
上流工程でこれらの誤りが発生し, 後工程まで残った場合, スケジュール, コストのインパクトが大きい. 上記のようなミスを上流工程にていかにして減らすかが重要な課題である.
(2) それらの問題点解決に CASE はどう寄与するか?
(1) 要求仕様の構造化により, 整合性を保つ.

(2) 変更が容易, 変更箇所が明白
(3) 図を用いた分かりやすい表現による要求の過読性を向上させる.
(4) データの流れを中心に分割することにより, ソフト設計への移行がし易い.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
CASE 導入に関しての問題点とその解決案を記述する.
(1) 導入コスト (経理部への説得)
・ 最近のソフト開発の動向 (特に海外) から説明をはじめる.
・ 信頼性向上, 後戻り作業なしによる生産性向上, 可読性の向上を強調する.
(2) 手法の教育 item・新人を起用する.
・ 熟練者一人のもとで小プロジェクトを 2 回経験させる.
(机上での勉強は不要, 実践なるのみ)
item(3)CASE ツールの選定基準まず第一に仕様を行なうことである. その際に以下の項目を主な評価基準とする.
・ コンフィギュレーション管理, 進歩管理, セキュリティ管理などのプロジェクト管理等の支援
・ 変更容易性 (オペレーション履歴, 自動フォーマット)
・ 他ツールとの互換性
・ 機能 (検証の厳密さ, 日本語)
・ カスタマイズ機能
・ マニュアルの整備/技術サポート体制
・ 稼働マシン,OS
・ プリントアウトの美しさ
・ 文書化
・ Ada 言語への設計支援

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ 他の会社での,CASE ツール適用 (使用) 状況.
・ CASE ベンダー, 他会社の方との顔つなぎ.
・ 手法の適用法.
2. 欲したが, 得られなかったもの (今後への期待)
・ 女性が少なかった.
3. お使いいただいた CASE ツールに対する要望
・ 早くマニュアルを整備し, 完成度を高めて欲しい.
4. 全体的な感想
・ 非常に有意義でした.
・ また来たい.
・ みんな燃えている人が多かった.

ふりがな えんどう すすむ
氏名 遠藤 晋

1. プロフィール

年齢:25 歳 性別:男
会社名:株式会社 アスキー 部門:OA システム開発部
これまでの主な開発経験

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
(2) それらの問題点解決に CASE はどう寄与するか?
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
私は,CASE については素人なので 1 から 3 までの設計設問についてはお答えできませんが, 弊社では将来的に CASE Tool を導入しようと計画しております. そのために勉強させていただきたくお願い致します.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ CASE ツールと言うよりはその裏にある理論 (方法論) にふれられた
(理解できたとは思っていないが, 今後の自分にインパクトをあたえるだろう)
2. 欲したが, 得られなかったもの (今後への期待)
・ 大きく分けて 3 つのグループに別れていたと思います.(SA/SD,OOA/OOD,PWB)
それらのすべてを体験したかったと思います.
・ でき上った物に対するディスカションもしてみたかった.

3. お使いいただいた CASE ツールに対する要望
OOAtools ですが, 今回はツールの機能を使いこなせなかったが (もしかしたらツールのすべてを使用したかもしれませんが) まだまだユーザインターフェースが未完成であると思う (Bver なのであたりまえ). また OOD/OOP にうつった場合に OOA で作成したものがはんえいされるのかどうかがわからなかった.(すいません要望になっていません)
4. 全体的な感想
・ 課題をやるのには時間が少なかったと感じています.
・ 2 でも書きましたが成果物に付してのディスカションをしたかった.
・ きたない字ですいません (FEP が欲しい…)

ふりがな いずみた あきよし
氏名 泉田 昭義

1. プロフィール

年齢:29 歳 性別:男
会社名:(株) 岩手電子計算機センター 部門:医療システム部医療大二課
これまでの主な開発経験
日立 VMS/ES ユーティリティ
社会保険庁 オンライン MPP
匠事会計システム バッチ ファイルメンテナンス

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
ソフトウェア開発ではあらゆる標準かが図られており, ドキュメント中心の作業となっているが, 新規開発ではよく参照するドキュメントも数年でその内容の信頼性がなくなっているように思われる. それは, プログラムの完成と同時に発生する仕様の変更に対するドキュメントのメンテナンスが行なわれていないからである.
(2) それらの問題点解決に CASE はどう寄与するか?
ドキュメントのメンテナンスに費やす時間の短縮, 作業の軽減を図り信頼性の高いドキュメントを生み出すことではないだろうか.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
設計段階で利用することの大切さを理解してもらえるように努力すべきではないだろうが.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
日々の仕事の中で 1 日に 1 個新しいことを覚えられれば良い 1 日だと思っていますが, ワークショップに参加していろいろなことを知ることが出来うれしく思っています. 又, 日頃勉強不足についても再認識させられ頑張らなければいけないなという気持をもつことができたことは大きかったと思います.
2. 欲したが, 得られなかったもの (今後への期待)
CASE Bench を選んだからいいというわけではないが, 今までにない Tool(私の知ってる範囲) だったのでビックリしていますが, 大変よいツールだと思います. 実業務でも使ってみたいが, 自分の担当業務 (保守業務が主) でうまく使えるかどうかは自信がない. 新システム作成の期会があれば, 使っていきたいツールである.
3. お使いいただいた CASE ツールに対する要望
良いツールと思いますが, 良い所がめだつとちょっと不便だと思うところがすごい欠点の様にみえてしまいます. 全体的な操作性のバランスをとっていただければ, 大変よいものになると思います.
欠点 ・文字入力がワープロより劣る.
・プリントアウトをハードコピーレベルでやってほしい.
4. 全体的な感想
実際に使用するためには, プロジェクトへの説明, 理解が必要であり, 古い頭 (上司) に対する理解が難しいのではないか. 導入に対して, 会社又は部しょでの環境がツールを使用する前の問題になるように思われる. その結果がどのようなものか, 開発工数や保守工数に値がほしいと思います. 現在の方法とツールを使ったらこうなるとか数字を出せるなら表にしてほしい.

ふりがな いたない ひとし
氏名 似内 均

1. プロフィール

年齢:28 歳 性別:男
会社名:(株) 岩手電子計算機センター 部門:公共システム部
これまでの主な開発経験

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
従来のソフトウェア開発では, ソフトウェアの計画・要求定義・設計・プログラミング・テスト・運用・保守といった一連の手続きが「ソフトウェアのライフサイクル」と考えられ順次開発工程を積重ねる方法がとられている. ところが, この方法では, 各工程の完成度, 品質が大きなウェイトを持っておりユーザーの要求があいまいなままプログラム作成に入ってしうと, 仕様の変更や設計の見直しなど無駄が多く発生しやすい.
(2) それらの問題点解決に CASE はどう寄与するか?
当初のシステム分析や要求定義の段階において, 仕様の変更・追加・削除が容易に行なえる.

3. アンケートの回答
1. ワークショップを通じて得られたもの (成果)
・ CASE ツールを実際に使ってみた事によりどの様なものかを理解する事が出来た.
・ オブジェクト指向について全くわからなかったし, 意識もなかったが, 方法論などがなんとなく理解出来た.
2. 欲したが, 得られなかったもの (今後への期待)
構造化手法に関してのツールを使えなかったために,SA・SD に関する理解度が薄かった.
3. お使いいただいた CASE ツールに対する要望
(PWB)
・ プリントアウトの機能が欲しい.
・ 多モデルの特性を理解する HELP 的機能がないために手順・選択かがわからない.
4. 全体的な感想
オブジェクト指向,CASE ツールに対して, 問題意識 (刺激) を持つ事が出来た.

ふりがな みかみ おさむ
氏名 三上 修

1. プロフィール
年齢:29 歳 性別:男
会社名:(株) 岩手電子計算機センター 部門:公共システム部公共第一課
これまでの主な開発経験
水道業務 ホスト,WS による水道料金管理システム
学校給食 WS による管理システム

2. ポジションステート メント
(1) ソフトウェア開発の上流工程における現状の問題点は?
現在文書や図形を中心とした階層的に手書き又はワープロで作成する手法を取っている. また, ユーザーの要求があいまいのまま, プログラム作成等に入ってしまっているため仕様の変更等に対応するのに時間がかかりすぎている. システム開発工程における手法が統一されていない為, ドキュメントを作成した後のメンテナンスがほとんどされていない.
(2) それらの問題点解決に CASE はどう寄与するか?
ドキュメントの作成や修正変更についやされる時間の短縮, 開発スタッフの人が変わっても, 一貫性のある開発, 拡張が行なえるのでは?
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
よくわかりませんが, ワープロみたいに,1 人に 1 つの WS,PC があれば普及すると思います.

3. アンケートの回答
1. ワークショップを通じて得られたもの (成果)
・ 方法論及び記法についての基本的な考え方
2. 欲したが, 得られなかったもの (今後への期待)
・ ツールの使いこなし.
3. お使いいただいた CASE ツールに対する要望
・ 現時点ではまんぞくしました, が OS が UNIX だけというのはざんねん
4. 全体的な感想
・ たいへん満足しました. 今後チャンスがあったらまた参加したい.
・ ベンダー 5 社が一同に集まって実習できた事がおもしろかった.

ふりがな さとう たくみ
氏名 佐藤 琢美

1. プロフィール
年齢:29 歳 性別:男
会社名:(株) 岩手電子計算機センター 部門:企画・ソフトウェア開発 ソフトウェア第 2 課
noindent これまでの主な開発経験

地銀第3次オンライン開発

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
(2) それらの問題点解決に CASE はどう寄与するか?
<分析, 及び設計作業を繰り返すことが出来ない!!>

現状我々は, ソフトウェアの開発において, 定められた各フェーズを定められた順序にしたがって開発する作業形態を取っている. たいてい現状のソフトウェア開発では, たとえば, システム分析フェーズではシステム分析以外の作業はやってはいけないような"おきて"(?) があり, そのフェーズの作業が完全に終了し,output に関してレビューがなされ, さらに承認されるまで次のフェーズであるシステム設計をはじめてはならないことになる. しかし, 現実には分析フェーズのすべての作業の完了の前に設計フェーズの一部に手を付けはじめるということは度々あるし, 分析作業中には必ず次の設計作業のことが頭の中にあったりする. 又, システム設計フェーズにおいても, やはり分析作業は依然として頭の中で継続しているのである. 現状では, そのような開発の試行手順を dump する思考錯誤可能な開発工程にはもちろんなっていないし, また, 紙とペンで表現するにはあまりにも煩わしく, 冗長な作業になる.

[CASE 環境では...]
要求分析フェーズでは適当なプロトタイピングツールを用いることにより, ユーザ要求の機能の一部, 又はすべてをシミュレートし, 作成しようとするシステムのモデルを基にして分析作業を行なうことが可能となるのではないだろうか. プロトタイピングツールの適用可能なシステムであるかは別の問題となると思うが, そのようにして引き出されるユーザのシステムに対する要求を図式文書かツール (DFD,DD,ERD) と呼ばれている各種ツール上のモデルに関連づけて活用されることにより, ユーザの要求をシステムに的確に反映するための手助けになってくれるものだと思う.

<上流工程作業は現状, あまり楽しい仕事ではない >

分析,output として次のようなものが通常考えられる.

システム要求定義書
システム仕様書
機能仕様書 etc.

これらの output を生み出す作業は, システム全体の生死を左右する工程であるのは誰もが認識しているし, また, 十分な期間がこの工程に割り当てられるのが通常である. しかし依然, 上流工程における何らかの問題点に起因した下流工程での失敗があとをたたないという状況もある. 幸い私は未だ, そのような重大な状況におかれたことはないのだが, 開発工程が下流に向かうほどその被害が大きいことは明らかである. 上流工程においてはかなりの時間と労力がペーパーワークに費やされているのが現状と思われるし, 私も含め多くの人間はそのような作業が退屈に感じていると思う. 全てがそのような状況から産まれる output ではないにしても, 仮にそのような output の情報からブレークダウンされる下流工程の作業は, 当然悲惨な状況が想像できる. やはり, かなりの時間と労力は, もっと知的であるべきだし, 本来の設計, 分析の業務に費やされて当然ではないだろうか.

[CASE 環境では...]
この問題において CASE を導入することによるメリットは, ペーパーワークが Computer 上での作業に支援されるということだけで十分ありがたいと私は考える. 実際はどうなのかというと私自身,CASE を適用したシステムの開発経験がゼロなので判断は出来ないが, 少なくともかなりの時間を本来の業務に費やすことが出来るのではないかという期待が持てる.

<ユーザに対して「仕様の凍結」を宣言しなければならない >

前述した問題点にそれぞれ関連するのだが, 現状の開発サイクル (シーケンス) では, 分析. 設計フェーズ (上流工程) と開発フェーズ (下流工程) との間にフィードバックの流れは原則としてない. そのため, しばしば「仕様の凍結」を宣言しなくてはいけない状況になることがある.「仕様の凍結は」は現状の問題点であるかどうかは判断しかねるが, あるていど規模の大きい開発になるとユーザ, もしくは開発チーム内からの仕様変更はプロジェクトの生死にか変わりかねない. しかし, 仕様変更に関して分析. 設計フェーズへのフィードバックを試みることはユーザにとっても, 又, より良いシステムを作りたいと思っている開発者にとっても有効であると思う. 現状は情報のフィードバックではなく, 分析. 設計のやり直しとなりそれにより発生するリスクから「仕様の凍結」を宣言することになる. このような開発は, ユーザのシステムへの要求に対する満足感を考えると, やはり必ずしも良い方法とは思えない.

[CASE 環境では...]
ソフトウェア開発のライフサイクルの中で,「仕様の凍結」は CASE の導入により無くなるわけでは決してないと思が, 各フェーズのフィードバックグループを快適に支援してくれるツールがあれば, 仕様変更に対する分析フェーズへのフィードバックはあたりまえにように行なわれるであろうと思う.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

これから導入を検討しようとい場合において, 決定権のある先進的な上司の理解が障害克服の第一歩であると考える.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)

同じソフトウェアという仕事をしているさまざまな人達との出合いによって, いろいろな話が聞けた.(いろいろな考え方があった) → 目ざすところは 1 つか?

2. **欲したが, 得られなかったもの (今後への期待)**
抽象化手法 (OOA) の手がかり. →トレーニングナビゲータツールの様なものがほしい.

3. **お使いいただいた CASE ツールに対する要望**
(Smalltalk-80 & G.G,OOATools) 十分使ったというよりも使う以前の問題だった……OOATools しか使っていない.(オブジェクト指向がまだまだわかっていない)

4. **全体的な感想**
全体的に楽しかった. しかし, グループ内で議論する時間がもっとほしかったと思う. (マシンから少しはなれて, 今自分達は何のために何をしようとしているのかをグループ内で再確認する時間)

ふりがな　まつもと こうぞう
氏名　松本 幸三

1. **プロフィール**

年齢:29 歳　　性別:男
会社名:(株) CSK　　部門:西日本産業システム第 2
これまでの主な開発経験
エキスパートシェルのツール開発
エキスパートシェルを利用したアプリケーションシステムの開発

2. ポジションステートメント

(1) **ソフトウェア開発の上流工程における現状の問題点は?**
標準化が遅れているため, 個人の技術にたよっている. このため, バラツキが大きい開発となる.

(2) **それらの問題点解決に CASE はどう寄与するか?**
個人差が少なくなる, 内容にモレが少なくなる.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**
上司への説得, 導入に対する熱意を失わない.

3. アンケートの回答

1. **ワークショップを通じて得られたもの (成果)**
オブジェクト指向が部品化, 再利用化に適しているとわかったこと. また, プロトタイピング手法を通じて人の思考を助けることがわかったこと.(インストラクターの青木のデモでわかった)

2. **欲したが, 得られなかったもの (今後への期待)**
1 の成果を自分の手で確認していきたいと考えています. また可能ならば, 実務上で使用していきたいと考えています.

3. **お使いいただいた CASE ツールに対する要望**
Smalltalk 上の CASE ツールの完成品 (FXIS から 12 月に完成予定) に触れてみたい.

4. **全体的な感想**
SA/SD,OOA/OOD など有用な考え方が提唱されているが, 自分自身に問題意識がなければ, 現状のまま推移していくだけだと考えます. このような機会を与えていただいて, 問題意識が深まり, ありがとうございました.

ふりがな　あおやなぎ かずひさ
氏名　青柳　和久

1. **プロフィール**

年齢:30 歳　　性別:男
会社名:(株) 東北 SRA　　部門:システム開発部
ここまでの主な開発経験
- 大学教務事務システム
- NTT 小局向交換機
- 銀行事務システム

2. ポジション・ステートメント

(1) **ソフトウェア開発の上流工程における現状の問題点は?**
上流工程 (調査分析, システム設計, 要求分析, システム分析などと呼ばれているもの) 自体が重視されずに, 下流工程に集中するし, またそうせざるおえない現実のスケジュール!

(2) **それらの問題点解決に CASE はどう寄与するか?**
ツールをうまく使いこなす事により上流工程の適切なスケジュール化が可能だと思います.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**
障害にもよるが, 金銭的なものはあきらめるしかない様な.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
SA/SD 手法以外での CASE ツールにふれたかったので, 目標は達せられました.
2. 欲したが, 得られなかったもの (今後への期待)
他のツールにふれる (概要説明はあったが…) 機会がなかった. とはいえ CASE ツールの発表会ではないのですからね.
3. お使いいただいた CASE ツールに対する要望
PWB にかぎりませんが,2 日間 (実質) では, 細かいエディタ操作等しか要望できることがない.
4. 全体的な感想
課題に対する成果はともかくとして (まとまる, まとまらない以外に…), この 4 日間で CASE を導入して実作業に生かせる様な気がしてきました.

ふりがな しぼうだ かずのり
氏名 思坊田 和典

1. プロフィール

年齢:29 歳 性別:男
会社名:日本電気ソフトウェア株式会社 部門:生産技術部
これまでの主な開発経験
・ ウォークスルー支援ツール
・ 構造化エディタ

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
ソフトウェア開発での上流工程では, 要求システムの機能を決定し, それを設計書にまとめることが作業の中心になる. そこで機能を決めるということは, 構築するシステムに必要なさまざまな要件を定義していくことを意味する. そして, それらの要件はお互いのに何らかの関係を持っている. これらの作業は現時点では文章によって表現されているため, 多分に曖昧さを含む結果となっている. また, システムへの要求は通常自然言語で行なわれており, それを図式かなど機械的な表現に置き換えるのも困難という状況がある. 第二は, 設計書が工程ないで閉じてしまうということである. ソフトウェア開発は, 情報の交換の繰り返しと言われている. そのため, 各工程ではレビュー等見直し作業をして, 作業の洩れや抜け, 誤りを防ごうとする. これらのミスは情報の変更過程に人間が介在するために発生するものが多い. 現在上流工程での設計者は文書で作成している. そのため, 次工程に情報を持ち込むには, 人間を介さざる追えなくなっていて, 直接次工程に持ち込むことが出来ない.
(2) それらの問題点解決に CASE はどう寄与するか?
機能がどのような要件で構成されているかを調べ, それを論理式で表現することで曖昧さを排除する. しかし, 自然言語での思考をすべて, 表現できるかどうかは疑問が残る. (補演習で, 実際にどう役立つのかを試してみたい)
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
CASE 導入での障害は, 戦場にいかにして CASE を定義させるかということである.CASE を導入すれば, 当然仕事のやりかたも変わるのであるから, それに対する反発が起こることは予想されることである. また, 強制的に使わせよとしても, 従来のやりかたにもどることも考えられる.CASE を導入するにあたって, それが本当に役立ち, 高価のあるものであることを立証し, 納得させることが重要である.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ CASE ツールへの興味をもつことができた
・ 構造化設計, データ中心の設計の意義が見えてきた
・ 他社の方の話をきけ, 自社の立場, 自分のレベルが認識できた
2. 欲したが, 得られなかったもの (今後への期待)
・ リバースエンジニアリング的アプローチ
3. お使いいただいた CASE ツールに対する要望
・ DFD 内に記述した名前 (データ名, プロセス名) を, キーボードからの入力フィールドにもってこれること
・ DFD 修正による,DD の自動修正機能
4. 全体的な感想
・ 参加してよかった
・ 方法論を充分理解しないまま参加したのはよくなかった
・ 方法論を理解してから,CASE ツールに向かうということをしなければ CASE ツールを充分に使いこなせない

ふりがな　まつうら みつや
氏名　松浦 光哉

1. プロフィール

年齢:27 歳　性別:男
会社名:中央システム (株)　部門:関西システム課
ここまでの主な開発経験

2. ポジション・ステートメント

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ SA/SD の理解
今後仕事を進める上でなんらかの手法 (法則) がえられたと思う
・ CASE ツールの使用方法
2. 欲したが，得られなかったもの (今後への期待)
リバースエンジニアリングへの使用
3. お使いいただいた CASE ツールに対する要望
リバースエンジニアリング
COBOL,FORTRAN 等から構造図が出来る様なツール
4. 全体的な感想
他の CASE ツールとの比較がもっと出来ればよかったと思います. 全体的に見れば有意義な時間がすごせたと思います.

ふりがな　ほんま ただし
氏名　本間 匡

1. プロフィール

年齢:27 歳　性別:男
会社名:中央システム (株)　部門:東京システム 5 課
ここまでの主な開発経験

2. ポジション・ステートメント

3. アンケートの回答

前雄介

1. ワークショップを通じて得られたもの (成果)
・ 方法論が多少なりとも理解出来た.
(CASE についての知識がまったくなかった)
2. 欲したが，得られなかったもの (今後への期待)
・ OOA Tools, グラファギアのツールがもう少し使いたかった.
(メニューが多くあったが, 使ったのはその中のいくつかであった)
3. お使いいただいた CASE ツールに対する要望
・ is-a,has-a の関係を示す線を引くとき, どうしても見づらくなってしまうところを何とかしてほしい.
・ あったのかも知れないが, 画面の縮小機能がほしいと感じた.
4. 全体的な感想
・ 課題をツールを用いて実際に分析してみましたが, その分析の一例を教えてほしかった.(比較してみたかった)

ふりがな　やましま としひこ
氏名　山島 利彦

1. プロフィール

年齢:28 歳　性別:男
会社名:日本エム・アイ・シー株式会社　部門:開発三部 第一課
これまでの主な開発経験
・ 都市銀行 第三次オンライン 「ホスト間インターフェース」
・ 都市銀行 Global Banking System [相場情報システム」

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
上流工程における最大の問題点は, ユーザーニーズが設計・仕様定義に正確に反映されていない事である. 現在システム仕様に関するユーザーとの意志の疎通は, 文書を中心としたドキュメントによって行なわれている.

個々のドキュメントは, それぞれSEによる独自の分析・設計手法により, 独自的な記述で作成されることが多く, 統一がとれていない. このため, ユーザーに仕様確認を行なっても理解されぬまま承認され, テスト・フェーズに (最悪の場合は, 導入フェーズ) になってから, 仕様の不備が表面化している. これらをふまえて, 下流工程までを含めた開発・保守作業の増化, 品質の低下にいかに対応していくかが, 今後の課題である.

(2) **それらの問題点解決にCASEはどう寄与するか?**

CASEの導入により, ユーザーの仕様確定作業への積極的参加が可能になることが予想される.

(理由)

(1) 豊富なグラフィック機能によりい, ドキュメント等を視覚的に作成することが可能となり, 確認が容易になる.

(2) 個々のツールが採用している分析・設計手法にしたがい上流工程をすすめることにとり, 統一された分析・設計が可能となる.

また, 現在の開発体制の中では採用には無理があるプロトタイプ技法の採用が現実的となり, 早期段階でのユーザーのニーズの検証が可能となる. 以上の理由により,CASE導入は (1) の問題点解決に十分効果を表すであろう.

(3)**CASE導入に際してさまざまな障害をどう克服すべきか?**

CASE導入にさいしての問題点として, 導入効果 (生産性・品質の向上) が現れるまでの時間がかかることが挙げられる. そこで, 導入効果を出来るだけ早く結果として現わすためには, 以下の対応が必要であると思われる.

(1) 現状の開発体制 (ライフサイクル, ドキュメント対系を含む) の見直しを考慮した導入計画の確立

(2) 現状の開発方法に満足している技術者の意識改革および啓蒙活動 (特に現場のマネージャーや年配技術者に対して)

(3) ツールで採用している分析・設計手法等に対する教育の徹底

なお, トップマネージャーには, 導入効果が現れるまで時間がかかることを認識してもらった上で, 導入する必要がある.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)

・ CASEツールの実体験 (お絵書き機能の現状把握)

・ SAに従った分析の実体験 (プロセス中心→データ中心)

2. 欲したが, 得られなかったもの (今後への期待)

・ SAからSDへの流れ

・ リバースエンジニアリングの現状

・ ワークステーション以外の環境への普及状況

3. お使いいただいたCASEツールに対する要望

・ 最適な形へのリフォーマッティング機能

・ OSを意識しない操作性

4. 全体的な感想

方法論, ツール, 記法の習得以上に社外技術者との意見交換ができたことが非常に有益な経験になった. 今後も今回の様な企画に積極的に参加することを心掛けたい.

ふりがな はま やすし
氏名 浜 恭士

1. プロフィール

年齢:24歳 性別:男
会社名:(株)SRA 部門:関西支社開発第1部
これまでの主な開発経験

・ プロッタープリンターの制御システム

・ 関西電力向けの, 事故復旧自動開閉器制御システム

2. ポジションステートメント

(1) **ソフトウェア開発の上流工程における現状の問題点は?**

私は, 経験が3年あまりですが, この1年弱ほど揉れているプロジェクトのお手伝いをしていました. プロジェクトへの参加が途中からということもあって仕事は, 設計等全くできずに, プログラミング以降でした. また. 納期直前ともなると, 時間がない為に設計なしで, 機能仕様書を眺めながら, プログラミングすることが精いっぱいでした. 上記のように, 設計もろくにしないために, ドキュメントが残らなかったり, テストを終らせてからドキュメントを作成するようなことが行なわれ, プログラムの, 引き継ぎをするにしてももれが生じてしまう. また, 最近のプロジェクトは, 納期までの期間が短すぎる. このような事が, 私の身の回りでは問題だと思われる.

(2) **それらの問題点解決にCASEはどう寄与するか?**

私は,CASEに対する知識がないため, なんとも言えない.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
同上

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
ツールの使い方というよりも, 構造化分析の手法が, かんぺきでないけど得られたと思う.
2. 欲したが, 得られなかったもの (今後への期待)
実際にツールを使っていき, シュミレーションを行いたい.
3. お使いいただいた CASE ツールに対する要望
(StP)
・ まだ, ごく一部の機能しか使っていないので, なんとも言えない.
4. 全体的な感想
・ 構造化分析の手法というのが, どの様なものなのかが, 完全でないけど, 認識できたことが, 今後の仕事のためになるだろうと思います.
・ ただ,(・手法の学ぶ時間が短いかなと思う.)
(・CASE ツールの使用時間も, もっと長い間, 使いたかった.)

ふりがな なかたに のぶこ
氏名 中谷 信子

1. プロフィール

年齢:25 歳 性別:女
会社名:(株)SRA 部門:関西支社開発第 2 部
これまでの主な開発経験
・ X.25 通信システム (パッケージ改造)
・ トークンバスのモニタシステム
・ X-Window(Sony Widget) を使ったユーザインターフェース

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
まず, 上流工程に参加できる適当な人材の不足が挙げられると思う. 適当な人材とは, 実際にプロジェクトのメンバとして設計を行なっていける能力を持つ者, および, メンバを統括してプロジェクト全体の把握を行なう管理者である. ソフトウェア業界が人材不足であることを考えても, 事態はかなり深刻である. 機能分担等を行ない, 同時進行でいくつかの作業が可能であるように管理者が考えたとしても, その作業を任せていける人材がいない. その結果, 過重な作業を任せる者が発生し, ストレスが溜る. 作業がうまく進まない. 作業のフォローに管理者が手伝わざるを得なくなる. といった事態にいたる. 下位レベルの者の育成の面から考えても, 過度な要求は, 本人にもプロジェクトの為にもならない. それにもか変わらず, そのリスクをあえて侵さなければならないところに問題があると思う. また, 開発期間の短さと, それに伴うその場当たり的な作業の進め方が挙げられると思う. 最近のプロジェクトは, 開発聞かんが短くなっているように思われる. そのため, コーディングすることが優先され, 本来なら, 先に作成しておくべきであるドキュメントが, コーディング終了後につじつまをあわせて書く, または, テストが終了してから書く等がしばしば行なわれる. これは, 能率的にドキュメントを作成する手段が普及していないせいもあると思われる. しかし, 各自の担当している作業の詳細が本人の頭の中にしかない, という状況は, コミュニケーションの時間の増大, 伝達の間違え等が発生しやすい状態を引き起こしている,
(2) それらの問題点解決に CASE はどう寄与するか?
現在までに CASE ツールを使ったことがないので, また CASE についての知識が不足しているので, なんともいえない. 今後考えていきたいと思う.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
同上.

3. アンケートの回答

林 明賢

1. ワークショップを通じて得られたもの (成果)
SA/SD の方法論を勉強した. しかし完全理解までまだ時間かかる.
2. 欲したが, 得られなかったもの (今後への期待)
StP の使い方を勉強したかった (社内では StP しかないので). 今回 1 グループ 9 人の制限で勉強ができなっかった.
3. お使いいただいた CASE ツールに対する要望
設計完了後, シュミレーションができる.(特に制御系のタイミング問題.)
4. 全体的な感想
・ 研修の時間が短すぎる.
・ 構造か設計は従来通りの設計時間がかかる. しかしテスト時間は短縮できると思われる.

ふりがな　うえだ けんいち
氏名　上田 賢一

1. プロフィール

年齢:28 歳　　性別:男
会社名:(株)SRA　　部門:特別開発第 3 部
これまでの主な開発経験
　証券アプリケーション

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
　ユーザーの業務知識に対するわれわれ開発担当者の知識不足による設計もれや計算ミスが多い. ユーザーの要件が把握できにくい. 設計技法の認識不足.
(2) それらの問題点解決に CASE はどう寄与するか?
　ユーザーに提出された形式での設計なので問題点解決に CASE Tool がどう寄与するのかははっきりわからない.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
　現在開発マシン等ユーザー先での仕事なのでこれを全てクリアできなければならない. 例えば,tool 導入のうえで tool の使用マシンが不適当ということも考えられるのではないだろうか? CASE tool を有意義に我々にも使用できるかどうかというてんからも参加してみたい.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ SA/SD によるデータ中心による設計手法とそれをもとにしたツールの存在.
・ CASE ツールなるものを使ってみたいと思う気持ち.
・ 勉強不足?
・ 現在使用している設計手法に対して少し疑問を持つことが出来た.
2. 欲したが, 得られなかったもの (今後への期待)
・ 手法とツールをもう少し理解したかった.
3. お使いいただいた CASE ツールに対する要望
・ 事務処理に対するツールの有用性を高くして欲しい.
・ もう少しマウスの動作だけで作業できると…と思う.(今回は特別な状況であったかも?)
4. 全体的な感想
・ CASE ツールに関する知識が全く白紙の状態で臨んだワークショップであったが, 今後の CASE ツールに対する考えが, もう少し勉強していこうと思える段階に自分の意識としてなっていることに参加した意義が見い出せたような気がする.
・ CASE ツールは要求分析や設計という上位行程のサポートということで, プロジェクトの対象は新規開発のプロジェクトだと思うのですが, 現状システムの保守作業を主としている私にとっては今の仕事には使える気がしないし, 事務系システムに対する CASE ツールの有用性も少し疑問を持っています.
・ 今後勉強していくに従い, 様々な問題が出てくると思いますが, 事務系システムに対して CASE ツールを有効につかえるように努力してみたい (環境面の問題もあると思うが) と思います.

ふりがな　たてやま じゅん
氏名　館山 純

1. プロフィール

年齢:27 歳　　性別:男
会社名:(株)SRA　　部門:特別開発第 1 部
これまでの主な開発経験
　COBOL による事務アプリ系業務の設計～保守

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
　現在, 私が担当しているシステムは某電力会社の工事管理業務であり,COBOL で約 10 万弱といった規模のものである. スケジュール的には非常に恵まれており, 設計から開発, テストまで余裕をもって行なえたのではあるが, リリースを間近に控えたここに来て色々と問題が発生している. そう, 仕様の洩れと仕様の変更である. つまり, 現在のソフトウェア開発 (方法, 体系) における最大の問題がこの点なのである. 基本設計の段階では, 当然全ての仕様を把握し, 確定させることは不可能である. また, 事務アプリ系の業務の場合, ユーザ・インターフェースが非常に重要なものであるため, 設計ドキュメント上だけでは実際の使い勝手がわからないといった点がこれらの問題の大きな要因となっていることは, 間違いないであろう.(実際, 仕様変更の理由としての大半を占めるのが,' 使いにくい',' わかりにくい' という理由である) こういった点が, 開発・テストの段階での無駄な (余分な) 作業を発生させているのである.

(2) それらの問題点解決に CASE はどう寄与するか?

本来,CASE の設計思想が現在の事務アプリ系業務にあまり適合しないというような感じはあ拭い切れないのであるが, 以下のような点では CASE の導入による効果が期待できると思われる.1 章で述べた問題のうち仕様の洩れという点に関しては, 多分に業務の内容, 及び知識に関係するものであり, 時には当のユーザでさえ知らないような業務が存在するケースさえある. こういった点を CASE ツールを用いることによって直接的に解決することは困難であると思われるが, 間接的にでも何らかの支援を期待できるのではないだろうか. また,2 番目の問題に関しては設計フェーズにおいてプロトタイプ的に仕様を決定し, それに対して検討を重ねていくということが容易に実現できればその効果は絶大なものとなろう.(この点が個人的にも CASE に対しても最も期待している部分である)

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

CASE の導入に当たっての障害として考えられることは多々あるが, それらを克服する最良の手段は, やはり実績の提示であろう. つまり, 投資以上の効果 (物理的, 非物理敵を問わず) が得られることを示すのが最も有効であることは間違いないであろう.

(4) さいごに

これまで. 色々なところで CASE ツールの効果ということを耳にしてきたが, 実際に自分で使用したこともなく, また, 事務アプリ系の業務でどのような効果があげられているかという点に関しての評価に触れる機会もなかったため, このような点から CASE ツールの実態に触れてみたい.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ CASE ツールによって何かが出来る (変わる?) という実感.
・ 恐らく最初で最後のコントロール系の分析が経験できた事.
2. 欲したが, 得られなかったもの (今後への期待)
・ 1 の答のうち' 何が出来るようになるか' という部分に対する具体的解答.
・ 方法論に対しての明らかに有効であるという確信.
3. お使いいただいた CASE ツールに対する要望
・ ある手法 (手順) にしばられないという姿整は維持してほしい.
・ 細かい部分に関しては, まだ発展途上のツールだと感じた.
4. 全体的な感想

今回のワークショップの企画自体はすばらしいものであったと思います. ただし, 主催者の意図したものになるためには, もう少し時間が必要なのでは?どうしても, 観点が「ツールの操作を習得する」という方向に向きがちだったように思われます.

ふりがな さかもと よしみい
氏名 坂本 善通

1. プロフィール

年齢:27 歳 性別:男
会社名:SRA 部門:特別開発第 1 部
これまでの主な開発経験

東京電力ホスト端末間 FD 転送システム	開発言語	COBOL
東京電力送電工事管理システム	開発言語	COBOL
電次連パソコン通信システム	開発言語	C

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

一般論ですが次の 3 点に収束されると思います.
1. 適用業務に対する業務知識の収得
2. 対客先, 及び対プログラム作成者へのプレゼンテーション技術
3. ドキュメントの整備

(2) それらの問題点解決に CASE はどう寄与するか?

この問いに回答する前に自分なりの CASE 環境の考え方を述べておきたいと思います. 私は,CASE 環境とは次のようなもの (あるべきもの) だと考えています.
1. 実行可能なプロトタイプ作成
2. 入出力仕様にともなってデータ構造の定義
3. 上記モデル作成にともなうドキュメントの自動作成

さて (1) 解決にあったての CASE の寄与するであろう点ですが,
(1)–1 についてはあまりというか (全然) 役に立ちそうにはありません. これだけではアプリケーション開発にあったての永遠の課題でしょう.

(1)–2 についてはかなりに点で寄与してくれると思います. 実際の開発に役立ち, その目的とするものをユーザに確認してもらうことによりユーザとシステム設計者間のずれを極力おさえる事ができると思います. 設計者とプログラム開発者との間も同様です.

(1)–3 についてもかなりの点で寄与してくれるものと思います. 複数の設計者がいる場合でも, 自動生成することにより, 設計者間の技量の相違に関わらず, 統一したドキュメント作成できます. また従来後回しに成りがちだった, ドキュメントの作成が開発と平行して, または開発に先んじて作成することが可能になります. さらに, 加えて一度作ったデータの再利用が可能であれば類似システムの開発が極めて容易になるでしょう.

(3)CASE **導入に際してさまざまな障害をどう克服すべきか**?

導入にあったては次の点が問題となると思われます.

1. 機器の導入
   CASE 環境を使用するに当たっては,CASE 本対及びそれが作動する環境 (ハード) が, 必要になりますがそれについては上司にお願いするしか仕様がないでしょう.
2. CASE 利用技術・知識の収得
   その初期利用に当たってはやはりかなりの知識が必要と思われますが, それについてもい証券名勉強するしか仕様がないでしょう. 他だその環境を大勢の人によって使ってもらうために, その適用を限定してより簡単なインターフェースを構築することも一つの手かも知れません.

3. **アンケートの回答**

1. **ワークショップを通じて得られたもの (成果)**
   構造化"分析"を実際に体験できたこと. 分析の延長上に設計があるのではなく, 分析の次の step に設計があるということを初めて感じることができた.
2. **欲したが, 得られなかったもの (今後への期待)**
   オブジェクト指向のほうもやってみたかった.
3. **お使いいただいた CASE ツールに対する要望**
   設計もサポートして欲しい.
4. **全体的な感想**
   同じ問題を両方 (構造化, オブジェクト指向) の手法でやってみたかった. 期間が短い.

ふりがな おくむら よしひこ
氏名 奥村 吉彦

1. **プロフィール**

年齢:29 歳 性別:男
会社名:株式会社 CSK 部門:教育事業部
これまでの主な開発経験
販売管理, 教育コース管理等の要求定義, 外部設計, 内部設計, プログラム開発

2. **ポジションステートメント**

(1) **ソフトウェア開発の上流工程における現状の問題点は**?

<<全体を通して>>

問題点 1: 各社プログラム開発局面でのプログラムに関する標準かは進んでいるが, 計画, 要件定義, 外部設計, 内部設計局面における開発手法や各局面でアウトプッとされるドキュメントの種類やフォーマットが統一されていないことが非常に多い.

問題点 2: 現在, プログラムの再利用, プログラムの共通かはかなり進んできてはいるが, 上流工程でのアウトプッとである成果物の再利用, 共有はほとんどされておらず, すべて新規作成というのが現状である.

問題点 3: 作業スケジュールの作成, 要員計画の立案, また, 作業進捗の管理において, 現状では, 過去に同一規模のシステム開発に従事したことのある人間がいるか, または, 正確なドキュメントが残っていない限りかなり困難である. 特に, システム規模の見積り, 作業スケジュールの作成は, 蓄積された経験により大きく左右されている. また, 作業の進捗管理も, 開発する規模が大きくなればなるほど難しい.

問題点 4: ドキュメントにかかる作業負荷が高く, 特にシステムの仕様変更によって発生する, ドキュメントの再作成はかなりシステム開発の作業の効率を悪くする大きな要因となっている.

問題点 5: 下工程であるプログラム開発では, プログラムのテストにより, 品質のテストが可能であるが, 上流工程では, システムの仕様の正確性や整合性を測る事がかなり難しい. ややもすると, 下工程に移ってから誤りが見つかり, 作業が後戻りするという事態をまねき開発計画をおくらせる大きな原因となることが多い.

<<設計, 要求定義局面>>

問題点 6: システム開発の対象業務担当者と, システム部門の人間, 下工程の作業担当が使う, システム分析におけるツール, ドキュメントが統一されていない場合情報が正確に伝達されにくくなり, 開発効率が悪化する.
(注) この局面では, 主な作業担当者は, 実践に業務を担当している人間となる. よって, 原稿システム

分析システム要件の発掘, システム要件の定義における作業の進め方, 及び成果物の種類, フォーマット及び成果物 (ドキュメント) の作成法等は全社で統一されいないと, 作業に混乱をまねくことになる.

<<外部設計局面>>

問題点 7: ユーザインターフェースで重要となる画面のデザインや記帳のデザインにおいて, ユーザのニーズを正確に把握しにくい. 特に, 画面については, 画面間の流れ, 操作性など紙に書かれた画面フォーマットや画面フローだけでは現実性に欠け, ユーザーにイメージをつかんでもらうのが非常に困難である.

(2) **それらの問題点解決に CASE はどう寄与するか?**

item1) 問題点 1 について CASE ツールによって直接解決されるものではないが,CASE ツールをシステム開発の全ライフサイクルに適用することにより, そのベースとなる開発手法が確立されることになる. 正確には, システム開発のライフサイクルの中に CASE ツールを組み込むためには, 事前にシステム開発の手法をしっかりと確立させておき, どの局面のどのような作業に CASE ツールを組み込むかを考える必要がある.

2) 問題点 2 について各局面で作成されたドキュメントがレポジトリーと呼ばれるデータベースの中に蓄積され, 再利用が可能となり, ソフトウェアの開発, 改良を容易に行なうことができる.

3) 問題点 3 について過去のシステム開発の各工程に費やした人月をレポジトリーに蓄積しておき, ある係数を変更することで自動的に作業スケジュールを作成することができる. また, 作業の進捗管理についても, 現状の作業進捗状況を入力する事により, 計画との誤差等を把握することが可能となる.

4) 問題点 4 について構造化ダイヤグラム等を描き, 図によるシステムの仕様書を作成することができる. また, ドキュメントの変更については, すでに作成され保管されているドキュメントを端末装置の画面上に呼び出すことにより変更し, 再保管しておくことができる. よって, 手作業ですべて新規に紙に書き直すという事はせずにすむため, 開発効率が向上する.

5) 問題点 5 についてレポジトリーに蓄積された構造化ダイヤグラム等のドキュメント類を表す. システム仕様の整合性, 完全性, 正確性を CASE ツールにより自動的にチェック, 分析する事ができるため, 下位工程に移ってからのシステム仕様の誤りが発見され作業が後戻りすることが少なくなる.

6) 問題点 6 いついてシステム分析, 新システムの仕様を,CASE ツールを使い構造化ダイヤグラムにより図を用い表現する事を全社で統一しておく事により, 担当者間の情報の伝達がスムースに行なわれる. また, ダイヤグラムの作成をワークステーション上で各担当者が行ない, レポジトリーに保管しておく事により, 他の担当者がその内容を照会したり, 再利用したりすることが可能となる.

7) 問題点 7 についてユーザインターフェースとなる画面, 帳票を CASE ツールのプロトタイピング機能を用い, ユーザに提示することにより, ユーザニーズを速く, 明確にする事ができる. またシミュレーション機能を用いる事により, システムが実際の環境でどのように動くのかを, 事前に知ることができ, コーディングに入る前に設計上の誤りを発見することが可能とる.

(3)CASE **導入に際してさまざまな障害をどう克服すべきか?**

現在, 我者では, アンダーセンコンサルタントの METHOD/1 とともに,CASE ツールである DESIGN/1 を導入し, システム開発のライフサイクルの中に組み込んでいる. 我社が DESIGN/1 だけでなく,METHOD/1 等の関連製品を導入するにさいして問題となった点, 及び考慮した点について以下に記述する.

1) 決定, 導入にさいしての障害となった事項
   (1) ソフトウェア開発者, システム開発にたずさわる業務担当者は,CASE を積極的に受け入れるか, 無関心か, もし, 無関心もしくは, 拒絶的であれば, どのように対応すれば良いか.
   (2) 工学のよさんと導入までかなりの時間が予想されるため経営管理者に対し, どのようにアプローチするか.
   (3) 導入決定後, 導入, 普及担当部門, 担当責任者をだれにするか. また,CASE についての管理方法, 担当者をだれにするか.
   (4) CASE 導入するにあたって, 現在のシステム開発の体系をどう変えていけばよいか, また, どの部分に CASE を適用させるのか.
   (5) CASE を導入し, 普及させるためには, どのような教育訓練と, だれを教育担当者にするか.

2) 決定導入にさいしての障害に対する対応
   (1)

前ページ 1) の (1),(2) に関しての対応として我社がとった方法としては,CASE を導入するのに積極的である人間 (私もそこに含まれていた) を中心として, 仮想のプロジェクトを作り,CASE ツールを使ったシステム開発を行ない, その成果を, 時間と人数により, 従来のシステム開発と比較した数字をレポートする方法と, 説明会を各プロジェクトの中心となる人に対し開催し, 実際に使ってもらう方法を取る事により, 対応した.

その後, この仮想のプロジェクトのメンバーが各プロジェクト (社内の主要な) に配置されたり, 教育訓練のメンバーとして, 社内での普及にたずさわる事になった. 私は, 後者の教育訓練を担当することになった.

   (2) 前ページ 1) の (3) に関して上述の内容にも関連するが,CASE について関心の高いメンバーを社内より選び出し, プロジェクトを作った. また, 別に SiSwo 担当する部門が同時期に作られたため, その部門が実質導入後の主な管理担当部門となった.
   (3) 前ページ 1) の (4) に関してシステムの開発体系は, すでに社内にあったが, 別に METHOD/1 という開発体系を導入したため, これをファンデーションとして, その局面の作業に CASE ツールをあてはめていく事になった.
   (4) 前ページ 1) の (5) に関して社内教育体系の中に METHOD/1 等の教育を導入し, 関連プロジェクトの人間を定期的に教育している. 教育担当者は, アンダーセンコンサルタントの社員と CSK 内の教育担当者

が担当している.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ 制御系の分析, 設計, 手法 … 従来の業務が事務処理系の分析, 設計が中心であったため.
・ CASE ツールの有用性と欠点
・ CASE ツール, 方法論に対する初心者の反応
・ ワークショップの運営計画
・ オブジェクト指向とは何か?ということの理解の糸口
2. 欲したが, 得られなかったもの (今後への期待)
事務処理系のための CASE ツールが市場に出ているが, それが今回のワークショップで出典されていないのではないか?今回の CASE ツールは主として, 制御系をターゲットにしていると思われる. ライフサイクル全体をサポートしている CASE ツールについても内容検討してみたかった.
3. お使いいただいた CASE ツールに対する要望
(PWB) 今後の改良を期待する.PWB の開発の方向性は良いと思う.
4. 全体的な感想
今後の勉強のきっかけを得た.(自分自身に不足しているものがいろいろと clear になった) 今後もぜひこのようなワークショップに参加したい.

ふりがな ダバン マ チュウ
氏名 Dabin Mattheieu

1. プロフィール

年齢:29 歳 性別:男
会社名:SHIZUOKA UNIVERSITY 部門:研究員
これまでの主な開発経験
* Study , specifications , development , installation and tests of graphic visualisation software in real and differed time of the flight parameters of the AIRBUS A320 (Plane) , for the testflight department of AEROSPATIALE . This project lasted more than one year and consisted of ~ 12 persons.
* During 6 months production of numerical analysis tools in CAD context for a civil engineering company.
* Responsible for the development of a graphic and syntaxic editor of SADT method on PC. 3 persons – 13000 source lines .

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
(2) それらの問題点解決に CASE はどう寄与するか?

In the first phares of a software development project it's often difficult to know if we are in the right direction , what have already done , how to check the consistency of our project and persent the project's advancement (to chief or customers if there are).

Methods of software engineering help us , therefore CASE tools . CASE can be viewed as an environment that support the software engineering pows with configuration management or documentation facilities ,software inspection and so ...

Now , because of the size of software project , the communication between the different participants of the project is more and more difficult and important . I think that the use of CASE enhance the communicative and standardizational aspect of information systems development effects .

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

I can speak about the easiness of making modification with CASE tools , in the way we are not afraid to do many attempts.

The quality of documentation which is important for customers.

At last but not the least, the use of CASE tool is often pleasant.

I must precise that I have especially and CASE tool as support of standard methods of software engineering like SADT , SA/SD , SA.RT , E/A ... Of course methods are more important than tools .

3. アンケートの回答

ふりがな おざわ ひでゆき
氏名 小澤 英志

1. プロフィール

年齢:27 歳 性別:男
会社名:(株)SRA 部門:システム開発 5 部

これまでの主な開発経験
新預為システムの開発
水平メールシステムの開発

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
・ 仕様の記述がほとんどの場合厳密ではない. 例えば基本設計者といっても明確な記述範囲があるわけではないから設計者, あるいはユーザによってカバーする範囲がことなる.
・ ユーザと設計者の間で共通の言葉を持って設計を進められることは少ないから, 思い込みや取り違えが起こりやすいし, 仕様に矛盾や洩れがあっても気付きにくい. 設計者とプログラマの間も同様である.
・ ウォークスルーが軽視されていて工程の関係等で充分な時間を取れないことが多いため仕様の洩れが見つかりにくい.
・ 設計仕様のあいまいさは, 結果として物作りになってからあるいは, 物作りを終えてから要求仕様, 基本設計にフィードバックされることが起こる.(できてしまった物が仕様となる)

(2) それらの問題点解決に CASE はどう寄与するか?
・ 設計時における問題は従来の設計手法による問題であると考える. CASE の基礎となっている構造化分析, 構造化設計の手法を用いれば, 仕様の曖昧さはかなりの程度厳密性をもたせることができる. ただ仕様書の作成量はかなりの負担となるから,CASE は, 仕様書作成を容易にすること, 設計時の仕様の矛盾を設計者が気付くようにすることに寄与できると考える.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
・ 実際に CASE を使用したことがないから, どのようなことが障害になるかわからない.

3. アンケートの回答

渡辺瑞枝

1. ワークショップを通じて得られたもの (成果)
・ " エンドユーザ" の立場からシステムを使うという貴重な体験をした.
・ 理想を描くのは簡単だが実現への一歩が難しいということがわかった.(経験・馴れがいかに自分の頭を固くしているかが分かった)

2. 欲したが, 得られなかったもの (今後への期待)
・やったことが, 本当に構造化だったのか, 分からない.・割りと仕事に近い雰囲気で作業が行なわれたが (納期に対する切迫間, 残業があった) もっとリラックスして, 自由に討論できる場であって欲しかった.
要求分析の場には, 開発者とユーザがいると思うので, ロールプレイのようなことをした方が面白かったのではないかと思う. 今回は, 最初から最後まで開発者集団=プロジェクトの中にいるようで気分が落ち込んだ.

3. お使いいただいた CASE ツールに対する要望
・ UI が今いちと思ったが (例えば操作性の統一性があまりないとか, モードの多用とか...), 自分も MMI 部の設計をやっているので, 他人事として非難できないものがある.(見に積まされる)
・ あまりツールがリコウだと人間がバカになるのでこの程度でいいのかな?

4. 全体的な感想
この 4 日間で, 手法, ツールをマスターし, 会社にそのスキルを持ち返ろうなどとは最初から期待していなかった. しかし, 今後の向学心, 興味, 理想追求へのひとつの点火となればいいな, とは思っていた. 今思えば, これも他力本願であった. やはり, 自分がまずやる気を起こさなければ, 何も良い結果はもたらされないと思う. 逆にやる気さえ起こせば, 今後も, 勉強できるチャンス, 手段はいくらでもあるだろう.(若手の会は, 年例制限があるので, これが最後でしょう)

ふりがな よしだ かずとも
氏名 吉田 一知

1. プロフィール

年齢:28 歳 性別:男
会社名:(株) 岩手電子計算機センター 部門:流通システム課
これまでの主な開発経験

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
(2) それらの問題点解決に CASE はどう寄与するか?
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
開発における工程と期間の問題とは, いつでも悩まされます. 現在の開発は, 分析→設計→製造→テストという各フェーズの工程をたどりその中での分析, 設計は, 工程の一つにすぎず一定の期間内で終了させなければなりなせん. 期間の終了=工程の終了的な考え方が支配的であり我々もそれに支配されている場合が多いです. またこのフェーズが完了しなければ次に進めないという作業方法の為でもあります. このため, この工程では常に時間と品質のジレンマに悩むのです. 結局この段階に起因する問題がテスト等で発覚したさい際の後戻りは製造

時に造り込まれたバグの比ではなく, また結果として, 仕様書=プログラムという原則は, しだいに崩れていきます. このため完成した時には既に保守性に問題をかかえたシステムになっているのです. こうした点に対してCASE ツールがどう寄与するかというよりも, 製造工程においては, ある程度自動化が進んだ今日, 開発者, 開発管理者のもつジレンマに答えるべく,CASE ツールが世に出てきたのは自然であり, この流れの中で我々がツールをどのように使いこなすべきかが問われると思います. 根底にはワークステーションの発達等のハード的要因, プロジェクト管理技術システムのライフサイクルのとらえ方等の進歩もあり, それらを充分理解してこそ,, 使いこなせるものであるという意識はあります. 導入に際しての云々今後の課題として, 今回私がもう一つ興味を持つのは, こうした管理技術の考え方が 1900 年だ以前半に見られた経営管理論, 組織論に似ていることです. 生産性追求から出発した理論は, その後, 人間性の追求へまで進歩していきました. とすれば, 今後の CASE の背景にある, 管理技術論も, 人間というものにぶつかるのでしょうか. マシンに使われているという感が否めない現状からの興味でもあります.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
CASE ツールという一つのわく組みで定義されたなかにも, それぞれ特徴があるというのが明確になりました. また, ベンダーの方々に直接, 接する事ができ良かった (ばくぜんと) かなっと…
具体的な成果は, これから自分自身が生み出す…のかな.
2. 欲したが, 得られなかったもの (今後への期待)
・ システムをライフサイクルとして見た場合にツールがどの程度有効なのか, 分析・設計よりもむしろ, そちらに期待していたが, 今回は"有効そうだ"で終わった.
3. お使いいただいた CASE ツールに対する要望
日本語化が, もう少し進んでくれれば良い.
4. 全体的な感想
CASE ツールに対して, 私はあまり期待していませんでしたが, 実際に接して, 部分的にはかなり使えるかなという印象です. ただし, 誰でもが使えるわけではなく, 使う方もかなり事前の武装 (勉強しないと使えないんですね) が必要であり, それなりの心がまえが必要であると感じます. 私自身は, 実務の中でこのツールを生かすとするなら, やはり, システムライフサイクルとしての有効性に期待したいです. 分析・設計の部分は, もっと' あいまいな部分' を残したまま作業ができる様な形にならないと…という感じです. 方法論についても, 結局のところ学問分野としての永遠のテーマであり統一理論としての帰結はないだろうし, それは学者 2 人 (学者みたいな人) にまかせればよいのでしょう. それをケースバイケースで生かすしかないのでしょうね. 今回運営に御苦労された皆さんに感謝します.

ふりがな さいとう もとし
氏名 斎藤 基

1. プロフィール

年齢:28 歳 性別:男
会社名:富士通ビーエスシー 部門:仙台支店 4S/W
これまでの主な開発経験

機種	OS	言語
・PFU1500	UAS	ASM,COBOL — 通信制御
・S3000 シリーズ	OVIS/S	COBOL — 画像処理, バッチ
・9450		BASIC — 画像処理, バッチ

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
手作業による設計工程において発生した問題は, そのまま下流工程に引継がれ, 品質の低下を起こしやすい. また, 作業見積りやプロジェクト計画等, 不十分なほど次工程への影響が大きく, 生産性の低下の原因になる.
(2) それらの問題点解決に CASE はどう寄与するか?
支援ツール等導入による設計工程のの自動化で, 最適化, 信頼性向上を図れる. また, 各工程 (設計–保守) の効率か, コストダウンが見込まれる.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
CASE ツール使用に当たっての手法, 方法論の必要性. ツール導入はむずかしい?(社風, 技術等)
2. 欲したが, 得られなかったもの (今後への期待)
仕様作成までの機能 (ツール) を体験したかった. オブジェクト指向についてはあまり理解できなかった.(もっと勉強したい.)
3. お使いいただいた CASE ツールに対する要望
成果物の出力 (プリントアウト) をもっと簡単にしてほしい.

4. 全体的な感想
　全体的にもっと時間をかけたかった.(方法論に対しての知識が少なかった) ツール使用法よりも, 手法, 方法論について勉強したかった.

ふりがな　もり しょうぞう
氏名　森 彰三

1. プロフィール

年齢:30 歳　　性別:男
会社名:富士通ビー・エス・シー　　部門:仙台支店 1S/W
これまでの主な開発経験

機種	OS	言語
・PFU1500	UAS	ASM,COBOL — バッチ, 通信制御
・S3000 シリーズ	OVIS/S	PL/I,COBOL — 制御監視, 画面処理, バッチ
・M シリーズ	AiM	PL/I,COBOL — 制御監視, 画面処理, バッチ

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
・ ユーザの要求分析, 及びシステム分析を手作業により行ない, 当工程での誤りが発生した場合, 下流王手いにそのまま引き継がれ, 品質が悪く, 出戻り作業の多いシステムとなる.
(2) それらの問題点解決に CASE はどう寄与するか?
・ 設計工程支援ツールによる自動化で, 品質の向上, 生産性の向上を計れる.
・ システム保守の効率化, コストダウン
・ ソフトウェアの再利用が, ディクショナリの登録により可能となり, 統合・再適化に寄与する.
・ ソフトウェアの自動生成による省力化, 信頼性の向上につながる.
(3)CASE 導入に際してさまざまな障害をどう克服すべきか?
・ CASE 導入に伴い, プログラマ・レスが実現される. 現プログラマ作業に従事している者の, 上位志向を行なう.
・ 当検収会などへの積極的な参画により,CASE ツール使用の啓蒙を行なう.
・ 試行的に CASE ツールを導入し, 生産性, 品質の比較を行なう.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
・ 前提知識が全くなかったが, 今回紹介されたツールがおおまかに理解できた.
2. 欲したが, 得られなかったもの (今後への期待)
・ SA/SD 手法は理解できたが, オブジェクト指向が理解できなかった.
・ ツール, 手法の実際値としての効果が分らない.
3. お使いいただいた CASE ツールに対する要望
(Teamwork)
・ 日本語化 (より多くの…)
・ 日本語入力の簡易化
4. 全体的な感想
　方法論を学んでから参加した方が, より有意義だったろう.

ふりがな　ちねん とうる
氏名　知念 徹

1. プロフィール

年齢:27 歳　　性別:男
会社名:三菱電機コントロールソフトウェア (株)　　部門:開発部
これまでの主な開発経験

1986〜1988	火力発電プロジェクト運転支援エキスパートシステム開発 (特に推論エンジンの設計に従事)
1988〜1990	知的 CAI システム開発に従事
1990〜	ソフトウェア開発支援環境整備, 及び CASE ツール開発に従事

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?
・ 教育!!
・ ソフトウェアに対する客先とメーカのそれぞれの認識のまずさ
(客先:ソフトはおまけ, メーカ:言われたものをつくれば良い)

・ 一階きりのソフトウェア開発

(2) **それらの問題点解決に CASE はどう寄与するか?**

直接的には寄与しない.CASE が導入されたとしても人間が頑張らなければ良いシステムは作れない. しかし, 作業見積, 工程設計, および設計結果の表現手段として, 人間が頑張るときのより所にはできるかもしれない. 特にソフトウェアのビジブル化を可能とできるのなら, それが CASE の最大の効果であろう. これはウォータフォール型開発からスパイラル開発への躍進を可能にする.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**

成功するという確信をもつことが CASE 導入の障害を克服することが最大の鍵である. CASE 導入には成功事例の作成が必須. いくら良い方法, 良いツールでも成功の見通しがなければ現場にとって何の価値もない. したがって, 並行開発によって CASE が従来手法より優れていることを (データにものを言わせて) 実証するしかない CASE ツールはあくまで支援の道具でしかない. それをむりやり導入しても, 方面上 CASE 手法で作業しているように見えるだけで, 人間はついて来ない.

3. アンケートの回答

1. **ワークショップを通じて得られたもの (成果)**

確かに SA/SD 手法を使うと, 従来試験時に発生する多くの問題をつぶすことができることを実感した. しかし, 本手法を忠実に実行することには多大な労力を必要とする. CASE ツールなくして SA/SD 手法は使いたくない. それだけ手法とツールの依存関係が強いということだ.

2. **欲したが, 得られなかったもの (今後への期待)**

OOA/OOD/OOP の一環した方法論を知る機会に恵まれなかったことが残念である. 根拠はないが,SA/SD の一歩先に OOA～P が存在する気がする.

3. **お使いいただいた CASE ツールに対する要望**

(Teamwork) 表面的なことのみについて

・ テキスト修正方法の改善

・ SA → SD 変換の完全なツール化 (現状 UNIX ベース)

・ テキスト一斉編集の完全なツール化 (現状 UNIX ベース) あまり, 高級な機能を設けるよりは,SA,SD の基本機能を充実していかれた方が良いと思う.

4. **全体的な感想**

少し尻切れとんぼな感じで終わってしまった. 残念だ.SA だけでなく,SD まで完全に終わらせてみたかった. その上で OOA,OOD,OOP についても考察したかった. しかし, 確かに SA/SD は有効だ. かつ, 大変だ. その大変さゆえに 1 人だけで SA/SD 手法で頑張ることは難しい. 回りを変えていくことが必要だ. 言いたいことの半分も言えない内にワークショップが終わってしまった.SA 以前の作業についても, もっと議論の余地が残されている. 次の機会が得られることを望む.

ふりがな　やなせ けんいち
氏名　柳瀬 健一

1. プロフィール

年齢:26 歳　　性別:男
会社名:株式会社 KCS　　部門:ソフトウェア技術部
これまでの主な開発経験
「視覚的表現入力による仕様支援記述ツール」の開発に協力
某大学向け学性教育 (プログラミング言語教育) 支援ツールの開発

2. ポジションステートメント

abstract

数年前から CASE に対する期待が高まり, そのツールも商品化され多くのサイトで利用されている. しかし, 実際に導入しようとすると現状では問題が多く導入に踏み切れないという意見が多い当方も,CASE ツールの導入を検討しているが大きな問題を抱えている. 今回のワークショップでは,CASE ツールの状況把握, 導入に伴う原理/技法の理解, 及び, 導入に際しての問題点のヒントを得たい.

(1) **ソフトウェア開発の上流工程における現状の問題点は?**

当社ではメインフレーム主導の情報処理, 運用, 開発を中心業務としてそのソフトウェア開発は 富士通の SDEM によるトップダウンの方法論を採用し, 事務処理系の開発を行なっている. その様な中, 私は (好運にも) ワークステーション・ベースの環境で作業を行なっており, そのハード環境は大変恵まれ, 従事者は 20 人強と少ないが 1 人 1 台の WS が確保されている. 私自身の開発方法は SDEM とは言い難いが, それに基ずくウォータフォール, もしくは, プロトタイピングの方式を行なっている. その様な環境における問題点を, 全体の, 私の周り (WS ベース), 私個人の 3 つに分けてその問題点を以下に述べる.

1.1 **全体の問題点**

(1) 十分に設計を行なう時間がない

契約時における見積りの甘さやプロジェクトメンバーの技術不足から, 設計を十分に行なう時間がない. または, すぐにプログラミングに走る.

(2) 作業の集中化
各プロジェクトに必ず 1 人は居るであろう優秀な人材に作業及び責任が集中する. 他の人は添え物?

(3) 設計は会議室
設計はすべて会議室等に集まって行なう. 誰かがまとめて黒板に書き, それを記録しさらにそれをワープロ等に入力する. ツールを用いるのはプログラミングから. あるメインフレーマ曰く: [欲しいツールは別にない!現状でよい]

(4) 標準かがなされていない
開発プロセス (サイクル) のすべてのプロセスに対して各自, 各プロジェクトにより使用マシン, 使用ツールが異なる為互換がない. まして, そのためのベース環境もない.

(5) 再利用しない
メインフレームの事務処理は歴史も古く, その形態もほとんど変化していない. 過去の事例で類似したシステム・機能が沢山あるはずなのに, ほとんど毎回ゼロから作る. 標準化が問題なのか, 個人の問題か.

(6) ウォークスルー, レビューの軽視
ユーザとのコミュニケーションはさる事ながら, プロジェクト内のコミュニケーションが少ない. [なあなあ] や [暗黙の了解] とやらで, 各自が作業を勝手に進める.

(7) 設計プロダクトの軽視
ウォータフォール設計にも関わらず, 設計プロダクトは納品前 (システム開発後) に作成する. まして, その時間題に気付く. また, ユーザからの設計プロダクトの要求がなければ, 設計プロダクトは一切作成しない. 頭の中と, すぐにゴミ箱行きのメモ書き.

(8) 良い設計とは
良い設計とは [CPU 時間を食わない設計] と思っている人が多い.

1.2 私の周りの問題点

(1) 要求分析/定義ができない.
   (a) 総合的な知識修得者が少ない.
   (b) 何をどのように分析/定義するか?
   (c) ウォークスルー, レビューを軽視する.
   ユーザ側のシステム化に対する認識不足とユーザに意見を統括する担当者の技術/知識部即から要求項目が全部出きらない (まとまらない) うちに基本設計に進んでしまう. 最悪の場合プログラミングに進んでしまう.

(2) 設計ができない

   (a) 何を考えればいいのか分からない. とにかく何かを曖昧にプログラミングする.
   (b) 仕様が曖昧なのにボトムアッププログラミングを行なう (?).
   (c) 設計プロダクトを把握していない (が分からない).
   (d) 必要な機能を作らず (考えず), 不必要な機能を考える.
   トップダウンの方法論は理解できるが, 実際にどのようにして細分化していけばよいのか具体的な手法が分からないのが本心かもしれない.

1.3 私の問題点

私の現状の設計スタイルは紙と鉛筆を元に,UNIX ワークステーション上のお絵書きツールや文書処理システム (正確に roff) により設計プロダクトを作成している (CASE ツールや部分的な設計支援ツールは用いていない). トップダウンの作業により, 仕様や決定事項等はすぐに文書化し, その主旨等を明確にして, 設計プロセスの要所・要所で出来る限りユーザレビューやウォークスルーを行ない, その機能 (システム) の仕様を明確にする事によってフィードバックが最小限に留まる様, 心がけて開発を行なっている. その様な私のプロセスで発生している問題点とそれに付随する期待を述べる.

(1) フィードバックの問題
頭で分析を行なっているとフィードバックは発生する. またユーザのわがままな仕様 (要求) 変更もある. その様な事態が発生した場合, 現状では, 手作業により関連する全ての仕様及びそのプロダクトの変更を行なっているが, この作業を簡単に行なうことが出来れば,,,.

(2) 検証 (チェック) の自動化/省力化
設計には図表がつき物であるが, 馬鹿チョン式のお絵書きツールを使っている為, 入力/修正には労力がかかる. さらに, 一つの図表内におけるシンタックス, 一貫性, 機能分解整合性などの検証は図表を見ながら頭でもやもやとチェックする. また, 複数の図表間においても同じである. このような図表はもちろん設計プロダクト (プロセス?) の検証 (チェック) を自動化/省力化したい. 又, そのことによりバックトラックを避けたい.

(3) 設計の技術移転
私の周りの問題点として [設計が出来ない] としたが, 私自身 [設計って何?] と聞かれても答えに苦しむ. 現在のノウハウ移転の方法は, 経験者との共同開発や次作業の具体的な指示などを行ない, 労力による経験を積ませて理解させている. [設計のための CASE ツール] の様な設計プロセス支援システムが欲しい.item(4) 再利用/標準化
WS ベースの実開発は 2〜3 年と少なくかこの事例は少ないが, 私自身, 設計プロダクトやソースレベルの再

利用は出来る限り行なっている. が, その基盤となる環境がない為探すのに苦労する (ディレクトリやファイルをゴソゴソ). また, 全体を見てもそれに類する広いブラックフォームもない.

(5) 強調支援
何処とも, ネットワークでつながった個人ベースの WS 上で作業しているが, 何故か設計ミーティングやレビューは会議室にこもってしまう. マシン上で支援して欲しい. しかし, メールや talk じゃ問題外, 実開発の協調って?

(6) 進捗管理が曖昧
現在のプロジェクト内の各メンバの進捗状況等の報告は自己申告である為, 正確な管理が出来ない. 納期前になって"まだ出来ていません"って事もあり得る. プロジェクト運用時にスケジュールや各メンバの作業分担/内容を基に, 各メンバの開発状況と連動し, その実態はもちろん"A さんは作業は送れている.XXX しなさい"の様な警告を報告してくれれば.

(7) プロトタイピング
システム化を理解していないユーザと設計者の一番の相違はマンマシンインターフェースである. ユーザは紙面では分からないし, 開発者が XLIB,Xt 等でゴリゴリ書くには時間がない. ユーザインタフェース等をプロトタイプ構築する為のツールがない.

(8) 設計は楽しくない!!
紙と鉛筆を頼りに頭でモヤモやと考えるので設計作業は楽しくない. 知的感性を刺激してくれる様な設計をしたい.

(2) **それらの問題点解決に CASE はどう寄与するか?**
上述した問題点をまとめると,(1) 設計技術の移転,(2) 標準化/再利用,(3) 検証,(4) 設計の入力, に分類できる.CASE ツールの定義が曖昧なので,CASE から離れるかもしれない.

(1) 設計技術の移転
上流 CASE ツールがリポジトリやライブラリを介して結合されれば, [お勉強ツール] としての設計プロセス支援的な方法論修得に利用できるであろう. しかし,CASE はユーザがプロセスを行なった結果に対してのみ支援している. 即ち, ナビゲーション的な機能がない. [現在はこういう状況で, 次に何を行なって下さい] の様に事前にシステムがプロセス支援的なサポートを行なってくれればと考える. これはツール間の手順のみならず, 個々のツール内部も含める.

(2) 標準化/再利用
リポジトリやライブラリがしっかりしておれば, 開発環境の広いブラックホームとしての基盤となる. ここの CASE ツールがこれらを介して自由に共有できれば理想的かな?

(3) 検証
今まで頭で行なっていた不確実な検証をシステムが自動的に行ない, ここのプロダクト内の検証 (シンタックス, 一貫性, 機能分解整合性等) やプロダクト間の検証が強化されれば, より使用が確実なものとなり, バックトラックは減少する.

(4) 設計情報の入力
設計情報の表現, 方法論としての各種図表等の入力は, その方法論に添わない馬鹿ちょんのお絵書きツールよりは使える. ユーザインタフェイスの問題.

(5) その他
実開発の強調支援はリポジトリ等の情報共有だけではないはず.CASE ツールを用いるとそのインターフェースや機能が, 使っていて [苦痛] から [楽しい] への移行することを望んでいる.

(3)**CASE 導入に際してさまざまな障害をどう克服すべきか?**
CASE ツールの導入を検討しているが, 色々な問題からリストが大きく上司を説得するのに困っている.

(1) 品質・生産性の向上
(2) の技術移転の問題とも関わってくるが, 果たして現状で CASE ツールを導入すればその唱い文句である生産性・品質の向上は本当に可能なのか. 確かに [設計のお勉強ツール] として割り切れば良いのかもしれないが, それだけでは導入に踏み切れない.

(2) 技術移転
設計がわからない利用者にとっては [お勉強ツール] として技術修得の威力を発揮するかもしれないが, 現在の設計者に今までとことなる方法論 (関連モデル, 関連 DB, データのモデル化,CASE ツールそのもの, 等) を移転する事が容易に出来るのだろうか? 特に何十年と経験を積んできたメインフレーマはその典型的な例ではないかと思う. 具体的な例が, [DFD を認めない] とか, 頭が堅い. それは経験により培って来た今まで/これからの知識が足元から崩れ落ちるから新しい方法論を認めないのではないかと思う. また, 新しい方法論は今までの [機能中心の設計] から [データ中心の設計] に移行している. これらの人達にいかにした理解してもらうかは大きな問題である.

(3) ライフサイクル
技術移転の問題と関わるが, 導入に際し現場のライフサイクルを変える事が難しい. 開発のプロセスモデルの規定も曖昧であり, 作業分割のルールも決まっていない. 要求定義/設計の分析/検証の曖昧に行ない. まずプロトタイピングを行なって実際に動いてから考える, の様な安易な開発者が多い.

(4) 失敗は成功のもと
ある論文に,

成功=ツールの合致度+方法論+教育+支援+担当者の能力・熱意

とあったが, 導入にはかなりのリスクを伴いそうだ. CASE ツールの普及人や利用者の努力・熱意により大きく左右される.

(5) リポジトリの共有

現状の CASE ツールは［上流から下流へのツール, 上流間のツールでリポジトリが共有出来ない場合もある］と聞いた事がある. 組織, プロジェクト, 個人に対応した複数の方法論やツールが必要であり, これらがうまく統合できない事になる. 同一ベンダはもちろんの異なるベンダ間のリポジトリ共有が実現されていなければ意味がない. 再利用も局所的になる.

(4) はともかくそれ以外の良い解決案が見当たらない. 例えば, 業務命令として［今日から CASE ツール XXX を用いないと設計及びそのプロダクトは認めない］の様な全体プロセス又は一部プロセスへの強制が必要なのだろうが? 導入時, 一時的に生産性は落ちるだろうが, 今までの様な, 紙と鉛筆の設計とフィードバックが発生した時の労力/お金の大きさを考えれば, 現状の設計スタイルからすれば多少なりとも向上すると信じている. 今までの［泥臭い体験と体力の設計］から［刺激を受ける知的な自動か設計］に移行したいものである.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)

StP の操作方法のみ!

2. 欲したが, 得られなかったもの (今後への期待)

・ 導入に際しての問題点 (技術移転, ライフサイクル, インテグレーション等) をいかにして解決するか等の具体的な討論をしたかった. ・いかにして現場に問題意識をうえつけるか?

3. お使いいただいた CASE ツールに対する要望

・ StP
  ・ アニメーション, ナビゲートなどの機能が欲しい.
  ・ 強調作業支援が弱い.
  ・ プロジェクト管理の機能がない.

4. 全体的な感想

方法論や技法などのかなり低レベルな話しが中心であったのが残念でした. 青木さんグループに参加したかった.

ふりがな つきやま ひでお
氏名 築山 秀夫

1. プロフィール

年齢:28 歳 性別:男
会社名:株式会社 KCS 部門:オープンシステム部環境開発課
これまでの主な開発経験
インテリジェントビル等のオフィス設計用エキスパートシステムの開発
エンジニアリング・ワークステーション+パーソナルコンピュータによる検査業務システム

2. ポジションステートメント

今回のワークショップに参加するにあたって (現在の私)

(1) CASE に関しては, ごく最近知った.

(2) 今回の参加目的は
  (1) CASE ツールの現状は
  (2) 導入に伴い基盤作りは (ソフトウェア工学環境)
  (3) CASE ツールの効果を知りたい
  (4) 私が抱えている問題点は CASE で解決するか
  各所からの参加社とディスカッションしたい

(3) 現在, 私はプロトタイピング方式でソフトウェアの作業を進めているが, ユーザーからの不満要求やソフトウェアの構成上の矛盾などが発生したときの対処等, 修正に苦労している.

2. 私の周囲の環境

(1) パソコン上でソフトウェアの開発をしている item(2) 同じプロジェクトに属する人たちが議事録, 仕様書, ソースプログラム及びライブラリ, 実行モジュールなどを参照するのがスムーズにできない. また, 変更などの報告が遅れると悲惨なことにも成る.

(3) 報告等の道具 (ツール) としては, 文書による回覧や電話を使うことが多く, 後の作業等に有用と思える電子メール等の活用は, 十分にされていない.

(4) 1 人 1 台の割合で配られたワークベンチによる分散環境が運営 (運用) され始めたのはよいが, 業務上パーソナルコンピュータ用のユーザーソフトウェアの開発を重視するあまりユーザー依存度, ターゲット意識の強い開発環境になりがちでœ(分良い (汎用的な) 開発環境とは言えない.

(5) 仕様書設計作業の段階で様々な矛盾を無くすのが重要と考えているが仕様書や設計作業に時間がかかりすぎ, もう一歩踏み込んだ工夫にいたらない.

3. 私の周囲の支援ツール

(1) なしに近いその中で YPS SDEM の声が細々と聞かれるメインフレームの開発の中で一部しか行なわれていない.

(2) 私も SDEM やその周辺に点在するツール依存して, 開発環境やプロジェクトが順調には進められているとはいいがたい.

(3) 自前で支援ツールを作っても, 前述のような周囲の状況もあり, また, 極一部でしか機能しない為その支援ツールが忘れられてしまう.

4.CASE ツールを教えて欲しい

いままで構築したソフトウェアの資産を生かす方向としてオブジェクト指向, プロジェクト指向が, 強調されることは分かるが, 前述の様に, 私もたずされるフィールド (現場) の事情が有るが, この様な状況の中でこそ,CASE が役に立つと思う貴方,CASE ツールの効果を教えて欲しい.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
   方法論もよんだだけでは, 理解できなかったがワークショップでかなり理解できた.
2. 欲したが, 得られなかったもの (今後への期待)
   他の CASE ツールも触れてみたかった.
3. お使いいただいた CASE ツールに対する要望
   StP 今回使った所では無いかな?
4. 全体的な感想
   CASE ツールは, 手法, 方法論をもっと知った上で使うワープロとしてとてもいいなと思った.

ふりがな たからだ まさふみ
氏名 財田 雅史

1. プロフィール

年齢:28 歳 性別:男
会社名:株式会社 KCS 部門:オープンシステム部環境開発課
これまでの主な開発経験
大型汎用機向け運用管理システム
UNIX ベースの分散型環境向け運用管理システム

2. ポジションステートメント

「CASEって, どんなもんやろ?」

私は入社 6 年めではあるがその半分以上は, プログラム開発からはなれた作業をしていたため, プログラム開発自体は少ない. そして, その経験も入社後 1 年少しの間の大型汎用機でのコボルによるコーディング作業 (プログラムのメンテナンスと改良) が大半で, 設計等を含むプログラム開発にたずさわるようになったのはつい最近のことです. また, CASE についても今まで耳にしたことはあるが, ソフトウェア開発や, 保守の仕事の自動化を支援するツールとして CASE ツールというものがあるという程度の認識しかもっておらず, 詳細は知りません.

ようするに, 私は

(1) ソフトウェア開発における上流工程における現状の問題点
(2) それらの問題点解決に CASE はどのように寄与するか
(3) CASE 導入にさいして様々な障害をどう克服するか

といった, ポイントについて意見を述べるという段階には, まだ至っておらず, 今これからそのような段階に踏み込んで行く立場にある者なのです.

それゆえ,CASE に対しては

- それが具体的にどのようなものであるか
- それを使うことにとって, どのように生産性があがるのか
- 使い勝手はどんなものか (便利ではあるが, その修得に労力を要するのは困る)
- 現状はどうなっているか
  (CASE ツールはおそらく各ベンダから多種他様なものが提供されているのでしょうが, それ羅書くツールを任意に組み合わせて使う (いいとこどり) というようなことが可能なのか, つまり, 各社のツールを統合するための共通の基盤作りのようなことがなされているのか等)

といった興味, 疑問ばかりですが. まさしく,「CASEって, どんなもんやろ?」です.

こんな私にとって今回のワークショップは「渡りに舟」でありますし,「百聞は一見にしかず」で,CASE に対する理解をてっとり早く深めたく思っております.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)

・ SA/SD の考え方の一部.(今までは言葉だけで, よく知らなかった. 一歩ふみこめたような気がする.)
・ 今後, このての手法等について勉強してみようという気になった.(実際の仕事では, 上流工程にかかわることがなかったので, 今一つ興味がなかった.)

2. **欲したが, 得られなかったもの (今後への期待)**
3. **お使いいただいた CASE ツールに対する要望**

・ エディタの機能アップ (カット&ペースト, アンドゥ等)
・ UNIX コマンドを意識せずに操作が行なえるようになればよりよい.

4. **全体的な感想**

もっと前提知識を無理をしてでも, つめこんでくるべきであった. そうすれば, もっと私にとってみのりのあるものになったと思い悔やまれる.

ふりがな こばやし とおる
氏名 小林 透

1. **プロフィール**

年齢:27 歳 性別:男
会社名:(株)NTT 部門:ソフトウェア研究所
これまでの主な開発経験
DIPS(NTT のメインフレーム) 上の分散型の開発環境 (特に対話型の開発支援ツール, コマンド) の開発を 3 年間

2. **ポジションステートメント**

(1) **ソフトウェア開発の上流工程における現状の問題点は**?

・ 現状, 上流工程に関しては, 各設計者のスキルに依存するところが大きいため, 設計者によっての生産性や品質の面でばらつきが生じ, 大規模なシステム開発の場合には特に, 全体の進捗管, 品質管理に支援を来す場合がある.
・ 現状, システム開発方法論が数多く存在するため, どの方法論を適用すべきかが重要な選択事項となる. また, 同じ方法論を適用しても設計者によって, 考え方が異なってしまうという問題点もある.
・ 上流工程での設計段階で, ユーザからのフィードバックやユーザとのレビューが十分得られないため, 上流工程での仕様バグ, 検討漏れ等が混入しやすい.
・ これまで, 設計者とコーダが事なる人間であったため, 設計者の考えが正しくコーダーに受け継がれず, その時点でバグが混入することがあった.
・ 上流工程での生産物は, おれまで, 設計者が手続き (せいぜい, ワープロ) したドキュメントしかなく, 仕様変更や設計変更に際して柔軟な対応が出来ず維持メンテ工程における生産性が劣化している.

(2) **それらの問題点解決に CASE はどう寄与するか**?

CASE 特に上流工程を支援する UPPER CASE は, ツールそのものが設計方法論を含んでいるため, ツールの仕様方法をマスターすることにより設計者のスキルによらず, 一定の生産性, 品質のを確保することが出来る.

・ CASE ツールを用いることによりビジュアル的に設計作業が行なわれるため, レビューの容易化, 早期の仕様確認が図れる. 場合によっては, プロトタイプによるシミュレーション等が可能であり, より品質の高いプロダクトを生成することが可能となる.
・ CASE では, 上流工程での設計情報を下流工程まで引き継いで, ソフトウェアの開発を一貫して支援するツールであるため, 開発工程において, 人為的なバグが混入する可能性が少なくなる. また, 設計工程で品質が上げられるため, 下流工程の工数 (コーディング, デバッグ等) が削除できる.
・ CASE での設計作業がそのまま設計ドキュメントと成り得るので,2 重手間が無く, ビジュアル的なドキュメントが作成できる.

(3)CASE **導入に際してさまざまな障害をどう克服すべきか**?

・ これまで, 培ってきた各設計者の方法論を変えて行かなければならない. そのために, 講習会, トレーニング等でこれまでの文化を CASE 文化に変えていかなければならない. また,CASE を利用することにより, どのくらい設計作業が楽になるのかを身をもって体験することが重要であると思う.
・ WS の低康価で, 購入しやすくはなっているが, これまでのハード資産を無駄にすることはできないので, これまでの資産を有効に活用しかつ WS の高級な MMI を利用して開発作業を行なえるような環境を構築していく必要があると思う

ふりがな　おおおか としゆき
氏名　大岡 俊之

1. プロフィール

年齢:30 歳　　性別: 男
会社名: 東北コンピュータ・サービス (株)　　部門: システム開発
これまでの主な開発経験
- 富士通マシン OS 基本ソフトウェア開発
- 東北電力アプリケーションシステム開発

2. ポジションステートメント

(1) ソフトウェア開発の上流工程における現状の問題点は?

(1) 業務知識のないシステム担当者とシステム知識のないユーザ部門とのシステム開発への意識の不一致及び工程見積りの誤り
(2) ユーザ部門のシステム開発へのニーズ及び問題点をはっきりと明確化しないまま, 開発工程スケジュールの都合上から下流工程へスケジュールを進めて, 後になり仕様変更が多く発生してしまう.

(2) それらの問題点解決に CASE はどう寄与するか?

(1) CASE ツールをより便利にし構造化分析を行なうことにより現状の業務分析やユーザ・ニーズがモデル化され明確になる.
(2) 上流工程から下流工程へ, 情報の受け渡しがスムーズになる.

(3)CASE 導入に際してさまざまな障害をどう克服すべきか?

(1) CASE 導入の問題点として現状のプロセスの流れからデータの流れに着目する構造化分析の技法が開発現場に受け入れられるか, また CASE ツール導入による費用対効果 (生産性等) があるのが等考えられるので全社をあげてトップダウン的に, 推進していく必要がある.

3. アンケートの回答

1. ワークショップを通じて得られたもの (成果)
- 色々レベルの人が使ってみた場合の意見.
- ソニーテクトロニクスの渡辺氏の方法論に対するアドバイス.(今まで SRA の人のコンサルしかなかったので)

2. 欲したが, 得られなかったもの (今後への期待)
- CASE ツールや方法論のすばらしさ.
- CASE ツールを使って行なう開発工程の進め方.

3. お使いいただいた CASE ツールに対する要望
- コマンドを意識させない操作性
- データディクショナリ作成のバックアップ

4. 全体的な感想

上に記入したとおりです.

NO! と言える CASE

1990年9月

平尾 一浩
Hirata Software Technology Co.,Ltd.

若手の会

若手の会とは、30才までの技術者があつまり、ある一つのテーマに沿って、それぞれの立場から議論を行うものである。今回で2度目の参加となるが、それも来年で参加資格が剥奪される運命にある。こういう差別があって良いのだろうか。

本当に集中討論

今回のワークショップでは各グループ毎に演習問題から1つを選択し、それぞれの CASE Tool を使用して要求の分析、設計を行うというものであった。こうなると、CASE を駆使してさらさらと・・・

実際は CASE云々という前に、方法論に振り回されたというところであった。前半はどのグループもディスプレイの前で、ではなく、 OHP とホワイト・ボードの前の議論に集中していた。つまり、使う以前に解決しなければならない問題が多過ぎたようだ。私のいたグループは特にひどかった。オブジェクト思考をわかっているメンバーが一人もいなかったため、何が正しくて、何が正しくないのか、全くわからないといった状況から脱することができなかった。実に机上の討論(ちなみに空論ではありません)となってしまった。

CASE

CASE を深く使用した訳ではないため、詳細に渡っての検討や感想を述べることはできない。しかし、只一つ言いたいのは、あまりにも利用者側に方法論の知識を必要とするシステムでは今後の普及に問題があるということである。 CASE を利用していくことで、今までの思考錯誤のフェーズが短時間に回せるのは確かで、だったら短時間に何回も繰り返し、より深い分析や、より良い設計を行っていったらいいという意見が聞こえてきそうだが、やはりその思考錯誤の段階でデータのチェックや矛盾点の検証だけではなく、思考途中にその考えかたではいけないんだという【ＮＯ】と言える CASE に期待してしまう。（タイトルを気にしてのこじつけに聞こえるが、それは少しだけである）勿論、全てにおいてそのようなことを期待している訳ではなくて、出だし、つまり CASE の導入を行ってから着地する間の部分へのサポートは是非欲しいと思う。

CASE の導入でユーザがハマってしまうのは CASE の導入で方法論や技法の導入までを当然のように期待してしまうためである。しかし、現状は、こんなことができます、あんなことができますといった羅列が多い。ツールだもん、そんなもんよって納得することもできるが、今後の導入を検討していく上では、どのツールが一番、着地がスムーズにいくかといったところを見るのではないかと思う。

プレゼンテーション時に会場からの質問で、バージョン管理の話題が出された。OS に依存したその管理がドキュメンテーションも含めて今後どのように展開していくのかも興味あるところである。

予感的中

今までの若手の会と違うところは、グループ毎にマシーンがあること・・・

案の定とでも申しましょうか、どのグループの皆様も会場から出ることなく（4日間とも）議論を行われていました。先にも書いたように前半はマシーンを使ってもいなかったが、目の前にどんと置いてあるとみんなかな縛りにあってしまうようだ。私は『CASE もいいけどやっぱ小岩井よねっ!』とばかりにうきうきと熊本から8時間(乗換時間含む)もかけて行ったのに!! まっ、それでもうまいものはたらふく食ったし、いいか。

熊本にも光を

若手の会は、そのほとんどが岩手で行われている。どうして東北ばっかなんだろう。九州にも悩める小羊は沢山いるのだ!

そこで SEA に一言いいたい。

熊本(いや九州)の若手技術者にも光を!!

第 8 回 夏のプログラミング・ワークショップ

Case Study of CASE Tools

-- 若手の会に参加して --

富永 伸哉
ヒラタ ソフトウェア テクノロジー株式会社
システム開発課
E-Mail : tommy@hst.co.jp

<< A 型の僕は... !! >>

このワークショップに参加する前に書いたポジションペーパにおいて、 CASE Tool についての幾つかのポイントを挙げておいたので、そのポイントについて以下に述べる。
(A 型ならではの書き出しである :-)

●各種 CASE Tool はどのようなものなのか！？

ベンダー 5社のうち 3社が構造化手法を支援するもので、構造化手法を支援しているものは、リアルタイムな制御系にも対応していた。時間の関係(方法論の習得に時間が取られてしまった)で、全ての機能を使いきる事は出来なかったが、概観図(Context Diagram) からデータ・フロー図、データ辞書(Data Dictionary)、構造図(Structure Charts)などを統合的に支援していると言った感じであった。
また、リバース・エンジニアリングの機能を提供しているツールがあったが、これを何に使うのかは疑問である。

グループ実習ということで、オブジェクト指向を支援している Grapher Gear とプロトタイプ手法を支援している PWB を使えなかったことが、非常に残念であった。
(生まれつきクジ運はよくない!!)

● CASE Tool の問題点は！？

ネットワーク機能の強化
実際の開発においては、グループまたはチームによる分析／設計作業を行う事が多く、そのために資源の共有やネットワーク上での整合性チェック機能などネットワーク機能を強化する必要がある。

Upper CASE と Lower CASE のリンク
現在の CASE Tool の殆どが上流工程を支援している Upeer CASE であるが、ソフトウェアのライフサイクルを考えた場合には、テスト環境やテスト技法など、上流工程から下流工程までを全体的に支援する必要があるのではないだろうか。

方法論の複数化
最後にわがままな要求ではあるが、一つの CASE Tool で複数の方法論を支援してくれれば非常にうれしいのだが。
分析/設計の際に、時には SA/SD で、またある時は OOA/OOD でといったように。

● CASE Tool を導入し、有効的に利用するには何をすれば良いのか！？

現在の CASE Tool は、その名の通り単なるツールでしかないので、そのツールを有効的に利用するためには、最低限、その基礎となる方法論は理解しておく必要がある。後は使い方の問題ではないだうか。ただのお絵書きツールやワープロとして使うのではなく、実際のソフトウェア開発(現場)において、とことん使いきる事によって CASE Tool がカスタマイズされ、生きたツールとなってくるのではないだろうか。

<< いや〜、ワークショップは楽しい !! >>

ここからが本音である。

今回初めて SEA のワークショップに参加したが、実に有意義なものであった。 3泊 4日という短い期間ではあったが、参加された方と色々な話をして多くの知識を吸収することができたし、ワークショップ夜間の部では、盛岡の美味しい日本酒を呑みながら聞いた業界の裏話も大変興味深いものであった。

また、開催地となった盛岡の街も非常に気に入ってしまった。わんこそばや冷麺といった郷土料理や立ち寄った食堂のおばちゃんから、なんとなくではあるが東北人のあったかさや底力(パワー)を感じた。

最後に、来年も『若手の会』に参加出来る事を願って .... おわり。

12. 資　料

ＣＡＳＥによるソフトウェア開発

落水　浩一郎

静岡大学　工学部　情報知識工学科

(1) CASE とは
(2) システム動作の表現法
(3) CASE ツール機能の現状
(4) 新しいソフトウェア開発の枠組と CASE ツール
(5) CASE はソフトウェア開発環境のフロントエンドである
(6) CASE と他のソフトウェアエンジニアリング諸技術
(7) CASE 導入にあたって
(8) まとめ… CASE の試用経験より

Koichiro OCHIMIZU

Department of Computer Science, Faculty of Engineering, Shizuoka University
3-5-1 Johoku, Hamamatsu, 432 JAPAN

1. CASEとは

CASEとは、Computer-Aided-Software-Engineering の略である。プロトタイピング技法の重要性の認識が深まるにつれ急速に充実してきた。開発対象システムの動作記述や解析を種々の図式表現に基づいて支援し、システム動作の可視化を図っている。CASEのねらいと効果は以下の通りである。

(1) コストの高いソフトウェア・エラーの発生源である要求定義や機能設計段階におけるエラーを早期に摘出することにより、失敗の危険性を減少させる。
(2) ユーザ要求を早期に確定させる
(3) 上流工程の標準化を促進する。

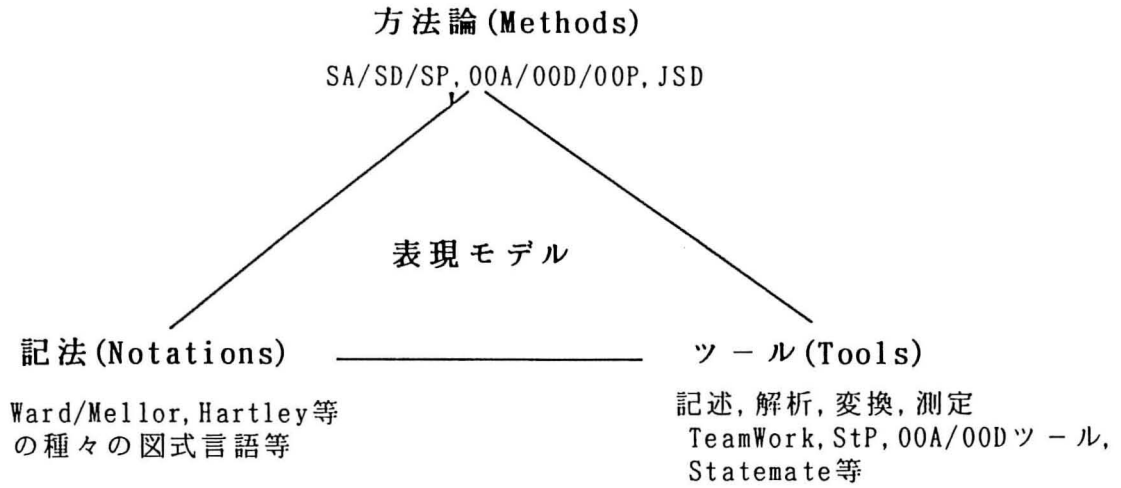


図1. CASEの構成

CASEは、方法論、記法、ツールの3つの要素からなる(図1)。

(1) 方法論

ソフトウェア開発の手順を与えるもので、開発局面の定義、局面間の移行支援法(構成管理等)、設計対象物の表現法と検証・検査法、チーム構成法等を定める。方法論は、再現性があり、教育可能であり、ソフトウェアツール群によって開発者の活動が自動化可能であり、ソフトウェアシステムの進化に対応できるものでなければならない[1]。

(2) 記法

要求仕様や、設計仕様を記述するための言語、70年代の記法がテストや変更が容易であるシステム構造を記述することに目標があったのに対して、80年代には、システム動作の記述・解析が可能であるような方向に進展した[2]。

(3) ツール

作成・編集、解析、計測、変換、保存等のソフトウェアエンジニアリング諸活動を支援するプログラム

方法論、記法、ツールいずれも対象分野の特徴に応じて、様々なものがある、対象分野の特性やソフトウェア開発の規模に応じて、まず方法論を選択し、次に記法を定め、最後にツールを選択することになる。このとき組織の作業形態や、計算機環境に応じてカスタマイズする必要がある。

2. システム動作の表現法

種々の設計方法論には、設計対象物(やその動作)を表現するための中心的な基礎概念がある。応用分野の特徴と、設計段階における解析の目標にあわせて、適切な基本概念を吟味・決定することは、システム設計段階における最も基本的な活動である。いくつかの例を紹介しよう。

2.1. データフロー

問題記述からデータと機能を抽出し、その接続関係を設計する。文献 [3] にあらわれる例題をもとに説明する。問題の概略は以下のとうりである。「ホストからネットワークを介して設定される制御パラメータをもとに、一定時間の間、製造プロセスの制御をおこなう。制御される量は、温度と pH 値である (図 2)」

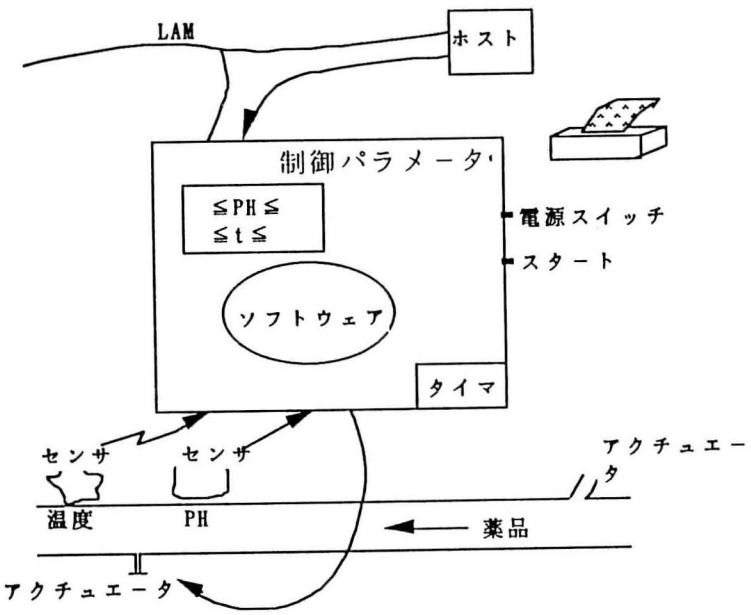


図2 プロセス制御の例題

問題から、図 3 に示すような機能とデータが認識される。機能は○印 (データ変換子とよぶ) で表現され、「設定パラメータを変更する」、「温度を設定値範囲内におさめるよう制御する」、「pH 値を設定値範囲内におさめるよう制御する」、「温度が設定値範囲内にあるかいなかを検査する」、「指定された時間間隔の終了を通知する」などがある。記号=はデータストアとよばれデータの生成と利用のタイミングが異なる時のデータ記憶の目的に使われる。矢印はデータとその流れの方向を示す。

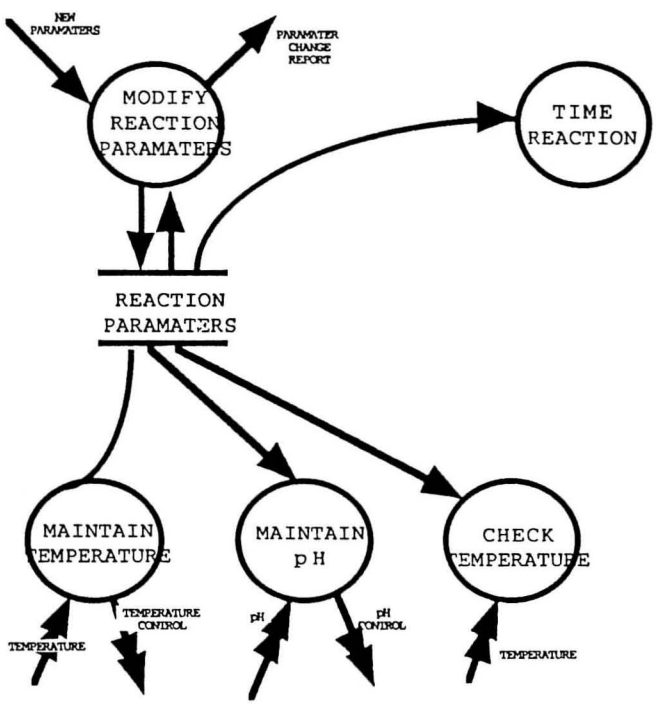


図3 データフローの例

例題がいきなりリアルタイム制御の例からはじまったので面くらわれたかと思う。図3で使用された記法は、事務処理システムの設計においては自然に利用できる。図4に示す業務フローやジョブフロー[4]は表現形式こそ違えデータフローそのものであることに注目して欲しい。

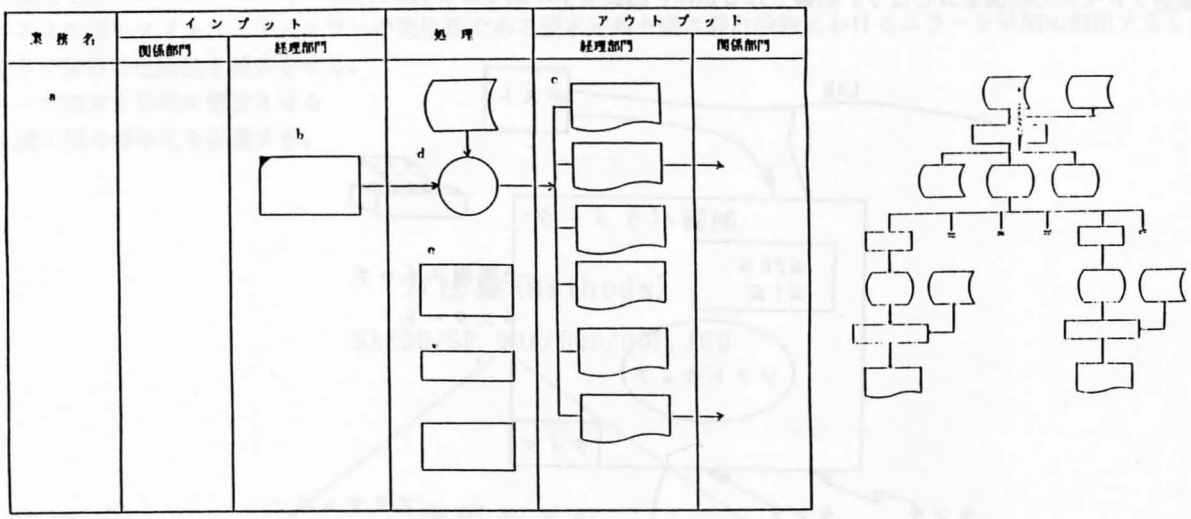


図4　業務フローとジョブフロー

一般に事務処理システムの設計は図5に示すようなデータの接続関係をTop-Downに認識していく過程としてとらえられる[5]。機能やデータの量が圧倒的に多いことが事務処理システムに特有の特徴であり、階層性をとりあつかう方法論、記法、ツールをさらに発展させる必要がある。

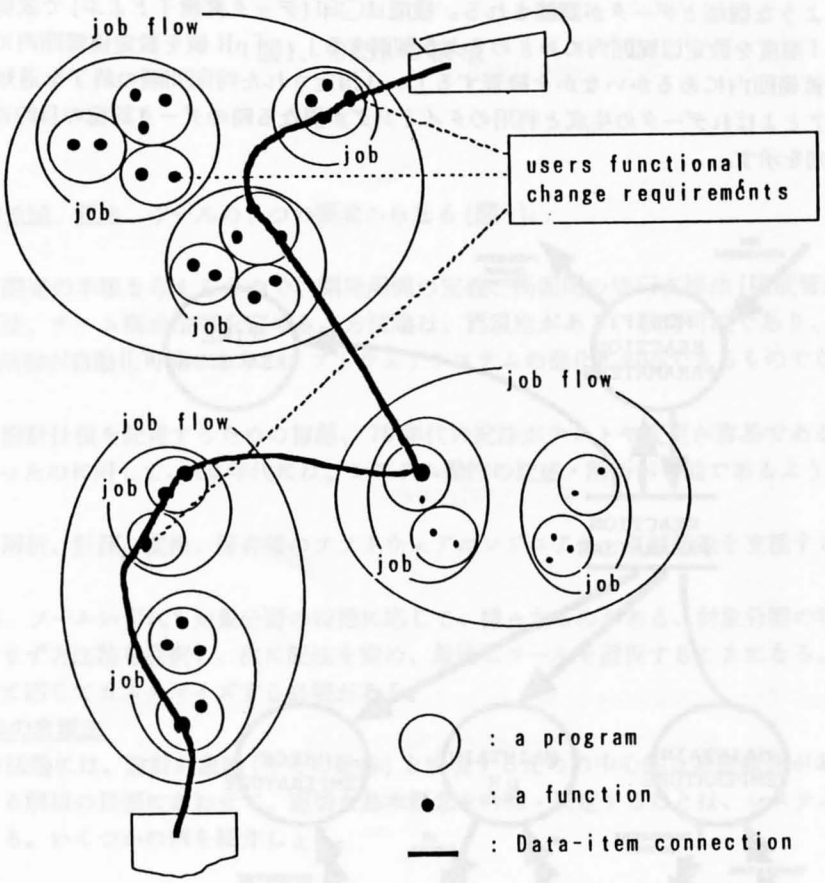


図5　事務処理システムの一般的構造

2.2. 有限オートマトン

有限オートマトンモデルにおいては、状態、事象、反応の3つの概念を基にした状態遷移図や状態遷移表を利用して、システムの動作モデルを記述する。図2の例題をふたたび利用して説明する。

状態は、「システムの動作パターンが固定されている時間間隔」として定義される。図6に示すように時間軸にそってシステムの動作イメージを想像した時、「電源がはいっていない(OFF)」、「制御プログラムが動作していない(IDLE)」、「温度範囲の検査結果に対する信号を待っている」、「制御中であり、制御時間の終了の通知(DONE), 異常状態の発生の通知を待っている(REACTING)」のような時間間隔が認識される。

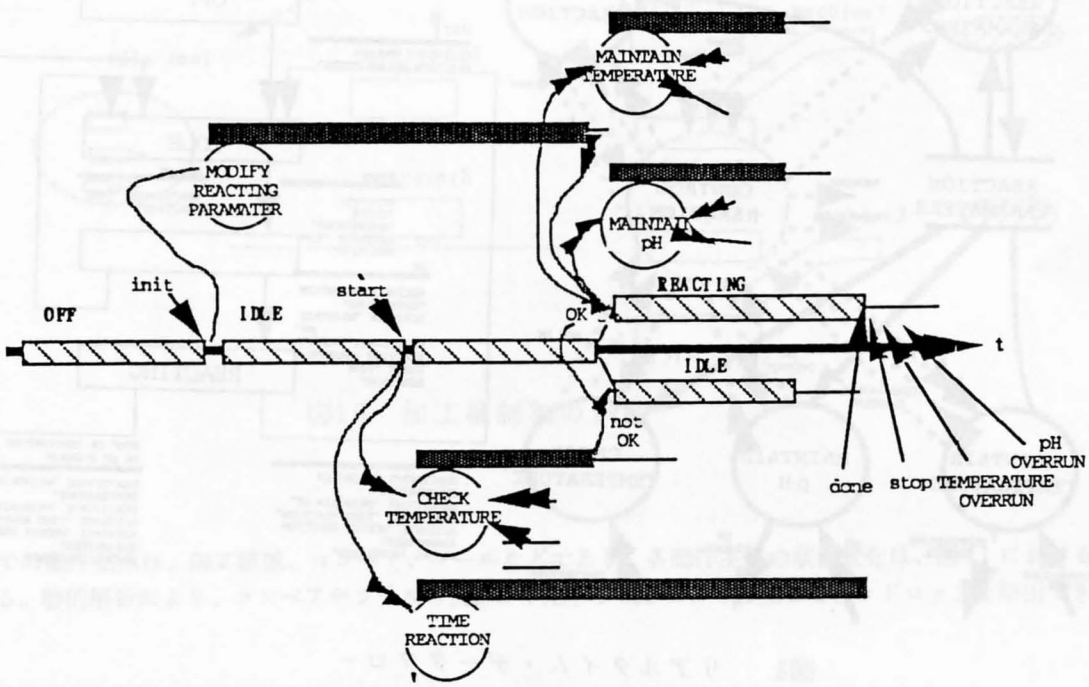


図6 時間軸にそったシステムの動作認識

状態遷移図ではこれを○印で表現する(図7)。このような状態は事象の発生や信号の到着によって解除され、指定された動作(機能)を実行したのち、再び新たな固定状態に推移する(図7).

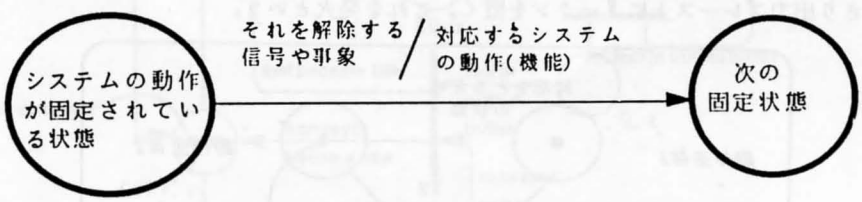


図7. 状態遷移図によるシステム動作の表現法

このようにシステムの状態を状態遷移図で記述し、シミュレーションにより動作解析を支援するのが有限オートマトンを利用した設計である。リアルタイムデータフローの手法は、図8に示すような制御変換子(Control Reaction)を導入することで、図3で認識された機能の起動タイミングとシステムの動作(状態遷移図)を結びつける。

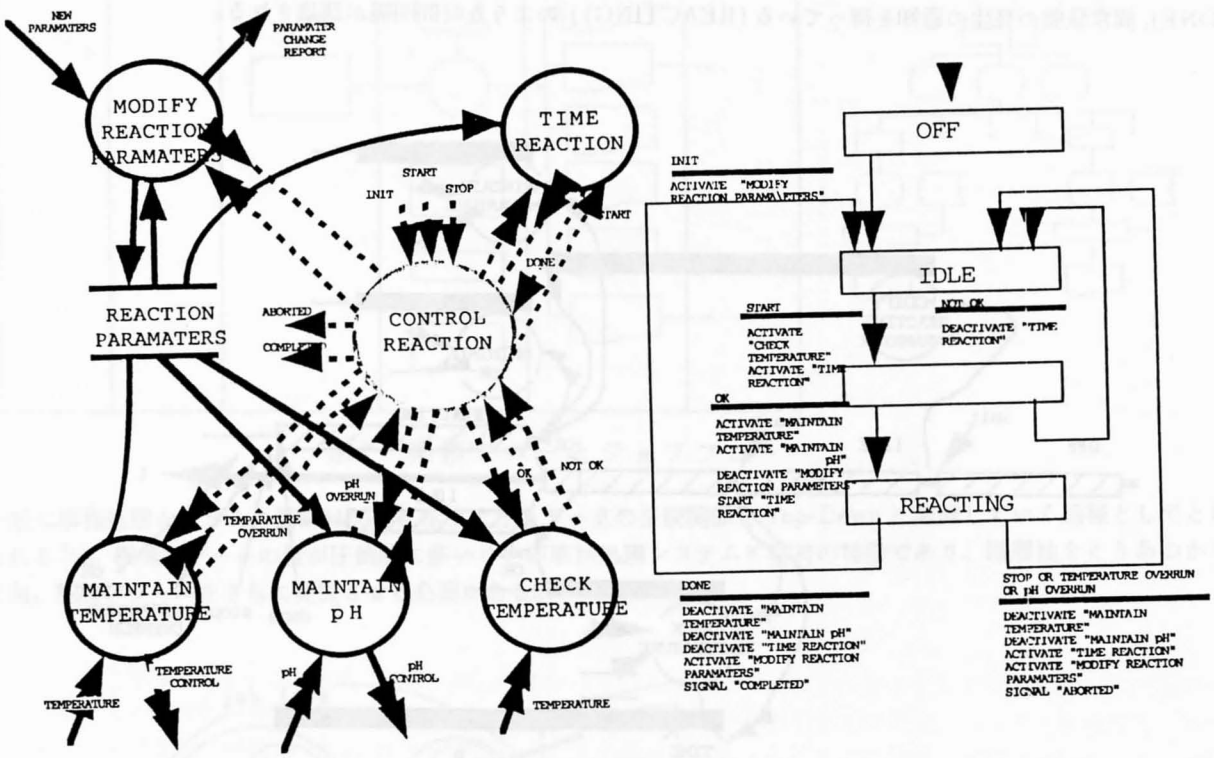


図8　リアルタイム・データフロー

2.3. ペトリネット

いくつかの並行動作する動作主体間の同期条件を明確にし、デッドロック検出などの静的解析やシミュレーションなどの動的解析を支援する。ペトリネットは図9に示す概念でシステム間の同期を表現する。図9においては2つの動作主体を仮定する。○印はブレースとよばれ状態遷移図と同様にそれぞれの動作主体の状態をあらわす。有限オートマトンの時とは異なり、動作主体の内部状態ではなく、他の動作主体と干渉する時点(同期待ちの状態)を定義することが多い。記号・はトークンとよばれ、ブレース中にトークンが存在することは、動作実体が現在その状態を保持していることを意味する。記号—はトランジションとよばれ、すべての入力ブレース上にトークンが置かれたときそのすべてのトークンをとりさり出力ブレース上にトークンを置く。これを発火という。

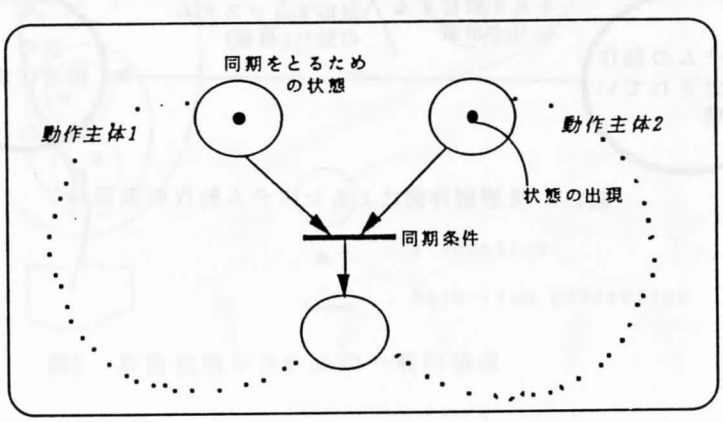


図9 ペトリネットによる同期の表現

文献 [6] にあらわれる例題にもとづいて具体例を説明する。図 10 において、加工機械は、コンベアによって運ばれ入力バッファ中におかれるツールを装着して加工をおこない、仕事が終了するとツールをとりはずして出力バッファに置き、新たなツール要求を出す。コンベアは、加工機械間で共用するためのツール置場と入力 / 出力バッファ間で、ツールを移送する。

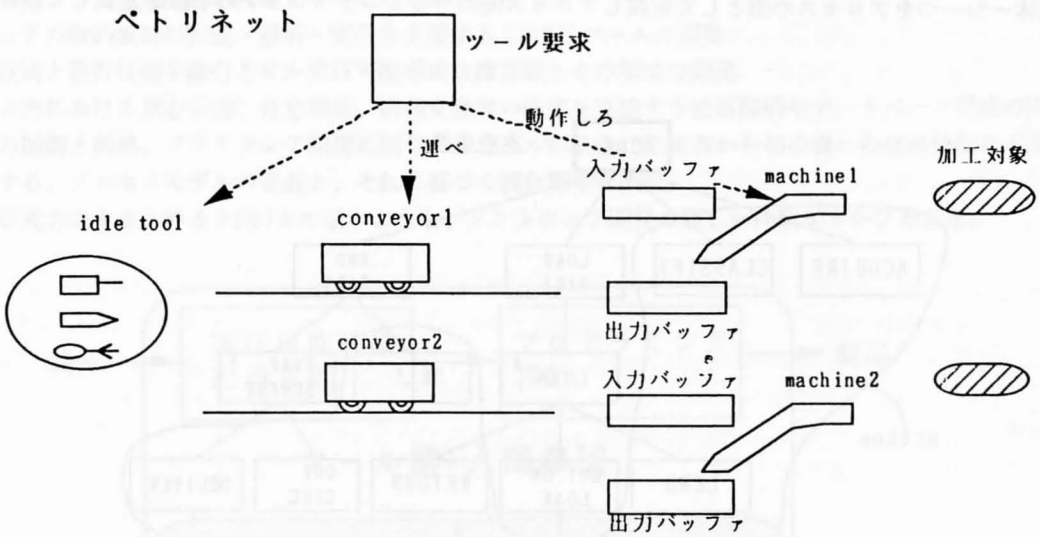


図10　加工機制御の例題

ここでの動作主体は、加工機械、コンベア、ツールなどであり、各動作主体の状態変化は、図 11 におけるループに対応する。静的解析により、コンベアやツールの資源が不足することによって生じるデッドロックを検出できる。

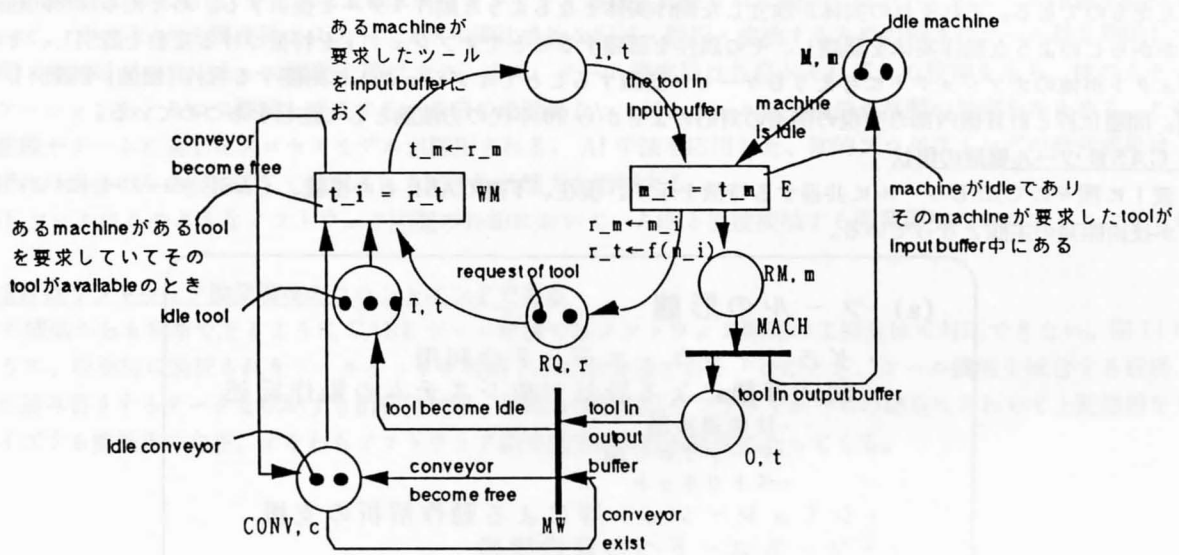


図11　ペトリネットによる動作表現例

2.4. 実体アクション図

動作記述の基礎として、機能中心、制御中心の記法を紹介してきた。ここからデータ中心の記法を紹介する。JSD法[7]では、あるデータオブジェクトの生涯を動作記述の単位とする。まず情報システムの管理の対象となるデータ(実体)を洗い出し、その実体の状態が変化する時点を実体に対する操作で識別する。そのような操作を時間系列で並べ木構造で構造化する。図書管理システムの場合、実体は本、操作は購入、貸し出し、返却、廃棄などになる(図12)。この実体一つ一つをプロセスの型として定義しプロセス間のコミュニケーションの手段を定義して設計は終了する。

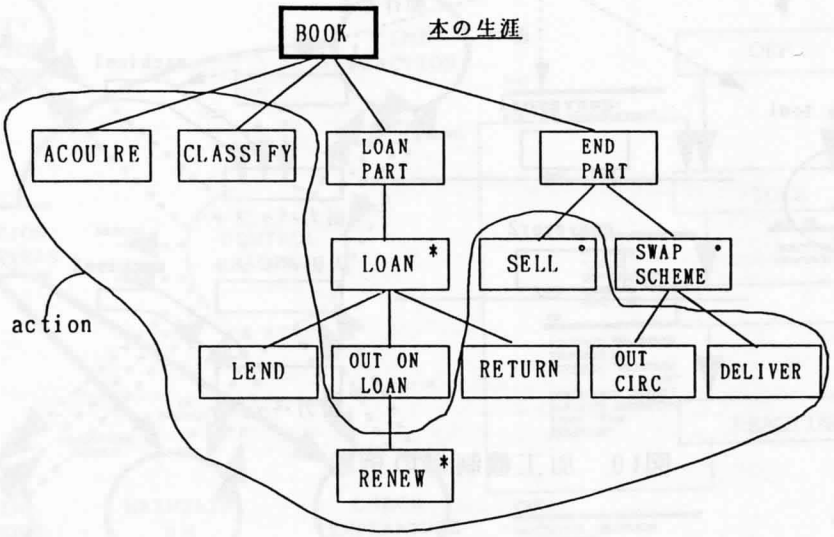


図12 実体アクション図

2.5. オブジェクト

オブジェクトとは、データと、その状態を変化させたり報告したりする操作を一つのカプセル化された実体としてとらえたものである。それぞれの実体が独立した動作実体となるような動作モデルを提供する。あたえられた問題記述のなかからこのような動作実体を認識し、その属性を認識することでオブジェクトを特徴づける変数を識別し、そのオブジェクトが他のオブジェクトに与えうるサービス認識することでオブジェクトに附随する操作(機能)を識別していく[8]。問題世界と計算機内部の実現の構造の対応のよさから90.年代の方法論として注目をあつめている。

3. CASEツール機能の現状

表1に種々のCASEツールに共通する特徴を示す。現在、下流CASEとの接続、CASEデータベースとの結合などが技術開発の主眼となっている。

(a) ツールの形態

・グラフィック・エディタの利用
・図式言語による設計対象システムの動作記述
  -状態遷移図
  -データフロー図
  -ペトリネット
・シミュレーション等による動作解析の支援
・ソースコードへの変換機構

(b) 稼働環境

・ワークステーションと
  ビットマップ・ディスプレイ

表1 CASEツールの共通特性

4. 新しいソフトウェア開発の枠組とCASEツール

ソフトウェア工学に関する研究は、ソフトウェアライフサイクルの概念の形成とその要素技術の確立からはじまり(70年代)、上流工程の形式化と自動化による革新を経て(80年代前半)、現在人間要因の分析と支援法の模索にまで研究対象領域が拡大されつつある(80年代後半)。その歴史は、他の工業製品の製造原理を転用することから始まり、ソフトウェア作成に関する認識の深まりにつれて独自の技術体系を構築しつつある過程として眺めることができる[9]。

現在、以下のような研究課題が目標となっている。

(1) ソフトウェアの動的振舞の記述・解析・実行を支援するCASEツールの開発
(2) 要求仕様技術と設計技術を融合させた実行可能形式仕様言語とその環境の開発
(3) 開発チーム内における意志伝達、合意形成、情報交換等の作業を支援する交信機構やデータベース構成の開発
(4) 集団活動の制御と調整、ソフトウェア開発に関与するツールの統合、熟練者から初心者への技術移転などを表現するなどを有する、プロセスモデルの定義と、それに基づく統合環境の開発。

これらの研究方向をまとめると図13に示すような、ソフトウェア開発の新しい枠組がうかびあがる。

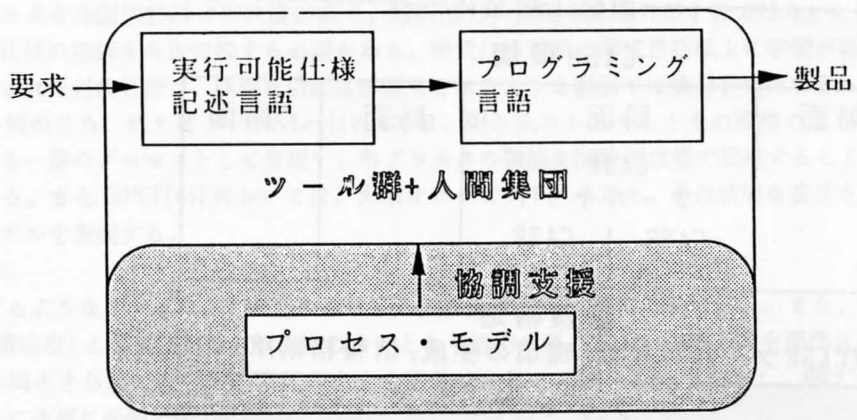


図13 ソフトウェア開発の新しい枠組み

図13において、ユーザー要求は、実行可能仕様記述言語や動作解析が可能な図式言語で記述され、解析、検証された後、設計情報(効率の考慮)を付加した後、プログラミング言語で記述されて製品となる。このような作業を進めるにあたって、プロジェクト構成員はソフトウェア中間生成物を記述・解析・変換するための種々のツール群を利用しつつ、人間・機械系が一体となって課題を達成する。また、チーム構成員は各自がそれぞれの役割をもち、他の人とコミュニケーションをとりつつ課題を達成する。各自の活動においては、ツール統合機能が活動の効率をたかめる。このために組織やチームに適したプロセスモデルが採用される。AI手法を応用した、知的プログラミングの研究成果は、上記いずれの場合にも必要に応じて適用され人間の知的能力を増幅する。

CASEツールはこのようなソフトウェア開発の枠組において、人間と直接接触する開発環境のフロントエンドである。

5. CASEはソフトウェア開発環境のフロントエンドである

前章の議論からも判断できるようにCASEツール単独ではソフトウェア開発の工程全体に対応できない。図14に示すように、局面毎に選択されたツールキットを利用するのが普通である。このとき、ツール機能を統合する機構、ツールが読み書きするデータを格納する情報格納庫、組織の作業形態やプラットホームの構造にあわせて上記機構をカスタマイズする機構等の存在、すなわちソフトウェア開発環境の存在が重要になってくる。

ソフトウェア開発環境とはツールの集合である。もちろん単なるツールの寄せ集めやライブラリを意味するのではなく、環境全体としては以下のようなサービスを環境利用者に提供しなければならない[10][11]。

(1) 集団活動の制御
(2) 熟練者の作業プロセスの再利用
(3) ソフトウェアプロジェクト構成員間の情報交換支援
(4) 開発対象物に対する目標と基準の設定、および個々人の活動の調整
(5) ユーザ固有の作業形態に適合した開発環境の生成

近年、このような統合環境の構築にあたってプロセスモデルに基づく統合環境の開発が注目を集めている。プロジェクト管理、文書管理、情報伝達管理の方式や、利用する方法論やCASEツールは、対象分野の特性、組織構成員の分散度、プロジェクトチームの規模、データの安全性や機密度、利用する計算機設備などによって大きく異なる。それぞれの開発形態に応じたプロセスモデルを環境内に内蔵し、上記のパラメータに適合する開発チームにマッチした環境を提供しようとする技術開発の試みが統合環境構築の目標である(図15)。

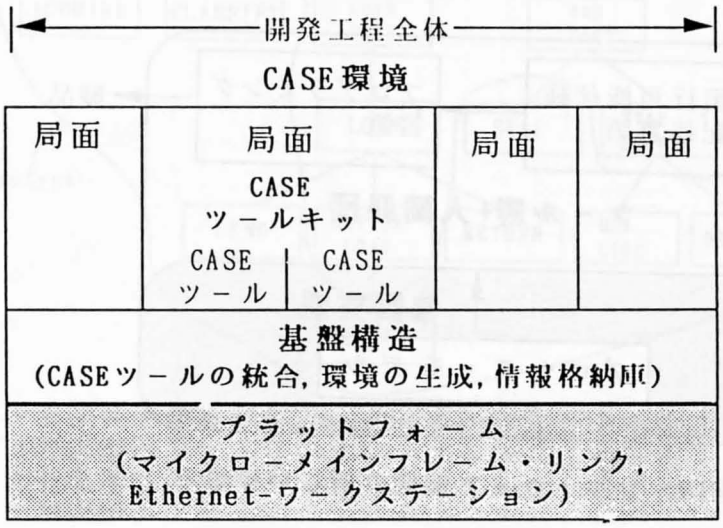


図14 CASE環境

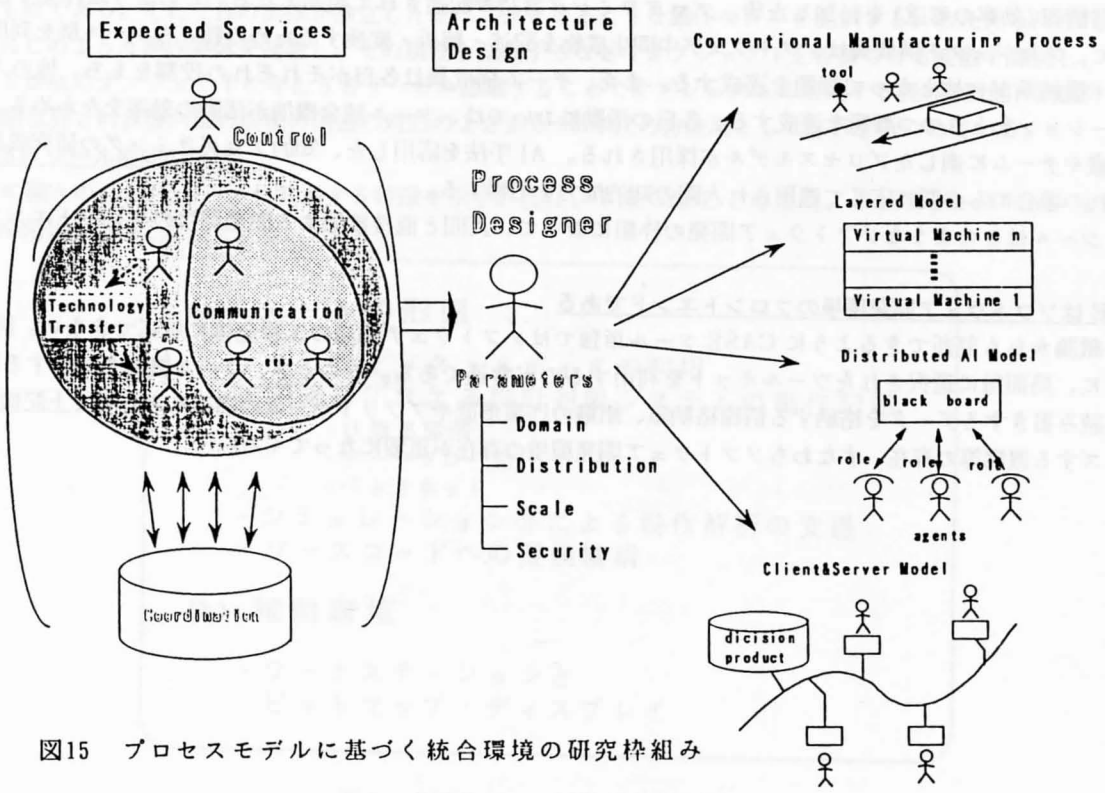


図15 プロセスモデルに基づく統合環境の研究枠組み

6. CASEと他のソフトウェアエンジニアリン諸技術

(1) 再利用技術

ソフトウェア再利用の活動は、ソフトウェア開発活動の一貫として位置づけられるべきである。ソフトウェア開発における再利用とは、ソフトウェア開発活動を通じて生みだされる情報の再利用である。すなわち再利用支援のツールは、同時に開発支援のツールでなければならない。再利用の活動は、対象物をさがし、それを理解し、現在の目的に適合するように変更し、全体に組み込むというプロセスをふむ。このためには、現在のCASEツールは情報格納庫中のある特徴をもつ対象物の検索の支援、構成管理と変更作業支援、統合化支援などの機能をさらに強化する必要がある。

(2) 保守技術

CASEはソフトウェアの動作解析を支援する。2章で述べた種々の動作表現モデルは、ソフトウェアの変更要求と変更箇所の結びつけ、波及効果の解析などの作業において、有用な情報をわれわれに提供するとともに、詳細な情報のコンテクストを与える。この意味において、変更の作業もまたソフトウエア開発活動の一貫であり　変更支援のツールは開発支援のツールでなければならない。

(3) 実行可能仕様

2章でのべたような図式言語は形式度が低く、解析能力、自動生成能力が十分ではない。操作的仕様技術等に基づく、実行可能仕様の技術も今後発展する必要がある。形式仕様言語は図式言語以上に学習が困難であり、ユーザ親和性が低い。限定された対象分野で、極端な信頼度が要求されるような場面では威力を発揮するものと思われる。

実行可能仕様のうち、例えば、PAISLey[12][13]は、組み込みシステムとその環境の双方を、たがいに干渉しあいながら動作する一群のプロセスとして表現し、各プロセスの機能を関数型言語で記述することによりプロトタイピングを支援している。またGIST[14]においては、大域オブジェクトベースと、その状態を変化させるデーモンの集合としてシステムモデルを表現する。

(4) 人間要因

下記に挙げるようなソフトウェア開発における人間の活動[15]は実に知的である。また、新しい状況に遭遇した時の柔軟性(臨機応変)と学習能力は、創造性のもととなり活力の源となる。表現形式を標準化し強制するだけのCASEツールでは結局あきられてしまう。下記のような活動における人間の知性を増幅し、弱さを補強してくれるようなCASEツールこそ真に必要である。

- 問題を分析しシステム・イメージを固める…対象分野の知識を統合しシステム設計へ写像する
- 問題を技術的解に変換する…新規システム特有の制約を考慮しつつ、設計問題解決の経験的知識を再利用する
- 設計結果をコーディングする…データ設計、アルゴリズム設計に関する経験的知識を有効に利用する
- 誤り検出の方針を定める…エラー現象と原因のつながりを発見するための戦略を状況に応じて選択する
- プログラムを理解する…開発者による意志決定の過程を再現し、変更箇所とその波及効果を読みきる

7. CASE導入にあたって

CASEツールの導入にあたって注意すべき点を導入計画から運用・評価にいたる時間の流れに沿ってまとめる[16]。

(1) ツールの導入と学習にかかるコストを過少評価してはならない。

(2) ツールを危機状態にあるプロジェクトの救世主として位置づけてはならない。過負荷状態や予算超過したプロジェクトは新技術導入の場としては不適切である。

(3) すべてのプロジェクトに同じツールを同じ方法で適用するのは誤りである。

(4) CASEツールは組織の環境にカスタマイズしてから利用すべきである。

(5) 仕事に適するツールを選んだとき、利用者に仕事の進め方がかわったことを通知すべきである。

(6) 学んだ人を次には教師にすることによって人材を養成する

(7) 仕事に適合するように選択されたツールは本来の意図どうりに利用されなければならない。しかし、おおくの場合開発者の意図によりねじまげられて使われることが多い。

(8) 単なる慣れによる慣習化をふせぐため定期的に効果を測定すること

(9) なにかトラブルがおこった時、ツールをスケープゴートにしないこと

(10) 生産性を上げるためのツールは、同じかそれ以下の資源量でいままで以上のことができる(量、質、時間)という観点から評価されるべきである。

(11) よく設定されたゴールをもたない、単なるツールや方法論の展示店になってはいけない

(12) 方法論やツールは小規模プロジェクトではうまくいく、しかし大規模プロジェクトではうまくいかない。パイロットプロジェクトでの試用によりどの分野での適用が適切であるかを判断すべきである。

(13) 組織は方法論をそのまま採用するのではなくカスタマイズする必要がある。このとき、ツールは媒体の役目をする

(14) 組織やその中の人々は劇的な変化をこのまない

(15) 生産性学習カーブは単調増加ではない。たちあがり、さちり、そして減衰する。

8. まとめ… CASEの試用経験より

1990年9月5日から8日にわたてソフトウェア技術者協会(SEA)の主催で開催されたCASEに関するワークショップの結果を報告することにより、CASE導入の効果と、問題点に対する生の声を紹介しまとめにかえる。

8.1. ワークショップの目的

演習問題を実際に解決する過程を通じて、設計方法論、CASEツールに対する理解を深める。

8.2. 参加者のプロフィル

年齢30才以下、経験2年以上という制約がある。

(1) 参加の動機(45人) CASEツールへの興味：仕事上の必要性：上司の指示 = 24：16：23

(2) 日常の業務分野事務処理系：制御系：その他 = 20：14：13

(3) その中でのあなたの役割要求分析：基本設計：詳細設計：プロジェクト管理：その他 = 24：31：28：13：20

8.3. 日程

(1) 9月5日・方法論講義 SA/SD, OOA/OOD 各1時間 計2時間

・CASEツール機能紹介 各社30分 5社 計3時間… グラファギァ(FXIS),PWB(PFU),Teamwork(東陽テクニカ)、StP(SRA),CaseBench(ソニーテクトロ)

(2) 9月6日・問題説明とグループ分け 事務系、制御系各2題 5グループ

(グループの構成) 参加者9名, コーディネータ1名, インストラクタ1名, 記録者1名

・グループ毎のツール機能、操作法詳細説明 2時間

・問題解決 3時間

・中間報告会 2時間

(3) 9月7日 終日問題解決

(4) 9月8日 最終報告およびパネル討論 3時間

8.4. ワークショップの評価(参加者からのアンケート集計結果)

以下のようなアンケートを最終日、パネル討論中に実施した。

ワークショップを通じてCASEツールに対する理解・認識がどのように
深まったかについてお尋ねします。御自由にお書き下さい。

1. ワークショップを通じて得られたもの(成果)
2. 欲っしたが、得られなかったもの(今後への期待)
3. お使いいただいたCASEツールに対する要望
4. 全体的な感想

---

(1). 成果について

方法論について

- 抽象化する時の発想について多くのヒントを得た(OOA)
- オブジェクト指向が部品化、再利用に適していることがわかった
- オブジェクト指向が入門程度理解できた
- オブジェクト指向、分析の概観と理解の糸口がみえた
- 構造化設計、データ中心の設計の意義が見えてきた
- SAに従った分析の実体験(プロセス中心からデータ中心)
- SA/SDに対する深い理解が得られた
- SA/SDによるデータ中心の設計手法の理解
- SA/SDの考え方を具体的に知ったこと
- SA/SDを使った実際の体験(誤解に気づいた)
- SA/SD手法を使うと従来試験時に発生する多くの問題をつぶせる。しかし理解に多大の労力を要する
- 方法論が多少なりとも理解できた
- いかにすぐれたツールであってもその方法論を知らないと宝の持ちぐされである
- CASEツール使用にあたっての手法、方法論の必要性の認識
- 今後仕事を進めていく上で、なんらかの手法(法則)があることを知ったこと
- 今回のように、分析に時間をかけてじっくり作業を進める方法を体験できてよかった

ツールについて

- 作図、編集が簡単にしかもきれいにできるのは感動した
- ツールへの嫌悪感が取り去られた、と同時に夢も消えた
- CASE ツールへの興味を持つことができた
- 現状の CASE ツールの機能の一部の理解
- お絵書き機能の現状把握
- 上流の CASE ツールの一部に接し、その現状の一端をのぞきみたこと
- CASE ツールで何ができるかということについて実感した
- 汎用性を考えるとまだまだ有用でない

全般的に

- CASE は有効である
- CASE ツール以前に人的な要素が成否を握っている
- エンドユーザの立場からシステムを使うという貴重な体験をした
- 経験、馴れがいかに自分の頭を固くしているかがわかった
- 各ベンダーの考え方の違いがはっきりした (方法論、ツール)
- 色々なベンダーさんのアドバイスが聞けたこと
- 社風、技術等の問題で CASE ツールの導入は難しい
- こういう有意義な会合はもっとどんどんやるべきだ
- 自分たちの保守性や目さきの利益を見てしまう傾向に対する再認識
- 他の会社での CASE ツールの使用状況

交流について

- 他社の方の話を聞け、自社の立場、自分のレベルが認識できた
- 同年代の他社技術者との交流
- 同年代の人の考え方
- さまざまな人々との出合い (目ざすところは一つか)
- ヒューマンリレーション

(2). 欲っしたが得られなかったもの (今後への期待)

方法論関係

- OOA/OOD/OOP のすべてを経験しその上で、方法論やツールの有用性を確認したかった
- OOA/OOD 全体の経験
- OOA における抽象化手法の手がかり (トレーニングツール)
- SA/SD と OOA/OOD の関係に対する将来展望
- 現実にそった方法論

有効性

- 事務系を本格的にサポートする CASE を体験したい
- 事務処理に対するツールの有用性
- 方法論が本当に実務に役に立つのか (従来の手法より金になるのか)
- ノウハウについてまとめたものがあればもっと簡単にはいっていけるのではないか
- 大規模開発に CASE は有効か
- リバースエンジニアリングの現状やワークステション以外の環境への普及状況等に対する解説
- リバースエンジニアリングへの適用法
- 保守の支援
- ライフサイクルとの対応を明確に
- CASE ツールや方法論のすばらしさ
- CASE が過去に積み上げてきたものを破壊するための勇気と力をあたえてくれるだけのパワーがあればよいと思います
- ハードのアーキテクチャ、ソフト、コストのトレードオフで仕様がボンボンと変化するような場合、プロトタイピングを繰り返しながら仕様をまとめていく段階をサポートするものがなかった

導入法

- 導入にあたっての前提条件をしりたい。例えば、要求分析にあたっては純粋なユーザ、ユーザから SA になりつつある人、SE, プログラマに適するツールはどれか。方法論や記法の教育問題、前提とするコンピュータに関する知識等

- 実際に使用するためには、プロジェクトへの説明、理解が必要である、古い顔(上司)に理解させることが難しい。効果について、開発工数や保守工数に関する値がほしい。
- いかにして現場に定着させるか

(3). CASEツールに対する要望

ツールのサービス

- リバースエンジニアリング的アプローチ
- ツールオリジナルだけでは満足できる環境は作れない。CASEツールはカスタマイズのお手伝いをして下さい
- 方法論の支援が不十分
- 思考のサポート
- チームで分析設計する場合のネットワークのサポート
- ソフトウェアライフサイクル全体の支援
- ある手法にしばられないという姿勢を維持して欲しい
- 多機種対応

ツールの機能

- DFD内に記述した名前(データ名、プロセス名)を、キーボードからの入力フィールドにもってこれること
- DFD修正によるDDの自動修正機能
- 最適な形へのリフォーマッティング機能
- HELP機能の充実
- 独学するには多くのドキュメント、マニュアルが必要だ
- レビューをサポートする機能が欲しい
- アニメーションやナビゲションの機能
- 設計完了後、シミュレーションができること、とくに制御系のタイミング問題

ツールの操作性

- エデイタの機能アップ(カット&ペースト、アンドウ)
- エディタを強化し、わずらわしさを感じさせないようにして欲しい
- is-a,has-aの関係を示す線を引くときどうしても見づらくなる
- 画面の縮少機能が欲しい
- OSを意識させない操作性
- UNIXを意識せずに使用できるように
- UIをもっとオブジェクト指向にして欲しい
- UIがいまいち。操作の統一性があまりなく、モードが多用されすぎる。ただし、あまりツールがリコウだと人間がバカになるのでこの程度でよいか?
- UIMをものすごく充実させて欲しい(ハードのアーキテクチャのシミュレーション等)
- UIがもうすこし良ければよいのに
- 性能、UI(特に日本語)の面での改善が必要
- 日本語がいまいち
- 日本語入力の簡易化
- 文字入力がワープロに劣る
- もうすこしマウスだけで操作できるようにして欲しい

その他

- 数カ月試用したい
- 保守業務に利用できるのか
- 価格が高い

(4). 全体的な感想

自分自身にとって

- 課題に対する成果はともかくとして、この4日間でCASEを導入して実作業にいかせる気がした
- 貴重な経験、情報が得られた
- 参加してよかったが、方法論を十分に理解しないまま参加したのはよくなかった。方法論を理解していなければCASEツールは使いこなせない
- 環境がないものにとっては整えるのは大変だができればこれから使ってみたい
- 自分も含めて参加者がまだまだ未熟である

- この4日間で手法、ツールをマスターし会社にそのスキルを持ち帰ろうなどとは最初から期待していなかった。しかし今後の向学心、興味、理想追求のひとつの点火となればいいなと思っていた。今思えばこれも他力本願であった。やはり自分がまずやる気をおこさなければ何も良い結果はもたらされない
- SA/SD,OOA/OOD ど有用な考え方が提唱されているが自分自身に問題意識がなければ現状のまま推移していくだけである
- 第一線で開発、保守に従事している他企業の若い人と接し、その考え方問題意識について話をする場をもててよかった
- 方法論、ツール、記法の習得以上に社外技術者との意見交換ができたことが非常に有益な経験になった
- CASE ツールを活用するとしたら、システムライフサイクルとしての有効性に期待したい。分析設計の部分はもっと`あいまいな部分を残したまま作業ができるような形にして欲しい
- 要求分析や設計の支援だけでなく、現状システムの保守にも使えるように
- ハード、ソフトの値段が高すぎるので自分には関係ない
- 課題を分析する際の参考例が欲しかった (比較のため)
- 手法 (含 CASE) と宗教の類似性を感じた
- 構造化設計は従来通りの設計時間がかかる、しかし、テスト時間は短縮できると思う

参考文献

[1] A.I. Wasserman, "The Ecology of Software Development Environments", Sofutware Development Environments, 1981, pp.47-52.

[2] W.W. Agresti 編,"New Paradigms For Software Development", IEEE Tutorial, ORDER No. 707.

[3] P.T Ward,"The Transformaion Schema: An Extension of the Data Flow Diagram to Represent Control and Timing", IEEE Trans. on SE, Vol. SE 12, No.2, pp.198-210.

[4] 野村、落水、"ソフトウェア保守における開発時文書構成の問題点"、静岡大学大学院電子科学研究科研究報告第8号, 1987, pp.59-64.

[5] 野村、落水、中村、小川、"保守手順に適合する文書情報の編制法'、情報処理学会論文誌、Vol. 27, No.9, 1987, pp.934-941.

[6] G. Bruno, G. Marchetto,"Process-Translatable Petri Nets for the Rapid Prototyping of Process Control Systems", IEEE Trans. on SE, Vol.SE-12, No. 2, 1986, pp.346-357.

[7] J. R Cameron,"An Overview of JSD", IEEE Trans. on SE, Vol.SE-12, No. 2, 1986, pp.222-240.

[8] S. シュレイアー、S.J. メラー著、本位田、山口訳、「オブジェクト指向システム分析」、啓学出版、1990.

[9] 落水、春原、「ソフトウェア工学」、情報処理学会30年のあゆみ第7章、pp.205-211.

[10] 落水,"ソフトウェア開発環境の新しい動向", bit, Vol.21, No. 1, 1987, pp.78-88.

[11] 落水,"ソフトウェアプロセスモデルに基づくソフトウエア開発支援環境 Vela", 日本ソフトウエア科学会第7回大会論文集、pp.205-208,1990.

[12] P. Zave,"An Operational Approach to Requirements Specification for Embedded Systems", IEEE Tutorial ORDER No. 707, pp.159-178,1986.

[13] E.F. Berliner and P. Zave,"An Experiment in Technology Transfer: PAISley Specification of Requirements for an Undersea Lightwave Cable System", 9th ICSE, pp.42-50, 1987.

[14] R. Balzer et al, 'Operational Specification as the Basis for Rapid Prototyping', IEEE Tutorial, ORDER No. 707, pp.116-132, 1986.

[15] B. Curtis 編、"Human Factors in Software Development", IEEE Tutorial, ORDER No. 557.

[16] Elliot J. Chikofsky,"How to Lose Productivity With Productivity Tools", IEEE COMPUTER SOCIETY PRESS TECHNOLOGY SERIES, SOFTWARE DEVELOPMENT COMPUTER-AIDED SOFTWARE ENGINEERING(CASE).

13. 参加者名簿

第8回 夏のプログラミング ワークショップ

委員会およびスタッフ

No	氏 名	所 属	役 職
委員会			
1	野村 行憲	岩手電子計算センター	実行委員長
2	落水 浩一郎	静岡大学	プログラム委員長
3	天池 学	カシオ計算機	プログラム委員
4	荻原 剛志	大阪大学	プログラム委員
5	佐藤 千明	長野県協同電算	プログラム委員
6	佐原 伸	SRA	プログラム委員
7	菅野 卓矢	岩手電子計算センター	プログラム委員
8	太田 剛	静岡大学	プログラム委員
3	天池 学	カシオ計算機	プログラム委員
協力企業			
9	青木 淳	富士ゼロックス情報システム	インストラクタ
10	足立 太郎	PFU	インストラクタ
11	桜井 麻里	SRA	インストラクタ
12	田中 耕市	東陽テクニカ	インストラクタ
13	渡部 健次	ソニーテクトロニクス	インストラクタ
スタッフ			
14	小澤 一彰	静岡大学	記録担当
15	下津 直武	静岡大学	記録担当
16	関谷 和愛	静岡大学	記録担当
17	中尾 博司	静岡大学	記録担当
18	古川 勝也	静岡大学	記録担当
19	方 学芬	静岡大学	記録担当
20	山本 喜美代	SRA	事務局

一般参加者

No	氏 名	所 属	グループ
1	田口 秀成	新日鉄情報通信システム	OOA/OOD
2	阿部 和弘	富士ゼロックス情報システム	CASE Bench
3	佐藤 哲太	富士ゼロックス情報システム	Teamwork
4	井上 忠則	シスプラン	OOA/OOD
5	平尾 一浩	ヒラタ ソフトウェア テクノロジー	OOA/OOD
6	富永 伸哉	ヒラタ ソフトウェア テクノロジー	Teamwork
7	木村 茂和	日商エレクトロニクス	PWB
8	満田 豊	シーイーシー	StP
9	刀根 利光	シーイーシー	Teamwork
10	池川 哲夫	アスキー	PWB
11	西垣 郁夫	東京電力	StP
12	清水 洋子	東芝	StP
13	高橋 秀行	SRA	OOA/OOD
14	飯塚 宣男	カシオ計算機	PWB
15	大篭 高之	日本システム	Teamwork
16	藤原 千尋	日本ラッド	CASE Bench
17	藤井 康範	MHI エアロスペース システムズ	PWB
18	遠藤 晋	アスキー	OOA/OOD
19	泉田 昭義	岩手電子計算センター	CASE Bench
20	似内 均	岩手電子計算センター	PWB
21	三上 修	岩手電子計算センター	StP
22	佐藤 琢美	岩手電子計算センター	OOA/OOD
23	松本 幸三	CSK	OOA/OOD
24	青柳 和久	東北 SRA	PWB
25	思坊田 和典	日本電気ソフトウェア	CASE Bench
26	松浦 光哉	中央システム	CASE Bench
27	本間 匡（代理: 前 雄介）	中央システム	OOA/OOD
28	山島 利彦	日本エムアイシー	StP
29	浜 恭士	SRA	StP
30	林 明賢（代理: 中谷 信子）	SRA	Teamwork
31	上田 賢一	SRA	CASE Bench
32	館山 純	SRA	PWB
33	阪本 善通	SRA	StP
34	奥村 吉彦	CSK	PWB
35	Dabin Matthieu	静岡大学	OOA/OOD
36	渡辺 瑞枝（代理: 小沢 英志）	SRA	Teamwork
37	吉田 一知	岩手電子計算センター	Teamwork
38	斎藤 基	富士通ビーエスシー	CASE Bench
39	森 彰三	富士通ビーエスシー	Teamwork
40	知念 徹	三菱電気コントロール ソフトウェア	Teamwork
41	柳瀬 健一	ケーシーエス	StP
42	築山 秀夫	ケーシーエス	StP
43	財田 雅史	ケーシーエス	CASE Bench
44	小林 透	NTT	PWB
45	大岡 俊之	東北コンピューターサービス	CASE Bench

Call for Papers

WICS'92

Wulumuqi (烏魯木斉) International CASE Symposium

発表者およびパネル討論参加者募集

1992 年 10 月 7 日 ～ 9 日
於： 中華人民共和国 新疆省 烏魯木斉市

主 催
ソフトウェア技術者協会 (SEA) & 西北大学 (中国·西安市)
協 賛
復旦大学 (中国·上海市) 烏魯木斉市科学技術委員会
Rocky Mountain Institute of Software Engineering

SEA では，1987 ～ 88 の両年に，上海ソフトウェア·センタと共催で上海市および杭州市において「日中ソフトウェア·シンポジウム」を開催し，日中ソフトウェア技術者·研究者の間の人的·技術的交流を深めてきました．天安門での不幸な事件のために予定より 1 年遅れて 1990 年に大阪で開かれた第 3 回目のシンポジウムでは，この場を単に日中間に閉じるのではなく，より開かれた国際技術ミーティングに発展させようという合意が形成され，それに基づいて，昨年秋，北京大学において，広義の CASE をテーマとする国際シンポジウム BICS'91 (Beijing International CASE Symposium) が開催され，大きな成功をおさめました．

今年は，会場をシルクロード西端に位置する新疆省の首都·烏魯木斉に移し，遥かな古代における東西文化交流の歴史にロマンチックな夢を駆せながら，来るべき新しい世紀を支えるためのソフトウェア技術について討論したいと考えています．3 日間にわたるシンポジウムの運営形式は，昨年の BICS のそれを踏襲して，招待講演，論文発表，グループ討論，パネルをミックスしたかたちを想定しています．プログラム委員会で考えているセッションおよびグループ討論のテーマは次の通りです：

- ☐ Formal Method ☐ Object Orientation ☐ CASE Environment Architecture
- ☐ AI/KB vs Software Engineering ☐ Software Process ☐ Groupware/CSCW
- ☐ Oriental Philosophy & Software Engineering ☐ Silkraod of Software Engineering

そこで，下記の要領により，論文発表またはパネル討論参加者を募集します．ふるって御応募ください．

応募要領

応募論文 (Full Paper または Extended Abstract) またはパネルへのPosition Statement を **1992 年 5 月 31 日**までに，日本側プログラム委員長·岸田孝一 (SRA) 宛に郵便または Fax または E-mail でお送りください．応募論文は 7 月上旬のプログラム委員会で審査し，結果を 7 月下旬までに，応募者全員にお知らせします．審査を通過した方々には，最終論文（予稿集にのせるためのカメラ·レディ原稿）を 8 月末までに仕上げていただくことになります．

シンポジウム·スタッフ

実行委員長	艾克剛 (西北大学·学長)	
プログラム委員長	岸田 孝一 (SRA)	張 然 (復旦大学)
プログラム委員	荒木 啓二郎 (九州大学)	William E. Riddle (Software Design & Analysis)
	熊谷 章 (PFU & 富士通)	馮 玉琳（中国科学技術大） 張 校正（新疆水利庁）
	二木厚吉（電総研） 他	

***** 応募の宛先および応募用のカバー·シートは 次ページ にあります *****

WICS'92

論文応募カバー・シート

氏　名：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿（ふりがな）＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

会社（大学）：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

部門（学部）：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿役　職：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

住　所：（〒　　）＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

TEL: [　　－　　－　　内線　　] FAX: [　　－　　－　　]

E-Mail: ＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

論文タイトル（予定):＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

Abstract または Position Statement：（論文の要旨を 10 行以内でまとめてください）

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

＿＿

問題点と分野： あなたの論文は予定されているセッション・テーマのどれに関連していますか？ あてはまるものにチェックしてください。

☐ Formal Method　☐ Object Orientation　☐ CASE Environment Architecture
☐ AI/KB vs Software Engineering　☐ Software Process　☐ Groupware/CSCW
☐ Oriental Philosophy & Software Engineering　☐ Silkraod of Software Engineering

カテゴリ： Paper Session での発表またはパネル討論のいずれを御希望ですか？（いずれかにチェックしてください）

☐論文発表　☐パネル討論

---------- 送 付 先 ----------

〒160 新宿区四谷 3 - 12　丸正ビル 5F
(株)SRA ソフトウェア工学研究所
岸田 孝一
TEL: 03 - 3357 - 1507　FAX: 03 - 3351 - 0880
E-MAIL: k2@sra.co.jp

********** 応募締め切り： **1992 年 5 月 31 日** **********

ソフトウェア技術者協会 (SEA)

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

その主な活動は，機関誌SEAMAILの発行，支部および研究分科会の運営，セミナー／ワークショップ／シンポジウムなどのイベントの開催，および内外の関係諸団体との交流です．発足当初約200人にすぎなかった会員数もその後飛躍的に増加し，現在，北は北海道から南は沖縄まで，1000余名を越えるメンバーを擁するにいたりました．法人賛助会員も約80社を数えます．支部は，東京以外に，関西，横浜，長野，名古屋，九州の各地区で設立されており，その他の地域でも設立準備をしています．分科会は，東京，関西，名古屋で，それぞれいくつかが活動しており，その他の支部でも，月例会やフォーラムが定期的に開催されています．

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

なお，会費は次の通りです： 個人正会員－年間7,000円（入会金3,000円）．法人賛助会員－年間1口あたり100,000円（入会金なし，賛助会員会社の社員は各種イベントに正会員に準ずる割引価格で参加できます）．

今回のイベントを機会に，あなたにもぜひ会員になっていただきいと思います．下記の申込書に必要事項を御記入の上，郵便またはFAXで事務局までお送りください．折り返し会費納入の手続き等について御連絡をさしあげます．

代表幹事： 熊谷章

常任幹事： 市川寛 落水浩一郎 岸田孝一 中野秀男 深瀬弘恭 松原友夫 山崎朝昭

幹事： 飯沢恒 筏井美枝子 上原憲二 臼井義美 奥山充 片山卓也 亀田繁 窪田芳夫 小林俊明
杉田義明 田中一夫 玉井哲雄 鳥居宏次 中谷多哉子 中山照章 野村敏次 野村行憲 浜野善吉
林香 平尾一浩 藤野晃延 二木厚吉 堀江進 本位田真一 盛田政敏 山崎利治 渡邊雄一

会計監事： 辻淳二 吉村成弘

分科会： 環境分科会 (SIGENV) 管理分科会 (SIGMAN) 教育分科会 (SIGEDU) ネットワーク分科会 (SIGNET)
法的保護分科会 (SIGSPL) 調査分科会 (SIGSURVEY)

SEA入会申込書

.....

正会員（入会金3,000円，年会費7,000円） 92-03

氏 名：_____ (ふりがな：_____)

生年月日：_____年_____月_____日 性別（男 女）血液型（A O B AB）

勤務先名：_____

所属・役職：_____

勤務先住所：（〒 ）_____

勤務先TEL：_____-_____-_____（内線_____）

勤務先FAX：_____-_____-_____

自宅住所：（〒 ）_____

自宅TEL：_____-_____-_____

連絡先（どちらかにチェック） □勤務先 □自宅

.....

賛助会員（年会費1口100,000円，何口でも可） 92-03

会社・団体名：_____

代表者氏名：_____ (ふりがな：_____)

連絡担当者：_____ (ふりがな：_____)

所属・役職：_____

住 所：（〒 ）_____

TEL：_____-_____-_____（内線_____） FAX：_____-_____-_____

申込口数：_____口

.....

申し込み先

〒160 東京都新宿区四谷3-12丸正ビル5F ソフトウェア技術者協会（TEL 03-3356-1077, FAX 03-3356-1072）

SEA 海外研修ツアー

The 4th International Symposium on Future Software Environments

(May 7 - 8, 1992: Sydney, Australia)

14th International Conference on Software Engineering

(May 11 - 15, 1992: Melbourne, Australia)

－ 参 加 者 募 集 －

主催: 近畿日本ツーリスト（株） 新宿南口支店

企画: ソフトウェア技術者協会（SEA）

恒例のソフトウェア工学国際会議(ICSE-14)が，今年はオーストラリアのメルボルンで開催されます．オーストリアでの開催とあって、キーノーターの一人に Queensland 大学の Lister 教授が顔を出すなどしており、興味を引かれるものがあります。また，ICSE の直前に ISFSE-4 (The 4th International Symposium on Future Software Environments) が，シドニーで行われます。ISFSE とは、これからのソフトウェア開発環境のあり方について，ワークショップ形式で討論しようというもので、米国、日本で行われていましたが、今回 ICSE にあわせて、オーストラリアで開催されることになりました。皆さんの希望にあわせられるように、A コース (ISFSE & ICSE 参加)、B コース (ICSE のみ参加) の２つのコースを用意しました。実施要領は以下に示す通りです．多くの方々の参加をお待ちします．

(1) 日 程

A コース		B コース
5/6(水)	成田 -> シドニー	
5/7(木)	ISFSE 参加	
5/8(金)	ISFSE 参加	
5/9(土)	技術訪問	成田 -> シドニー -> メルボルン
5/10(日)	シドニー -> メルボルン	自由行動
5/11(月)	ICSE Tutorial 参加（オプション）	同左
5/12(火)	ICSE Tutorial 参加（オプション）	同左
5/13(水)	ICSE 参加	同左
5/14(木)	ICSE 参加	同左
5/15(金)	ICSE 参加	同左
5/16(土)	メルボルン -> シドニー	同左
5/17(日)	シドニー -> 成田	同左

(2) Symposium/Conference Program

概要は別添のパンフレットをご覧下さい．参加を申込まれた方には，後ほど詳細な Advance Program をお配りします．

(3) 参加費： 裏面の申し込み用紙の参加費の欄を御参照ください。

コース別、会議へのエントリーの代行有無、ICSE Tutorial 参加の有無により参加費が異なっています。ただし，それは 15 名以上の参加申込を前提とした料金です．これには，航空運賃（エコノミー・クラス），バス代，ホテル代（2 名 1 室）が含まれています．個室を希望される場合には追加料金（A コース:76,000 円、B コース:63,000 円）が必要になります。

(4) 申込み方法:

裏面の申込書に必要事項をご記入の上，郵便または FAX で近畿日本ツーリストへお送り下さい．

申込みおよび問い合わせ先:

〒 160 東京都新宿区西新宿 1 - 18 - 5 甲新ビル

近畿日本ツーリスト（株）新宿南口支店 （担当：亀田）

TEL: 03 - 3342 - 0751 FAX: 03 - 3342 - 0806

SEA 海外研修（*ISFSE/ICSE*）ツアー 参加申込書（申込日：_____ 月 _____ 日）

氏名

（日本語）---------- （英語）----------

会社（学校）名

（日本語）---------- （英語）----------

部門名

（日本語）---------- （英語）----------

役職

（日本語）---------- （英語）----------

住所

（日本語）〒----------

（英語）----------

TEL：_____ - _____ - _____（内線 _____ ）

FAX：_____ - _____ - _____

.....

参加費

A コースか *B* コースを選択の上、ご希望の項目にチェックをして合計金額をご記入下さい。

A コース

料金種別	金額（円）	チェック欄
基本料金：航空運賃，バス代，ホテル代（2名 1室），*ISFSE* 参加費 *(65,000* 円*)* を含んでいます	*465,000*	
ICSE 参加申し込み代行：会議，*WELCOME RECEPTION*，*CONFERENCE BANQUET* の参加費を含んでいます。*ICSE* には色々なオプションがあります。ご希望の方は参加要項を取り寄せ、御自分でお申し込み下さい。	*80,000*	
個室希望	*76,000*	

B コース

料金種別	金額（円）	チェック欄
基本料金：航空運賃，バス代，ホテル代（2名 1室）を含んでいます	*350,000*	
ICSE 参加申し込み代行：会議，*WELCOME RECEPTION*，*CONFERENCE BANQUET* の参加費を含んでいます。*ICSE* には色々なオプションがあります。ご希望の方は参加要項を取り寄せ、御自分でお申し込み下さい。	*80,000*	
個室希望	*63,000*	

Tutorial

参加申し込みの代行を希望される方で、かつ *Tutorial* 参加も希望される方のみご記入下さい。御自分で *ICSE* に申し込まれる方は、*Tutorial* の申し込みも御自分でお願い致します。別添のパンフレットを参照の上、*ICSE* の *Tutorial* を選び番号とタイトルをご記入下さい。1日に1つの *Tutorial* にしか参加できません。

日付	番号	タイトル	金額
11 MAY			*40,000*
12 MAY			*40,000*

参加費合計 ---------- 円

International Symposia on Future Software Environments

7-8 May 1992

Sydney, Australia

****

Sponsors:

Rocky Mountain Institute of Software Engineering (Boulder, Colorado USA)
Software Designer's Associates Consortium (Tokyo, Japan)
Software Engineers Association (Tokyo, Japan)

Purpose:

The International Symposia on Future Software Environments have been organized to provide an interactive forum for discussion of leading edge topics in the area of software environments.

Structure:

Each symposia is structured into two days of four sessions each. Invited presentations are given at the beginning of each session to set the stage for a focused discussion of concrete issues. The second half of each session is devoted to open discussion of the issues.

The focus of the sessions varies from symposium to symposium.

A typical set of session topics is: process issues, overall system issues, environment issues, and supporting mechanisms.

Participants:

Attendance is limited to approximately 40 people.

All participants are expected to actively engage in the discussions.

ISFSE-4

The 4th International Symposium on Future Software Environments will be held in Sydney, Australia, 7-8 May 1992.

This is just before the 14th International Conference on Software Engineering that will be held in Melbourne the following week.

The focus for ISFSE-4 will be Environment Evolution.

The sessions will address:

- in what ways do software processes evolve?
- in what ways do software environments evolve?
- what are the requirements and design constraints upon software environments to support this evolution?
- what environment componentry and environment building technology is needed to meet these requirements design constraints?

Fee:

The fee for ISFSE-4 is ¥65,000.

Attendees are expected to pay their own travel, lodging and meal expenses.

14th International Conference on Software Engineering

11-15 May 1992

Melbourne,Australia

****

TUTORIAL PROGRAM:

MONDAY 11 MAY 1992

1: Software Reliability Engineering *(John D. Musa)*

2: State-of-the-Art in Environment Support for Configuration Management *(Susan Dart and Peter Feiler)*

3: An Introduction to Computer Security *(Richard A. Kemmerer)*

4: CASE Tools for Requirements Engineering and Software Design *(John W. Brackett)*

TUESDAY 12 MAY 1992

5: Applying Total Quality Management(TQT) to Software Systems *(Michael S. Deutsch)*

6: Software Re-engineering *(Robert S. Arnold)*

7: Exception Handling, or "When Bad Things Happen to Good Programs" *(Chuck Howell)*

8: Object Management for Software Environments *(Jack C. Wileden)*

CONFERENCE PROGRAM:

WEDNESDAY 13 MAY 1992

KEYNOTE ADDRESS

Moving Toward Safeware: How Do We Get There From Here?

PROF.N. LEVESON, University of California, Irvine(USA)

SESSION

MEASUREMENT AND PREDICTION

SOFTWARE ANALYSIS AND TESTING

ENVIRONMENT SUPPORT

THURSDAY 14 MAY 1992

KEYNOTE ADDRESS

Trusted Computing Systems: The ProCos experience

-PROG.D.BJORNER, United Nations University - IIST MACAU

SESSION

SOFTWARE EVOLUTION

REAL-TIME SYSTEMS

FORMAL METHODS

STATE-OF-THE-ART REPORT

FRIDAY 15 MAY 1992

KEYNOTE ADDRESS

Design of Dependable Real-Time Systems

PROF.ANDREW LISTER, University of Queensland, Australia

SESSION

SOFTWARE PROCESSES

TOOLS

SOFTWARE RE-USE

AWARD: Best paper of ICSE 4. STATE-OF-THE-ART lecture by recipient

Programming-in-the-Large: Past, Present and Future,

W.TICHY, University of Karlsruhe

ソフトウェア技術者協会
〒160 東京都新宿区四谷3-12 丸正ビル5F
TEL.03-3356-1077　FAX.03-3356-1072