

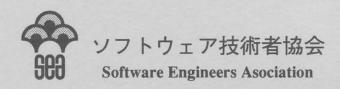
Newsletter from Software Engineers Association

Volume 8, Number

11-12 April, 1994

目 次

編集部から		1		
題 11 回 夏のプログラミング・ワークショップ報告		3		
まえがき	中来田 秀樹	4		
プログラム		5		
参加者名簿		6		
ディベートの報告		7		
はじめに	大塚 理恵	8		
オブジェクト指向	広沢 博	10		
プロジェクト管理	外岡 徹	12		
CASE	棚橋 巡治	21		
開発プロセス	一原 武司	28		
ポジション・ステートメントと感想		31		
SEA Seminar & Forum November'93 の記録				
セミナー		62		
パネル討論		64		
Call for Participation/Papers		34		
Kunming International CASE Symposium		2		
ソフトウェア・シンポジウム'94		79		
First World Congress for Software Quality		83		



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、900 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 40 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、東北の各地区で設立されており、その他の地域でも設立準備をしています。分科会は、東京、関西、名古屋で、それぞれいくつかが活動しており、その他の支部でも、月例会やフォーラムが定期的に開催されています。

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

代表幹事: 中野秀男

常任幹事: 岸田孝一 熊谷章 玉井哲雄 深瀬弘恭 堀江進 山崎利治

幹事: 筏井美枝子 市川寛 伊藤昌夫 臼井義美 大塚理恵 大場充 菊地俊彰 君島浩 窪田芳夫 小林俊明

坂本啓司 杉田義明 武田淳男 田中一夫 鳥居宏次 中来田秀樹 中谷多哉子 西武進 野村敏次

野村行憲 盛田政敏 平尾一浩 藤野晃延 二木厚吉 松原友夫 山崎朝昭 渡邉雄一

会計監事: 辻淳二 吉村成弘

分科会世話人 環境分科会(SIGENV):田中慎一郎 渡邊雄一

管理分科会(SIGMAN):野々下幸治

教育分科会 (SIGEDU): 杉田義明 中園順三

ネットワーク分科会 (SIGNET): 大塚理恵 小林俊明 人見庸

調查分科会 (SIGSURVEY): 岸田孝一 野村敏次

支部世話人 関西支部:臼井義美 中野秀男 盛田政敏

横浜支部:藤野晃延 北條正顕 野中哲 松下和隆

長野支部:市川寛 佐藤千明

名古屋支部: 筏井美枝子 鈴木智 平田淳史

九州支部:平尾一浩

東北支部: 菊地俊彰 和田勇

贊助会員会社:NTTソフトウェア研究所 NTT九州技術開発センタ PFU SRA アスキー

エスケーディ オムロンソフトウェア カシオ計算機 キャノン新川崎事業所 さくらケーシーエス

サン・ビルド印刷 ジェーエムエーシステムズ ジャストシステム

セントラル・コンピュータ・サービス ダイキン工業 テクノバ ニコンシステム

ニッセイコンピュータ ムラタシステム リコーシステム開発

安川電機 古河インフォメーション・テクノロジー 構造計画研究所 三菱電機セミコンダクタソフトウェア

三菱電機メカトロニクスソフトウェア 三菱電機関西コンピュータシステム

新日鉄情報通信システム 新日本製鉄エレクトロニクス研究所 池上通信機 中央システム

辻システム計画事務所 東芝アドバンストシステム 東電ソフトウェア 東北コンピュータ·サービス

SRA東北 日本NCD 日本ユニシス・ソフトウェア 日本情報システムサービス

日本電気ソフトウェア 日立エンジニアリング 富士ゼロックス情報システム 富士写真フィルム

富士通 富士通エフ・アイ・ピー オムロン (以上44社)

SEAMAIL Vol. 8, No. 11-12 1994年4月11日発行 編集人 岸田孝一

発行人 ソフトウェア技術者協会 (SEA)

〒160 東京都新宿区四谷3-12 丸正ビル5F

TEL: 03-3356-1077 FAX: 03-3356-1072

印刷所 サンビルド印刷株式会社 〒162 東京都新宿区築地町8番地 定価 1,000円 (禁無断転載)

編集部から

公

Vol.8 の最終号をお届けします.

公公

この編集前記を書いているのは、3月の最終土曜日、実は、今月は年度末で忙しいので、この号は4月に入ってからゆっくり編集しようと考えていたのですが、木曜日の Seminar & Forum のあとの幹事会の席上で、大塚理恵さんから、昨秋の「若手の会」のレポートが、ドカンと完成原稿で手渡されたのです。

2 2 2

そこで、予定を急きょ変更して、とりあえず、このワークショップ報告特集で Vol.8 をめでたく打ち上げにし、いま手持ちの原稿を Vol.9, No.1 にまとめようということになりました。 大塚さん、またその他の PC メンバのみなさん、ごくろうさまでした。

公公公公

去年の若手の会は、初めてディベイト形式を採用するということで、準備その他も大変だったようですが、折りあしく「戦後最大級」とかいう振れ込みの大型台風が会期中に北陸地方を襲撃しそうになり、そのため予定を1日切り上げて終了するというアクシデントに見舞われました。

4444

で、相談の結果、その分の予算の余りを利用し、台風の影に吹き飛ばされた「フォーマル・メソッド」のチュトリアルを、Seminar & Forum の形で、11 月に東京で開催しました。このレポートには、そのフォーラムの記録も収められています。

ところで、今年の若手の会は、開催場所を 3年ぶりに、なつかしい盛岡に戻し、CSCW またはグループウェアをテーマに開催すべく、いま、準備が進行中です。実行委員長は、筏井美枝子・野村行憲のお 2人、プログラム委員長は熊谷章さん、桜の花の散るころには、Call for Paper が出るでしょう。

Preliminary Call for Papers Kunming International CASE Symposium '94

November 16-19, 1994 Kunming City, Yunnan Province, China

Sponsored by SEA(Japan) and UNU/IIST(Macau)

KICS'94 is the 4th International CASE Symposium in China organized by Software Engineers Association of Japan in cooperation with several academic institutions; the 1st was in Beijing '91, the 2nd was in Urmuqi '92, and the 3rd was in Taian'93.

The symposiums brought together software practitioners, researchers, managers, and educators, from a variety of organizations across the countries to examine the state of software engineering technology and practice in light of CASE in a broader sense.

All attendees will have the opportunity to actively participate: exchange ideas, obtain information, and gain insight about software engineering activities across the countries. We are anticipating an exciting experience. Participation is open to industry, government, and academia. We welcome and encourage your participation.

In this year's KICS'94, we are planning to focus our main attention on the idea of "Domain-Orientation" in a boarder sense. All participants are expected to explain how he/she apply his/her research/technique/ experience to domains. Another typical question expected to be answered might be "whether his/her research/technique/experience is domain dependent or independent, and why it is so?".

All participants should submit full paper or extended abstract to the either of program cochairs by July 15th. Authors will be notified by the end of August whether their paper is accepted or not. Someone may send position statement for panel sessions instead of paper. All submissions are expected to clarify topics which you are interested in. The following list gives the typical candidates of topics.

Topics:

- Domains Analysis
- · Object Orientation and Domains
- · Domain Oriented Architecture
- · Domain Oriented Software Process
- · CASE and Domains
- Domain Orientation and Independence

CALENDAR:

Paper deadline:

July 15, 1994

Acceptance notified:

August 31, 1994

ORGANIZING COMMITTEE:

General Chair:

Kouichi Kishida (SRA, Japan)

Program Co-Chairs:

Huowang Chen (CIT, China)

PC Members:

Keijiro Araki (NAIST, Japan)

Kegang Hao (North-West Univ, China) Kyochul Kang (POSTECH, Korea) Tomoo Matsubara (Peopleware, Japan)

Toshiharu Yamasaki (Free, Japan)

Treasurers:

Yoshiaki Sugita (Nihon NCD, Japan)

Jingshan LO (Kunming Univ, China)

Kokichi Futatsugi (JAIST, Japan) Dines Bjorner (UNU/IIST, Macau)

Dehua Ju (ECUST, China) Akira Kumagai (PFU, Japan) Baile Xi (Fudan Univ, China)

Ran Zhang (Fudan Univ, China)

PAPERS SUBMITTED TO:

Local Arrangement:

Huowang Chen

Department of Computer Science Changsha Institute of Technology

Changsha, Hunan 410073, China Tel: +86-731-443-4601 ext 3021 Fax: +86-731-442-9355

Kokichi Futatsugi

Japan Advanced Institute of Science and Technology Tatsunokuchi-machi, Ishikawa 923-12, Japan

Tel: +81-761-51-1255 FAX: +81-761-51-1116 E-mail: kokichi@jaist.ac.jp

第11回

SEA夏のプログラミング・ワークショップ **ダウンサイジング時代における未来を求めて** - ダウンサイジング時代の重要性を探る! -

報告書

1993年9月1日(水)~9月3日(金) 於:石川県教育会館(石川県金沢市)

ソフトウェア技術者協会

まえがき

"第11回 夏のプログラミング・ワークショップ"のプログラム委員を 私が担当することになり、一番の問題はこの不況の中で色々な会社の若 手の人達を集めるにはどうすれば良いのか?と言うことで悩みました。

日本の一般社会生活で馴染みの少ないディベート形式で最近の話題のタイトルを使い、若者がはたして集まってくれるか? 出来る限り楽しく、有意義な時間を過ごしてもらおうとプログラム委員全員でタイトルを決め、いざ、実行してみますと観測史上最大の台風の上陸と言うアクシデントにもめげず、ディベート中は各人の個性が出てきてするどく突っ込む人あり、突っ込まれる人ありで、ディベートの重要性、各ディベートの内容の勉強不足を感じてもらえたと考えています。

実行委員長 中来田 秀樹 (ネクストファウンデーション)

プログラム

9月1日(水)

12:30 集合(石川県教育会館)

12:30~14:30 参加者自己紹介

14:30~17:15 ディベートの説明(大塚 理恵)

PCによるチュートリアル

オブジェクト指向(中谷 多哉予

プロジェクト管理(佐原 伸)

CASE(東田 雅宏)

開発プロセス(伊藤 昌夫)

17:15~18:00 移動、チェックイン (ホテル坂)

18:00~21:00 懇親会(しゃもじ屋)

21:00~ 作戦会議 (グループ毎)

9月2日(木)

9:00~13:00 作戦会議 (グループ毎) 、昼食

13:00~14:30 ディベート:オブジェクト指向

14:30~16:00 ディベート: プロジェクト管理

16:00~17:00 招待講演 「ISO 9000-3をめぐって」松原 友夫氏

18:00~ 作戦会議、食事 (グループ毎)

9月3日(金)

9:00~13:00 作戦会議 (グループ毎)、昼食

13:00~14:30 ディベート: CASE

14:30~16:00 ディベート: 開発プロセス

◇台風接近のためディベート終了後、会を解散しました。

 \Diamond 9月4日午前中に「フォーマルメソッド」に関するチュートリアルとパネル討論を予定していましたが、'94年11月27日にSEA Seminar & Forumとして開催されました。

第11回SEA夏のプログラミング・ワークショップ

参加者名簿

No.	氏名	会社	ク゛ルーフ゜	備考
I	中来田 秀樹	ネクストファウンデーション	D	実行委員長
II	大塚 理恵	RSK	D	編集委員
III	伊藤 昌夫	MHIエアロスペースシステムズ	A	プログラム委員
IA	菊地 俊彰	東北電力	В	プログラム委員
V	佐原 伸	SRA	С	プログラム委員
VI	中谷 多哉子	富士ゼロックス情報システム	A	プログラム委員
VII	東田 雅宏	PFU	В	プロデラム委員
VIII	松原 友夫	Office Peopleware		オフ゛サ゛ーハ゛ー
IX	山崎 利治	フリー		オブ゛サ゛ーハ゛ー
0 1	藤本 泰	中電技術コンサルタント	Α	
0 2	大野 修一	日本NCD	В	
0 3	中尾 吉克	高岳製作所	A	
0 4	坂口 達司	高岳製作所	С	
0 5	高村 智子	山一情報システム	Α	
0 6	堀井 剛	シーズ・ラボ	С	
0 7	根岸 易世	ニコンシステム	В	
0 8	一原 武司	ジェーエムエーシステムズ	D	
0 9	伊藤 武司	システム・エンジニアリング・サービス	D	
10	丹羽 徹	オムロン	В	
1 1	広沢 博	オムロン	A	
1 2	菊地 義幸	フリー	D	
1 3	棚橋 巡治	SRA	В	
14	外岡 徹	SRA	С	
1 5	沖 さとみ	MH I エアロスペースシステムズ	С	
1 6	加藤 昌也	MH I エアロスペースシステムズ	A	
1 7	荻原 郁子	SRA	В	
18	神崎 秀樹	アドバンスト・ソフトウェア・インスティチュート	С	
19	横井 伸司	日本アイ・ビー・エム	D	
20	芳里 嘉昭	日立ソフトウェアエンジニアリング	D	

第11回SEA夏のプログラミング・ワークショップ 若手の会ディベートの報告

はじめに オブジェクト指向 プロジェクト管理 CASE 開発プロセス

若手の会ディベート はじめに

大塚 理恵(RSK)

今回の若手の会では、グループ討論の発表をディベートで行いました。

ここではディベートについて、開催までの準備、 ルール、結果などを報告します。

1. 事前準備

(1)論題を決める

応募のポジションステートメントに、

- オブジェクト指向
- フォーマルメソッド
- 開発プロセス
- CASE
- プロジェクト管理
- · QC, TQC

の中から関心のあるもの2つ選び、日頃思っているこ おく。これは肯定側、否定側とも同じ。 とを述べてもらう。この2つがディベートするときの 分野の候補となる。 立論 1名

一方、プログラム委員会で論題を決め、それぞれど の分野に該当するか検討した。

論題は

● 開発標準は悪である

(開発プロセス)

SEは会社に出る必要はない

(プロジェクト管理)

オブジェクト指向は悪である

(オブジェクト指向)

● 上流CASEは必要である

(CASE)

を用意した。

現地での作戦会議の前、「開発標準は悪である」で は議論するのに範囲が広すぎるので、

大規模システム開発において 開発標準はいらない

と変更した。

(2)対戦方法を決める

ディベート1試合に講評までいれると1時間弱かかりそうなので、一つの論題に対し肯否入れ替えることなく、各論題とも1試合のみとした。

時間を考慮し全部で4試合、応募者数20名を4グループとし、1グループあたり2論題とした。 どのグループも両日試合がある。 応募状況を見て次のような分野の組み合わせを考えた。

- オブジェクト指向とCASE
- CASEとプロジェクト管理
- オブジェクト指向と開発プロセス
- 開発プロセスとプロジェクト管理

(3)参加者への事前連絡

現地で初めてディベートの論題を知らされ、短い時間にまとめるのは難しいので、約1ヶ月前、参加者には受け持つディベートの論題を連絡した。

このとき参考図書のリストを添付した。 (数名は下調べに利用された模様。)

また自分が肯定側になるか否定側になるかは、**準備**中は両方検討してほしいので、現地で決定すると伝えている。

2. ディベートの形式

(1)試合での役割

次のような発表のときの役割を作戦会議中に決めて おく。これは肯定側、否定側とも同じ。

 立論
 1名

 反対尋問
 2名

 反駁(最終弁論)
 1名

 記録係
 1名

(2)試合の流れ

判定+講評

次のような流れで行なった。[]内は持ち時間。

肯定側立論		[4分]
否定側立論		[4分]
作戦タイム		[1分]
第1否定側反対尋問	(否定→肯定)	[2分]
作戦タイム		[1分]
第1肯定側反対尋問	(肯定→否定)	[2分]
作戦タイム		[1分]
第2否定側反対尋問	(否定→肯定)	[2分]
作戦タイム		[1分]
第2肯定側反対尋問	(肯定→否定)	[2分]
作戦タイム		[2分]
否定側反駁		[2分]
肯定側反駁		[2分]

(3) 各役割について

し、論題支持を主張する。

今回の論題には「悪である」とか「いらない」とい のデータを利用するグループがあった。 う否定的なものがあり、論題の支持も混乱しがちだっ た。

反対尋問は、相手側立論者に対して疑問点、不明な 点を質問する。

第1否定側反対尋問では否定側から肯定側立論者に 質問する。ルール上、肯定側立論者が応えられないと きは他のメンバーが代弁することを認めた。(否定側 の質問についても同様。)

反駁 (最終弁論) では尋問等で明らかになった相手 の議論の間違いを証明し、自分の主張をまとめ論点を 繰り返す。

若手の会では議論を活発に行ないたいと考えたの で、反対尋問を2回、反駁を1回とし、作戦タイムの 回数も多くとっている。

記録係は討論の内容を報告書にまとめる。

判定と講評はプログラム委員(PC)が行い、立論、 **尋問などの役割や総合的印象について5段階評価で判** 定した。

(4) 発表資料

各グループ、模造紙2枚程度に、論題と現状分析、 結論などをまとめ、試合前に壁に貼る。

事前に参考資料として配布していた「ソフトウェア 立論は、論題に対する自グループでの問題分析を通 開発プロセスの実態調査(ソフトウェア・シンポジウ ム'92)」のデータや、PCが行ったチュートリアルで

3. 作戦会議

作戦会議に入る前、対戦相手同士で肯定、否定を決 定した。

各グループには1、2名のPCが入り、論題につい て議論する。試合は午後からで、それまで場所は自 由。

4. 本番

下の表が各グループでの役割と判定(得点)である。 結果として若手の会では

- SEは会社に出る必要はない
- オブジェクト指向は悪である
- 上流CASEは必要である
- 大規模システム開発において

開発標準はいらない

という論題すべて肯定側で主張したグループが勝っ

★続いて、実際に行われた4つのディベートの報告 を各グループの記録係にしていただきます。テープか らの採録です。

役割と判定

実施日	9月2日				9月3日			
論題	オブジェク である	ト指向は悪	SEは会社にない	出る必要は	上流CASEは	必要である	大規模シス 開発標準は	テム開発に いらない
グループ	В	Α	D	С	В	С	Α	D
肯定/否定	肯定	否定	肯定	否定	肯定	否定	肯定	否定
立論	荻原	高村	菊地	神崎	大野	外岡	高村	芳里
反対尋問1	棚橋	加藤	芳里	堀井	根岸	坂口	加藤	菊地
反対尋問2	丹羽	中尾	一原	神	荻原	神	中尾	横井
反駁	根岸	広沢	横井	坂口	棚橋	神崎	広沢	伊藤
記録	大野		伊藤	外岡	丹羽	堀井		一原
担当PC	東田 菊地	中谷 伊藤	中来田 大塚	佐原	東田 菊地	佐原	中谷 伊藤	中来田 大塚
判定	94	86	100. 5	100	110	76	95	87

オブジェクト指向 論題

「 オブジェクト指向は悪である 」

記録 広沢 博(オムロン)

◎肯定側立論

定義:

オブジェクト指向は私達をだましているということ が悪であると定義。

理由:

多くの書籍の中には、ブジェクト指向は柔軟性があると記述されているが、実際にはそうではない。

また、再利用性があるといわれているが、そうではない。〇〇でいうところの、閉じた再利用、開かれた再利用はできるかも知れないが、一般的にいうところの再利用はできないのではないか。

さらに、オブジェクト指向は万能であるといわれているが、それも嘘である。

それに加えて、クラスを生成できる人も非常に少な い。

論拠:

柔軟性があるのは、クラスライブラリを新規構築するときのみである。もし、下層のサブクラスで不具合が発生した場合、それがスーパークラスに影響を与えるものであれば、構造ががんじがらめになっている。

再利用性を調べてみたところ、1割にも満たない。

クラスライブラリとして発表されているもので成功 しているものは、画面管理のものだけである。という ことは、すべての物をオブジェクト指向で捉えるの は、無理があるのではないか。

クラスを生成する人が少ないのは、それに対応した 思考が必要であり、そのためには訓練が必要である。 そうした、思考を身につけることがむかない人もいる。

このような点から見ると、オブジェクト指向を用いて新しく行っていくと、うまくいく、再利用ができる、柔軟性があるといって、私達に過剰な期待をさせているが、実際にはその期待に答えていない。

だから、オブジェクト指向は悪であると肯定した。

◎否定側立論

GUI構築に便利

分析設計に有利

保守に強い

という点から、論題を否定する。

理由と論拠:

GUI構築に便利なのは、メッセージパッシングの機構を持っていることである。具体的な例としては、ウィンドウ画面のボタンにあらわれている。つまり、ウィンドウ・ライブラリ等で実際に使われ有利性が認められている。

分析設計に有利というのは、要求分析によって開発 者とエンドユーザーのギャプを小さくできるというこ である。

これは、実際のプロジェクトで使用されている。今までの場合、場馴れした開発者がエンドユーザーに理解できるかたちで説明していたが、オブジェクト指向を用いた場合、現状の業務をオブジェクト化していくことで、開発者とユーザーとの間で言葉があうことが利点である。

保守に強いというのは、従来の方法のプログラムでは、データと機能の配置がばらばらである。しかし、オブジェクト指向を用いると、データと機能の配置が整理され、相互の関係が明確になる。この事は、設計の途中段階だけではなく、後の追加や保守を行う作業でも有用である。実際に過去の類似したプロジェクトをオブジェクトして捉えた場合、明確な追加および保守が行える。

以上の点から、論題を否定する。

◎否定側反対尋問1

否定側:再利用性がないと言われると同時に、論拠でGUIでは実際に再利用されていると言われましたが、それは再利用されていることになるのではないか。

肯定側:私達の主張としては、再利用が成功しているのはGUI関連しかないのではないかということであり、それが1割である。とすれば、万物に有効であると言われるオブジェクト指向が、1割しか再利用できないというのは、私達を騙しているとしか言えない。

否:それはものの見方だと考える。1割ということは、10個作成したもののうち1個は再利用できるということである。この面からすると十分再利用性があると言えるのではないか。

肯:何割できれば再利用性があるという定義は人に より異なる。しかし、再利用性が高いといわれて私達だからオブジェクト指向は無い方がよいと理解してよ が受ける印象は1割では少ない。事実その数値を聞い てがっかりする人が多い。

実際のプログラムで再利用性が1割と聞いた場 合、それなりの評価がされるのか。

否:現状1割もできてないと言われているのだか ら、それに比べれば十分な値と考える。

肯:そうは思わない。

(時間切れ)

◎肯定侧反対尋問1

肯:まず、再利用が1割ではなく、資料では3割で あったことを訂正しておく。

さて、要求分析において、開発者とエンドユー ザーとのギャップを小さくできるとあるが、これはプ ロトタイピングのような話も含むのか。

否:はい。

肯:では、要求分析のギャップは小さくなったとし て、要求分析から下流工程に移す部分のギャップはど うしら良いのか。

否:ギャップがあるとの認識での質問か。

肯:あると思うので、どうするのか。

否:ギャップは無いと思うので、その点については 回答できない。

肯:ギャップが無いというのは、オブジェクト指向 の言語を分析のアプローチに使うからこそ無いので あって、現状の言語でプログラムされたものの中に利 益を出すものがあると思われるか。

否: 先程の具体例のなかで、実際に上流をオブジェ クト指向で、下流をC言語で行ったものがある。

肯:それは、努力でギャップを埋めたのではない か。

(時間切れ)

◎否定側反対尋問2

否:定義に騙しているから悪であるとしているが、 いか。オブジェクト指向は存在してもよいのか。

肯:他にオブジェクト指向より優れた設計方法があ れば、いらないと考える。しかし、現状の各種設計方 法の中で、どれが良いとは言えない。だから、例え悪 であっても、存在してはいけないとは言えない。

否:では、訓練が必要であるとあるが、これは他の 手法の初期導入時にでもおこることであり、オブジェ クト指向に限ったことではないではないか。

肯:確かに限ったことではない。だが、オブジェク ト指向の売りものはクラスライブラリを構築すること にある。このとき、良いクラスライブラリを作るには 抽象化がきれいに行われなければならない。そうする ことによって、初めて再利用性が出てくる。

否:と言うことは人の問題ということになるのか。 (時間切れ)

◎肯定側反対尋問2

肯:ひとまとまりとして考えやすい物をオブジェク トとして扱えるから、非常に有利だろうとあるが、ひ とまとまりとして考え難いものはオブジェクトとして 扱えないと考えてよいか。

データと機能が整理されてないと言われるが、オブ ジェクト指向を用いたから整理されるのではなく、い ままで、整理を行っていなかっただけではないか。

否:これについてはそうであるかもしれない。しか し、適応できる部分があると言うことは、少なくとも 悪ではないと言える。

ベストではないが、悪ではない。

肯: 先のひとまとまりで考えやすい、考えにくいに 関する質問内容を認めたものと判断する。

否:何が考え難いものかは、人により異なる。

肯:人によって異なるということは、人によって分 析のしかたも異なるということか。とすれば明らかに 利用者が分析者の考え方が理解できないと利用できな いということになる。

否:しかし、それはオブジェクト指向にかぎったことではない。

肯:かぎったことではないということは、オブジェクト指向も利用できないもののひとつになる。

(時間切れ)

◎否定側反駁

私達はオブジェクト指向がベストなものであるとは 言っていない。 現状の設計手法にベストなものは無 い。

先程述べた通り、データと機能を整理し隠ぺいして、オブジェクトしていく手法がオブジェクト指向の中の有効な1手法として存在するなら、オブジェクト指向は悪では無いと考える。

ベターなものが悪であるかどうかではなく、ベター のものを踏み台としてよりベターの手法を選択するの である。よって、オブジェクト指向の存在は悪ではな いと考える。

◎肯定側反駁

私達はオブジェクト指向の存在を否定しているのではない。一般的に言われていること、またそれから伝わるイメージが悪であると考える。

オブジェクト指向を習得するためには、今までの考え方を変えなければならない。だとすれば、今まで以上に、習得のためのコストが必要になり、これは悪であると言わざるを得ない。

また、オブジェクト指向を用いて物事を簡単に捉えることができるとある。確かに全体を見た場合はそうかもしれない、しかし内部に立ち入った場合どうなるのか、オブジェクトを生成した場合、それらがどのように関係しているのかを使用する側も知らなければならない。そのような問題をどのように解決するのかは述べられていない。これもオブジェクト指向は良いとされている面からすればまったく逆のことである。

以上のように一般的にオブジェクト指向と言われる言葉からの印象は我々を騙している。

 $\Rightarrow \star \Rightarrow \star$

プロジェクト管理 論題

「SEは会社に出る必要はない」

記録 外岡 徹(SRA)

*** 模造紙の掲示 *** Dグループ (肯定側)

論題

「SEは会社に出る必要はない」

肯定側の主張

・SEは仕事の内容によって、作業の場所、環境を 選べるべきである

定義

SE = ソフトウェア・エンジニア (自己管理能力のある人)

理由と論拠

- SEには、色々な作業がある
- 場所を選んだほうが能率のよい作業もある
- ・最もやり易い所で作業を行なった方が能率がよい
- ・SEの作業には、頭脳労働の部分がある
- ・SEを勤務形態で評価してはいけない

プラン

- ・社内教育の充実
- ・評価基準の多様化
- ・作業環境の整備

结验

勤務地を拘束しない事によって、生産性、利益の向 上につながる

メンバ

立論者 : 菊池 義幸 反対尋問1: 芳里 嘉昭 反対尋問2: 一原 武司

反駁 :横井 伸司(リーダー)

書記 : 伊藤 武司

*** 掲示 ***

Cグループ (否定側)

論題

「SEは会社に出る必要はない」

否定側の主張

・上記論題を否定する

理由と論拠

- ・結束したチーム作りができないーコミュニケーションの限界
- ・精神的な負担
- ・アメリカと日本の社会環境の違い
 - アメリカでのデータ、理論が 必ずしも日本で通用しない
 - 例)・管理されないと仕事ができない
 - ・日本ではOJTの割合が多い
- ・自宅で開発する為の十分な環境を作れない
 - ファイル管理
 - セキュリティの問題
 - 通信スピードの問題
 - ネットワーク技術の不足

結論

一部を除いて、SEは会社に出る必要がある

メンバ

立論者 :神崎 秀樹

反対尋問1:堀井 剛

反対尋問2:沖 さとみ

反駁

: 坂口 達司 (リーダー)

書記

:外岡 徹

*** 意気込と自己紹介 *** Dグループ (肯定側)

菊池:

昨日車で来まして遅刻しまして...

意気込なんですが、SEの作業場選択の自由を与えて欲しいと思います。その観点でこの主張を行なって行きたいと思います。

芳里:

日立ソフトウェアエンジニアリングの芳里と申します。先ほどのPCの方々の色々な好評を参考に頑張って反対尋問を行ないたいと思います。

一原:

株式会社JMAシステムズの一原と申します。照れ 屋であがりしょうなもので上手く質問出来るとは思え ないんですけど、一応反駁となっていますから出来る だけの事はやろうと思っています。

横井:

日本 I B M の横井です。この論題に関して色々議論 するうちに奥が深いなという事が判かって来まして、 その結果、我々は結構良い主張をするのではないかと いうふうに...(笑)

伊藤:

システムエンジニアリングサービスの伊藤と申しま す。頼もしい先輩方がいるので絶対勝てると思いま す。

Cグループ (否定側)

神崎:

アドバンスト・ソフトウエア・インスティチュートの神崎です。僕自身は、会社に出なくても仕事がやりたい派なんですけど、実際問題それで上手くいけるかどうかという問題点が現在かなりまだあると思うんですけども、そのへんを重点に言っていきたいと思います。

堀井:

シーズラボの堀井です。負けないように頑張りたいと思います。

沖:

MHIエアロスペースシステムズの沖です。現実的に、会社に出なくて出来るかどうかという点で主張していきたいと思います。

坂口:

高岳製作所の坂口です。佐原さんが会社に行きた い、というような議論をしたいと思います。 (笑)

外岡:

SRAの外岡です。いつも会社で電話がうるさくて、誰もとってくれないのでいつも作業が中断されてますけども、まあそれはそれ、これはこれとして否定側の主張をしてゆきたいと思います。

*** 立論 *** (4分) Dグループ (肯定側)

では、SEは会社に出る必要はないという論題につきまして肯定側の形で意見を述べさせて頂きたいと思います。

まず、その主張なんですけれどもSEは仕事の内容によって作業の場所、環境を選べるべきである。つまりなにも会社だけが作業の場所ではないという観点にたってお話したいと思います。まずその条件みたいな形なんですけども、その本人の意志によって働くところが選べる。それからあと、そのときにプロジェクトに支障をきたさない。ということを前提として折り込んでおきたいと思います。

では、定義としましてSEとはなんぞやということで、SEとはこちらでは簡単にソフトウェアエンジニアリングという風に解釈しております。で括弧で大書きでかいてありますけども、自己管理能力のある人、つまり入社して2年目であろうが5年目であろうが自己管理能力があればその人はもうSEと呼んで良いのではないか、というような形でとらえております。

それから理由と論拠なんですが、まずSEには色々な作業があります。昨日のお話にも色々あったと思います。例えば設計をするとか計画的にプロジェクトを作る、ドキュメントを作る、テストする、レビューするとか色々なものがありましたけれど、そのような作業ある。ということは、それぞれによって場所を選んだほうが能率が良い作業というものが当然あります。作業によって場所を選ぶ。またはこの場所ではこういう作業をやったほうが良い。まあ逆の事も言えますけど、そういうふうな関係があります。

それで、最もやり易い所で作業を行なったほうが能 率が良い。まあこれは読んで字のごとくです。

それからあと、さっきのSEには頭脳労働の部分がある。一生懸命残業をやって肉体労働の部分もありますけれども、基本的には「考える」という仕事におきましては頭脳労働の部分ということが言えると思います。

それから次の、勤務形態で評価してはいけない。つまり、いつも上司の目の前にいるからといってその人が「仕事が出来る」という事ではない。

このような理由をもちまして、SEはわざわざ会社に詰めている必要はない。これは必ずしもということで、会社に出たい人は出ても構いません。その人の自由です。

今の実態ですとかを考えますと、実際には会社に来ないと出来ない、あるいは会社に来た方が良いというような意見で会社側は言って来ると思いますけれど、実際に会社に来なくてもどこでやっても出来るという形に持って行くためにはプランとしましてこのような3つ、社内教育の充実とかあるいは評価基準の多様化。評価というものをまた別の見方で行なってみたら

どうでしょうか。それから作業環境の整備。これは会社におきましてもそうですし、もちろん自宅というような意味においても使って良い。このようなことをやっていく事が必要ではないでしょうか。

今までの結論にあたる部分なんですけど、勤務地を 拘束しないことで生産性とか利益の向上につながる、 というのが結論になっております。つまり、その人が 最適な場所で、最適に仕事を行なう。そうすれば能率 が上がりますよ。会社としては利潤を追及することが 一番である。それが目的である。その為には能力を十 分発揮するような環境用意してやることが必要であ る、と言うふうに思っています。

以上で肯定側の立論を終わりにしたいと思います。

*** 立論 *** (4分) Cグループ (否定側)

これより否定側の立論を行ないます。論題は「SE は会社に出る必要はない」で、我々はこれを否定しま す。

まず、否定する理由と根拠ですけれども、まず第一に結束したチーム作りが出来ない。これはコミュケーション、結局チームの各員がばらばらになりますので、これはデマルコの「チーム殺しの秘訣」の中でもありますけれども作業場所を分散することによってプロジェクトチームは結束できない、ということがあります。チームが結束できないということはプロジェクトが成功しにくい訳ですから、会社に出たほうが良い。

次に精神的な負担ですけれども、会社に行かない事により、人とのコミュニケーション、雑談なども含めて減りますので、会社から見捨てられているのではないかという疎外感などが生まれる可能性があります。これはゼロックス社のサテライトオフィスの実験によって明らかになっています。また、一般に日本の技術者などは会社以外でのコミュニケーションが少なく、結構会社などの人付き合いを止めてしまうと家に閉じこもりっぱなしになる可能性がかなり高いのではないかと・・・。そうすると気が滅入って生産性が下がるなどの弊害がでてくるのではないでしょうか。

第三にプロジェクト管理のよく出版されている本などは、データとか理論というものは基本的にアメリカのものでありますので、それがそのまま日本に通用するかどうかというのは、通用しないと我々は思っていまして、まずどういった点かといいますと、一般に管理されないと仕事ができない。

これは、日本人は小学校のころから管理教育に慣ら されていますから、先生や管理者に見られていないと 仕事が出来ない。誰も見ていないとさぼってしまうと か、自己管理能力がアメリカ人と比べて不足している のではないか、ということがあります。

また、大半の技術者というのはソフトウェアの仕事をしていながらソフトの本は読んだことがないとか、そういう人が実際問題かなり多いと思われます。そういうような人が自宅で自由に仕事をやって本当に仕事、プロジェクトをこなして行けるかというのは疑問であります。

また日本の会社というのは生涯雇用が一般的でありますので、仕事をいっしょにしながら技術レベルの高い技術者から低いレベルの技術者へ技術の伝達とか教育などが行なわれていますので、もしこれをバラバラにしてしまうと技術の低い技術者への技術伝達が上手く行かないと思います。

また自宅でやる場合はファイル管理システムやセキュリティ、ネットワークを組むにしても通信システムのスループットなどの問題が現在ありますので、結論として一部...

*** 反対尋問1 *** (1分) Cグループ (否定側)

質問者一否定側

返答者一肯定側

質問者:

それでは、否定側の反対尋問を行ないます。

まず、定義の所「SEとはソフトウェア・エンジニアリング(自己管理能力のある人)」とありますが、自己管理能力がある人とは誰が決めるのでしょうか?会社なのか、自分なのか?

返答者:

基本的に会社の方で、この人は自己管理能力がある という形で評価で決めていると思います。

質問者:

それでは、そういう人が日本人でソフトウェアエンジニアリングというかソフトウェア関係の仕事をしている人で、割合と言いますか... どれ位いますか?

返答者:

それは各社によってまちまちだと思います。能力の ある人間がいっぱい集まっている会社は高いと思いま すし、そうではない会社は低いと思います。

質問者:

それでは、作業場所についてなんですけど、ここで 言われている作業場所というのは、会社以外というこ となんですけど具体的にどの辺を、自宅なのか、そう でないのか... (1分経過)

*** 反対尋問1 ***(1分) Dグループ(肯定側)

質問者一肯定側

返答者一否定側

質問者:

肯定側の反対尋問を行ないます。

コニュニケートの限界というお話が立論でありましたけれども、毎日会社で顔を合わせないとコミュニケーションは取れないのでしょうか?

返答者:

緻密なコミュニケーションは取れないということです。

質問者:

それでは次に移ります。

日本人は自己管理能力が不足しているというような 事がでていましたけれども、自己管理能力があれば、 充実していれば、会社に必ずしも出社する必要はな い、という事でよろしいでしょうか?

返答者:

大きな要素の1つです。 (音声不鮮明の為,半分想像)

.

質問者:

以上です。

(笑:時間を大きく残して終わってしまった為)

質問ではないのですが、コミュニケーションでした ち電子メール等の手段はあると思うんですけど... 。 あと、人との雑談が減るという話がありましたけれ ども、人との雑談による生産性の低下というのも結構 あると思うのですけれども、それについては?

返答者:

生産性に低下はありますけれども、逆にこういう部分から例えば新しいひらめきであるとか、バグの発見であるとか、意志疎通が出来ていない部分が明らかになったりする事はありますので、雑談する時間がそのままマイナスになるとは思っていません。

質問者:

どうもありがとうございました。

*** 反対尋問2 *** (1分) Cグループ (否定側)

質問者一否定側

返答者一肯定側

質問者:

それでは否定側の反対尋問を行ないます。さきほど の続きになりますが、勤務地を選べるという事ですけ れども、勤務地は「自宅」としてよろしいですか? (「自宅」以降は想像)

返答者:

それだけではありませんけれどもそれも1つです。

質問者:

その他にどこがありますか?

返答者:

昨日のお話にもありましたけれども、会社以外と言ったら全てが場所となりまして、具体的には自宅というのがありますし、夏場でしたら涼しいペンションとかそういう所も1つの場所ですし、その人が働きやすい所がその人の職場になります。

質問者:

例えば自宅で勤務するという場合に、自宅でやる場合にはそれなりの設備が必要ですし、ネットワークを使ったりすればそれなりの通信費用が掛かると思いますけれども、それでも利益が上がると言うことがよくわかりませんが?

返答者:

例えば自宅で、「パソコンがなければ仕事が出来ない」という仕事が全てではないと思います。SEの仕事の中でいろいろな仕事がある、という事を先ほど説明したと思いますけれども、そのうちのパソコン等マシンがなくても出来る仕事はあります。ですからその仕事をやればいいと思います。

質問者:

社内の教育について質問したいのですけど、ここで 挙げている社内の教育というのはセミナーとか、そう いうものでしょうか?

返答者:

社内教育と言いますと、色々なものがあると思いま す。セミナーもその1つです。

質問者:

実際に今まで自分が経験としてセミナーで得たことと、実際の業務で他の人を見て、OJTで得たことと、どちらが大きいですか?

返答者:

こういうふうに皆さんと会っているセミナー形式の 方が得るものが大きいと思います。 (笑)

質問者:

仕事をして先輩方の行動を見ている方が良かったと 思います。終わりです。

*** 反対尋問2 ***(1分) Dグループ(肯定側)

質問者一肯定側

返答者一否定側

質問者:

反対尋問2を行ないます。

先ほどのお話で、日本人には自己管理能力のある人 が少ないとおっしゃいまいした。と言うことは、自己 管理能力のある人は多少はいる?

返答者:

多少はいますね。

質問者:

それでよろしいですね? 自己管理能力のある人も 会社に絶対に来なければいけない?

返答者:

結論では「一部を除いて」と言ってありますので、 (笑) 要するに自己管理能力があって、技術レベル も高く、しかもネットワークやファイル管理、セキュ リティの問題がクリア出来るような場合であれば会社 に来る必要はない。

質問者:

では、そちらにも一部の人は会社に来る必要がな い。ただ一般的ではない。

その一部の人というのは、いま、「人」ですよね。 開発工程ではなく「人」だと言いましたよね?

迈答者:

一部の人ですね。

質問者:

一部の人というのは全工程に於いて会社に来なくて いいと結論づけていますか?

返答者:

一部の人で一部の業務です。

質問者:

そうですね。一部の人で一部の業務。そうですか。

返答者:

当然ミーティングなどは会社に来なくてはいけない。

質問者:

それで、お話の途中に日本人は管理されないと仕事が出来ないとおっしゃいました。アメリカなどと比べて...。ということはアメリカ人よりも日本人のSEの方が品質が悪い、とおっしゃっているのですか?

返答者:

品質が悪い...品質をどう定義するかということによりますが、「品質」=「自己管理能力」であればそういうことになります。

返答者2:

今の意見については日本の教育制度上そういう傾向 が多いのではないかとそういう風に考えています。

質問者:

であれば、教育制度が同じであれば結局皆が同じような育ち方をしているから全員同じに、いっしょくたに考えて良いとおっしゃるのですか?

返答者2:

管理教育という事を問題にしているのです。

*** 反駁 *** (2分) Cグループ (否定側)

否定側の反駁を行います。

先ほど、機械を使わない仕事もあると言いましたが、そこから考えれば人と会うことが多くなり、結局は会社に出て来なければならないという面が多くなると思います。

また、いますぐ環境をそろえるというのは、はっきり言っていますぐという事には絶対に出来ないと思います。数年後に関しては私は分かりませんけれども、 現時点では出来ないです。

日本人に関しては、高校生の時と大学生の時どちらが勉強したかと言えば、多分僕の感じるところでは皆さん高校生の時に一番勉強したと思います。これは管理教育の元である為に、SEは管理の元でなければある程度は出来ないと思うのでSEは会社に出て来る必要があると思います。

*** 反駁 *** (2分)

Dグループ (肯定側)

肯定側の最終弁論を行ないます。

まず、我々の主張を確認しておきたいのですけど、 我々は皆会社に来るなと言っている訳ではなくて、会 社以外でもっと仕事の能率が上がる人は自由度を持た せよう、という主張をしている訳です。そういう人が どういう人かというのは、我々はSEであって自己管 理能力のある人で、本人が会社以外の方が生産性が上 がると考えている人。で、その人が会社にいないこと によってプロジェクトに支障がない、という条件が満 たすのであれば、より生産性の上がる場所で仕事をす る方が良い、という主張をしています。

そもそも企業とか会社というのは営利団体ですから、生産性とか利益の追及というのが第一義になります。向こうの主張のように結束したチーム作りが出来ないというのがありましたが、結束するというのはそれによって生産性とか利潤を追及する手段である訳です。だからそれ以外に「場所を変える」というもっと生産性の上がる手段があるのであれば、そちらを追及するというのが当然の事であるという風に考えています。その為に会社というのは、こういう社内教育の支援であるとか、評価基準の多様化、作業環境の整備というような事を支援して、社員がより良い環境で、より生産性が上がるという職場を選べるような支援をこれから続けていくべきだと思います。

先ほど主張が有りましたけど、今すぐ無理というのではなくて、今すぐ無理かもしれなかったら今から試行して段々改善をして行くというプロセスを始めるべきであると思います。

最後に言われました管理教育。日本人は管理教育であるというという話がありましたが、日本の企業もこういう仕事の形態を取ることによって、回りの自宅とかで仕事をしている人から啓発を受けて、皆が管理教育から抜けていくような雰囲気が作れるのではないか、という風に考えています。

(次頁に続く)

*** 肯定 ***

		肯定側	否定側
中来田		1 3	14 0
中谷		11.5	15 O
伊藤	0	1 2	1 1

東田	0	1 6	1 5	
菊池	0	1 7	1 4	
佐原		1 5	16 O	
大塚	0	1 6	1 5	
				_
合計	1	00.5	100	

中来田:

3 対 4 で肯定ですね。微妙に0.5 の僅差で肯定の勝です。

*** 講評 ***

中来田:

私は13対14で否定側有利と書いたんですけれども、尋問の仕方が、初めの沖さんですか?が、誠に素晴らしかったので、そこの点の一点差で否定側の方が有利なようです。ただ、最後の反駁所がですね、まあどちらもなんですけど時間を有利に使ってなかったのでそれが残念に思いました。

あと、利益追及って話なんですけれども、実際に佐原さんほどではないんですけれど、結構分散型で私の会社はやってるんですけど、なかなか利益は上がらない。

現実はですね。 (笑) その辺の現実の問題の数字 なんかを出せば、うまくすればもっと否定側は有利に なったような気がします。以上です。

中谷:

えーと、.5 というのがありますけども、反対尋問とかやりますよね。反対尋問の質問に対して、答える側が上手く答えてない場合には減点をさせていただきました。

ということで、質問しながらそれに答えた内容が的を射ているか、射ていないか。それから自分の主張がある訳なんですけれども、自分の主張ではなくて相手の主張に対して賛成してしまうような答えをしてしまった場合には減点となります。

ということで色々計算していったんですけれども、 反対尋問の2番目ですね。沖さんの質問はとても良かったと思います。それからあと、反駁の方の横井 さんは時間を有効に使っていたと私は思いまして、5 点を付けさせてもらいました。それに対してやはり、 時間は短い訳ですからその中の時間を有効に使ってですね、質問した場合に返って来る答えはイエスかノーかですよね。その答えによってどうやって攻撃しようかというような所まで考えて質問をすれば、もっと私 は否定側の方がですね、点数が良くなったのではない かなという気がします。以上です。

伊藤:

えーとですね、私さっきより低いんですが、全体的付け方が...。それで山崎さんもおしゃってたんですが、これパッと見るとですね、上だけ見ると違う全然正反対のテーマみたいなんですが、結局はですね結論は「一部を除いてSEは一部を除いて会社に来る必要がある」という事で、その一部っていうのは説明の中では自己管理能力っていう事だと思います。

それは肯定側もやはり「自己管理能力のある人は出る必要はない」という意見だったんで、結局方向的な所では何も違っていなかったというので、議論になっていなかったような気がします。

どこで差異を出していたかというと、否定側が管理 教育で肯定側が・・・生産性と利益という話だったん ですが、私は総合的に肯定が勝ったのは、管理教育だ から自己管理の出来る人は少ないという所へ持って 行った所がまず失敗だったのではないかと私は思いま す。

もうちょっと自己管理能力一向こうの方がですね、 あれだけでかく主張してるので、では自己管理能力と いうのはいったい何だというところをもっと突っ込ん でおけば、完全に逆転出来たんじゃないかと思うんで すが。

それから肯定側の方で言いますと、プランが余りに も弱いような気がしました。例えば社内教育。要する に会社の存在、会社の集合作業そのものを否定するよ うな主張にもかかわらず、社内教育という言い方は なんかすごく違和感がありました。まあそんなところ です。

東田:

まあ、伊藤さんと同じような意見、感想を持っているんですが、やっぱり議論になってなかったかなという印象があります。

で、反駁以前までは僕は否定側が勝っていたんです が、最後のまとめの段階で否定側は上手くまとめられ なかったということです。

否定側は結構内容を見ると質の良い材料を持っていたという気がしたんです。それがまあ、はずされてしまったんだと思うし、肯定側まあ、全体的に最後まで上手くまとめたから、1点多く結果が付いたんですけれども、なんとなく綺麗にまとまっていただけで、内容的にそういうインパクトがあるような成果が見られなかったのは残念でした。

菊地

具体的に先程1回試合をみているので、先程よりは だいぶ良くなったなというのが全体的な印象です。そ れから、最初パッと見て気付いたのは否定側の方で、「一部を除いて」という結論が出ている訳ですね。これはすごく弱いと思います。そこを突かれると弱いなあと思ってもう、最後にやっぱり突かれてメロメロになってしまいましたね。

最初みんなで話し合ってた所に「一部を除いて」という結論が出てもですね、そんなことないよ、全然関係ない、出なくちゃいけないんだ、っていうくらい主張しないと最初から臨む時点でもう負けてるんじゃないかなという気がします。

それからあと、反対尋問。肯定から否定なんですけど、芳里さん二つくらいポンポンとやって時間余しましたよね。時間余るのは余るんで良いと思うんですけど、あるからとその後慌ててした質問がちょっとまずかったんじゃないかなと思います。 みれは、あの時点で切ったほうが良かったと思います。 それからですね、結構反対尋問もそれぞれ自分の方に引き込もうで、なかなか引き込まれなかった。これは質問自体はさっきより大分良くなったと思うんですけど、答え方も結構良くなって来て簡単に相手の方にのらないようで、やり取りは結構良かったんじゃないかと思います。

ただ、ちょっと気になったのは沖さんが、何気なくだと思うんですけど「そうですか」って言っちゃったんですね。あれは相手の主張を認めることになってしまうんです。質問しちゃって相手が答えて、「つい」でも「そうですか」と言っちゃうと、もう相手が正しいですよって言っちゃう訳ですよね。そういう余り不用意に言わない方が良いんじゃないかと思います。

あと最後の反駁。これ、名前こそ「反駁」ってなってますけども最終弁論なんですよね。なんで、必ずしも相手に対する反駁という形じゃなくて、もう一回最後に自分の主張を繰り返す訳です。そういう点から言うと、否定側の方が相手に対する反駁が二つで、自分の主張を繰り返したのが一つ位ですけど、そういう形になっちゃったんですね。そうじゃなくて別に相手に対する反駁っていうのは必ずしも言わなくて、もう一回自分の主張を繰り返すと。

で、その自分の主張を繰り返したんですけど、その時も「あ、しまった」というような表情が完全にでてるんですよね。その時点でもう負けだと思いました。そう思ってはいても全然顔には出さないでやるべきだと思います。その点肯定側の方は、自分達の主張も最初に言ってもう一回繰り返してますし、相手に対する反駁もみんなやっていて、もう他のPCの皆さんも最後の最終弁論の所でかなり肯定側に良い点付けてこういう結果になったと思います。以上です。

佐原:

私の方からなんですが、たぶんですね、これ、最後がなければ否定側の圧勝だったと思うんですね。何故かというと、肯定側の立論自体が私のチュートリアルの範囲を出てなくて、否定側は全部予想してたことですね。あそこで逆に否定側が主張している事はですね、最後まで一つも否定されなかったんですよ。だから最後の反駁の時では、あの主張をもう一回繰り返せばよかったんですね。繰り返してしかもそれは肯定側は一つも否定できませんでしたね、という事を言えば良かったんです。

私が菊池さんと違ったのは「一部を除いて」というのを入れておいても何等おかしくないと思うんですね。つまり、例外的に一人でも自宅で勤務していたらそれはダメよ、って言うのはそれはさすがに条件として厳しすぎるので、つまり、殆どの技術者は会社に出るべきだよ、というのは何等おかしい主張ではないと思うんですね。

つまり当然条件関争ですからこれは、両方とも、こちらの方は「かなりのソフトウェア・エンジニアは会社に出なくて良いよ」。こちらは、「いや、殆どは出なきゃダメだよ」と言っている訳ですから、やっぱりそこにかなりのパーセンテージの違いがあるわけで、そこ明確に主張すれば良かったと思うんですね。で、基本的にはですから否定側の方が全部あちらを攻撃することになっていて、しかも自己否定されてないから、それさえ言っていれば良かった。

しかもですね、反対尋問の時に殆ど肯定側が自殺点に近いようなのが2回ともあって、かつですね、否定側の質問の時に「パソコンなくても出来る」って言っちゃいましたよね。あそこを突かれるとですね、非常にヤバイな思ったら沖さんが確か突かなかったんですね

何故かというと、パソコンなくても出来るって事は、かと言ってビジネス用の連絡はしなきゃいけないですから、電子メールとか何にもなしだと結局電話とか掛かり始めるんですね。それは、何の事はない、会社にいるのと大差ない訳ですよ。だから緊急ではなくても、一日とか半日単位では連絡取りたいというのはやっぱり仕事の中でかなりあって、明らかにパソコンなり道具がないとちょっときついですね。つまり、自宅でやっているんのに会社にいるのと同じ仕事になってしまう。そこをこちらは突くべきだったんです。

日本の管理教育に関してですね、主張したところは 実は、肯定側は別にたいして突いてきていないから、 そこを敢えて言う必要はなかったですね。で、肯定側 が言ったミスは、自宅の開発環境は今時点では実は、 たとえば、私なんかUNIXのシステム管理者もやった し、Macintoshの事も良く知っていて、その間のネッ トワークも結構出来るので、何とか自宅で出来るよ、 という主張だった筈なんですね。あのBの主張は... .。それを見るとこちらの肯定側の主張にはそれが あんまり入ってなかったんですね。整備出来るとは 言っても実は、否定側の主張は整備するって言ったっ てそれは何年もの教育とかですね、それから今より I SDNなどもうちょっと普及したりとか、そういうふ うにならないと

・・・ (テープ切れ)・・・

数年間は、だから数年間はダメと言ったので、数年というとちょっと弱いですね。まあ実質的には出来るかもしれないけど。

山崎:

結構今のスタイルというのは、私はかなり強引な・・の・・・だと思うんですね。というのは、一つは「SEは会社に出る必要はない」なんていうのの否定は何というのは一つ大問題で。それで、そのパーセントの違いだなんていうのは、殆ど意味をなさないんじゃないかと。

それともう一つは、例えばよくソフトウェアハウスの仕事として、要員派遣か、受託業務かという様な話がありましたね。そうだとすると、受託業務というのをみんなやれば良い訳で、何も媒体がどうこうとかいうことはあんまりない訳ですね。それから二日に一回連絡しなきゃダメだとか、一ヶ月にいっぺんは連絡するだろうなんて事は意味無くて、仕事が出来た時に、要するに納期が遅れてなければ、そして出来たものがその注文通りの物であれば良いだろう、なんて言う事が出来る訳でしょ?

(注:「・・の・・・」はどうしても聞き取れず、 想像もつきません)

佐原:

いや、それはだから特殊な業務に関してそうだけれども、全体としては出来ない訳じゃないんだから。要はソフトウェア技術者というのはいろんな業務が含まれていますから、その中でやはり大多数の人がどうなのか、とうのを言わないと、例外は、どっちの議論にも例外的な所はありますから。そういうことで言うと、100%出なくて良いというのとどちらも成り立たないと思います。

山崎:

いや、そういう議論なんじゃ...。

佐原:

そういう話だと、両方ともそこを突かれたら、どち らもどうしようもないですね。

山崎:

うん。どうしようもない、というような**事**を言って いるんじゃないかと。

佐原:

だからパーセンテージで良いんじゃないかと。

山崎:

いや、そうすると論理というものが訳わからなくなりますでしょ?

(いつのまにかPC同志の議論に...)

山崎:

あの、必要とか必然とか可能だとかっていうのは、 ものすごく難しい論理でしょ?自然言語なんかをどう 理解するかっていうのは一番難しい訳だから、そうい うのをちょっと付け加えておくとたちまち引っ掛か る、ていう風になる。

菊地:

そもそもこのテーマになんですけど、PCの間でも 反応が別れていたんですね。一番難しいんじゃないか と。本当はもっと抽象的な(注:違うような気が)か ちっとしたテーマでやるべきなんですけど、難しいで すよ、今SEをテーマでやるには。それを敢えてやっ てるんで、ディベートとしては難しいところがあった と思います。でもその割りには良くやったな、という 気がしますけど。

大塚:

「SEは会社に出る必要はない」って、一番心配していた論題です。これで若手がしゃべれるのかって、一番心配してたネタなんですけど、それでも皆さん良く意見が出ていた様なのでその点すごく安心しました。

否定側のほうからも指摘されましたけれども、あそ こに定義の所で「自己管理能力があるSE」と括弧書 きで書いてしまったんですけど、あそこで言いたい定 義っていうのは、SEっていうのはソフトウェア・ エンジニアの略称だよという事だけで、自己管理能力 がある人というのは主張の所でちょっと付け加えた かったんですね。但しSEっていうのはこの場合は、 我々は多少は自己管理能力、多少プロジェクト管理の 技法とかある程度の能力を持っていると。自分で進捗 管理くらい出来るとか。管理者の側から見ると、社員 がどこで働いているのか判らない訳ですよね。だから 今よりももっと人を管理する能力というか、プロジェ クト管理を細かくやらないとなかなか難しいのかなと 思って、プロジェクトリーダーにとっても教育はもっ と必要になるんじゃないか、といった意味での社内教 育です、模造紙に書いてもらったのは。そういった点 を会社側からも補助というか支援すべきでしょうと主 張したかった訳です。

佐原:

私が肯定側だったら、「自己管理能力がある人」っ て仮定は外すね。だって自己管理能力がなくたって自 宅でやったって良いじゃないですか。 (笑) つまり自 己管理能力がない人は会社に来たって同じですよ。

大塚:

だから敢えて、否定側から聞かれたら言って良いの かなと思ってたんです。あとやっぱり反駁ですか。反 駁としての役割を果たされていたので点数を良く付け ています。

中来田:

はい。どうもありがとうございました。

 $\triangle \times \triangle \times \triangle$

CASE 論題 「上流CASEは必要である」

記録 棚橋 巡治(SRA)

● 論題

「上流CASEは必要である」

● 肯定側 (Bチーム) 配役

立論:

大野 修一

反対尋問1: 根岸 易世

反対尋問2: 荻原 郁子

最終弁論:

棚橋 巡治

書記:

丹羽 徹

● 否定側 (Cチーム) 配役

立論:

外岡 徹

反对尋問1: 坂口 達司

反対尋問2: 沖 さとみ

最終弁論:

神崎 秀樹

書記:

堀井 剛

肯定側模造紙の内容

- 1. 論題 「上流CASEは必要である」
- 2. 主張 論題を肯定します

3. 理由と論拠

(1) 事実

- ・ソフト開発の規模は年々大きくなってきている.
- ・ソフトが社会に及ぼす影響は多大である.
- ・人間は誤りを犯すものである.
- ・ソフト開発における主要な問題の発生源は上流工 程である.
- ・上流工程の誤りは下流工程に大きな影響を与え る.

(2) 論拠

- ・コンピュータの誤動作は社会に多大な不利益とな る.
 - ・品質向上のためには上流工程が重要である.

4. 結論

『社会全体の利益のためにはソフトの品質向上が不可欠であり、よって上流CASEが必要である!』

● 否定側模造紙の内容

- 1. 論題 「上流CASEは必要である」
- 2. 主張 「今ある上流 CASEツールは使えない」

3. 理由と論拠

- ・方法論における仕様記述に問題がある.
- 開発規模の限界
- ・方法論自体が発展途上であり、CASEツールが 追いつけない.

例) OOA

・一度導入したら、乗りかえが困難

4. 結論

- ・今のCASEツールのミクロの進歩よりも 別のマクロの進歩を選択するべき。
- ・現在のCASEツールを多額のコストとリスクを 負ってまで導入する意味はない。

● テープから採録 [肯定側、意気込み]

大野:

日本NCDの大野です。グループ・リーダーをやっていて、昨日はちょっと皆さんに遠慮して引っ込んでいたんで一番前の席にいます。頑張ります。

根岸:

ニコンシステムの根岸です。今回の場合は、昨日と 違ってだいぶ、追風的なところがあるんですけど、 まぁこれに気を緩めないで頑張りたいと思います。

荻原:

SRAの荻原です。CASEは必要です。 会場(笑)

棚橋:

SRAの棚橋です。すんなりまとまるかなっと思ったんですが、ちょっと紛糾しましたがなんとかまとまって同意に持ち込めたと思っています。頑張りたいと思います。

丹羽:

オムロンの丹羽です。このまま勢いに乗って連勝したいと思います。よろしくお願いします。

[否定側、意気込み]

外岡:

SRAの外岡です。ぱっとあそこを見ると、なんか、うまく行きそうな気もするんですけど、昨日の件もありますので、頑張りたいと思います。

坂口:

高岡製作所の坂口です。昨日は足を引っ張ったんで、きょうは気持ち良く帰りたいと思います。

会場 (笑)

沖:

MHIエアロスペースシステムズの沖です。今日は 勝ちたい(違うかも)と思います。

神崎:

アドバンスドソフトウェアインスティチュートの神 崎です。緊張しています。

堀井:

シーズラボの堀井です。個人的な意見として、CA SEは必要だと思います。

会場 (爆笑)

[肯定側、立論]

大野:

Bチームは、上流CASEは必要である、という論題で主張いたします。あ、失礼しました、上流CASEは必要である、というのを肯定します。

理由と論拠ですが、現在のソフトウェアの開発は規 模が年々大きくなってきている。それから、ソフト ウェアが社会に及ぼす影響が多大である。あと、人間 は誤りを犯すものである。

ソフトウェア開発における主要な問題点の発生源は 上流工程にある、上流工程の誤りは下流工程に大きな 影響を与える、というような事実がある。

これから導き出される事が、コンピュータの誤動作 というのは、社会に非常な、社会に多大な不利益とな るということです。これを解決するために、品質向 上、品質向上のためには上流工程が重要である。

こういった上流工程をきちんとやるためには上流CASEを導入してですね、上流CASEが必要でしっかりした設計をするが必要であるということですね。

で、事実のところなんですが、資料としてSEAが アンケートを出したんですが、その中(ページをめく る音)ソフトウェアのプロジェクトにおいて、主要な 問題というのはどこから起因しているのか、というこ とがあるんですが、これが全体設計、要求定義、プロ ジェクト管理、それからレビューと検査、この四つが大きな要因だということですね。

そしてもっとも重要な問題というところですが、これは要求仕様が曖昧であること、それから、要求仕様が変更されること、こういった上流工程の重要な部分が問題となっているということです。

結論として、社会全体の利益のためにソフトウェアの品質向上が不可欠です。ということは品質を向上するために上流CASEが必要であるということです。

[否定側、立論]

外岡:

論題、上流CASEは必要である、という論題についての我々否定側の主張は、現在の上流CASEが使えない、そういう立場でお話したいと思います。

理由と論拠ですけれども、まず一つ、今ある上流CASEというのはほとんどが、SAとかOOAの方法 論をベースとしたCASEだと思いますけれども、これらの方法論を、最後まで、一番小さなプロセスまで 落した時に、仕様記述の問題があります。

この仕様記述というのは、まずこれらの方法論では、自然言語で書きなさいと言っているんですけれども、まず、我々はどうしてSA、SDなんかを使ったり、OOAを使ったりするのかというと、その大きな原因の一つというのは、分析で何を行なうかというのを明確して完全にすると、これが大きな要素だと思うんですけれども、それなのに、ここの部分を自然言語で書かなくてはいけなくて、結局はここに曖昧さが残ってしまう。それを完全に排除しきれるのは、今のところないんじゃないかと、そう言う立場です。

もう一つ、開発規模の限界というのがあります。

これも、CASEツール自体の問題ですけれども、 リポジトリの限界。これは、詳しいデータはちょっと ありませんけれども、まぁ大体数十万ステップぐらい になると、もう、ちょっと追い付けなかったり、処理 が極端に重くなってしまったり、そう言うことがあり ます。

また、OOAなんかのCASEツールでは、リポジトリ自体がなかなかサポートしきれていない、ただのお絵書きツールになっちゃっている、というところもあるみたいです。

もう一つ、方法論自体が現在発展途上であり、CA SEツールが追い付いていけない。 たとえば、OOAなんかの話しですけれども、現在、どんどん変わっているというのと、はっきりしたどれが正しいとか、どれが最高であるのいうのが今まだ、議論されている途中だと思います。

しかも、ほとんどがアメリカで議論されていることで、日本で使われているCASEというのの大部分も、これもアメリカが作ったものを日本語化して、まぁアメリカとは限りませんけど、日本語化しています。ということは、方法論自体のお話で、少し変わった新しいものが出来た。それをCASEツールに対応するのにタイムラグがあります。それをまた日本語化するのにタイムラグがあります。

こうすると、新しい方法論自体をCASEツールに 一旦導入してしまうと、すぐに使って行くことが出来 ない。こういったマイナス点が大きいと思います。

またもう一つ、CASEツールの導入というのは、 ずいぶん大きなコストとリスクがあると思いますけれ ども、これを一度導入してしまったら、やっぱりやめ た、というのは、導入した方としてはなかなか踏み切 れないと思います。

結論です。

今のCASEツールの、現在のCASEツールのミクロ的な小さな修正をゆっくり待って行くよりも、全く別の大きな、何か新しいものを選択したほうが良いのではないかと思います。 (制限時間の音)

[反対尋問その1、否定→肯定]

坂口:

今回、...(聞き取れず)の92年6月のレポートから、上流CASEツールは生産性は2、3年、CASEを入れる以前のものより生産性は上がらない、保守性も上がらない、と出ているのですが、それだけで確かに品質向上はありますが、それだけでCASEが必要なのですか。

大野:

もう一回。

坂口:

2,3年は生産性は上がらないんです。

上流工程が大切なのは分かりますが、それだけで品質向上が得られるのですか。

大野:

得られますよね、分析と設計をしっかりやりますから。

外岡:

上流工程が大切なのは分かるのですけれども、

大野:

はい。

外岡:

そのことが、CASEツールを使うことによって、 なのですか。上流工程が大切だっていうのは、CAS Eツールを使わなくても、

大野:

じゃぁ他に方法は何かありますか。

会場(ざわめく)

棚橋:

上流工程をきちんとやるためには、手間が必要なのです。これからのソフトウェアは、段々複雑になる。 その複雑さに追いついて行くためには、人間の能力だけでは追い付かないのです。そのためには、(制限時間の音)

[反対尋問その1、肯定→否定]

根岸:

先ほどの尋問の中で、まず一つお尋ねしたいのは、 そちらが言ったことなのですけれども、2,3年は生 産性が上がらないとおっしゃりました。では、その後 は生産性が上がるわけですね。

外岡:

それについてはですね。

根岸:

YESかNOかで、できれば。

外岡:

わたしには、ここでは分かりません。

根岸:

では、先ほど述べられた中で、上流工程が大切だ、というのは認めていただけますね。

外岡:

はい。

根岸:

では、次にもう一つ、やはり同じように、品質の向上性が認められる、というのも認めていただいたわけですね。

外岡:

え、品質向上が認められる、って、誰が言ったんで したっけ、却下します。

会場 (爆笑)

根岸:

我々の主張(違うかも)の中で、そちらが言っていたような、今あるCASEツールのことを云々していますか、

外岡:

現状の、

根岸:

我々の今主張している中でです。

外岡:

もう一度お願いします。

根岸:

今我々が主張している、ここに紙に書いてある中で、現在のCASEツールを問題としていますか。

外岡:

そうです。現在又は、極めて近い将来のことを考え ています。

根岸:

次に、我々が書いている中に、問題としている中 に、方法論云々の話しがありますか。

方法論が、発展途上であるかとか、そう言うことを 問題としていますか。

外岡:

いませんが、

根岸:

いませんね。

外岡:

上流工程というのは、上流工程のCASEツールというのは、必ず方法論に乗っかってなくちゃいけないと思います。 (制限時間の音)

[反対尋問その2、否定→肯定]

油:

それでは、反対、否定側の反対尋問を行ないますが、まず、なんかそちらの論拠からいくと、上流工程が大切なのは分かるんですけれども、上流CASEを使えば必ず上流工程がきちんと出来るということですか。

大野:

そうです、曖昧さがなくなるわけですよね、

沖:

上流CASEというのは、方法論に基づいて作られていて、それなりに方法論を理解していないとただの

お絵書きツールとしか使えないと思いますが、その辺はどうでしょうか。

大野:

お絵書きツールですね。ということは逆に言うと、 方法論を知っていれば、非常に有効なツールになると いうことですよね。

沖:

方法論をちゃんと知るためには、それなりの教育を したりとかが、大切で、そう簡単に導入すれば使える というものではないのではないでしょうか。

大野:

ないですね、でもそれは、当然それなりの訓練を積めば良いわけで、CASEが不要だということではないですよね。

沖:

実際にデータがあるのですけれども、それなりに高いって言うか、良いって言うツールを買えば、それに比例して教育費もかかるんですけれども、その何年も、生産性が上がるまでに時間がかかって、時間がかかって、生産性が上がるころには、その方法論、またその、導入した人が使えるとは限らない、ということはあるんじゃないかと思うのですが、

大野:

確かに教育にはコストがかかります、時間もかかります。けど、上流工程でバグが少なくなれば、下流で発生する、時間というのは回収出来ると思います。

沖:

だから、それがそう言う風になるまでに時間がか かって、(制限時間の音)

[反対尋問その2、肯定→否定]

荻原:

では、肯定側の反対尋問を行ないます。まず、C チームの方では、上流CASEツールは使えないとい う主張ですけれども、上流CASEは必要ではないと はおっしゃっていないですよね。

外岡:

はい、いいですか。

荻原:

はい。

外岡:

我々の立場は、現在又は近い将来の、現状のCAS Eツールが使えないと言っているわけで、なにも未来 の素晴らしいCASEを否定するわけではありません。

荻原:

で、それから先ほど、いろいろ反対理由を述べられているのですが、お話を聞いていると、例えば、今サポートしている方法論は発展途上だから使えない、というような意見がありますが、例えばSAなりSDを導入して、既に自分たちの方法論として、確立している企業でも、現状のCASEツールは導入しても意味がないとお思いですか。

外岡:

えっとですね、うまく使っているところがあるということは事実ですけれども、実際に普及しているというのは、日本の調査では、実用はわずかに3%です。 ほとんど、普及していないと、考えられて良いでしょう。

荻原:

となると、昨日、一昨日でしたっけ、プロセス成熟 度のお話がありましたが、97%のものは、レベル1 である、ということですよね。で、CASEツール、 CASEを導入して有効なのはレベルが3に上がって から、始めて有効になる、という事実を全くそこで反 映させている、ということであり、それはもう、CA SEが必要か必要でないか、以前の問題だと思うんで すね。

外岡:

で、その、レベル1の我々に、CASEを導入する 必要がないと、ということですか。

荻原:

だけども、導入する必要性とCASEの必要性は違いますよね。(制限時間の音)

[否定側、最終弁論]

神崎:

肯定側がまず前提としている上流CASEというのは、理想論であることを向うも認めていると思いますけれども、そう言う理想的なものは、あればいいに決まっているのは明らかなことであって、我々は、実際、上流CASEというツールが現在、いいものがない。

また、向うの方でも上流工程がきちっと出来ていなければ、工数が大きくなるという部分がありますけれども、現在のCASEツールで、最終的に仕様記述が曖昧な部分がまだ残されて、その部分は下流工程まで残されるわけです。そうすると結局、それだけ大きなフィードバックがかかって、工数大になります。

その辺なんかは、フォーマルメソッド等の手法を取り入れることによって、上流工程でもっときちんとした、仕様記述をすることによって、上流CASEを使わなくても出来るんじゃないでしょうか。

また、CASEは導入してから効果が出るのに、非常にコストとか、時間がかかりますんで、上流CASE、というか、また、先ほど言った、導入してからの乗換えであるとかバージョンアップ等に非常に時間がかかることが予想されます。

こうして見る場合、CASEを使わないでCASEを使わない開発における生産性のアップも当然考えられるわけで、一度CASEツールを導入して、しがらみになってしまって、結局CASEを使わない手法に追い越されるという可能性は否めない。で、以上で我々は、(制限時間の音)

[肯定側、最終弁論]

棚橋:

そちら側もおっしゃられています様に、CASE ツール、現状のCASEツールの問題点は、いろいろ 述べていらっしゃいます。

でも、そこの中でありましたように3年後には、大 丈夫になるぞというお話もありました。と、言うこと なんですが、それの代わりにいろいろ方法論がある、 と、おっしゃいましたが、どの方法論にしても、CA SEですので、コンピュータ・エイデット、えっと、 コンピュータがどの方法論を支援してもいいんです、 CASEというのは、コンピュータがそれを支援する からいいんです。

なぜ、コンピュータで支援することに重要性があるかというと、人間はミスを犯すものですから、コンピュータの支援によってミスを減らして、単純なミスですね。機械的なミスはコンピュータの支援によって減らすことによって、人間がもっとコンピュータの検出できないようなミスの方に集中出来るようにしなければならない。

なぜそこまでしてミスを減らさなければならないか、重大な目標があるわけです。コンピュータの誤動作は、社会に多大な影響を及ぼします。かつ、現在ほとんどの現代社会においては、コンピュータが多用されています。そこに、コンピュータの信頼性が必要なわけですね。

そのコンピュータの品質、信頼性を向上させるためには、上流工程が必要です、上流工程が上昇しますと、上流工程をきちんと実施するには、コンピュータによる支援が必要になります、上流CASEが必要になるわけですね。(制限時間の音)

司会:

結果発表です。全て肯定側です。 肯定側(ぱちぱち)

[講評]

菊地:

まず立論のところで、両方とも言いたいことがあるんですけど。これはあの模造紙に書いてあるんですけれども、項目がありますよね(違うかも)、論題があって、主張があって、理由があって、結論がある。

で、項目を埋めるわけじゃなくって、これはディベートの本にある、ストーリーって言うもの、ようするに、...(聞き取れず)に書くだけじゃなくて、ストーリーとして継っていないと。

そう言う面から見ますと、肯定側の方が、大体分か るんですけれども。立論の話しを聞いて、模造紙を見 ると、流れって言うのがはっきり見えないんじゃない かと。

それから、否定側の方なんですけども、論題は上流 CASEが必要である、ってなっている、それを否定 するわけ。そうすると、普通に考えると、必要でな い、となるわけですけれども、使えないとなってま す。

必要でないのと使えないというのは、紛らわしいというのか、本当にそう言えるのかどうか、疑問がある。でそれは、後から尋問の方でも肯定側から出ていましたね。

それからあと、最後の結論て言うのは、まぁ、必要でない、というようなことを表していると思うんですけど、ストーリーとして見た場合に、まず、論題と主張との関係って、いうのが、ちょっと疑問があるかなぁ、と思いますし、主張からその理由というのは、それはまぁ、分かると思うんですけれども、その理由からその結論が果たして出てくるだろうか。

そこの理由って言うのは使えないって言うことですよね、で、結論を見ると、使えないって言うよりは、 必要でないって言うのに近い内容になっている。そう 言うとこで、ストーリーとかが弱い。

まぁ、ストーリー云々とか、PCの説明とかもかなかったかも知れない、ことはあります。それからですね、最初の否定側の反対尋問が、...(聞き取れず)でね、ころころころころかわっちゃって、それで1分間の作戦タイムの後なんですけど、質問しかけて...(聞き取れず)。その辺はもう少し整理してやっていただきたい。

それから肯定から否定への反対尋問のところで、最 初のところで答えたのに対して、さっきこう言いまし たねと言って、却下しましたね。これはまずいと思いますね。

その時点で、尋問としては、さっき言ったのを肯定 します。って言うのは、だめだと、失言だとか言うの は、突いてきますから、それは、その時点では、やり 取りとしては負けだと。

それからですね、二回目の反対尋問は、結構どっちも良かったんじゃないかと思いますけど、最初のに比べますと。ただやっぱり、もう少し自分の方に引っ張り込むか、相手をどんどん攻撃してやっつけちゃうか、そこまで行かないといけないんですけど、行きかけて時間なくなっちゃうとか、あと質問考えてなかったりで、もうすこし... (聞き取れず)のほうが良かったんじゃないかと。

あと、最後の最終弁論ですか。否定側の方で理想論 だからいらないんじゃないかというのは、理想論だか ら必要でないというのは、良く分かんないです。別 に、理想であっても、必要性ある場合っていうのは、 ありますよね。それからあと、

佐原:

それは本来、理想と現実は別の話しで、理想的なCASEツールとか、そういうものを考えたら何の意味もなくて、そこを突かないと駄目ね、はっきりさせないと、ああ、ごめんなさい。

萄地

2,3年間は生産性が上がらない、それはそれでいいんですけど、その後どうなるか言わないと、攻撃したことにはならない。2,3年は上がんないんだけど、その後良くなるとか、その、2,3年悪いのをひきずって、ずっとおかしくなりますとか、その辺まで言わないと、攻撃したことにならない。えーっと、だいたい以上です。

伊藤:

内容的って言うか、良く勉強したな、と思うのは、 否定側の方なんです。だから、多分、ディベートには 負けたのだけれども、色々調べてきたなと、私は、否 定側の方を、ディベートは別にしてね、評価します。

で、例えばですね、肯定側はあそこに書かれた内容のことを言われていたんだけれども、主張は上流CASEは必要である、ってことで、それに関わることって、結局、何もないわけですよ。あそこにあるのは単に上流工程が大事ってことしか言ってないんですよね、で、かろうじて論拠の最後になって、キチッとした上流工程を実施するためには、上流CASEが要る、って書いてあって、それって結局、論題と同じわけなんですよね。

ということは、あそこは結局、上流CASE、上流 工程とCASEの関わりについて何も言っていない。

で、気が付いたんですが、昨日我々のチームがなぜ 負けたかって言うのを、今日この議論を見てようやく 分かりました。とりあえず、あそこにはわけの分から ないことを書いておく。

会場 (爆笑)

議論の運び方とか言うのは、明らかに、肯定側の方が上手ですね、と言うことでCASEを作っている身からすると、非常に楽しく聞かせてもらいました。ありがとう。

東田:

まぁ色々皆さん講評してしまったので、一つだけ言うと、どちらも提示されている事実とか、論拠って言うのをうまく使って質問したり反論したり、あまりしていなかったと思います。

たとえば、この事実はこう言う理由でどうなんですか、って言うような質問で、その事実を叩き潰せれば、それだけでもう、かなりポイントが稼げるんですね、で、答える時も、この事実、潰されていない事実から、その質問にうまく答られれば、ずっと信憑性が高まるわけだから、その辺もうちょっと、その場で、相手に質問されて答える時に、頭の中で考えるんじゃなくて、書いてある内容を使って負かす。

聞いている側は、分かりやすいという気がしました。

大塚:

話し方の分かりやすさからというと、やはり肯定側の方で。

否定の側の立論で、自然言語だから曖昧だとか、仕 様記述言語のことを持ち出されていましたよね。

多分山崎さんから、そう言ったお話を聞かれている と思うので、仕様記述言語の方の議論を、もっと盛り 上げていただければ、こちらにとっても、面白い話し が聞けたんじゃないかと思いました。

司会:

はい、どうもご苦労さまでした、あ、佐原さん、

佐原:

敗軍の将と言うことで。

肯定側の立論って言うのは、あれは、ほぼ予想通り であったと。 上流CASEは必要であると言う論題は、ディベートの題としては良くなくて、上流CASEを我社にいれるとか、そう言うことだと分かりやすかったね。

が、必要であるって言うと、もう明らかに理想的な、だから相手の土俵では戦わずに、まず現状のCASEって言うのはSAとかOOAをベースにしてるんだから、それは、方法論的に、こういった仕様記述が曖昧なんだと、それは今のやり方と変わんなくてね、だめじゃないですか、それからCASE自体も、色々問題があって、そう簡単には使えないね。かと言って、CASEがマイナスにになるかとまではちょっと言えないですよね。

だからマイナスじゃない、ミクロの進歩かも知れないけどもっと、フォーマルな仕様記述言語を使えば、もっとマクロな進歩が出来るのであれば、上位CAS Eってのは必要でない、使われないという言い方をしようとした。

まず戦場を限定して、戦いやすいところで戦おうと したんですけども、結局立論の時から反対尋問の1ま で、こちらが何もそう言う意図が言えなくて、最終弁 論で、あそこで始めて、こちらの主張ができたので、 なかなか戦えなかったなと。

もう一つは、やっぱり、途中で突こうとしていたけども、上流工程が大事だと言っているけど、やっぱり、上流CASEがそれに応えるものだと言うことは、何にも言っていない。

そこをどんどん、突けば多分、肯定側の方は、私の判断だと、CASEは使ったことないんじゃないかなと思って、技術的なところで、仕様記述が曖昧で、こう言う問題があるんだけど、どうするんだと言われたら、結構オタオタとし始めるんではないかなと、そこを突けなかったって言うのは、具体例まで行けば、結構戦えたかなぁと気がします。

山崎:

審判団が声が大きいとなんて言うのは、面白くないですね。主張をどうするって話しと全然違って、犬の 喧嘩みたいな気がする。

司会:

はい、どうもありがとうございました。 一同(拍手)



プロジェクト管理 論題

「SEは会社に出る必要はない」

記録 一原 武司(JMAS)

肯定側立論

(1) 論題

「大規模システム開発において、開発標準はいらない!」

(2) 主張

大規模システム開発において、 開発標準はいらない.

(3) 論題の言葉の定義

開発標準・・・開発の流れを定めるもので、改定 が行われにくいもの.

(4) 理由と論拠

★現状

開発標準はあまり使われていない.

- 1. 標準的な開発プロセスが定義されていない。
 - 54% · · · SS'92 (サンプル数 500)
 - 20% ・・・ 若手'93 (サンプル数 10)
- 2. 定義されているが利用されていない.
 - 6 3 % · · · SEAMAIL Vol. 7 No. 1

(サンプル数 262)

67% ・・・ 若手'93 (サンプル数 10)

★論拠

○現状の標準はプロジェクト管理者がスケジュール を管理する為に作られている.



「プロダクト中心」

○ひとつの静的な標準を多様なプロジェクトにあて はめている.

問題領域によって開発の仕方が異なる.

ex)

制御系 ---+

事務処理系 -+

ではどうすればよいのか (プラン)

「プロセス中心」 (プロセスのモデル化)

(5) 結論

大規模システム開発において開発標準は役にたたない.

否定側立論

(1) 論題

「大規模システム開発において開発標準はいらない」

(2) 主張

大規模システム開発において開発標準は必要である.

(3) 理由と論拠

- 1. 開発者の共通理解を得るため.
- 2. 進捗管理を正しく行うため.
- 3. 結果を評価するため.
- 4. 組織全体のレベル向上のため.
- 5. 属人性からの脱却のため.

(プラン)

・開発標準自体の見直しも必要である.

(4) 結論

- ●システム開発失敗のリスク軽減.
- ●部門全体のレベル向上.

するために,

「開発標準は必要である.」

否定侧反对尋問

Q1. 立場を明確にして下さい.

a. 開発標準は必要であるが現状の開発標準では役にたたない、b. 開発標準そのものが必要ない、のどちらですか.

- A1. 今の標準は必要でない.
- Q2. それは開発標準そのものが必要でないと言っているのではないと解釈してよろしいですか.
- A 2. これから出てくる理想の標準を今話題にするのはおかしいと考える.
- Q3. それではこれからできるであろう理想の標準 にのっとて開発したほうがいいと考えますか.
- A3. もしそれができればそれにのっとて行ったほうがいいと考えるが、現状はそんなものできないであろう考える.
- Q4. 開発標準そのものが不必要だとはいってない のですね.
- A 4. 開発標準とはわれわれが定義している様に「開発の流れを定めるもので、改定が行われにくいもの.」と言うものだからこのような開発標準は必要ない.

肯定側反対尋問

- Q1. 自分の会社に開発標準というものはあるか.
- A1. あります.
- Q2. では、その開発標準にのっとて開発を行っているか.
 - A1. 基本的には開発標準にのっとて行っている.
- Q3. それぞれの開発にあった開発標準を用意しているのか.

否定侧反对尋問

- Q1. 我々は、システム開発失敗のリスク軽減につ ながると考えるがどう考えますか.
- A1. プロジェクトにあった開発標準を考えて行けばもっとリスクは軽減されると考える.
- Q2. 開発標準は必要ないというが実際の開発の仕方などは誰がどの様にして決めるのですか.
- A2. プロジェクトリーダなどが世の中でいいと言われていることの中から選んで決める.
- Q3. プロジェクトリーダによって開発のしかたは 異なるのか.
 - A3. はい.

Q4. 開発期間中にプロジェクトリーダが変わった 場合, あらたなリーダのやり方で開発をやり直すの か.

A 4. そのプロジェクトに合ったやり方であればい ちがいに変わるとはいえない.

Q 5. ではどうしてそれを開発標準といわないのか。

A 5. それは個々のプロジェクトによって手順てきなものを持たせれば良いのであって特に開発標準とは言わない.

肯定侧反对尋問

Q1. 開発標準の見直しが必要といっているがそういう意味でこちら考えとおなじか.

A1. そうですね.

Q2. 開発者の共通理解を得る, 属人性からの脱却 とあるが開発標準はホームを喪失と考えるか.

A2. そうですね.

Q3. それは我々の主張と一致すると言うことで反 対
尋問を終了する.

最終弁論(否定側)

我々は,「大規模システム開発において開発標準は 必要である.」と主張してきた.

肯定側もこちらの意見と同様といったのでこの主張 は認められたと考える.

現在の開発状況は日々の開発におわれ開発全体のレベル向上に欠けているのでは無かろうか.

また開発規模の拡大、システムの複雑化に個人レベルでは追いつけない.

開発標準があれば可能である.

しかし、開発標準そのものを見直す事も必要である.

最終弁論(肯定側)

我々は、「大規模システム開発において開発標準は いらない、」と主張してきた.

開発標準を実際ほとんどの開発が使用していない.

それはプロダクト中心だからである.

品質,生産性を向上したければ,プロセス中心でなければならない.

これは、標準と言う1個に固定したものでは無くて、プロジェクトにあう手順であれば良い.

プロジェクトのあった開発手順さえあれば、開発標準は要らない.

 $\triangle \times \triangle \times \triangle$

第11回SEA夏のプログラミング・ワークショップ ポジション・ステートメントと感想

Aグループ

01 藤本 泰

性別:男 年齢:33 血液型:A 会社:中電技術コンサルタント(株) 部門:技術開発本部 情報事業部

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

コンピュータにより統合化されたエンジニアリング 業務システムの開発を約6年間手がける。(某自動車 メーカの開発部門に所属)

今春より現会社に入社し、エンジニアリング業務シ ステムのコンサルティングを担当する。

(2) 現在利用している開発環境

社内:大型汎用機+WS端末 EWS,パソコンをLAN接続 主なクライアント:大型汎用機+WS端末 WS,パソコンをLAN接続

(3) 関心のあるテーマ2つ

■ オブジェクト指向

非定型業務の多いエンジニアリング分野における データベース化には特に有効であると思っている。

約5年前頃からその基本的な考え方が提唱されてきたが、そろそろ、実務に耐え得るDBMS等が出回っているようで今後の業務で実用化を検討していきたい。

QC, TQC

ソフトウエアの品質保証についてISO9000-3などで 規格化され、ソフトウエア開発の品質管理が今後重要 視されてくるであろうと思う。QC, TQCなどの品質管 理手法及び品質管理活動は従来よりハードウエア・ メーカでは盛んに取り組まれてきたが、設計作業が重 点を占めるソフトウエア分野への具体的な適用には困 難な面があると思う。

QC、TQCの手法を過去の開発事例の解析に適用し、 ノウハウを可視化した上で、設計段階に品質を作り込むという考えの品質機能展開の手法を適用する事で最 終的に品質保証につながっていくのではないかと思 う。

(初日のみ参加)

03 中尾 吉克

性別:男 年齡:28 血液型:A

会社:(株)高岳製作所

部門:開発部 システム技術開発課 特記事項:タバコの煙にかなり弱い

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

- ・電力向監視制御システムの設計・制作
- ・業務機械化システムの分析
- ・社内向UNIXプラットフォームの分析・設計

(2) 現在利用している開発環境

- · Sun Sparc Station 2 + X terminal
- Ethernet + TCP/IP
- · X11R4 (OPEN LOOK & Motif)
- ・HP9000シリーズ,ストラタスXAシリーズ

(3) 関心のあるテーマ2つ

■ オブジェクト指向

~当社の実状~

オブジェクト指向×××(プログラミング,分析,設計etc)と言われて,かなりの月日がたつ中で,当社でも数年前から様々な調査が行なわれてきた.

しかし、なかなか実業務としては取り扱わない、そ こには, 例えば, C++では遅いとか言うだけでなく, わざわざC言語で済むものをという保守的な意見や、 何かわかりにくい(今までのプログラマ)などの意見も 多く, いやむしろ現実的な評価よりこのような現場中 心的な物の見方のために、仮に高い評価を得ていたと しても採用されなかったのかもしれない. しかし, こ こに来てソフトウェア設計におけるエンドユーザ指 向, ソフトウェア規模の増大, 複雑化に伴う信頼性, 保守性の強化, C/Sシステムで代表される分散ネット ワーク・システムの増加などにより、今までの単純な ウォータ・フォール・モデルのシステム構築ではとて も対応しきれず、またより人間に近いモデルによるシ ステムを構築していくという意味においても、オブ ジェクト指向プログラミングというより、オブジェク ト指向分析・設計というものが重要でないか、また個 人の思考そのものもオブジェクト指向のような捉え方 をしなければならないのではないかというように少し づつではあるが変ってきている. 上記のような考え方 で、実際に分析・設計まではオブジェクト指向的な方 法で開発を進めていき、製作においては今までどおり にC言語などで開発をしていくということを行なって いる会社があるということを聞いたことがある. この ようにオブジェクト指向という考え方は、1つの小さなワク(プログラミング)からより上流の工程をも抱括して、さらにはマルチメディア技術を始めとするあらゆるソフトウェア技術の根本となりうるのではないかと思う。

■ プロジェクト管理

~実務と講習から~

プロジェクト管理、そこには様々な管理(ここで詳細でいう必要もないと思う)を内抱したものから成り立つ、その1つひとつを扱うには、難解な手法と多くの経験を必要とすると感じる、とくに私のような、コンピュータの"コ"の字も知らないものが、数百K~千数百Kステップにもなるソフトウェア開発のプロジェクトを管理している人たちを見ると、まるで神様のような存在として写る、そこには、数百人の人(プロパーと外注を含む)がいて、また様々な異機種のH/Wがあり、人と機械をまこと巧みにあやつるさまは、複雑なからくり人形を扱う人形師のごとくである。こういう人たちを神と呼ばずして(神といっているが、自分は無信仰、無神論者だけど・・・)何といえばよいか、しかし、現実は人形を操るような簡単なものではないようだ。

ところで、プロジェクト管理を評価する上で、大きく2つのことを考える必要があると思う。1つにはプロジェクト管理を行なうための手法、また1つは、プロジェクトを行なうのは人間であるという点からの人間管理的な側面の2つを考える必要があると思う。

まず、手法であるがこれは比較的簡単(人間管理的な側面と比較してということだが)であり、理論的・定量的に扱え、またそれぞれの管理を存在し、実際にかなり有効なものとしてひろく広まっているようである。例えば、コスト見積もりのCOCOMO、スケジュール管理のPERTやCPMなどである。こういった手法の上では、プロジェクト管理はほぼ完全に行え、動作していくかに見える。そして、上長(システムをよく知らない人ほど)はプロジェクトがうまく行くと感違いしてしまう。

しかし、現実には、2つ目の側面である人の問題が入ると混沌とした暗の中にさまよい始めてしまううだ、プロジェクト管理はしなければならないし、実際に行なっている、プロジェクトを成功させるために様々な管理手法を用いるが、一番大切なことは、そのプロジェクト組織にいる人をどう操作していくかがイントのようだ、個人の能力が高く、コミュニケーションもうまく行けば、管理の多少の悪さを取消してしまうこともよくあるようである、プロジェクト管理の評価ということにはならないかもしれないが、まず、管理面(もちろん、技術面を含めた)という土俵をしっかりと標準化なりを行い、顧客、外注、プロパー

という様々な人の間を人間的側面からうまく操作できたプロジェクト管理が良いプロジェクト管理と言えるのではないかと思う.

2. 感 想

今回の若手の会の参加の目的は、世間の注目をあび ている技術の掌握と、他社の方々の技術力の自身との 違いを見たいということで、参加しました。今回の4 つのテーマ(プロジェクト管理、開発プロセス、CA SE、オブジェクト指向)は、当社においても議論の 的になっているのですが、なかなか明確な結論がでな いまま世間の動向に任せてしまっているという感じで す。その中で、今回のセミナーに参加し、感じたこと は、自分自身の技術力の低さ、問題としている次元の 低さを目の当たりにし、ひどくカルチャーショックを 受けたことは認めざるを得ないところです。また、自 分自身に対する自信の喪失と今まで行なってきたこと への不信感に陥っているところも、実はあります。こ ういった反面、自分の性格にもよるのでしょうが、今 回のテーマを非常に興味深く、これらの世界に足を踏 み入れたいという衝動にもかられています。今後の個 人的な技術修得の分野が明確になったような気がしま

さらに、このようなセミナーだけでなく、様々な分 科会にも参加し、いろいろな話を聞いてみたいと思う ようになりました。

最後に、良い意味での失意と興味を与えてくれる場所を設けて下さった、中来田実行委員長、グループAのPCである中谷さん、伊藤さん、さらに各実行委員の方々、グループAのみなさんに深く感謝します。

05 高村 智子

性別:女 年齡:27 血液型:AB

会社:山一情報システム(株)

部門:開発第1部

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

汎用機による証券業務のアプリケーション設計・開発を入社以来4年弱担当清算のシステム、特に ATMのシステムであるため主にオンライン処理が中心であった。今年に入りシステム再構築の分析業務を担当し、オブジェクト指向技術を用いた分析作業を勉強しながら進めている。

(2) 現在利用している開発環境

ハード的には、開発専用の大型汎用機を用いて、COBOL, FORTRANの言語を使ったプログラムを2~3人に1台の端末を使用して開発している。コンパイルや

テスト環境等は社内で簡単なユーティリティーが準備 されており、ある程度はコマンドを覚えなくても開発 できる環境となっている。

最近では、8人程度に1台のパソコンを導入して一部で4GLを使用したシステム開発も進んでいるが、まだ社内OAにとどまるか、ネットワーク対応のないものでおわっている。最新動向にあわせた環境を作るには費用がかかりすぎるため、整備が遅れているのかと思われる。

開発方法論としては、MIND-SHとHI-PASEをアレンジしたAURORAという手順を作成して使用しているが、過去の資料が不足している事などにより分析のための調査等に非常に無駄が多い。まだ自己流の人も多い。私としても、分析等については経験不足のため方法論をうまくつかいこなしていないと思う。

(3) 関心のあるテーマ2つ

■ オブジェクト指向

オブジェクト指向による分析作業は、ユーザーから 業務の現状を聞き出すのにはとても良い手法と感じ る。特に分析経験の少ない人間が行う場合は、かなり 有効に思う。また、分析後システムの規模について計 るときにも有効であり経験の少ない若手が中心のプロ ジェクトチームでは今後この手法が望ましいと思う。

しかし、実装の段階を考えるとパフォーマンスが悪くなってきたり、ファイル設計で無理が生じたりするという不備が有るように思うが、私自信が今までの汎用機での頭でしか設計できないせいもあるかと思うのでオブジェクト指向を用いた設計開発につながるスムーズな手順があるとよいと思う。

■ プロジェクト管理

システムの規模が大規模化しているため、開発方法 論に基づいたプロジェクト管理が良いとされている。 私たちの会社でもMIND-SAをアレンジしたAURORAとい う開発方法論に従って行っている。ところどころ自己 流になってしまうが、進捗をはかることに絞ればうま くいっているのかとも思う。しかし、実際の進捗より も計画の方がうまくできていないように思う。決まっ た見積もり技法がないため、かなりいい加減で人に よって異なる点でプロジェクト管理の甘さを感じる。

2. 感 想

今回の研修で中心となったことは、ディベートに勝ったことであった様に感じてしまいました。私にとっては大変良い勉強になり、結果として良かったのかもしれませんが、もう少しオブジェクト指向について等他社の話を聞いてみたり、わからないところを聞いてみたかったと思います。(しかしその前に、自分が勉強不足で用語など全然知らない事に気づいた状態にいる事は大問題であるが・・・)

私自身はあまりもの事を深く追求せずに生きてきたのですが、PCの方々は遊びでも仕事でも興味をもった事はとことんやりぬくという姿勢が羨ましく、また、今後見習いたいと思いました。

私は今、オブジェクト指向を用いてシステムの再構築を行なうプロジェクトにいて大変めぐまれていると思っています。今はこれが興味のある事ですので、成功にむけてがんばっていきます。

その前に、1週間ハワイに行ってまいります。成田で書いているので内容があまりないことをお許し下さい。

最後に事務局の皆様、お世話になりありがとうございました。

11 広沢 博

性別:男 年齡:28 血液型:A

会社:オムロン(株) 部門:EFTS統轄事業部

開発センタ第2開発室第1開発課

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容 モニタソフト変更作業 言語ライブラリ修正作業 デバッガテスト作業

(2) 現在利用している開発環境 ワークステーション・ネットワーク環境 PCセルフ開発環境

(3) 関心のあるテーマ2つ

■ プロジェクト管理

プロダクトの品質は、このプロジェクト管理の手法 のレベルアップが必要と言われるが、その背景には固 有の技術に頼るところも多いのではないか。

ならば、その技術修得もプロジェクト管理の中の重要な一要素と考え、その具体的な手法を確立しなければならないのではないか。

CASE

便利なツールのように思われているところもあるが、その原因は導入にあるのではないか。CASEは利用を論じるよりも、導入の段階の理解を促すための手法を明確にしていかなければ、十分にその効力を目に見える形にできないのではないか。

2. 感 想

前日に「明日金沢へ出張」と言われ、準備などできないまま参加した若手の会。しかし、参加して良かったと今思っています。

会社の中では、せいぜい数個のソフトの世界しかなく、それ以外の世界は未知のものでした。と言うことは、私にとって今回の会は、未知との遭遇だったのです。その未知との遭遇で得られたことは、これから私はソフトウェア(ただし、今私が触れているもの)を、どのように見ていくのか、触っていくのか、使っていくのかという、いくつかの課題を自分自身に与えることができたことです。

世の中で、ソフトウェアの技術が叫ばれ、今回もそういう話題がディベートの論題となりましたが、そういう問題(特に、ソフトウェアの技術とは何だろうということ)を、セクション内等でディスカッションしてみたいと思います。(そういうことが、メールやニュースシステムでできる環境が、オムロンにはあるんです。いいでしょ。これからは、ネットワークの有効活用時代ですよ。)

グループのみなさんには無知なリーダーで御迷惑をおかけしましたが、私は得るものが多く、これからの会社での生活が楽しくなりそうです。またみなさん方とディスカッションできる日を楽しみに筆 (キーボード)をおきます。

最後に実行委員長、PCの方々、実行委員の方々 に、深く感謝いたします。

16 加藤 昌也

性別:男 年齡:26 血液型:B

会社:MHIエアロスペースシステムズ(株)

部門:技術部開発グループ

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容
 - · 人工知能試験研究
 - ・社内用ソフトウェア開発支援ツールの構築
- (2) 現在利用している開発環境
 - ・WS (HP9000シリーズ400)
- (3) 関心のあるテーマ2つ
- オブジェクト指向

私が最初にオブジェクト指向に触れたのは、C++を使ったときである。このときは、「オブジェクト指向言語」といっても機能は増えたけれどもC言語とあまり変らないのではないか、と感じた。このときのC++に対する印象がそのままオブジェクト指向に対する印象となって、オブジェクト指向は今までのやり方と比

べても、それほど使って嬉しいものでない、というのが私のオブジェクト指向に対する第一印象だった。 もっとも、このときはオブジェクト指向がなんたるか をほとんど知らなかったのだが、その時よりは多少は 知識も増えた。今でも、この印象はほとんど変ってい ない。

なぜかというと、今オブジェクト指向がもてはやさ れているが、現状では理論ばかりが先走りすぎてい て、実際の環境がついてきていないような感じがする からである。オブジェクト指向に関する本はたくさん 出されているが、実際に使ってみようと思っても、サ ポートしてくれる手頃なツールがないとか、今までに はない苦労をしなければならない(設計方法は大きく 変るためそれに伴う苦労)とかで、なかなか最初の1 歩が踏み出せない。もっと、OOAD用のCASEツールとか 言語等が出てきて普及しなければ、少し前のAIブーム と同じで単なるブームでおわってしまう可能性が高 い。特に下流工程でのサポート (言語、デバッガ等) が普及しなければ本当にブームだけで終ってしまうと 考えている。なぜなら、オブジェクト指向は継承、多 相性等、従来の言語にはない表現方法を用いているた め、それを実現できる機能を有する言語がないとその 有効性は発揮できないからである。

例えば、オブジェクト指向を用いて、分析、設計を行い、ふつうのC言語でコーディングを行うことを考えてみる。まず、コーディング時において、設計書においては、継承情報ということで復数箇所に記述されている情報を一ケ所にまとめるという無駄な労力を費やすことになる。

また、保守の段階では、上記の作業の結果として、 ドキュメントとコードの表現が異なることになるの で、ドキュメントとコードの対応付けに余分な労力を 費やすことになる。このように、いくら設計時に威力 を発揮できるとしても、後々苦労しなくてはならない のでは、全体的にみると実用性に欠けると言わざるを 得ないからである。

オブジェクト指向についてよく聞く話で、オブジェクト指向を用いると再利用性が高まり開発効率が向上する、というものがあるが、私はそうは思わない。その理由は2つある。

1つめは、再利用を考慮したものを作るには、その場限りのものを作るのと比べて、かなりの労力、時間がかかるということである。この開発効率向上という話は、再利用できる部品が存在するという前提で成り立つものであり、再利用可能な部品を作るのに費やす時間のことを考えれば、全体で見るとそれほど変らないと思う。2つめは、すでに再利用可能な部品があることが明確な場合は別として、あるかどうか判らない物を既存の部品から探しだすのには、新たに作るのと同じぐらい時間がかかるということである。

smalltalkのクラスライブラリのように何百もある中から、自分が必要としている物を見付けだすにはかなりの時間を要することは容易に想像できる。そのうえ、もし必要なものが見付からず、類似のものに手を加えるとか、新たに作るとなると、最初から新たに作った方が早く作れるかもしれない。このようなことを考えると、開発効率が向上するのは非常に運が良かった場合だけで、たいていはオブジェクト指向を用いなかった場合と変らないのではないかと思う。

CASE

CASEについての私のイメージは、ある手法を用いてシステムを構築する時に紙と鉛筆ではなく、ディスプレイ、キーボード、マウスを用いて行える物、というものである。そういう意味では、ワープロ、図形エディタ、デバッガ、等を組み合わせるだけで、十分CASEといえると考えている。

つまり、いわゆるCASEツールという名前で市販されているツールが出てくる前の作業方法でも、CASEを用いてソフトウェア開発をおこなっていたと、私は思っている。しかし、このような形では、データの管理は作る人間が行わなければならないし、各ツール間でデータのやりとりは望めないし、同じデータを何度も入力しなければならないかもしれない、といった欠点がある。その反面、このやり方では、設計手法に縛られずに自由に作業ができるという、大きな利点がある。

現在市販されているCASEツールは、これらの長所短所が、完全に逆になっている。ツールを使用している分には、ある程度はデータ管理はツールが行ってくれる。しかし、これらツールは特定の手法に縛られてしまう。さらに、元にしている手法を忠実に守っているため、融通がきかないと言う点も、欠点としてあげられると思う。例えば、手法の持っている表記方法ではどうしても表現しきれないためコメントを付け加えたいのだが、CASEツールではそれができない、といったことが考えられるが、私の経験からいうと、このようなコメントが後で重要になってくることがよくある。この点が、私が、現在のCASEツールに対して持っている一番の不満であり、この点を考慮すればもっと使いやすいCASEツールができるのではないかと考えている。

例えば、記述用に2つのレイヤーを用意しておいて、1つには手法に基づいた図を記述し、もう一方にコメントを記述できるようにする、といった方法は、比較的実現しやすく、かつ有効な方法だと思う。また、この考え方は邪道かもしてないが、手法のカスタマイズ機能を付けると良いのではないかと思う。さまざまな設計手法が出されているが、これは裏をかえせば、どの手法も万能ではないということを表している

といえる。したがって、そのような手法を元にしているCASEも万能ではないということになる。そこで、ユーザーが自分たちの方針にあうように表記方法等をカスタマイズすることができれば、より使いやすいものになる。

このようにCASEはもっと柔軟なものであるべきだと 考えている。

2. 感 想

今回初めて「若手の会」に参加したのですが、全く 堅苦しいところもなく、楽しく学ぶことができまし た。

今回のディベートは初めてだったこともあって、多 少の不安はあったが、なんとかなるだろう、という軽 い気持ちで望んだが、それほど甘いものではなかっ た。

何を論点にすればいいのか見極めれない、一言でいうとその倍以上言い返されてそれに反論できない、 等、ディベートをする以前に、如何に自分がテーマに 対する勉強が足りないかを思い知らされた。

ディベートについては、状況に応じて話す内容が変化していくため、かなり難しいと感じたが、やってみると意外と面白いので、機会があれはやってみたい。



Bグループ

02 大野 修一

性別:男 年齡:28 血液型:AB

会社:(株)日本NCD

部門:大阪支社 技術

特記事項:タバコの煙にかなり弱い

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

リレーショナル・データベース 4GLのサポート・エンジニアリング Xターミナルのサポート・エンジニア

(2) 現在利用している開発環境

NEWS, SUN IPX +Xターミナル

(3) 関心のあるテーマ2つ

■ オブジェクト指向

オブジェクト指向が発表されてかなり時間が立っている。私自身も興味があり色々勉強してみてインへリタンス,カプセル化などの考え方があるというのは理解できた。でも実際にシステム(プログラム)をどの様に構築するのか、また、いままでの方法と比べてどのくらい良くなっているのか掴めていない。

CASE

4GLをサポートしていたときに興味があり自社の CASEツール(もどき)を使用してみたことがある。

これは下流のCASEツールであったが私が使用しているレベルでは非常に便利に思えた、しかしデータ入力を漢字で行なうために時間が掛かってしかたがなかった。このような事は構築するシステムが大きくなればなるほど手間をとられるような気がする。

あとCASEツールの有効性を説明する時に困ったことがある、それは、CASEの効果はすぐに現れないこと、 CASEの効果を測定する方法がないこと、結構コン ピュータ資源を消費することであった。

個人的には非常に興味があるがそれだけでは企業で 通用しない(あたりまえ)ので企業(システム部)を どうやったら説得できるか悩んでいたが、転職により 4GLのサポートから離れたため結論は出ていない。

2. 感想

私はSEAに入会してかなり長いのですが、今回のようなワークショップに参加するのは初めてでした。

私が選んだテーマは「オブジェクト指向」と「CASE」で、しかもディベート形式という事で、事前に勉強はしたものの金沢に着くまでの間に自分の意見をまとめながら「これでやっていけるのだろうか?」と不安になっていました。でも、いざ始まってみると「やるしかないんだ!」と開きなおり、ワークショップ終了まで十分に楽しむことができました。

いつもと違って今回のワークショップはテーマではなくてディベートをメインに考え、ディベートの方法、相手の欠点を突く、相手を自分達の土俵に引っ張り込むなど、普段実行していてもあまり意識していない討論の方法を学ぶことができました。特に肯定側・否定側が完全に分かれ相手を論破するために知恵を絞りテーマを深く追求することは今までのワークショップになかったことだと思います。

私は少なくても1年に1回はこのようなセミナーに 参加してまったくの他人と接し自分の知識・考えをリ フレッシュしていきたいと思います。

それから、チーム・リーダーを任せていただいて有 難うございました。非力なリーダーで申し訳なかった と思います。やはり、日頃の作業の進め方がこういう ところで現れてくるのですね、少し改善しなければな らないと思っています。

あとワークショップと直接関係は無いですが、私を含めてBチームのメンバーで瞬間的に漢字が書けない人が何人か見受けられました。私は普段電子辞書を持ち歩いているのですが、文明病(ワープロ病?)というのでしょうか、漢字や意味は理解しているのですが「書けない」状態に陥ることが多く、ちょっと問題かな?と思っています。

最後になりましたがPCとして東田@PFUさん、 菊池@東北電力さんがBチームの迷える小羊たちに救 いの手を差し延べてくれました、感謝しております。

ありがとうございました。

07 根岸 易世

性別:男 年齡:28 血液型:B

会社:(株)ニコンシステム

部門:第1システム本部第3部

特記事項:酒が飲めない

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

MS-Windows を使用した機器制御のためのユーザインタフェース

(2) 現在利用している開発環境

CPU:NEC PC-9801 DA OS:MS-DOS MS-Windows 言語:MS-C

(3) 関心のあるテーマ2つ

■ オブジェクト指向

熟練を必要とし、多角的な視点による評価をしなく てはならない.

導入直後はむしろ生産性は悪くなる.

■ フォーマルメソッド

2. 感 想

終わってみれば...

このワークショップの締切の当日、上司から「こんなのがあるんだけど、行くよネッ!」と"ネ"を強く言われてしまった。まぁ、ここしばらく何の講習会にも行っていないし、と自分に言い訳をしつつ、「それでは行かさせていただきます…」と渋々。実を言うと、自分程度のレベルでまともなディベートが出来るのだろうかと不安であった。

不安と言うものは、たいてい的中してしまうもので、回りのレベルの高いこと。聞くとみんな難しくて、必死で理解しようとしていた。さらに弱り目にタタリ目、ディベートでは「ココー番」の時の弱さが出てしまい、発言の時には緊張がいきなりピークに達してしまい、言っていることがメチャクチャになってしまった。最近は随分と図太くなって、大抵のことは平静を装っていられると思っていたのだが、まだまだ修行が足りないな。

しかし、終わってみての本音を言ってしまえば「楽しかった」の一言に尽きる。ディベート中の心地よい緊張感(発言の時を除く)、そして何よりも、ディベートのためのグループ内の作戦会議での白熱した意見の交換が(内容はともかく)素晴らしかった。

様々な視点からの意見・論点、当然ぶつかりあうものもあるが、それらが互いの論理の欠点を指摘・補足しあいながら一つのまとまった?結論へと収束していく様は、何にも得難い快感を感じる。

こんなことで快感を感じるなと言われそうであるが、考えてみれば、ある目的を持ち、様々な情報からひとつの結論を導くことは、モノ造りと通ずるものがある。消費する側の要求を目的とし、形状や機構、素

材の特性をひとつひとつ吟味し、そのなかから目的に合ったものをピックアップする。それを組み合わせることによって結論となるモノを造り上げていく。作戦会議で行った情報の収集・分析・論理の組み立て・結論の導出と合致する。技術者と言われている人が全全とは言わないが、モノを造ることに感動や快感を感じたの道を選んだはずである。私自身、まだまだないがソフトウェアというモノを造ることに感動を持成ないがソフトウェアというモノを造ることに感動を持ったからである。そう考えれば議論しあうことに、快感や喜びを感じたとしても不思議はない。参加したやも技術者である以上、やはり同じようなことを感じたとしてもではないかと思う。結構、みんな熱中してやってたもんな…。

また、1グループの5人は数としては少ないが、それでもやはり、意見を交換しあう中で、自分とは立場を違えた意見に数多く出会え、思考の新陳代謝を行えた。一人で(たま~にだが)勉強なんぞしているときには、思考の方向が固まってしまい、その脱出に非常に苦労する(出来ないときのほうが多い)ものであるが、自分とはまったく異なる考え方が刺激となり、パチンコ玉のようにはじかれ、あっさりと脱出をはかり、広範囲にわたる考察ができてしまう。それに加え、意見を交換しあう中から自然発生?した世間話(言い方が悪いが)は、会社の中に閉じこもっていては絶対得られない貴重な体験を擬似的にさせてくれ、休止ぎみであった好奇心を活性化させてくれた。

さて、意見の交換は各自の持っている知識の共有化 と視点の違いを認識する等のために、必要不可欠なこ とである。今思えば、ここで細心の注意を払わなくて はならないことがあった。議論が発散しすぎないこ と、こだわりすぎないことである。情報は多いほうが いいが、多すぎると取捨選択の判断が複雑になり、場 合によっては人間の能力を越えてしまう。こだわりや 議論の固着が出てしまえば、情報が減るとともに、視 点が固定してしまうために論理の弱点を見落としてし まう。したがって、全員が同じような方向を向き、あ る幅を持つ帯の上で縦横無人に交差しあうのが理想的 なのだが、これが結構難しい。これに関しては、PC の方が議論に参加しながらうまいこと制御してくれた と感謝しています。話がつまった時にはネタを提供し てくれましたし、発散ぎみになったときには方向を示 して頂けました。

「ディベートを通し、テーマとなっているものの認識を深める」のが目的であったが、それはそれとして、それ以外で得られたものの方が印象に強い。例えば、ディベートそのものに関しても、実際にやってみ

て、一般に言われている「ルールに従った討論」以上 に、表現方法や情報分析の訓練、さらにはモノ造り全 体を通しての思考回路の訓練に有効なこと(特に新人 研修でやると良さそうだ)。自分とは異なった環境で 生活している人と話を出来たこと自体、重要な収穫と も言える。

最近、ほとんど会社の外へ出ることが無かったため、ある意味で、保守的な状態であったと言える。今回、このワークショップに参加した(させられた?)ことで、この状態を打破でき、またモノを造ることに感動することが出来そうだ。今後とも機会が有ればまた参加したい。

また、本来グループ活動は苦手なうえ、勉強不足のせいでだいぶ足を引張ってしまったと思いますが、それにもかかわらず、私の決してまともとは言えない性格を受け入れ、短い期間とはいえ、一緒に活動してくれたBグループみなさんに感謝します。特にPCとして参加して頂いた東田さんには、解散後の観光でガイドまでしていただき、本当にお世話になりました。ありがとうごさいました。

10 丹羽 徹

性別:男 年齡:28 血液型:A

会社:オムロン(株)

部門:技術本部開発支援センタLSA

1. ポジション・ステートメント

- (1) 現在までの主な仕事の内容
 - 1). 社内外向けソフトウェアの開発/管理 (担当したおもなシステム)
 - ・キーボードオペレーションシステム
 - ・運用支援ツール
 - ・メッセージハンドリングシステム
- 社外システムのサポート
 WS販売支援のためのシステムサポート。
- 3). 社内へのCASEの普及

(社内のソフトウェア開発にCASEを導入するための以下の取り組み)

- ・ツールの導入/サポート
- ・方法論の教育
- ・適用方法の考案

(2) 現在利用している開発環境

1). ハードウェア LUNA 2 SPARC

2). OS

UNIX

3). おもなアプリケーション Emacs

(3) 関心のあるテーマ2つ

■ オブジェクト指向

mh

- ・現在のところ最も期待されている次世代のソフトウェア開発パラダイムである。
- ・どちらかというと、プログラミングが中心で議論 が進んでいる。
- ・利点は認めるが、企業での実開発への導入にはま だまだ課題が多過ぎる。

CASE

- ・開発工程に正しく取り入れられた時には相当な成果を発揮する。
- ・分析工程の重要さが理解され難い。
- ・ツールが誇張され過ぎる。
- ・小規模システムで導入するのは難しい。

2. 感 想

今回このプログラミングワークショップに参加した 感想を以下に述べる。

(1)作戦会議にて

- ・現在さまざまなポジションにおられる方々の意見 や内輪話などが聞けて非常に参考になった。毎晩遅くまでの議論であったが、眠気を感じさせないぐら い、議論は充実していたと思う。
- ・みなそれぞれ立場が違うためか、なかなか意見が まとまらなかった。そのためか、作戦会議は、若 干時間切れの感もあった。
- ・「主張」から「結論」までの話しの筋を通すのは 難しいことと、データによる裏付けは重要である ことを実感した(特に肯定側)。

(2)ディベート中

- ・今回ディベートというものを初めて体験した。
- ・一回当たり、僅か1分という作戦タイムで、意見をまとめるのは酷である。経験がないというハンデもあるが、前準備が相当必要である印象を受けた。

- ・一応、我々のチームは、話しをつなげて尋問して いたそうである(緊張のため、あまり覚えていな い)。
- ・否定側は、あの手この手で攻めてくるので、テーマの知識だけでは苦しいのがよくわかった。
- ・第3者として見ていると、話し手の論理のおかしな、穴などがよくわかる。

しかし実際にやると、なかなかそうはいかない。 こを克服するためには、やはり訓練か…。

(3)終ってから

- ・いつもあるテーマに対して、「こう攻められた時 にどう答えるか」など、自分なりに意見をまとめ ておくことが重要である。現在は、あるテーマに 対して自分の持っている考えが漠然としているこ とが多い。自分なりにしっかり意見をまとめてお かないと、考えをアピールできないことを今回痛 感じた。
- ・一般に肯定は難しい。しかし、善し悪しは別にして、今回は与えられたテーマに対して、自分達に有利なように最初の主張を行なうことができたのが、勝因の一つであると思う。
- ・ディベートを2回行なって、少しはディベートに慣れたと思う。しかし、「本当のディベート」というものに関しては、自分ではまだよく理解していない。今回、「本当のディベートが」を知るためのお手本になるもの(ex.ビデオ、PCによる実践等)があれば良かったと思う。
- ・ディベートとは、恐ろしい武器である。問題点があっても、それを相手に悟られないように、話術巧みに進めれば勝ちである。まあ、それにいかに引っかからないようにして、相手にボロを出させるかが対抗側の役割であろうが…。
- ・中身はともかく、とにかく2勝したので、勝利の 美酒が飲みたかった。今となっては、これだけが 心残りである。なぐさめとしては、台風で中止に なる前の夜に、後先考えず、我々のチームに割り 当てられた食費を派手に使い果たしてしまったこ とか…。
- ・日程的には辛かったが、非常に充実した3日間 だった。機会があればまた参加したいと思う。

4. 最後に

・今回のプログラミングワークショップで、ディベートに関していろいろ助言を頂き、夜遅くまで議論につき合って頂いたPCの方々、私の支離滅裂な意見にもかかわらず、それに対して議論を続けて頂いたBグループの皆様、技術講演で貴重なお話をして頂いた松原先生、およびその他関係者の方々に感謝致します。ありがとうございました。

13 棚橋 巡治

性別:男 年齡:27 血液型:A

会社:(株)SRA

部門:システム開発第5部

特記事項:タバコの煙は好きではない

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

主な仕事は受託ソフトウェアの開発です。過去に開 発に参加したシステムには、

- ・交換機の制御端末のソフトウェア
- ・テレビ局向け番組送出システムのソフトウェア
- ・既存 OS へ TCP/IP による通信機能の移植が、あります。

(2) 現在利用している開発環境

UNIX ワークステーション上で、X Window System、emacs エディタを用いて、C 言語による開発を行なっている。

(3) 関心のあるテーマ2つ

■ オブジェクト指向

オブジェクト指向と言われると"オブジェクト指向言語"ぐらいしか思いつきません。そして、そのオブジェクト指向言語も、C++をちょっと触ったことがあるだけです。

そのちょっと触ったC++言語の感想ですが、C言語によるプログラム開発でいつもやっているような事、

- ・関連のあるデータを構造体としてまとめる。
- ・特定の構造体を扱う関数群一式を準備する。
- を、言語の機能で明示的に出来ることが便利だと思ったぐらいで、別段「オブジェクト指向のご利益」を感じたわけではありませんでした。

ただ、C++ではなくて、別のオブジェクト指向言語だとご利益があるのかも知れませんが、他は知らないので分かりません。

■品質管理(TQC 他)

品質管理の話しですと、最近騒がれている"ISO900-3"が心配です。

私の現在の業務は、ソフトウェアの受託開発が主な のですが、近い将来にその開発契約書に、

・ I S O 9 O O O - 3 に基づいた開発を行なうこ と

と言うような、条項が書かれる日が来るような気がし ています。

と言って、恐れているだけではいけないので、ISO9000-3の内容を吟味して、それが我々の味方であるのか敵であるのか見極めて、対処を考えていきたいと思っています。

2. 感 想

今回のワークショップがきかっけで、いろんなこと を思ったり考えたりしました。それからいくつか挙げ ると、

プロセス成熟度について

専務がプロセスがなんだかんだと、言っていたのを、プロセス、って何だろう、ps で出て来るのとは違うんだろうなぁ、と思っていたのだが、このワークショップの予習で「アメリカの事例に学ぶ」を読んで、やっと何の事だかわかった。(しかし、97%がレベル1じゃぁ、ピンとないわけだ、冗談じゃなくて、本当にプロセ非成熟度、とかやってもらわなないと、実感できない、と思う。)

CASEについて

ディベートで、上流CASEは必要である、とか力説していたが、実際のところは、CASEのCの字も使っていない。

でも、ちょうど、近くのマシンにCASEツールをインストールしたところなので、ちょっといじってみようかな、などと考えている。

ただ、ディベートの敵方が、初期の学習コストがかかる、と、力説していたので、「ちょっといじって」ってわけにはいかないんだろうなぁ、

まずは、本買ってきて勉強するようですね。

てなところです、今回のワークショップでは、いろいろな方にお世話になりました、ここであらためてお礼を申し上げます。

17 荻原 郁子

性別:女 年齡:27 血液型:0

会社:(株)SRA

部門:システム開発第4部

特記事項:タバコの煙は好きではない/酒が飲めない(アルコールは駄目ですが、宴会は好きです)

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

・パソコン LAN システムユーザインタフェースの 開発

- アセンブリ言語オプティマイザの開発
- ・NFS4.0 ポーティング及び高速化
- ·OAシステム帳票作成
- ・新人研修教育スタッフ

(2) 現在利用している開発環境

- ・NFS & YP によるネットワーク環境
- ・ドキュメント作成は、troff 及び LaTeX を利用
- ・特にCASE ツールなどは利用していない

(3) 関心のあるテーマ2つ

■ オブジェクト指向

オブジェクト指向によるプログラミングは、

- ・抽象データ型によるデータと手続きのカプセル 化により、情報の局所化や、情報の隠蔽が可能 になる。
- ・インヘリタンスにより、差分プログラミングが可能になる。
- ・ポルモフィズムにより、さまざまな種類のデータを統一的に扱うことが可能になる。 などの利点がある。

これらの利点は、今まで私自身が行ってきた、手続き型のプログラミングに比べ、保守性や生産性などの 面において非常に効果が現れるであろう。

しかし、オブジェクト指向の採用をするにあたって は、多くの解決されるべき問題点がある様に思われ る。

まず、今までのシステム開発工程は、ウォータフォール・モデルによって進められていることである。

おそらくオブジェクト指向による場合には、この方 法は成り立たないであろう。しかし、現場にはプロ ジェクト運営のノウハウが全くない。

また、普遍的なクラスを作る為には、抽象能力が必要であり、この能力を育てる(あるいは発掘)する必要がある。

さらに、従来までに培ってきた開発技法を一切排除 して、一からオブジェクト指向で開発するのには、リ スクが大き過ぎる。従来技術とうまく融合させて、適 所に採用していく能力も必要であろう。

さまざまなクラスライブラリが、市販され始めて来 ているが、これにあわせてオブジェクト指向の CASE tool が現れれば、もう少し、開発現場でも採り入れ 易くなるのではないだろうか。

CASE

システム開発の各工程を自動化する為の仕組みを持った、様々なCASE 製品が販売され、現場での利用機会も増えてきているようである。CASE 製品が支援している方法や技法が、構造化分析や構造化設計をベースとしているものが多い為、これらを習得している者にとっては、さほど違和感なく導入できると思われる。人に依る作業がなくなる為に、システムの整合性や完全性が高められ、生産性や保守性も向上する。

また、CASE の中核となるリポジトリにより、開発の各工程で、整合性をチェックしたり、有機的なリンクを張ったりすることが出来る為、各工程間に跨る情報の参照が非常に楽になるのではないだろうか。

上記の様な利点があるにも関わらず、現場への CASE 環境の導入は、ようやく普及し始めたと言った ところのようである。

この原因を考えるに、CASE 環境を導入してシステム開発を行う場合、立ち上がりにかかる工数が、非常に大きい為であろう。また、CASE の導入によってどの程度の生産性が挙げられるのか、不明確である事も挙げられる。どの程度のリスクを想定すべきか、失敗した時の修復はどの程度必要か、そういったものが未知である為に、導入に踏み切れないプロジェクトが多いのではないかと思われる。

CASE の導入は、いわば新しい文化の導入であると考えられる為、開発担当者には、CASE tool が支援するモデル、構造化分析/構造化設計などをきちんと教育し基礎を固める必要があり、またマネジメント担当者には、CASE 導入による生産性と品質の向上を理解させる必要があろう。

CASE 環境でソフトウェア・ライフサイクル全体を カバーする為には、いくつかの使い易い製品を組み合 わせる必要がある。従って、リポジトリの内部構造の 仕様が公開され、共有できるものでなければ、現場と しては、工程に継目が出来てしまい、重複作業を強い られることになるのではないかと思われる。現在、内 部仕様が公開されているかどうかは不明であるが、も し、まだ公開されていないのであれば、是非とも標準 化を望みたい。

2. 感 想

(1) テーマ

今回取り上げられたテーマのうち、私の担当した テーマは

- ・オブジェクト指向は悪である
- ・上流 CASE は必要である

の2点であった。

現在仕事として取り組んでいる訳ではないが、非常に興味あるテーマである為、実際に取り組んでいる 方々から、どの様な情報が得られるのかが楽しみで あった。

「悪である/悪でない」、もしくは、「必要である/ 必要でない」という観点からの考察は、通常あまり注 意を払わない様な点をじっくり考えることになり大変 良い経験となった。普段、何かが必要だと思っている 場合、その理由を考えることはあっても、そこで敢え て「必要ではない」という考察を行うことはあまりな いからである。

(2) 形式に関して

今回行ったディベート形式というのは、初の試みと のことであるが、良い点も、あまり良いとはいえない 点もあったように思う。

良い点は、肯定/否定で意見を戦わせる為、自分の 普段捉えているのとは違う側面からも、テーマを眺め る必要が出来たということである。また、たとえその 意見が自分の本心とは違っていても、相手を納得させ る為には、自分の中で論理的に話を進められる様にし ておかなければならず、積極的な参加が出来た。

一方悪い点は、まず、ディベートの形式や勝負に拘り過ぎて、テーマの本質から外れてしまいがちだった点である。グループでの話し合いで、メンバの経験談やそのテーマに関する捉え方などから、良い話を聞けていたとしても、結局『ディベート』に立ち返ってしまい、途中で途切れてしまうことも多かった。

限られた時間の中では、どの様な形式を採用しても 出てくる感想ではあるかも知れないが、とても残念で あった。

それから、担当テーマ以外のディベートを傍観しているだけでは、その方々が取り扱ったテーマに関してはあまり理解が出来なかった点である。発表後、質問や意見などを取り扱う時間をとっても良かったのではないだろうか。

また、PC の方々には、ディベートの形式に対する 評価以外にも、テーマそのものに対する意見や感想な ども述べて貰えたらもっと良かったように思われる。 例えば、自分達が『事実』として取り上げた内容が妥 当であるかとか、世の中ではこんな風に唱われてるけ ど実際はどうだとか、そういう面からの評価もあれば 良かったように思う。

(3) その他

他の会社の方々と、きちんとしたテーマを持って具体的な話をする機会が得られたことが、何よりの成果だと思っている。特に、専門的な知識を持ったPC の方々やメンバも経験のある方々だった為、豊富な経験談を聞くことが出来た(その方々にとってみれば、私

からはあまり新しい情報を得られなかったことになる訳で、少し申し訳ないのだが)。

ディベートの経験も、きっと今後うまく利用できそうに思われる。特に、こちらの提案をことごとく否定する様なお客さんに遭った様な場合に、有効ではないかと思っている。

今回は、金沢で行われた為、私にとっては間接的な学習が、後2つ出来た。

それは、金沢城を一周するのには 40 分かかり、お城に入るのには石川門以外にないこと、そして、飛行機の乗り方である。:-) これらの知識は、今後大いに役立つと思う。今度金沢に行った時には、石川門を逃さずに入城できることであろう。

 $\triangle \times \triangle \times \triangle$

Cグループ

04 坂口 達司

性別:男 年齡:25 血液型:A

会社:(株)高岳製作所

部門:開発部 システム技術開発課 特記事項:タバコの煙にかなり弱い

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容 プログラミング開発
- (2) 現在利用している開発環境Sun Sp2hp, Stratus

(3) 関心のあるテーマ2つ

■ CASEについて

現在、いろいろなCASEツールが世の中に出てきています. 以前のものに比べると、格段にカスタマイズの柔軟性が増し、より多くのユーザの要求を満たすものが増えてきたと思います.

つい最近も,2つのCASEツールのデモンストレーションを見学しました. どれも,カラフルで,1時間 ぐらい使用すれば,ある程度のレベルまでつかいこなせる感じがしました.

しかし、どのCASEツールを取ってみても、「帯に短し、タスキに長し」と言った感じです。また、価格的に高いなあと思います。もう少し安価ならば、自社にも導入してもらえるでしょう。たぶん? 現段階においては、各領域ごとに必要なCASEツールを使用していけば、かなり満足できるものがあると思います。そして、ツールの機能も次第に向上していくでしょう。

良いシステムを作るためには、CASEツールはもっと 必要になっていくだろう.

■ QC. TQCについて

QCの目的は、「顧客の要求する品質を確保することができるように企業の品質目標を定め、これを合理的かつ経済的に達成するものである.」

この目的があったからこそ, 現在の高機能, 高品質な日本製製品が, 日本中はおろか世界中に出ているのだと思います.

しかし、人間の欲求とは、常に上へ上へと向けられており、新しい技術、新しい商品を要求しています. お客様が買うのはより良い商品であり、そのような商 品を作るためにもQCは必要不可欠である. 利益もそこから生まれてくるものであると思います.

ここまで、口でとやかく言うことは実に簡単である と思います。実際に"全員参加による経営"と言う考 え方自体は大変にすばらしいと思います。自分自身、 会社でQCの勉強を受けましたが、体質改善には至らな いと思いました。それは、自分が経営に参加している と言うものが感じ取れなかったためでしょう。それ は、QC教育が入社して2ヶ月目だったからでしょうか。

2. 感 想

今回若手の会に参加するにあたり、「不安」と「緊張」を抱いていました。その「不安」とは、「ほとんど予習というものをやっていなかった」ことと、「全く知らないメンバーとどこまで成果があげられるのか?」がありました。前者の「不安」だけが見事に的中してしまいました。そして、それプラス「緊張」とで「ここ一番に弱い自分を露呈してしまい、勝ちゲームを一転、負けゲームにしてしまい、グループの各メンバーとPCの佐原さんにはなんとお詫びしてよいか・・・。

後者の「不安」については、知らない同士でも同じ 目的を持つことにより、まとまりを持ってでき、自分 自身ここまでできるのかと感心しました。また、方法 論では、いろいろな勉強ができ、今後生かしていくこ とができると思います。

年齢的にも同じくらいで、同じ業界の人たちと作業 しましたがいろいろな人がいて、いろいろな考えがあ るのだなと実感しました。

06 堀井 剛

性別:男 年齡:25 血液型:AB

会社:(株)シーズ・ラボ

部門:システム部第2グループ

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容 地図情報システムのAP開発ほか
- (2) 現在利用している開発環境
 - DEC 3000-400 AXP OSF/1
 - Sun SPARC station2 Sun OS
 - IRIS Indigo HP-LIX
 - PC-98 MS-DOS
- (3) 関心のあるテーマ2つ
- オブジェクト指向

オブジェクト指向を用いたシステムは、拡張性・信頼性に優れているという記事を雑誌でよくみかける.

システムにとって信頼性は最も重要であることから、 今後オブジェクト指向はシステム開発手法として一般 化すると思う.

CASE

現状では、上流・下流CASEツールの使用により、開発効率が向上するか疑問であるが、リバースエンジニアリング機能を備えたCASEツールは、既存のシステムを利用し再構築することから、現状でも使えるのではないだろうか。

2. 感 想

- ワークショップについて -

今回はじめて、ワークショップに参加しました。

今まで受講したセミナーは、講師->受講者の一方 通行的なものでしたが、今回のワークショップでは、 同世代の同業の方々と交流を持て有意義で貴重な体験 となりました。

(台風により日程が短縮されたのが残念です...)

- 作戦会議について -

自分の考えと同時に相手の立場になって物事を考えることは、両方の考え方の良い面、悪い面が見え、 物事をより広く、深く理解することができました。

今後は、このような物事の考え方を身に付け、仕事 に役立てていきたいと思います。

また、グループ内の討論においては、自分の無知さ と経験不足を痛感しました。

ー ディベートについて -

残念ながら勝つことはできませんでしたが、良い経 験になりました...

14 外岡 徹

性別:男 年齡:27 血液型:A

会社:(株)SRA

部門:システム開発第1部 特記事項:酒が飲めない

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

ASCII COBOL、COBOL85による汎用機上の事務アプリケーションプログラミング、設計。

IE設計技法によるサブシステムの基本設計(実現を前提としない設計)と、従来技法との比較評価。

ORACLE、SQL * Formsによる分散環境の事務アプリケーション(設計、開発)

(2) 現在利用している開発環境

Macintosh Quadra950/700 + Sony News 十数台 他(社内)

UNISYS U6000 + TOSHIBA EWS + PC9801 (現プロジェクト環境 (客先))

(3) 関心のあるテーマ2つ

■ オブジェクト指向

私はオブジェクト指向分析/設計の実践経験はないし、オブジェクト指向言語に至っては全く知らない。

そういう私がオブジェクト指向についての意見なん てそうそう言えるものではないが、実際にオブジェク ト指向を導入した時に現われる問題点について考えて みたい。

現在我々が設計、開発を請負い仕事を進めて行くと き常に問題となる事は、たとえば重要なことが決定さ れていないままスケジュール上次の行程に進んでし まったり、曖昧な要求を元に設計をすすめなければな らなかったりすることである。こういった問題はオ ジェクト指向の分析/設計によって解決あるいは改善 されるだろうと思う。

しかしオブジェクト指向の本質的な効果とは、分析 段階からユーザと共に業務の分析を行い、設計し、オ ブジェクト指向言語によって実現されることによって 現われるものではないだろうか?

もちろん実現手段を考えずにまずオブジェクトとその関係にのみ着目する分析や視覚に訴え誤解を招きにくい表現方法、3つの異なった視点から分析を行う..など分析方法それ自身に効果があるということは疑うまでもないが、多くの汎用クラスライブラリがあり、CASEツールも整備されて...という前提条件があってこそではないだろうか?とも思うのである。

問題は現在の設計/開発行程からオブジェクト指向に徐々に移行しようと試み、オブジェクト指向のある部分のみを取り入れようとした時にどういった事が起こり、どう対処していったら良いのだろうか?ということである。おそらく近い将来、多くの開発者が直面する問題ではないだろうか?

完全にオブジェクト指向で設計された仕様を非オブ ジェクト指向の手続き型言語で実現しようとした時の 問題。オブジェクト指向分析のドキュメントから旧来 のドキュメン(設計書)を作っていかなければならな い場合。オブジェクト図でユーザと話ができない時。 分析段階で今まで以上の時間を掛けることが許されな い場合。既存のシステムの一部をオブジェクト指向で 作り変えたいと言う要求にたいしてはどうだろうか?

その他ユーザ側からみれば基本検討から予算獲得までのタイミング、開発側としては受注の形態から見積りまで多くの部分で勝手が違って来るのではないかと思う。

このような、少し中途半端な形でオブジェクト指向 分析/設計を行う機会がこれから増えてきそうな気配 である。

CASE

システムのメンテナンス業務において頭を悩ませるのは、仕様書と実際のプログラムが合っていない、どのような経緯でその仕様が確定したのか分からない、などといったことだろうか。その結果、役に立たない仕様書は見捨てられ、開発当初から関わっていた"事情が分かる"技術者はいつまでもそのシステムから離れられなくなってしまう。

かくしてそのシステムのことなら何でも知っている"職人"の誕生である。

従って、私が一番望むCASEツールとは、仕様書とプログラムが強く結び付いているものである。

仕様書の中でも分析、基本設計、詳細設計の仕様書が結びついており、対応する言語やベースとなる環境を選ばない。

しかしこのようなCASEツールは特に汎用機の世界ではなかなかお目に掛かれないようである。これまでの業務上使用の機会のあったCASEツールはそれぞれ便利ではあったが、あくまでも"作成"の為のツールであったように思う。バージョン管理や検討経緯の管理を含めたCASEツールが整備され、正しく運用されるならば、前述の問題はかなり改善されるのではないかと思うのだが...。

2. 感 想

今回このワークショップに初めて参加した訳ですが、なかなか有意義な3日間でした。機会があればまた参加してみたいと思っています。

今回のワークショップでは、「ディベートは与えられた論題について議論をしたり、色々な角度から検討する事が一番の目的であり、ディベートそのものは主目的ではない」という思いが2戦2敗の原因だったかもしれません。作戦会議が論拠に集中してディベートの戦略、戦術が足りませんでした。

確かに今回のディベートは形式であって目的ではな かったと思いますが、我々SEにとってかなり重要な テクニックであると感じました。顧客の要求を引き出したり、収束しない仕様変更をあきらめさせたり、私たち(特に私のような事務アプリケーション屋)の仕事には「話し合い」が重要です。ここでより有利な状況を作り出す為には戦略や戦術、素早い状況判断と対応などディベートのテクニックが役立つのではないかと思ったわけです。少なくともつまらない研修を受けさせるよりディベートの訓練を行なった方が会社の利益になるでしょう。

台風13号の接近により4日のスケジュールがキャンセルされてしまった訳ですが、残念だったのは3日の夜までもキャンセルされてしまったことです。ディベートの準備に時間がかかり、グループのメンバー同志ゆっくりと話をする機会が少なかったので・・・。

4日の午前中JRが止まってしまったため、その日に帰るのを早々にあきらめて、週末の観光客となりました。こちらはうれしい台風の影響でした。

15 沖 さとみ

性別:女 年齡:27 血液型:B

会社:MHIエアロスペースシステムズ(株)

部門:技術部制御システムーグループ

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容 航空機のシミュレーションシステムの開発
- (2) 現在利用している開発環境
 - ・マシン HPワークステーション
 - OS UNIX
 - ・使用言語 FORTRAN、C
- (3) 関心のあるテーマ2つ

CASE

現在、私が参加しているプロジェクトの開発工程に 並行して、社内でCASEツールの開発が行われていま す。

このCASEツールは、以下に示す様に、概要設計を除く全工程をサポートしています。

- ・基本設計支援ツール(モジュール階層図/イン ターフェース、データテーブル概要/詳細作成)
- ・詳細設計支援ツール(擬似コードからのHIPO及びソースのスケルトンの作成)
- コーディング支援ツール
- ・検証支援ツール(モジュール単体/結合試験)

管理支援ツール

私自身は、世間に普及しているCASEツールを使用した経験がないので一般的なCASEツールについてはよくわかりませんが、社内で開発したCASEツールを使ってきて感じたこと、思ったことを並べてみます。

・ドキュメント作成にかける時間が短くなる、ドキュメント作成作業が楽になる。

CASEツールを使用しない場合、基本設計書、詳細設計書等のドキュメントをワープロで作成していました。このようなドキュメントは通常罫線が使われていて、ワープロ作業が非常に重労働です。後になって間違いを見つけたり、修正を行わなければならなくなった時などは、私自身がは設計作業をしているのかどうかわからなくなってしまい、本当に悲しくなります。その点、CASEツールを使った場合、いかにして見た目に綺麗なドキュメントを作るかという余分な労力を使うことなく、設計作業に専念することができました。特にHIPOのようにワープロで作成するのが非常に困難なドキュメントを、擬似コードから生成することができるというのは、非常に便利でした。

・各開発工程における成果物の整合がとれる。

各開発工程での成果物がデータベースとしてあるため、他の工程で生成された設計結果との照らし合わせが可能となります。例えば、モジュールインターフェースの入出力データと、HIPOに書かれている入出力データが違うというようなことも設計の過程で気付くので、詳細設計でのデータの漏れは防ぐことができるし、逆に基本設計における漏れも早期に発見することができました。

・擬似コードを使うことによって、設計作業効率が 向上する。

実際にモジュール内の処理を設計する場合、HIPOのような絵を書きながら処理の流れを考えるというのは、考えることよりも絵を書くことが主になってしまい、なんとなくじっくりと取り組むことができません。かといって基本設計書から突然コーディングをするというのは無理があります。詳細設計段階では擬似コードを使用したのですが、モジュール内のアルゴリズムを考えるには非常に気軽で、またアルゴリズムを考えることに専念できたように思います。

■ フォーマルメソッド

設計作業を行っていて感じることは、設計仕様書を理解するのが難しいということです。社内的には仕様書の規約があるのですが、それは、章立てや紙面のフォーマット等の見た目についてのみのように思います。仕様をどのように記述するかについては、設計担当課、仕様作成者によって随分と差があって、各仕様

書毎にその仕様書の"読み方(理解の仕方)"があって、コツを掴む必要があるのです。そういう意味で、仕様書は誰が作成しても同じようなレベルの仕様書が出来上がるということ、また仕様書の中で何をきちんと定義すればよいのか、何が大切なのかということがわかりやすいことが大事だと思います。フォーマルメソッドというのは、これらの問題の解決の糸口として、今後勉強したいと思っています。

2. 感 想

- ・私は人前で話すのが不得手なので、今回のディベートはよい勉強になりました。ゲーム感覚でディベートができるくらいになりたいです。
- ・事前の下調べがたりなかったせいもあって、ずっとディベートの準備で精一杯で、参加した皆さんとざっくばらんに話があまりできなかったのは、とても 残念でした。
- ・台風のために途中で中止になってしまい、最終日の「フォーマルメソッド入門」がなかったのは、非常に残念でした。CASEの論題でディベートを行なったときに否定側の私たちグループ3は、上流CASEにかわる方法として、フォーマルメソッドを提案しましたが、私自身、これから勉強してみたいと思います。
- ・懇親会で行なった居酒屋(しゃもじ??? 名前 は忘れてしまいました)のウニやお魚がとてもおしし かったです。

18 神崎 秀樹

性別:男 年齡:27 血液型:0

会社:(有)アドパンスト・ソフトウェア・

インスティチュート

特記事項:タバコの煙に弱い/バスに乗ると乗り物 酔いし易い

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容

某豊田系の会社の、エアジェット織機周辺 (表示、 制御、通信) 装置のソフト開発を行なっています。

(2) 現在利用している開発環境

マ シ ン / O S は 、 VAX/VMS, PC9801/MS-DOS, IBM-PC/Windows言語は、アセンブラ、C、デバッグは、ICE、ロジックアナライザ

(3) 関心のあるテーマ2つ

■ オブジェクト指向 ■ CASE

今までに、ソフトを書いたり、ソフト屋さんにソフトを書いてもらったり(仕様書を書いて)してきた経験から、僕は「ソフトウエアの業界は機械やハードウエア等と比べると、一部を除いて、まだまだ経験も浅く洗練されていない」と強く感じています。なぜなら、ちょっとしたプログラムを作りたいと思った場合、大抵の場合において多くの工数(お金)が必要となり、かつ、作ったプログラムは使い捨て(再利用はほとんど不可能)で、あげくのはてにはバグがなかなか無くならないという事態になります。

よく、プログラムを作る立場の人が「仕様が不明確で、仕様のバグがあるからソフトがうまく出来ない」と言うようなことを言われますが(事実僕も思った)、最近仕様書を書く立場になって思うのですが、ソフトを作る前に完璧な仕様書を書くのは大変困難で、はっきり言って自分でソフトを完成させた方が早いと思います。それは「現在のソフト屋さんは全てをゼロから作らなければならなく、仕様書を書く人はその全てを明確に記す必要と、ソフト屋さんに正確に伝える必要があるからです。」最近では「仕様は、ソフト屋さんと共に作り、仕様書はソフトが完成したときに完成すればよい」と考えています。

しかし、これでは生産効率が上がりません、現在の ソフト設計(作成)をハードウエアの設計として考え ると、トランジスタと抵抗とダイオードだけで回路を 作るようなものだと思います。

このため十人ソフトを作れば、十通りのソフトが出来ます。しかし現在のハードウエアの業界では、極端な例かも知れませんがPCなどは、どのメーカーのマザーボードもほとんど同じです。

これは現在のPCの周辺回路がほとんどワンチップ化されているためできることだと思います。これをソフトウエアとして実現することが、オブジェクト指向やCASEの究極の姿ではないでしょうか?

ソフトがIC化されれば、ソフト屋さんは、その部分のコーディングとデバックから解放されるのはもちろん、仕様作成者はIC化されたソフトを組み合わせて出来る仕様を作れば良くなります。

仕様の内容が制限されることはデメリットと考える人もいると思いますが、僕はメリットだと思います。なぜなら何も制約条件の無い状態で全てを決定するのは、憂柔不断な僕にとって、非常に困難なことなのです。(例えば、自動車を右ハンドルか左ハンドルかのどちらかに決めろといわれても決められないと思うのです。 = どちらでも良い。)これは、WindowsのVisual Basicの考え方に近いと(僕は)思います。

筆不精の為、なんとなく取り留めが無くなってきた様なので、この辺りで結論をまとめますとオブジェクト指向にしろCASEにしろ、ローカル(部署内だけと

か)に標準を作っている間は、あまり大きな発展は無いと思います。ソフトウエアの部品 (IC) が一般に供給される様になり、世界的に共用され、地位を得るようになったら、ソフト開発は大きく変わると思います。

2. 感 想

つい1ヶ月と2週間ばかり前、我が社の社長から「都合が悪くなければ、これに行ってこい。」と言われ、訳もわからず「ディベート」と「CASEツール」の本を流し読みし、この「若手の会」に参加した私ですが、全体の感想としては「なかなかGOOD」だと思いました。

そもそも、私は会社の中でも異端児的な存在で、大学は機械工学出身、ソフトウエアの会社に入ったものの、ソフトは余りやらず(別に拒んだ訳ではなく、自然に・・)開発環境の整備(例えば:コンピュータ室のレイアウト設計、電源整備、PCの予算取りから導入、ソフトのインストール)が会社での作業の半分を占め、最近では制御装置のハード設計のフォローが主になっています(事実この半年ソフトに触れていない)。

こんな私ですから、はたして皆さんの知識について 行けるかどうか一抹の不安を抱いていた訳ですが。 P C各位や講師の方々に色々な事を教えていただき、何 とか無事にディベートを行う事ができました。 (結果 は 0 勝 2 敗でしたが)

さて、今回初めて「ディベート」を経験した訳ですが、このディベート「なかなか奥が深いな」と感じました。ディベートをやろうと思えば、その論題について色々な方向からの知識が必要なうえに、プレゼンテーションの要素も必要になり、なおかつ機転を効かせた反論が必要になります。こういう事で日頃から鍛えておけば、将来役に立つ事は間違いないでしょう。また私の場合、先に書いたように日頃から、上司に対するプレゼンテーション的な業務が結構あるため(何か新しい事をしたい場合上司の説得が必要になるのです)、今回のディベートは大変役に立ちました。

ただ、今回のディベートをやっていたときに多少気 になった事が1つ2つありました。

1つは「論題」が少々抽象的な場合があり、いざ肯定側と否定側で立論してみると、論点がうまくぶつからず、面白みに欠けたかなーと思いました。2つ目は立論から反対尋問までの作戦タイムの時間が短いと思った事です。それはチームの各メンバーの考えが違う場合、1分間では相手の立論をどう崩すのかをチームとして組み立てられず、反対尋問1、反対尋問2、反駁とがうまくつながらないような気がし、多少残念な気がしました。

最後に、今回の若手の会は台風の接近もあり、色々と思い出深い会になりました。いろいろ教えてくださったPC各位、事務局の方々大変ありがとうございました。

 $\Rightarrow \star \Rightarrow \Rightarrow \Rightarrow$

Dグループ

08 一原 武司

性別:男 年齢:25 血液型:AB 会社:(株)ジェーエムエーシステムズ

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容
- (2) 現在利用している開発環境
- (3) 関心のあるテーマ2つ
- QC, TQC

現在,提供時にソフトウェアの品質を計るのに良く 開発途中のバグ数を用いています.

ところが、このバグ数が規定数 (開発工程ごとにステップあたりのバグ数が決められている) より少ないとテストが不充分と言われ、バグ数が多いと品質が悪いと言われます.

上記の品質を計る方法はおかしいと思います.

と言うのは、まだ経験の浅い人の作った生産物と経 験豊富な人が作った生産物ではバグ数が違うのは当然 です。

それをレビュで一定の品質にするのですからそこで 出たバグ数を品質判定基準にするべきでは無いと思い ます.

■ プロジェクト管理

現在のプロジェクト管理者の多くは、現在のプロジェクトの進行のみに気を奪われがちの人が多い様に 思います.

と言うのは開発終了後、次開発を行っている最中 (それが結構忙しいとします)に前開発の障害が来た 場合、障害調査をできる人(技術的な問題で)が限ら れ、その人は今の今回開発中の作業と、障害対応で てんてこまいしていまうケースが良くあります.

つまり、プロジェクト管理というのは現在のみに気をとらわれない事が大切だと思います.

2. 感 想

ディベートの内容等につては、他の方が述べて下 さっていると思いますので私は全体の感想を述べさせ て頂きます.

ワークショップはセミナーの様に受け身では無く参加者がそれぞれ活発に意見交換をし積極的に参加できる機会として非常によいと思いました.

また, それぞれで活躍なさっている方々とお話もできるため, 今の自分に何が不足しているか, 是から何

をしなくてはいけないかなどが明確になり非常に勉強 になりました.

ただ残念なのは、開催地である金沢に台風が接近したため、最終日の予定がキャンセルされたのは残念でした(個人的に楽しみにしていました).

09 伊藤 武司

性別:男 年齢:23 血液型:B

会社:システム・エンジニアリング・サービス(株)

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容

今年,入社したばかりでまだ仕事らしきことはほとんどしてないのですが、社内のワークステーション管理チームに属しており,社内のシステム管理の仕事を少ししました.

(2) 現在利用している開発環境

現在はSunのSparcStation10で仕事を行っています.

(3) 関心のあるテーマ2つ

先ほども言いましたが,入社して日も浅く急きょ参加することになりまして,これらのテーマに対する意見を述べることができません。

今回の合宿でこれらの事を勉強できれば幸いと思います。

2. 感 想

今回のプログラミングワークショップに参加するように言われたのは、ワークショップ開催の2日前でした。その翌日の夜行バスであわただしく松山を出ました。そういうこともあり、このワークショップが何をするものかもあまりわからないまま参加しました。

このワークショップではディベートを行なうということで、そのテーマを見てみると、プロジェクト管理と開発プロセスということなのですが、私にとっては耳なじみのないものでした。私は今年の4月に入社したばかりの新人で、入社してからの5ケ月間は教育・研修を受けることがほとんどで、仕事と言えば社内のシステム管理を少しばかりしたにすぎないくらいでした。それゆえ開発など行なったことがないため、参加することに対し不安がいっぱいでした。

実際にディベートの準備が始まり、まず「SEは会社に出る必要はない」という論題を肯定することになったのですが、周りの先輩方が活発に意見を出し合う中、私は聴いているだけで、意見を出すことができませんでした。話が進んでいく中、私もSEを目指す者としていろいろ貴重な意見を聴くことができまし

た。結局、勝敗は私達肯定側が勝ったのですが、SE が会社に出る必要があるのかないのか、どちら側の意 見も正しく思え、私自身の結論は出すことができま せんでした。

翌日のディベートの論題は「大規模システムにおいて開発標準は必要でない」ということで私達は否定側に立ちました。開発標準は必要であるということを主張すればよいということで私達は勝利を確信したのですが、最終弁論に立った私が用意してもらった原稿を読んだだけに過ぎず、そういうこともあり負けてしまいました。私自身開発標準は必要であると思ってますが、それをディベートの中で意見することができず残念に思ってます。

このレポート自体、感想を述べたにすぎないのですが、現時点では、この程度のことしか書くことができませんが、今後開発業務に携わっていく上で、今回のワークショプの経験を生かしていきたいと思います。

12 菊地 義幸

性別:男 年齡:30 血液型:A

会社:フリー

特記事項:タバコの煙にかなり弱い/酒が飲めない

- 1. ポジション・ステートメント
- (1) 現在までの主な仕事の内容
 - ・開発環境メンテナンス
 - ・アプリケーショントラブルサポート
 - ・ライブラリ管理 etc.
- (2) 現在利用している開発環境

MVS/ESA MVS/XA

- (3) 関心のあるテーマ2つ
- ■開発プロセス ■プロジェクト管理

私は仕事柄「大型汎用コンピュータ下でのバッチシステム」しか知りません.

この会を通じて是非とも「分散システム」「ダウンサイジング」等の話しを聞きたいと思います.

2. 感 想

SEAの活動に今回初めて参加させて頂きましたが、なかなか良い企画ではなかったかと思います。これからの技術者は単に「テクニカル・スキル」が高ければ優秀な評価を得ていた時代から、自分の作ったプロダクトがいかにすばらしいモノであるかを相手(ユーザー)に説明する力、すなわち「プレゼンテー

ション能力」も同様に重要であると認識されつつある 時代に来たと思うからです。その意味からも社内教育 のカリキュラムでもなかなか行われないであろう今回 のディベートという企画はとても面白いものだったと いう気がします。

ただ、3泊4日の企画としてはディベートだけで終わってしまった感しかないのが残念です。台風で1日短くなったにせよ、もう少し講演とかフリーディスカッション等を組み込んだプログラムにしてほしかったです。今回のワークショップで何をやりたかったのか、その趣旨が私にはわかりませんでした。テーマを明確にし、そのテーマに沿った公演を聞き、ディベートの論題もそのテーマに関連したものを取り上げればより有意義なワークショップになったのでは、と思います。

さて、ここで、ディベートの中で開発プロセスの論題となった「開発標準」について個人的見解を述べたいと思います。結論から先に申しますと「標準化は必要である」と思ってます。以下のその理由を3点ほど掲げてみましょう。

まず第1に、標準化が無いと複数人数による開発では個人差を調整する時間や労力が全開発工程でかなりのウェイトを占めてしまう、ということです。技術者たちはそれぞれの経験から仕事の進め方や作り方について独自のノウハウをまさに十人十色で持っています。その人たちの自由裁量に任せて開発したら、確かに個々のプロダクトはすばらしいモノになるかもしれませんが、全体としてみたら使い物にならないという事態になりかねません。したがって各開発フェーズ毎のプロダクトの取り決めや開発プロセスにおける一定の方向づけ(これこそまさに標準化そのもの)が必要だと思います。

第2に、標準化が有ることで開発工程および人事の 両面における管理が容易になる、ということです。開 発期間が長く要員が多いほど、また高度な知識を必要 とするプロジェクトであればあるほど、要員の入れ替 えが発生した時の技術レベル維持を考えると開発要員 の確保が難しくなるのはいうまでもありません。その 時にプロセス・プロダクト両方においてある程度の標 準化がなされていればスムーズに進むだけでなく、プロダクトの質も従来と変わらぬ水準のものができるか らです。

第3に、標準化はユーザーにとってシステムの理解 (評価)を容易にする、ということです。標準化はな にも作り手のためだけではありません。そのシステム を発注したクライアント側(ユーザー)の立場に立て ば、そのシステムの使い勝手とそれをどう評価するか (監査)ということが重要なのです。その時に評価す 回

る対象として各フェーズ毎のプロダクトや開発プロセスのドキュメントを参考にします。それゆえ、クライアント側でも自社の監査基準に則った標準化を制定しますし、それに則した開発を依頼してくるのだと思います。

以上、標準化の必要性について述べましたが、これには当然反論もおありでしょう。それにつきましては 次回のワークショップの時にじっくりと伺うことと致 します。では次の機会まで・・・。

20 横井 伸司

性別:男 年齡:33 血液型:

会社:日本アイ・ビー・エム(株)

部門:SE研究所

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

- ・知識ベース・システム (エキスパートシステム, スケジューリングシステム)
- ・ユーザインターフェイス
- ・テストツール
- ·開発方法論

(2) 現在利用している開発環境 0S/2

(3) 関心のあるテーマ2つ

■ プロジェクト管理

開発方法論は、アプリケーション開発の生産性向上、開発期間短縮に寄与するものと期待している.

■ フォーマルメソッド

自然言語のあいまい性を補うために必要であると考 えている.

2. 感 想

今回のプログラミングワークショップは、北陸金沢の風光明媚な地で行われたにも拘らず、戦後最大級といわれる台風13号の影響で早めに切り上げることになったのは残念であった。私は、初めての参加で前回までどの様な活動が行われていたのかは知らないが、今回のディベートの採用は面白い試みであると思う。

ディベートは今回が最初の経験であった。プレゼン テーションは今までに社内外において何回も経験して いるが、それとはひと味違った印象をもった。

プレゼンテーションでは、自説を論理的に体系立 て、客観的事実と関連づけることによって相手を説得 するという作業を行うが、ディベートの場合には、こ れに加えて1) 敵陣の立論内容及び立論の筋道、立脚している事実を瞬時に把握すること、2) これに基づいて敵陣への有効な反駁と自陣の適切な防衛を考えること、3) ディベート過程の議論を反映して最終弁論の内容を組み立てること、が要求される。さらに、これらのことを限られた実時間のうちに処理しなければならないところも困難の理由であろう。

今回の経験で感じたのは、ディベートに勝つ秘訣は、ディベートの最中に考えることを最小限に抑えることである。そのためには、議論に関連する様々な事項の因果関係をあらかじめメモの形でまとめておき、ディベート中はどれをどの順序で話すか程度を決めればよいように準備しておくことが必要である。さらに、一般論として、日頃から様々な事象を論理的に整理しておくことの重要性を再認識した次第である。

さて、今後のことを考えると、ディベートにはいく つか改善点があると思う。

ディベートは開発方法論と同様、知識、経験、訓練が必要である。それ無しにディベートの形式だけを強行するのはルールを知らずに試合をするのと同じようなものではないか。また、PCの役割も不明確なところがあった。

つまり、ディベートの審判なのか、それともコメンテータなのかということである。審判に徹するのならば議論の論理性を重視すべきであろうし、コメンテータならばゲームとしての論理性より議論の実質(議論されていることの内容)を重視すべきであろう。(余談。米国の陪審裁判はディベートに似ていると感じたが、これも陪審員は素人であり今回のディベートと同様の問題を抱えているであろう)

論題の設定についても再考の余地があると思われる。今回のディベートで議論が噛み合っていないと感じたのは私だけではないであろう。原因の一つは、「. . . は悪」という形式の設定である。この場合、否定は「悪ではない」となり必ずしも「善」とはならないし、否定側は否定の否定となり議論の内容を分かり難くするのではないだろうか。「. . . すべし」の形式の設定の方が穏当であろう。論題の中身に関しても、余りに過激なものは合理・不合理がはっきりし過ぎている感があり、有効な議論を行うという立場からは面白味にかけるものとなる恐れがあろう。

以下、今回のワークショップで、ディベートに参加 しなかったテーマに関連して少し意見を述べる。

・オブジェクト指向

私は、オブジェクト指向アプローチによる、企業モデル作成、要件分析等の作業を経験したことがある。

そのときの経験では、業務内容を担当者から聴取して モデルを作成していく過程で、自然な記述が行えると いう印象をもった。この点に関しては、「オブジェク ト指向による分析作業は、ユーザーから業務の現状を 関き出すのにはとてもよい手法と感じる(高村)」と 同感である。

モデル作成や要件分析の作業では、一度で完成品ができるわけでなく、本質的に試行錯誤が必要である。 この過程においても、オブジェクト指向アプローチによるモジュラリティの高さは寄与するものと思う。

・CASEツール

ツールはあくまで「思考のための道具」であり、人間の能力を拡大するものである。従って、思考にまで介入するツールは、「小さな親切大きなお世話」だと思う。

所詮、「ドローイングエディタは考える力に、チェック機構は注意深さに、設計方法論は経験に、く決っして>勝てない(伊藤)」のだから、CASEツールは分を弁えて、単純作業や繰り返し作業、関連情報の検索表示等に如何に寄与できるかを考えるべきであろう。「CASEツールを使った場合、いかにして見た目にきれいなドキュメントを作るかという余分な労力を使うことなく、設計作業に専念(沖)」できたり、「モジュールインターフェースの入出力データを、HIPOに書かれている入出力データが違うというようなことも設計の過程で気づくので、...早期に発見(沖)」できたりということは、CASEツールの正しい役立ち方であると私は思っている。

21 芳里 嘉昭

性別:男 年齡:28 血液型:0

会社:日立ソフトウェアエンジニアリング

部門:北陸開発センタ基本ソフト第2グループ

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

- a) パソコン上の遺伝子情報解析ソフトウェア開発
- b) UNIX上のユーザインタフェース作成ツールの開発

(2) 現在利用している開発環境

HITACHI 3050シリーズ(UNIX System V)を使用

- (3) 関心のあるテーマ2つ
- オブジェクト指向重要であるが、難しい。

■ 開発プロセス

設計 --> 製造 --> テスト という単純な流れによる開発はソフトウェアには不向き

2. 感 想

ワークショップ参加の感想は、ディベートはしん どい、に尽きる。短い時間で他人によく理解できるよ うに分かり易く話し、さらに短時間に相手の発言を理 解しそれに対する反論を展開するというのは(自分に は)相当に難しい。こりゃ修行して出直さないかん な、などと考えている。

PCの方々によるディベートテーマの概説は、勉強になると同時に自分の勉強不足を痛感した。例えば「開発プロセス」で述べられた4つのモデルを私は一切知らなかった。(これがディベートの敗因かもしれない)

ともあれこの3日間は非常に刺激になりました。実 行委員の皆様、参加者の皆様、どうもありがとうござ いました。あの状況では仕方ありませんが、最終日の 「フォーマルメソッド入門」が中止になったのは残念 でした。

 $\triangle \times \triangle \times \triangle$

プログラム委員

I 中来田 秀樹

会社: (有) ネクストファウンデーション

ワークショップのプログラム委員長を指名されてしまい、やれる事だけをやってなんとか終了させる事ができました。

みなさん、ご苦労さまでした。

私個人としては、この業界がやっと成熟することができそうな可能性が出て来始めているように感じてきており、ゆっくりと若手の人がこれから辿るであろう道を模索して欲しいと思っていた頃のディベート形式でのワークショップでしたので、参加された人達に勉強の必要性と、まだまだやらないといけない事がある事を知ってもらって帰ってもらえたのではないかと考えます。

発狂しそうな状態で、なんでも良いから動いていればよかった時代から、目的を持って各自が適切に役割を自己認識して動かないとやっていけない時代になると思います。もっと落ち着いて、何が本質なのかを見ることができる人々を増やしていきたい、そう言う意味からも、ディベートと言うのは良い経験をしたのではないかと思います。

ただ、残念なのは、ディベートの為の作戦会議が必要ない最後の夜が、観測史上初の台風のために解散となり、みんなと宴会が出来なかったことと、最終日に「フォーマルメソッド」に関するフォーラムを開けなかった事の2点です。若手の方の中でフォーマルメソッドの言葉も聞いたことがない人がかなり居ると思い、その人達に、多少なりとも知って帰ってもらえたらと思っていたのです。

結果論になってしまいがちですが、台風も結局、予 測された程の悪影響もなく、無事でホッとしていま す。

さて、SEA はこれからも意味のある面白いことを やっていきますので、どしどしと参加してみてくださ い。

II 大塚 理恵

性別:女 血液型:0

会社:RSK

部門:技術開発部

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

- ・汎用機、EWSのDBMSシステム管理、開発支援
- ・パッケージソフトの社内向けマニュアル作成
- ・事務処理システム分析設計(開発技法の適用)の支援
- 社内開発標準の策定

(2) 現在利用している開発環境

サポート中のプロジェクトでは現在は分析段階にありPS/55 + WindowsでCASEツールを動かしている。

下流はオンライン画面設計、帳票設計等のユーティリティーを使い、最終的には汎用機で動かす。

(3) 関心のあるテーマ2つ

■ オブジェクト指向

「オブジェクト指向を『大規模事務処理システム』 の開発に適用すべし」なのでしょうか?

現在かかわっているシステム再構築では、何といってもオブジェクト指向を標榜しているので、もし問題 領域の整理や業務の分析がうまくいっているとした ら、それはオブジェクト指向のおかげだとどこかの雑 誌に発表するかもしれない。そうではない。どちらか と言えば構造化手法というのだろう。

しかし相変わらず機能重視であっても、OMTの本のおかげで分析という工程が見えてきたのは大変効果があったと思う。

CASE

これまでは汎用機用CASE(下流の支援)がほとんどであり、導入にはかなり慎重で、導入後も社内普及のための教育やユーザー・サポートに力をいれてきた。

しかし最近の上流CASEのためにはツールの使用法だけでなく、前提となる開発方法論を教育しないと普及・定着は難しい。(誰が教育するのか/できるのか難しい)

さらにWS + LANを前提とした製品が増えるであろうが、ノウハウの乏しい環境設定もユーザー主体で行なえるような体制作りが必要になると思う。

上流CASEには機能的に版管理、リポジトリなどいろいろ期待してはいるが、現状ではとりあえず作図専用ツールである。

2. 感 想

(1)プログラム委員

こういうワークショップに会社から参加させてもらえるSEは、恵まれている方かと思います。これまでの若手の会はどれもユニークなテーマで開催されてきましたが、みなさんの参加目的は一体何なのだろう?と疑問でした。職場によってセミナーの参加には、それなりの目的、意義を求められると思います。(弊社では管理者の推薦が必要です。)

今回は一体何がみなさんのお役に立ったかわかりませんが、職場へのフィードバックのついでに若手の会のPRもよろしくお願いします。

さてプログラム委員としての準備期間は約半年でしたが、本当に慌ただしく過ぎました。

個人的に他のPCと連絡を取り合うため自宅で通信できる環境作りに始まり、論題探し、参加者探し、ディベートとは?と準備しているとあっという間に8月です。PCに割り当てられたテーマを十分検討することもできず開催日を迎えました。

初日自己紹介を始めるまでみなさん結構きつい表情で、一見緊張しているようでしたので、4日まで大丈夫かと焦りましたが、おかげさまで無事(?)終えることができました。

(2)ディベートはおもしろかった?

最近では研修にディベートを取り入れている企業も あるそうですし、学生時代、英会話サークル等でディ ベートしたことがある方もおられたかもしれません。

全くノウハウがなく困っていたところ、偶然NIFTY の掲示板でディベートに関するホームパーティを見つけました。ここでセミナー、書籍、ビデオ等の情報を教えていただき、なんとか形だけは準備できました。

一つ二つ言い訳させてください。現地でディベートのルール(役割、発表順など)を調整したため、事前に作成し皆さんに配付したディベート関係の資料、用紙は使いにくくなりました。ごめんなさい。

また、私はタイムキーパーをしながら皆さんの議論 を理解していくほど賢くありませんので、審判からは ずれるべきでした。(雑用が思い浮かんだり、こなし たりと結構忙しかったし、予備のテープが途中で止 まっていたのには焦りました。)

ところで私にディベートなど能力、性格的に合わないことは明らかでしょう。日頃からゲーム類は勝敗にこだわるし、「論理性」と縁がない。今回参加者にディベートする意図がわかっていただけるか心配でし

たが、初回のディベートから活発に発言されるのを見 て半年間の不安も消えました。

毎回ディベートの講評をプログラム委員から一方的に言われるだけでは、ストレスがたまったのではないでしょうか。(表情でわかりました。)作戦会議に議論したことを十分発表できなかった人も多いはずですし、聴衆側でも質問や意見を持たれたと思います。そのため全員で議論する時間をとりたかったのですが、時間配分を誤りました。夜、プログラム委員とともに再び議論し納得されていればよいのですが、次の論題の準備に追われ、余裕がなかったかもしれません。

(3) Dグループは要領がよかった?

オブジェクト指向、CASEでの応募者が多い中、 当グループのプロジェクト管理、開発プロセス(また はQC)で応募されたのは少数であり、ポジションス テートメントだけでは先行き不安でした。実際、夜は 翌日のための議論はあまりしなかった。しかし午前の 作戦会議ともなると、グループ運営での自分の役割を 心得ているかのように議論し、模造紙にまとめ始め た。

議論は十分でなきかったかもしれないが、市街観光 を他グループより上手にこなし、楽しみ、とりあえず 時間内に準備できたのは要領がよかったのかもしれま せん。(一番楽しんでいたのは大塚です。)

結局「SEは会社に出る必要はない」(勝)、「開発標準はいらない」(負)という判定をいただいた。

伊藤さん:新人ですが全員に発表して欲しかったので反駁をしてもらいました。「SEは会社に出る必要はない」となりましたがワークショップ終了後、出社してくれるか心配です。

市原さん:(若いのに)プロジェクト管理の面では苦 労されたようです。「気使いより頭を使う」SEで頑 張ってください。

芳里さん:作戦会議中、メンバーが勝手にしゃべり 出す中、自主的に書記を務めてくれた。立論を繰返し 練習するなど真剣に取り組まれました。

菊地さん:大規模システムの開発・保守の長い経験を語ってくれました。('94年4月から2年間ニュージーランドに留学されるそうです。)

横井さん:フォーマルメソッドで申し込まれたのに 残念でした。しかもプロジェクト管理の方は関心な かったようですが、一番ベテランらしい雑談(経験 談)を聞かせていただきました。

(4)昨年/今年のPCへ

昨年の若手の会PC田中様をはじめSIGENVの 皆様には、今年のPC発足直後から開催に至るまで運 営面の他、様々なアドバイスをいただきました。

力作の若手の会報告書(SEAMIL)を読んだときは「プログラム委員って何をするのか知らないけど、とにかく大変そう。」という印象でした。反省点を参考にするつもりでしたが、例えばグループ以外のメンバーとのコミュニケーションなど、今回は実質2日でうまくいったかどうか。(逆にディベートの対戦相手に悪い印象を持たれてなければよいのですが。)

今年の大先輩のPCの皆様、お世話になりました。PC meetingの進め方など手際のよさは勉強になりました。

また何かのワークショップに応募しますので、私に 手伝えそうなことなら気軽に使ってください。

(5)最後に、若手の会報告書編集について

9月3日の解散後、参加者には9月10日までに感想文をグループのリーダーに提出していただきました。また記録係の4人にはディベートをテープから採録していただきました。11月のフォーラムのテープからの採録を待って報告書編集に取り掛かりましたが、年内にはほとんどの原稿が集まりました。

ご協力ありがとうございました。

III 伊藤 昌夫

年齡:34 性別:男 血液型:AB

会社:MHIエアロスペースシステムズ(株)

部門:技術部開発グループ

特記事項:煙草が吸えないと弱い

- 1. ポジション・ステートメント
- (1)現在までの主な仕事の内容 ソフトウェア開発支援環境の構築 社内管理システムの構築(NeXTを用いて)
- (2) 現在利用している開発環境NeXT(会社でも、そして家でも)
- (3) 関心のあるテーマ2つ
- CASE

- プログラマが必要としている道具-

乱暴に書けば、何も必須なものはないのだろうが、 これでは余りに乱暴なのでもう少し、常識的な範疇で 考えてみよう。 コンパイラ、エディタ、デバッガ。それから、ドキュメントように文書整形ツール位か。更にプログラムの差分を取ったり、shellのように自動実行するためのユーティリティの類か。これらは、必須ではないが、あったほうが便利である。

CASEはどうであろうか。ここに来るとあったほうが便利と云う比重も低くなる。それと同時に、今のCASEが生みだす幻想は、良心的なプログラマを混乱させる。

CASEに良く云われる事で、「これを使う事で、誰でもプログラミングできるようになります。」と云うのがある。果たして、このような道具を我々は欲しているのだろうか。

例えば、次のような事で考えてみる。遊園地にある ゴーカートは誰でも乗れる。子供から老人まで誰でも 乗れる。今、私と例えばF1ドライバのプロストがこ のゴーカートで競争したらどうだろうか。もちろん、 プロストが勝つだろうが、その差は2倍もないであろ う。では、レーシングカーで競争したらどうだろう。 たちどころに数倍の差がつくであろうし、もしかする と私はスタートすらできないかも知れない。

本来プロの道具とは、こうあるべきである。職業としてプログラムを作る人間が如何にしたら大きな能力 向上を得ることができるかが、先ず考えられるべきで ある。

自分自身も世の中でCASEと呼ばれる類のものを作っていて、常に思うのはこのことである。ドローイングツールが重要なのではない。チェック機構が重要なのではない。設計方法論すら、このコンテキストの中では不要である。何故ならば、ドローイングエディタは考える力に、チェック機構は注意深さに、設計方法論は経験に、<決っして>勝てないのである。

CASEが、考える力や注意深さや経験を必要としないで使えるものだとしたら、それはゴーカートであるに過ぎない。CASEに価値があるとしたら、充分な考える力と注意深さと経験を有している人間が、その力を数十倍にも発揮できるものでなくてはならない。更に、そのプロの道具としてのCASEは、F1カーと同様に、使う人や環境によりカスタマイズされる必要があり、そうあるべきだ。そのカスタマイズに耐え得るようなメタなCASEが作れるかどうかが、本来的な道具としてのCASEが生き残れるか否かの条件である、と考えている。

■ 開発プロセス

◇プロセスそしてオブジェクト

開発プロセスに関して最も分かりやすいモデルは (モデルとして分かりやすいと云う意味に注意)、 WBS的な木構造であろう。 ソフトウェア開発と云う大きなプロセスを次々と ブレークダウンしていく。実体として意味があるのは 葉になるプロセスであり、これを詳細記述する事であ たかも全体が記述できた様に思える。

このモデルは、モデルの分かりやすさと云う点において有効であるが、 いざ記述するとなると様々な問題がある。例えば、プロセスの動作順序について明示されていない。また、仮に記述するとしても、例えばモニタープロセスのような幾つかのプロセスにまたがって動作する並行プロセスをきちんと記述できない。或いはプロセスの再利用と云う側面が、モデルとして考慮されていない。

ところで、我々は最近こういった木構造ではないモデルに親しみ始めている。所謂オブジェクト指向と云われるモデルである。

オブジェクト指向とのアナロジーで云えば、属性は ワークプロダクトやアーティファクトと云ったデータ へ、プロセスの類似性は、is_aの関係へと還元で きる。

また、木構造で問題になる並行プロセスに関しても、充分に対応できる。

私自身の経験で云えば、このモデルは、木構造モデルに対して記述能力、実作業との適合性と云った面で 優れている。

◇プロセスそしてツール

プロセスとツールとの間には、何等かの写像関係が欲しい。そうすることによって、開発プロセスのメトリックスを収集する手だても、あるいはプロセス統合と云った視点からのユーザの利益も増す。

個人的には、その関連はプロセスの意味によっての み整理可能であると思っている。プロセスの意味と は、論理的に記述可能なレベル、即ち「データ構造を 定める」と云った事であり、このことはプロセス粒度 も、そしてツール粒度も又定める事になる。

◇プロセスそして人

プロセスの設計は、何かを作ると云う時間的な <線>を定めるものであり、今までの、管理がそうで あった様に、いついつまでに何かを作ると云ういわば <点>を定める事とと質的に異なる。

その意味において、開発プロセスに関する議論は、 端的に云えば、そして各プログラマレベルから云えば 管理強化と云った側面に展開されかれない。または、 部署ごとのインターラクションを考える場合、それは 政治的なものとならざる負えない。

こういった危うさを開発プロセスの議論は内包している。

こういった危うさに対して、我々は何ができるかと いう事も又、興味深いテーマであると思っている。

2. 感 想

土曜の朝、AM3時頃から殆ど動かないNHKの台 風速報を朝までずっと見ていた。

ベッドで横になりながら、ボーッと色々考えていた。今回は、前回の岡崎に比べるとかなり悔いが残った。

チームを組んだみんなは、非常に意欲的に、新しい事を知ろうとしていたと思う。そのことに対して、 応えられたのかと、何度も考える。

前回もそうだったが、若手の会は恐ろしいと思う。何か、全面的に、私自身が問われているような気がする。素直に聞かれると、答えられなかったりするし、素直に納得されるとオイオイと思ってしまう。日常の仕事の場だと、お互いにコンテキストを理解しているし、何かあったら「取り敢えず」とか「後で」とか云うセリフが通用する。それができないだけに、恐ろしい。

高村さんは、2度のディベートとも、難しいまとめ役をこなしていた。真面目できちんと仕事をこなす方で、その対局にいる私は、見事な反面教師になったでしょう。中尾さんや、広沢さんは、積極的にディベート時の受け答えについて考えていた。特に、広沢さんには、私の雑談でのプロセスの話が面白いといって下さったのだが、私はかなり一般とは違ったプロセスの捉え方をしているので、要注意。加藤は、私の仕事の同僚だから、良いですね。あと藤本さんは、途中で帰られて残念でした。

次の日、資料館や美術館を、強風の中、夕方まで歩き回っていた。文学資料館で、西田幾多郎の自筆原稿を初めて見た。彼の文章は、悪文と云われたりするが、彼の思想と同様に、私は好きだ。何故か、北陸の冬を知らない人間に分るまいと、同じ北陸の人間として思ったりもする。

ディベートと云うのは、所謂、論理的に話すと云う 事なのだろうけど、実は私が最も苦手な事で、そうい う点でも皆さんに迷惑をおかけしたのだろうと思う。 でも、少しへりくつをこねると、西田の最後の言葉、 「論理は、思惟の本質からでなければならない」と云 う事になるか。 どちらにしろ、皆さんご苦労さまでした。

IV 菊地 俊彰

性別:男 年齡:26 血液型:0

会社:東北電力株式会社 部門:応用技術研究所

1. ポジション・ステートメント

(1) 現在までの主な仕事の内容

入社時は通信関係の研究をしてました。具体的には ISDN、SDH等。

研究所に移ってからはマルチメディア、ヒューマン インターフェースを研究してます。

(2) 現在利用している開発環境

主に SparcStation 2、他に NEWS3865, Mac IIci。 プログラミングはしてないので、ほとんど OpenWindows です。

文書は Global View が多いですね。

(3) 関心のあるテーマ2つ

私はプログラム開発の現場にいたことがないので、 よくわかりません。

学生のときの研究室ではオブジェクト指向をやって る人もいて、私の使ってるマシンには ObjectWorks がのってますがほとんど使ってません。

CASE についても今の研究室でそれがらみのことを やってますが、最初言われていたほどじゃないかもね というような気がしています。

V 佐原 伸

性別:男 年齡:41(42歳直前) 血液型:B

会社: (株) SRA

部門:プロジェクト推進グループ

特記事項:医者から、タバコの煙は禁止されている &酒は控えるように言われているが少々たしなむ

1. ポジション・ステートメント

(1) 現在までの主な仕事

証券会社情報システム作成・保守、 ビジネスグラフライブラリー作成・保守、

ソフトウェア開発支援ツール作成・保守、

ソフトウェア開発環境作成、

ソフトウェア保守環境作成、

構造化分析/設計コンサルティング・セミナー講師、

商品企画・技術支援部署マネージャー、

オブジェクト指向分析/設計コンサルティング・セミナー講師、

ソフトウェア工学全般のコンサルティング・セミナー講師

(2) 現在利用している開発環境

ソフトウェア環境:

(自宅)

OS = Mac OS

プレゼンテーションソフト = Persuation

+ DeltaGraph

ワープロ = ASLEdit+ + YooEdit + 2ByWord

+ FlushWriter

辞書 = Grolier百科辞典 + Webster + 広辞苑

+ 研究社英和和英中辞典 + Syokendai

ペイント = Image

言語 = Standard ML + MacOBJ + MCL2.0

+ Smalltalk/V + MPW C++/C/Pascal

CASE = OOATool + ObjectCastLight

+ Entity-Relationship

DBMS = HyperCard + 4th Dimension

+ ForceEngine

表計算 = Excel (デモ版)

通信 = TerminalJ + Navigator + CoMix

+ ComNifty

シミュレーション = Extend + STELLA

知識ベースツール = Babylon

(会社)

OS = Mac OS + UNIX4. 3BSD

ワープロ = SoloWriter + vi + TeX + DWB

ペイント = MacDraw

言語 = UNIX C + MacOBJ

+ ObjectWorks Smalltalk

CASE = 明 + StP

DBMS = HyperCard + 4th Dimension

表計算 = WingZ

通信 = Telnet

ハードウェア環境:

(自宅)

Macintosh Qudra700 = 20MB + 1.7GB HD

+ 256MB MO + CD-ROM + 14.4K FAX Modem

+ Image Reader + Laser Printer,

その他Macintosh多数

(会社)

Macintosh Powerbook 170 = 8MB + 40MB HD, NEWS-1850 = 32MB + 2GB HD + ...

(3) 関心のあるテーマ2つ

■オブジェクト指向

オブジェクト指向分析/設計は役立つが、仕様記述がまだ曖昧なまま放置されている。

また、オブジェクト指向分析/設計は、構造化分析/ 設計やデータ中心設計の拡張であり、「オブジェクト 指向は役立つ」に決まっている。

オブジェクト指向CASEは、ハイパーテキスト機能が 軽視されている。

HyperCard程度のハイパーテキスト機能があれば、同じ機能でも随分と役立つのに...。

オブジェクト指向言語としてC++を使うのは不幸である。SmalltalkやEiffelを用途に合わせて使うべきだろう。

■フォーマルメソッド

分析手法やソフトウェア工学/科学を教えたりコン サルティングしていく中で、数学をベースとした形式 的仕様記述は確かに重要だと思い始めている。

2. 感想

今回のワークショップでは、「オブジェクト指向は 悪である」と「CASEツールは役に立たない」とい う論題を主張するグループのサポートを行ったが、さ すがにこれは骨が折れた。

そもそも、我々プログラム委員会で決めた論題の設 定が、ディベートに慣れていないせいか、少し間違え ていたようである。

「オブジェクト指向を今採用すべきでない」とか「CASEツールを今導入すべきでない」といったように、いつどうするかという風に論題を設定しないと、「理想のCASEツール」といったように存在しもしない理想状態について「良い」と主張されると反論できなくなってしまう。

とはいっても、実際のディベートでは、相手グループのミスにより何回も付け込めるチャンスがあったのだが、我がチームはそこに気づかず、連敗してしまった。

当初の予想通り、若手の技術者は(というよりは日本人のほとんどか:-))ディベートが苦手なようである。

前夜作戦を練った通りに主張するのはある程度こなせるが、相手が口を滑らしたミスを突いて矛盾を指摘することがほとんどなかった。

また、間違っても良いから自分の考えた意見を言う ということが少なく、従ってディベートの前の作戦会 議時間はむなしく過ぎて行くことが多かった。 我々中年技術者は、酒を飲んで酔った勢いでアイデアを出し合い、それを素面のときに整理するという特技を習得しているのだが、せっかく日本酒の本場石川県に来ていながら、宴会での消費酒量も少なかったのが、宴会担当プログラム委員としては残念である。

今回のワークショップに当たって、ディベートの本を2冊ほど読んだが、北岡俊明氏の本「ビジネスディベートの方法と技術、産能大学出版部」は、ディベートの技術の解説はともかく、ディベートの例に出てくる議論の展開は噴飯ものでなかなか笑えた。

従って、まともなディベートの勉強をするなら松本 通弘氏の本「これがディベートのやり方だ!、中経出 版」を読むことをおすすめする。

VI 中谷 多哉子

性別:女 血液型:AB

会社:富士ゼロックス情報システム(株)

部門:アドバンストシステム部 職 業:夜学に通う勤労学生

1. ポジション・ステートメント

(1) 現在までの主な仕事

過過過過去:天文学者を夢見て毎日星を見る 過過過過去:漫画家を目指しスケッチブックを持ち 歩くが、権威の一言で挫折。サラリーマンとな る....

過過過去:橋梁の線形計算、応力度等計算および描 画システム開発

過過去:宇宙遊泳を夢見て海の世界へ走る、走る、 走る、飛び込む、泳ぐ、潜る。

過去: 部品再利用によるソースコードジェネレータ 開発、秘書支援システム開発

近過去:オブジェクト指向開発環境開発、00A,00D 教育コース

現在:セミナー講師、学生

(2) 現在利用している開発環境

(自宅)

PC9801F2一式(百万もしたのに!今は火が入らない)、MAC Centris610

HP550C、FAX-MODEM(まだ一度もFAXとして使われたことがない)、

MAC SE-30...すでに養子としての行き先が決まってい

ワープロは今はなきSONYのProduce!....液晶ディスプレイが見にくい...

(会社)

Mac IIci...遅い!

(3) 関心のあるテーマ2つ

■開発プロセス、プロジェクト管理

プロジェクトが成功しないのは、SA/SDとか00とかによるのではなくて、もっと根源的な問題です。

学生のプログラマ (アマ) と社会人のプログラマ (プロ) との違いは、どれだけプロセスを管理しているかの違いだと思っています。

信頼性を高めるための手法は確立しているのか、スケジュールを管理するためにどのような方策を練っているのか。

スケジュールが遅れた場合の対策は何か。ソフトウェアは工業製品とは違い人間的な要素の影響が多いものです。それをどこまで理解して受け入れていくのかがプロの腕の見せ所であると思います。

さて、参加者はプロか?アマか?

00については世の中が動き出した感じがしています。はたして、今後00が持つ長所を日本の技術者は十分使いこなしていけるのでしょうか。導入を検討している企業は、ソフトウェア開発が抱えている問題のどの位を解決し、または解決していないのでしょうか。

00は本当に小さな問題を解決する手段でしかないため、過剰な期待を持つと絶望してしまうと思います。 00が善であることは否定しませんが、問題の本質が見えていない利用者にとっては、さらに問題を複雑にしてしまう恐れがあるため「悪」にもなりうると考えます。

問題が複雑になるとは、次のようなものです。

「00を導入したパイロットプロジェクトが全然終わらない。この原因は技術者の未熟か、ドキュメントの不整備か、プロセスの誤りか、見積りの誤りか、チームの人間関係の問題か。初期の仕様が曖昧だったのか。.....それとも00が原因か?」

VII 東田 雅宏

性別:男 年齡:31 血液型:A

会社: (株)PFU

特記事項: 酒は純米酒、タバコはマルボロ・ライト。

最近の動向: 学生

1. ポジションステートメント

(1) 現在までの主な仕事 要求仕様定義分析ツールの作成

(2) 現在利用している開発環境について

現在、何も開発していないので、利用している計算 機環境を紹介する。

自宅では、ZEOS 486DX2-66で MS-DOS v6.0(+日本語環境) + Windows3.1 をベースに、Freelance Graphics, Photo CD Accessなどのアプリケーションを利用している。言語は、Visual Basic, Borland C++などを使用。

学校では、Mac, UNIX WSから、並列マシンまで自由 に使用できる。

たいていのものはあるが、何故かSmalltalk80がいまのところない。

(3) 関心のあるテーマ2つ

■ オブジェクト指向

オブジェクト指向設計は、いままで培ってきた設計作法を網羅・包括した現時点での集大成といえるであろう。このことから、オブジェクト指向がいままでの設計手法の原理に対してメタな概念であると考えて不思議ではないし、また、そう考えることは私にとって自然なことである。

「今日のオブジェクトは明日のオブジェクトではない。」という静的には不安定であるが、実は動的には安定しているものこそが、オブジェクトである。このことがわかるためにとるべき行動はオブジェクト指向を実践するしかない。さて、実践するために何が必要で何が提供されているであろうか。「設計方法論はOMTが評判がいいね。ウチの環境では C++が使えるよ。」これではダメ。もっと、オブジェクトを実感できる環境が必要である。悲しいかな、こうした環境は、今も昔も Smalltalk環境しかない。

オブジェクト指向が明らかにされたという印象は今のところないし、将来もないであろう。そして、誰に とっても自然なことになる。

CASE

要求仕様定義支援ツールを作っていた手前、無視する訳にはいかない。

CASEの話をすると必ず出てくるのが、「最終的にプログラムを出力する」のかという間であるが、このような質問をする意図はなんであろうか。

まさか、世の要求すべてを記述できるプログラミング言語があると思っているわけではあるまい。いずれにしても、上流CASEに対する要求はボトムアップな傾向にある。この傾向は、要求者を無視した態度である。手に入れたものと本当に欲しかったものの違いと違いの生じた理由を明らかにすることが、要求者にとって最重要な仕事である。

実現可能性を検証しながら要求を定義していく過程を支援するようなCASEがあれば、要求者は精神的にも時間的にも楽になれる。さらに、要求者と開発者間のコミュニケーションを支援することになる。ついでに、要求者をユーザにすれば、最初の間に答える必要がなくなるので、上流CASEベンダも、あぁ嬉しい。

2. 感 想

ディベート形式で行なうということで、始まるまではどうなることか検討もつかず緊張しましたが、いざ、始まってみると慣れないながらも皆さんそれなりにさまになっていて嬉しくなりました。

今回のワークショップでは技術的細目を対象にしづらかったので、具体的に何かを得られたという印象は薄かったのですが、今回の各テーマに対して、より深い興味とより見識のある意見を持てるようになったと思います。感謝。

私はグループBの担当でした。作戦会議ではグループ全員が積極的に意見をだしあって発表内容をまとめていたし、ディベートでも各自の役割分担を自信を持って果たしたと思います。おかげさまで、相談役の立場として非常に気が楽でした。

「オブジェクト指向は悪である」の作戦会議では、ひと昔前までほとんどの人が使わなかった「オブジェクト指向」という言葉を今では多くの人がそれぞれのイメージで使っているという事実を痛感しました。 Smalltalkerとしてそれら個々のイメージは衝撃でした。

「オブジェクト指向」という言葉がコンセンサスを 得られるのはいつの日になるのでしょう。その時、私 が「オブジェクト指向」という言葉を捨てることにな らなければ良いのですが。

金沢へ来て半年になりますが、久しぶりに企業の 人々とお酒を飲めて感激しました。台風にも負けず滞 在した人と金沢周辺をまわったりして楽しい思い出が できました。

実行委員長の中来田さん、プログラム委員の皆さん ご苦労さまでした。

皆さんのおかげでワークショップは大成功でした。

VIII 松原友夫

性別:男 年齡:64

会社:幻の"Office Peopleware"

特記事項:たばこはのまない。酒はやめている。

役 職:小使いから社長まで

1. ポジションステートメント

(1) 現在までの主な仕事の内容

コンピューターとの付き合いは、第一世代から。

先日外人5人を科学博物館に連れていったら(江戸博物館に連れて行くつもりが、着いた日に行ってしまったというので仕方なく)何と私がかつて付き合った機械達がそこの地下室に置いてあった!

一見学者風に見えるらしいが、まったくの Practitionerで、ほとんどあらゆる種類のプロジェク トを経験した。ということは、失敗も沢山やったとい うこと。

プロジェクトを一巡したところでスタッフ部門に移 り、科学的管理の仕掛け作りを経験。特に会社創立期 の基礎的な仕掛け作りは良い経験になった。

約1年半前に会社を辞めて、コンサルタント業を始めた。今は会社にいたときよりも忙しい毎日を送っている。

(2) 現在利用している開発環境

自宅には、息子の分を合わせると、やたらに沢山の Von umann Maあるのにわれながら驚いた。

まともなのは、Quadra 700, PowerBook 170, ThinkPad 220, だが、その他に博物館行きに近い PC8001, PC9801VX2, Mac SE, などがあり、Mac E[9K/SE はいまだに通信専用に使っている。

その他ゲテモノとして、Portfolioというアメリカ で買ってきた小さなノート型のマシンがある。

我が家の設備の特徴は、音の関係のデバイス(シンセサイザー、MIDI)やソフトがかなりあることで、そのほかスキャナーやCD-ROM, Photoshopなど、芸術関係の楽しみ道具は結構揃えてある。もちろん商売道具としての Presentation 用 ソ フ ト、 Persuation, PowerPointも使っている。ということは、あまりプログラミングには使っていないことになる。

(3) 関心のあるテーマ2つ

どんなテーマでも、人や社会がからむ視点で見ることに興味がある。

たとえばISO 9000にからむ騒ぎを、人の愚かさという観点から冷ややかに見ている。(これについては今 SEAMAILに原稿を書いているのでそれを読んで下さい *) 少なくとも、SEAの催しに集まる人たちは、もっと 賢くなりましょう。

(*SEAMAIL Volume 8, Number 4 「ISO 9000-3をめぐって」に掲載されました--編集)

 $\triangle \star \triangle \star \triangle$

SEA Seminar & Forum

1993年11月12日 13:00~17:00 於:芝グランドプラザ

- 1. $t = t (13:00 \sim 14:30)$
 - 「 フォーマルメソッドの重要性 」 山崎利治(フリー) 「 オブジェクト指向とフォーマルメソッド 」 二木厚吉(JAIST) 「 CASEとフォーマルメソッド 」 佐原伸(SRA)
- 2. パネル討論(15:00~17:00)

「CASE、オブジェクト指向、フォーマルメソッド」 パネリスト:山崎利治(フリー)、二木厚吉(JAIST)、佐原伸(SRA) コーディネータ:中来田秀樹(ネクストファウンデーション)

記録:東田雅弘、大塚理恵

1. セミナー

岸田:

今月のフォーラムはフォーマルアプローチ、オブ ジェクト指向、CASEの三大話です。

元々、9月に金沢で開催された若手の会でやる予 定だったものですが、戦後、最大の台風に脅かされ て、一日早く終らせたので、その代わりに今日この フォーラムで行なうことになりました。それでは、セ ミナーを始めます。

元ユニシス、今、自由プログラマの山崎さんから フォーマルアプローチについてお話して頂きます。

山崎:

形式的ソフトウェア開発方法入門のための前口上を 述べます。

形式的方法は二つの用語<<形式的>>と<<方法>>を 含んでいます。

形式的とは、対象についての議論がその内容や意味 に言及しないで形式だけに頼ってできることをいい、 方法とは、一般的な術や方便ではない客観的な認識と 論理による目標達成への道程をいいます。

ソフトウェア開発の形式的方法は、したがって、ソフトウェア作成のための原理・理論をもち、この原理 の適用手段である技術が存在し、さらにその適用を円 滑にする道具を用意し、これらを選択利用する手順を 提供するものです。

その目的は正しいプログラムを作成することにあります。

多くの実用ソフトウェアはFORTRAN、COBOL、Cなど の命令型言語によって作成されています。

ところが、この命令型言語は参照の不透明さをも ち、これがプログラムの性質についての議論を難しく しています。

参照の透明さとは、文の一部を等価なもので置換えても文全体の意味が変わらない性質をいい、そうでない性質を不透明といいます。

命令型言語の代入文は不透明にします。そこで、プログラムが正しいなどプログラムの性質を議論するために、参照の透明な言語でこの仕様をプログラムが満たすことを証明する必要が生じたわけです。これをプログラムの検証といいますが、形式的方法はこの検証をやりやすくする枠組みを与えるものです。

お目にかけた例はごく短いPascalプログラムでしたが、プログラムの字面だけではそれが何をするものか

全くわかりません。これに前件と後件という仕様を与 えると明確になります。検証は省略しましたが、形式 的方法はそんな狙いをもつものです。

述べたかったことは、参照の透明な記述はずっと仕様らしいもので、MLやPrologによるプログラムは仕様であるといえますということです。

(質問はなし)

二木:

「形式」という言葉のイメージが良くないので 「フォーマル」と呼ぶことにしてます。

ソフトウェアの形式化は宿命的な問題です。

計算機科学の人気が下がってきていて、今は、アプリケーション指向に興味が移ってきていますし、ソフトウェア技術としてオブジェクト指向は流行っています。

オブジェクト指向についての説明は十人十色ですが、個人的に、Meyerの「構造化された抽象データ型」という説明がよいと思っています。 もちろん、「抽象データ型」についての説明がなされないと説明したことになりませんが。

抽象データ型は数学の世界では代数で表現でき、厳密な定義を与えられるものです。「形式的」を進めるための道具立てとして、形式仕様を記述するための言語があります。

インプリメンテーションをもって仕様とするという 立場もあり、「形式的」には多い。個人的にはモデル 指向である。

アメリカは軍事関係でAdaによる抽象データ型の蓄 積があるはず。

人間の指向・行動はきっちりと整理することは不可能なように、形式化の対象としてきっちりと整理すべきかそうでないかを見究めなければいけないかもしれない。

質問:

…メンテナンスをターゲット…

二木:

…オブジェクトは寿命が長い。複雑さの増大に対 応するため。

佐原:

SA/SDをはじめた頃はこれは大変な作業だと思った。 その後、OOA/OODをやってさらに大変だと思った。そ して、フォーマルメソッドをやってみてもっと大変 だった。

つまり、やらなければならないことがたくさん見え てくるということ。

SRAの出張精算プログラムを例題にMLという言語で記述したが、他の方法論では出てこなかった細かい点まで見えてきた。

フォーマルメソッドですべてのことが記述できないとしても、それは他のメソッドにもいえることで、例えば、OOA/OODでの図書館問題の職員と利用者に関する問題などがある。

結論としては、フォーマルメソッドは勉強しておい た方が良い。

マカオの大学で中国の鉄道システムすべてをフォーマルメソッドで記述するというプロジェクトをやっていて、それを見せて貰ったら、実にきっちりと記述されていて読んでみて思わず納得した。

オブジェクト図で描こうとしたらあいまいになってしまう。

フォーマルメソッドの教育について。コンピュータ・サイエンスのトレーニングコースを実施している。文部省にだまされていて難しいと思っていたが、つまり、試験ができないと理解していないという評価によってであるが、我々はカンニングもなんでもありだから、実際は思ったより簡単に理解できる。だから、心配せずに勉強した方が良い。

フォーマルメソッドはちゃんと書かないと中途半端だということがすぐわかる。オブジェクト図は未完成でもそれが完成なのか未完成なのかわかりづらい。二木先生はケースバイケースだといったが、私はどんな場合でも使えるという結論。

質問:

出張精算の例題の話は、仕様をきっちりと書いたという印象しかないが。

佐原:

出張日数とはどんなものかということについての議 論が起こる可能性がフォーマルメソッドにはある。

山崎:

関数型言語で書いたままがよい仕様となるとは限らないが、参照の透明さがそれを仕様にしているといえる。

質問:

例題は合計を求めるという目的がはっきりしている 問題でそうでない問題はどうか。

佐原:

問題の不明確さはメソッドに依存したものではない。ただし、その不明確さに気づく、あるいは、どの 程度不明確なのかを知る道具としてフォーマルメソッ ドは優れている。

武器を多く持つことが技術者の能力を向上させる。 一人で10年やっても、それは天才たちの永年の蓄積に はかなわないし、技術者レベルとしてでなく、研究あ るいは職人の意味の修練にしかならない。

(セミナー終わり)

2. パネル討論

中来田:

「CASE、オブジェクト指向、フォーマルメソッド」 ということで、パネルディスカッションをします。

パネラーの方はすでにそれぞれの立場で発表したの で質疑応答形式で進めます。

質問ありませんでしょうか。

会場A:

二木先生の話の中にラピッドプロトタイピングが何 回か出てきたのですが、それと今日の関係がよくわか らなかったので、どういう関係があるのか。

二木:

オブジェクト指向とフォーマルメソッドとCASEということで、最終的にCASEというか、計算機にこういったことを支援させるということを考えたときに、どういう形になるのかということで最後のOHP 2枚をお話ししました。

最終的に形式仕様というのは、かなり意味がきっちり定義されているわけで、ある見方からすると、そういう難しいことやらなくても、先ほどからMLで書いたという話題が佐原さんから提供されていますが、現在のものよりはかなりハイレベルなラングッジという見方もできるわけです。

つまり仕様記述言語というものは何なのかということなのですが。

仕様記述したときに、それは後でちょっと話題にしたら面白いと思いますが、仕様を書いた時にそれを動かすのがいいのか、動かさないでドキュメントだけして眺めるのがいいのかという議論があります。

私の立場では、やっぱりちょっと動かしてみて、 そこでリクワイアメント・スペシフィケーションの チェックとか、デザイン・スペシフィケーションの チェックをやった方がおもしろいんじゃないかとい う、そういうことのからみでラピッドプロトタイピン グというのを申し上げたんです。

仕様を書くということで、完全に動かない仕様を書くという立場は当然あるようですね。先ほど話に出ていたヨーロッパのグループには特にそれを主張しているグループがあります。

つまり仕様を書くときに動かそうと思って書くのは 邪道である、そういう気持ちで書いたらいい仕様はで きない。仕様は動かないものとして、しっかり数学的 なアルゴリズムを理解して書きなさいと。 私自身は仕様を書いてexecutableにやろうということでプロジェクトをやっているものですから、そういう見方からすると形式仕様を書くということは、通常の言葉ではラピッドプロトタイピングというのとちょっと雰囲気が違うかもしれません。

プログラム言語で書くときはだいぶ上のレベルで計算して、計算機の上で記号実行してみて、ラピッドプロトタイピングをして仕様の要件をかためていくという行為とつながっていくのではないか。それが結局がフォーマルメソッドに基づくCASEということになるのではないか。

ただ現在のCASEというのは、そういうことちょっと できるのもある、できないのも多い。

フォーマルメソッドに基づくとなれば、先ほどちょっと佐原さんのお話に出てきたように、現在の CASEでやっているよりもっと書き込んで動かす。

そういうコンテキストで申し上げたわけで、私はそ ういう方がいちおう狙って行く方向としては面白いの ではないか、見込みがあるのではないかと思ってい る。

会場A:

佐原さんのお話は、逆にそういう方法をとるってこ とはすごく手間がかかるというか。

私も現実の仕事をしていて、いかにエンドユーザーがうなずいてくれるかという提案をする立場から言うと、ラピッドプロトタイピングというのは、仕様とかそういうものをきっちり定義する前に、いかにしてできあがったものはこういうものだというのを見せられるかという解釈をしているものですから。

そういうものとはちょっと違う意味というふうに理 解すればよろしいですか。

二木:

ラピッドプロトタイピングは当然、仕様がかたまらない部分である程度動かしてみて、ユーザーの要求をとるというのが大きなことだと思うのですが、佐原さんのところの議論であったと思いますが、書き込むというときに何を意味するのかですね。

本来仕様を書こうというときには、決まってない部分は書かないでおこうというのをむしろ積極的にやりたい。

そうしますと通常のプログラム言語と仕様記述言語 の考え方の上での一番大きな差は何かというと、要求 仕様のレベルとか設計仕様のレベルというレベルより 上で、必要なことだけを書いて、かつそのレベルで チェックできるというのが理想なわけです。 私が申し上げたのは、これは将来の話ですので、無 責任なことを言っていると思われるかもしれません。 そういうことはできるのではないか。

つまり、後に残しておいた方がいいことはそこで書 く必要はない。

ただ書いた限りは何らかの整合性とか、完全性とか ということをやりたい。そのためには自然語で書いて いたのではしょうがなくて。

例えばドメインといいますか、ユーザーが持っているようなレベルであるモデルを組み上げて、そのモデルの上で、その抽象化のレベルできっちり書いて。ただ書いた限りのことに関して動かしてみる、そういう考え方で、基本的な考え方の相違はないと思います。

山崎:

ラピッドプロトタイピングというのもいろいろあると思います。

ユーザーにとって、例えば特にインターフェースといいますか、そういう機械と我々との界面に、どんな操作によって、ある種の目的が達成できるかというようなところに重点を置くものもあれば、わりとややこしい計算を探索的に調べるというものもあります。

それでいろんなタイプのプロトタイピングがあって、例えばある種の形式仕様はそのまま実行できることから、時間的な効率は必ずしもよくないけれど、課題のサイズが小さいから、ある種の成果が得られるという課題もあるわけです。

その実行可能な仕様というのが、そのままラピッド プロトタイピングにつながります。

例えばMLやMirandaなどの関数型言語を使い、 Prologなどの論理型言語をそのまま使い、わりと普及 しているStructured Analisys、Structured Designの データフローダイアグラムのバブルのミニスペックを 書くといい。

そのミニスペックの部分を例えば佐原さんはたまた まMLで書かれた。

それはそのまま実行できちゃうわけです。 これはラピッドプロトタイピングです。

また X Windowや Motifを使ってインターフェース を簡便に組み上げる。深刻なアルゴリズムは今から検 討して作るけれども、とにかくこんなシステムができればどうでしょう、というラピッドプロトタイピングもありえると思います。

で様々あって、ラピッドプロトタイピングの道具として見れば Motifや X Windowは随分いい道具でしょうし、MLや Mirandaも、それから Prologなんかも、随分いい道具であるいえると思います。

それが正々堂々としたある種の、整然としたフォーマルメソッド、特定の固有名詞を持つようなフォーマルメソッドを採用してやった場合にもかなりいろいろできるといえます。

二木先生のOBJを主体としたシステムもそういった ことで、ひとつの強力な道具を提供してもらえます。

中来田

プロトタイピングといっても、一般的にどのところをとっていうかという問題になるのですね。質問の方で言われたのは、ユーザーインターフェース的なところのプロトタイピングとか、ユーザーに見えるところでの問題ですけれども、その内部の機構のプロトタイピングという意味では、こういう仕様記述の方でのプロトタイピングが使えるという話で。

佐原:

車で言えば外観のボディのプロトタイプというのがあるわけですね。

それからエンジンはエンジンで当然いろいろシミュレーションして。

今の仕様記述言語とか、OBJとかフォーマルな方の CASEは、エンジンの方のシミュレーションとか、そっ ちのプロトタイピングにはたぶん向いている。

だけどMLなんかでも、グラフィックなインターフェースを持った、ライブラリ持ったようなものもあって、そちらだと多分両方できるようにはなってくるとは思う。

中来田:

よろしいですか。他の方は。

会場の中谷さん:

先ほど二木先生が、決まっていないことは無理に上流で決めないで、決めなければならないときに決めればいい。決まっていることであれば、それを確かめたいということで形式的に使用できる、そういうところに使っていきたいという話をされたのですけど。

プロセスの過程で、何を決まっていないこととして 後に回すかとか、こういったことは今決めておかない と後々大変だとか、そういったところの見極めという のは、ちょっと今日のテーマとはずれるかもしれま せんが、どういったところでやるのか、ちょっと悩ん でいるのですが。

二木:

そこに関してフォーマルメソッドを使ったからなに かということ、あまり期待されるのは。確かに新しい 見方を与えると思います。

ですがそれはかなりユニバーサルな問題で、ドメイン分析とか、そういったものが話題になってきています。フォーマルメソッドを使うときそこら辺がどうなるのかというのは、私はまだ報告も聞いていないし

今フォーマルメソッドが話題になってますので、少 しそちらの方から何かあるかということを考えて言わ せていただくと、まずモデルを立てなきゃいけないで すね。

決めなくていいことは後に延ばして、今考えるべき ことをちゃんとチェックしたりしたときに何にもない 空中では議論できないですから。

さっきスタックなんてやりましたけど、そこのレベルだけでとどまっているととらえられるとよくないのですが。

こんな絵を見せましたが、これを見てこの裏にある モデルを理解されると、スタックという概念に対し て、もうほとんどこれはこれ以上にそぎ落としようが ないというくらいに理解できるというものなんです。

結局こういったものは、スタックの積まれる内容とか、スタックをアレイで実現するのか、リストで実現するのかということと一切関係なしに、スタックのスタックである性質をかなり抽象的にとらえて、かつ、スタックに一番最後に積んだものは確かに最初に出てくるかというようなことに関しては、ちゃんと検証できるようなものを与えてるわけです。

これはスタックなんて簡単なもので、データ構造だからできているのですが、これが例えばリアルタイムなんとか処理というドメインに関して同じことやりたいとすると、これに相当するモデルを作らなきゃいけない。

そのモデルの作り方というのは、ドメインの知識を ちゃんと持っていて、それまでそういう修練をつんで からでないとできないと思いますけど。

例えばMLでもいい、OBJでもいいし、数学的な集合とか関数とか、例えば代数なんていうアイディアを持っていたとすると、そのアイディアに、カテゴリでもいいですが、そういうものに習熟していると、そこのところにパッとモデル化できるかもしれない。

例えばちょうどこういうふうにモデル化したのと同じように、空中で自然言語で議論したのでは不可能なような、そのレベルだけに特有な性質が検証できる。

それを作り上げる技術が一番難しいところで、それができる人は、たぶんSE 1000人いても10人いるのか、100人いて1人のことかと思いますが、それは誰かがやらなきゃいけないので、そういう人に関しては、そういったものをちゃんと理解されていることは役に立つと思います。

山崎:

そのエキスパートシステムを作るために、エキスパートからどうやって知識を聞き出すか、AI華やかなりし頃問題になったわけですが、それと全く同じなんだと思います。たぶんそれができればSEなんだと思います。

我々が相手にする客はそれを日常業務としていて も、その分野の内容を話せないわけです。

だからそれを聞き出す能力がないとだめなんです。 聞き出す能力はどこから生まれるかですけれども。 それはどういうことなんですかね。やっぱり理論形成 能力でしかなくて。そういうのは何を勉強すると身に 付くかということがあるんですが、わかりません。

二木:

数学を勉強すれば役に立ちますが、そういう理論形 成能力は。

山崎:

どうなんでしょう、本当の数学は別でしょうけれど も、数学の周辺で数学理論を作る人達の態度なんかは 参考になるかもしれないと思います。

中来田:

SEの技術レベルという話が出てきましたが、数年前のバブルのはじける前の状況ではプログラマが数年経つと自動的にSEになったのですが、そういうSEに、CASEからオブジェクト指向、フォーマルメソッドに対して、どう適用していかないと取り残されるかとか、そういうものがもしあれば順番に言っていっていただきたいのですが、お願いします。

山崎:

今日、昔の仲間の峰尾さんがおいでになるのを知らなかったのですが、峰尾さんはRAISEやRAISEの支援ツールを随分お使いになっているので、その辺の感想をうかがってはいかがですか。

会場の峰尾さん:

私は日本ユニシスの峰尾ですが、いろいろフォーマルメソッドをかじっていまして、非常に苦労しているという立場でちょっと感想くらい。

フォーマルメソッドをかじっていますと、なんていうかよくこの世界ではいろいろ流行があって、何かがあるとぱっと広がって、すぐにああ、あれは使い物にならないという格好でぽしゃっていくというのがだいたいこの業界の歴史ではあるので、フォーマルメソッドに関してはそうさせたくはないという思いが非常に強いですね。

そんなに流行らないだろうと思っていますが、でも やっぱり悪しき傾向も出てきたという感じもしており まして、なんとかしていきたいと。

こんな例があったのですが。私の会社で、面識はなかったのですが、

An Inrtoduction to Discrete Mathematics ですか、 Zの教科書を一人で読んだのだけど、いいかどうかわ からないのでちょっとコメントを付けてくれよという ので手紙が来たのですけど。

Zというのは忘れていたので復習しながら少し読ませてもらったのですが、何が書いてあるかわからない。というのは、皆さん裸のZの記述を見ても、やっぱりそれはよくわからない。

何を書きたいかということが、絵をモチーフにした ような説明が書いてあったのですが、基本的にそれが よくわからない。

したがって何をやりたいかということがわからない ので、記述されていることがどういうことか、理解で きない。

結局のところ、よく理解できませんということでコメントを返すしかなかったのですけどね。

これは最も基本となる、何を書きたいかということ 自身がわかってないというか、記述されていないと判 断せざるをえないですね。

日本語でも絵は多少役に立ちますが、きちんと書けないので、Zで書こうとしてもやっぱり書けないのじゃないか、という感じがします。

さつき図書館の問題で、利用者と職員は共通でない ということに気が付かないで前は記述を書いていたけ れども、今度はフォーマルメソッドでやるとそれに気 が付いたり、あるいはちゃんと書けるようになったと いうお話もありましたが、どうかなという感じが ちょっとしてまして。

フォーマルメソッドで書いたら、そういうふうに気が付いたという方は、やっぱり自然言語でそれに気が付かなくてはいけないわけで。

何でやったとしても、気が付かなきゃいけないわけで、そういう方はフォーマルメソッドやるというのは止めてほしいって感じが実はするわけです。

これはフォーマルメソッドの問題でないと思います。ドメインを分かってないと、何でやっても駄目なんだと思います。

だからスタックのドメインが分かっている人は、モデルをフォーマルに書けるということがあるんです。 わからない人は、やっぱり書けない。どんな言語であったとしても書けない。

フォーマルであれば書けるという話では全然なくて、やっぱりだめなんです。

基本的にフォーマルにやればうまくいくという話じゃなくて、ソフトウェアエンジニアリングに対する 円満な幅広い知識がないと、何を使ってもうまくいかないということですね。

一番いいのはやっぱりやってみるという、とにかく 言語をひとつ書いてみるということがいいと思いま す。

非常に苦労することがあると思いますけど。それで 失敗するかもしれないですが、失敗したときに、それ はフォーマルメソッドあるいはフォーマルランゲッジ が悪いということではなくて、別の問題があるのでは ないかと思います。

そういうことによってむしろソフトウェアの難しさに気が付くと言いますか、何を考えなきゃいかんかというとに、ひょっとすると気が付くかもかもしれない。

最近ヨーロッパのペーパーを見ましたが、随分理論では進んでいますが、経験というか応用という部分では、それほど進んでいるわけでもないみたいで、何年かかけていけば少し追い着くのじゃないかという気もしますので、少し気長にやりましょうという、そんな感じです。

二木

峰尾さんは本質的な問題をおっしゃっていらっしゃ ると思います。

ドメインの話と、つまり実際のソフトウェアエンジニアの素養としてのモデルに対する総合的な知識のバランスの問題と、使うツールというのは本当に効果があるか、ツールでどうなのか、ということだと思います。それは簡単に答えが出る問題とは思わないですが。

ツールはいいものを使ったときにどういう効果があるかとか、教育でこういうものを教育したときに効果があるのかどうかとか、そういう話があって。

じゃ優位の差はないのかという話になると思いますね。優位の差があるかどうかは、客観的に評価しにくいと思います。

ただ確実に、コントロールのストラクチャの構造から始まって、データに行って、抽象データ型に行ってオブジェクトという、そういう発展の中でかなり、ツールに対する理解とか、いいツールとか概念とかというのに関して、人間の方も進歩はしている。

ですからそこら辺は、いいものはいいと評価する事も努力しなきゃいけないのかな。

つまり、フォーマルメソッドで Z で書いたからって 全面解決には絶対ならないけども、かといって Z と同 等の、いわゆる数学の本を読んで数学のロジックの本 を読んで、ロジックの書き方を習熟すれば、 Z と同じ ことができるかというと、やっぱり Z はある程度そこ で何か残したものがあるのじゃないか。

同じ数学のノーテーションなんですけど。

モジュール化の仕方とか、基本的なモデル化の仕方に対するスタイルとか。 2のグループのやった仕事というのは、全然ああいうものを知らない人が、First Order Logicを習って集合論を習ったよりは、プラスアルファがあるのではないかという気がする。

ですから、ソフトウェア作る人みんながそういう違いを分かる必要はないけれども、開発に携わる人とか少なくとも新人教育をする人とか、そういう人は違いを伝えていくということも必要なのかなという気がした。

ただ確かにツールはツールでしかないけれども、 ツールの違いというのもCASEだけでなくて概念とか方 法論とか、広い意味でのツールで。

OMTの本が出て、オブジェクト指向方法論が整理されてわかったと思った人もそれは、何人かはいるんだと思います。ですからやっぱり彼らが本を書いたのは、評価されなければいけないわけで。

そういう違いにもやっぱり、センシティブになって いく必要があるのではないか、というのが私の意見で す。

山崎:

言語というのは思考の道具でもあるわけですね。 プログラミングとか仕様記述とかが頭の仕事だとすれば、頭の使い方を規制する面が極めて強くあるわけです。

だから例えば自然言語でプログラムないしはアルゴリズムを記述するよりは、プログラム言語で書いた方がはるかに楽なわけです。

それはアルゴリズム記述という面で、我々が今まで 考えたいわば知識の集積があって、そのための仕掛け を記述言語がいっぱい持っているからなんです。 ですから逆にプログラム言語や仕様記述言語によって仕様の書き方や、プログラムの仕方を学ぶ点が極めて多くあるわけで峰尾さんの今の意見は、ちょっと乱暴すぎるのではという気がします。

それと峰尾さんRAISEのツールをお使いになっているでしょう、それの印象を後で付け加えてほしいのですけど。

佐原:

私自信はいろんなツールとか方法論をずっと使って きた。

峰尾さんのおっしゃったように多分あまり優秀な、 天才的、ストールマンみたいに道具何もなしでUNIXの 端末さえあればすごいソフトウェアを作る、そういう 部分は明らかにないですが、道具を使うとそれを通し て元の方を、概念とか理論を理解するということは結 構できて、逆にそれじゃないとできなかった。

多分コンパイラーも、yaccとかいう道具に出会って 初めて、それを使う上でああこういうところに引っ掛 かる、なぜかな、ということで逆に理論の方を、逆に いうと興味を持ち理解できる。

だからフォーマルスペックもフォーマルメソッドとかも、それと同じことだと思います。道具を知っているいろ興味が出てくるし、今まで気が付かなかった側面に気が付く。

自然言語の仕様でそこまで気づけと言われると、気づく人もいると思いますが、気づかない人もかなりいて、道具ってのはだいたいできない人のための救いの道具じゃないかなという気がする。

だからそういう意味じゃフォーマルスペックって難しくなくて、難しくないと言い切るとあれですけど、できない人ができるようになる可能性のある、道具じゃないかなって気がします。

実際に現場でコンサルティングとかやっていると、 自然言語のスペックでどういうことが起きているかと いうと、データフロー図もそうなんですが、構造化分 析ですら役に立つことは何かというと、例えばここに アウトプットがありますね。

自然言語で書いてあると、入力がないのにアウト プット出そうとしている人がいる。極端な話、そうで すね。

これは単純だからすぐわかりますけど、もっと大きなレベルでしたって、入力がないのにこういう結果を 出したい、ということがあって、それはデータフロー 図とかツールを使えば、一目瞭然に、入力がないのに どうやってこの結果出すんですかということを指摘で きる。

だけどそれをぐちゃぐちゃと自然言語で書いていると、何もわからない。

それと同じことをフォーマルスペックについても、 私自信は感じた。

しこしことやっていかなければいけないってことは 確か。

中来田:

MLにしてもZにしても他のものにしても、本に書いてあるものだけを見るとすごく理解が難しい。

実際に動いているのを見ると理解が早いのですが、動くものがないとか、あとはもっと低レベルなものの例題というのが余りにも少な過ぎるような気がするけれども、そういうのは言い逃れですかね。

SEというのはもっと勉強していって、そのくらいの ものはもっと簡単に理解していかないとSEにはならな いのでしょうか。

佐原:

私は現場にいるSEの責任だと思います。

車でどこにドライブするかということは我々の課題でね、目的地がわかりませんけど、ってエンジン作っている人に言ってもそれは。

一木:

だからドメインのノレッジというか、通常の言語のライブラリみたいなものですね。ライブラリみたいなものは、例えばZなりVDMみたいな比較的本が書かれたりするようなものに関しては、あることはあります。

ただ先程から議論になっているとおり、抽象度の高いところで議論しようとしているから、何が書いてあるかというとマップが書いてあったり、ファンクションが書いてあったり、なんだこりゃ役に立たない、おれの知っている世界と全然違う。

それで知っている世界からそこにどうやってやるのかというと、さっきのスタックというのもかなり抽象化された概念ですけど、もうちょっと日付の話とか本当の話をどういうふうにそれにマップするのかというのは、佐原さんおっしゃるとおりそれはSEの仕事でしょうけど、ただ両方整備しないといけないですね。

我々の方でそういうライブラリを作ろうということ をやって、ある人がそういうことをやるということに なっている。 ただ僕はいろんなことやるときに、常に両輪で、両 方やらなければいけないので、一人ではできないとい うこともある。

ですからこういう場でいろいろ議論して、お互い得意なところで議論しできる人がやる、お互いに連絡とるという、そういうことの活動をしていかないと。

中来田:

オブジェクト指向的なものというかSmalltalkとか そういうものだと、ライブラリが存在する世界にいき なり入れますね。

佐原:

オブジェクト指向だってドメインレベルのライブラリはない。

二木:

GUIのあたりでは、Smalltalkとか、一番最初からそういうものがあって。ですから何かドメインを設けてそこでのことになると、ライブラリ的なものを組み上げていくという努力はすごく大事。

山崎:

それと我々がやっている仕事、ちょっとした仕事が 本当はとても難しいのだと思いますね。

例えば要領よく、すぐわかるような仕様をどう書けばいいかというとそれが難しい。

例えばアラームウォッチだって、ステーツチャート で書いたのを見ても結構大変でしょう。

電話機を作るとき、状態遷移図だけがスタンダードエンジニアリングツールだそうで、そうすると家庭用の普通の受話器だって600ステーツ位になる。

600ステーツあると一覧性なんかもちろんないわけだし、正しいかどうかそれを見て議論できない。作ってみて使ってみてのものらしい。

するとなんで、例えばステートチャートを使わないの、ということを思います。

あるいはそれを最近のミルナー先生達のモービルプロセスかなんかで書いてみたらなんて思います。これはやっぱり大変ですね。

課題がそれぐらいに難しい。一見易しく見えてても、きっちり書こうと思ったら、理論をちゃんと作らないといかん。

だからそういうのをちょっと覗いてうまく利用できるものじゃないと思います。

ですから見かけの易しさと、それを何かしっかりした、例えば微積分みたいなものにするというのはとて も難しいと思う。 ある意味では微積分だって、エンジニアとそうじゃない人との識別の道具であったりしたわけでしょう。 そういう意味ではプログラミングの世界ですと、プロとアマの差がないですね。

例えば微積分というようなクラシファイアがないで すね。

例えばモービルプロセスがわかってないと仲間にし てやらない、というようなことがあると随分違うかも しれませんけど、今のところそうじゃない。

そうじゃない方がお互い幸せかもしれませんので。 ですから、見かけによらず本質はすごく難しいこと なんじゃないか。

中来田:

他に質問は。

山崎:

ちょっとごめんなさい。先程のフォーマルメソッド の支援ツールというのは、あまりポピュラーになって ないんじゃないかと思う。

峰尾さんが一生懸命使ってまして。

峰尾:

RAISEをやっていたのでRAISEツールというのを購入 して。それデンマークで売り出されていますけど。

フォーマルメソッド in ヨーロッパ '93、というののレクチャーノートが出ていますけど、この後ろを見ますと、その場で展示されたツールとして20か30ぐらいあります。

ョーロッパの方では随分盛んにやられているという ことがあります。前にそこに行ったときに、B と RAISE がよかったので、そのうちRAISEを買ったので すが。

やっぱり僕らだめプログラマにとっては、ツールというのは必須じゃないかという感じがします。

これもやっぱり言語ですから動かしてみないといかんていう面があります。

最近COBOLの教科書がよくなって、読めば結構わかるというのもあるかもしれませんけど、昔はわからないとすぐサンプルプログラム作ってマシンルームに行って流してみて、ああこうだったのかとか、わからないことを確認するとか、そういう覚え方をしたと思います。

似たような面はあると思います。読んでもシンタックスなんてきっちり頭の中に入るわけじゃないですから。

それと、比べながらこうだったのかということを確認できる、という点が一つありますね。

そういう意味で非常に常識的な使い方でありますが、シンタックスチェッカというのは絶対必要だと思います。

それから、だいたいタイプチェッカはあります。 これは型というのがこういう言語では大きな意味を 持ってますから。

タイプチェックは私の使っているツールに関しては、かなりしっかりやれてます。結構型の間違いというのはありますので、それは随分助けになる。

言語にもよりますけど、型というのはだんだん複雑化するというか、結構難しいですから、そういうタイプチェッカがあると非常に参考になりますね。

あとそのツールではトランスレーターがありますが、書かれた仕様をいわゆるC++かあるいはAdaに変換するというツールがありますけど、これはそれなりに制約はある。全てが変換できるわけではない。

implicitな表現に関しては、変換はちょっと難しい。その部分に関しては、別の表現で書き直すというようなことが必要ですね。

Justificationという証明系がありますが、これは 私はまだやってないのですが、最近翻訳された2の教 科書によりますと、証明系というのは5年間はまずお 遊びと思わなきゃいけない、ということが後ろの方に 書いてありましたけど、まだそういう意味じゃ正式な リリース版ではありませんので、証明系というのがど の程度支援するかちょっとわからない。

今年はその辺のペーパーがフォーマルメソッド in ョーロッパに載っている。そういうツールもポチポチ出てきたと言えますね。

VDMもそうですし、Z、LOTOS もそうですし、商業的なツールとそれからパブリックドメインとして登録されてるのと、イギリスの大学の実験ツールがだいたいありますので、ちょっとお買いになるか、それもひとつ覚えるのにいいかと思います。

やっぱりツールというのは、僕らだめプログラマー にとっては不可欠だなという気がしてます。

山崎:

さっきのご発言とは違いますね。

中来田:

やはりツールとして動かしてみないとわからない。

峰尾:

書いてみたものがあっているかどうか、自信が持て ないですね。 せめてシンタックスでもあっていることを確認した い、というようなことがあります。

私はMirandaなんて言語を変換してやろうとしているのですけど、それをプロトタイプとして動かしてみて動けば、安心した、ということが一応言える。

佐原:

データフロー図、CASEでちょっと簡単なおもちゃのようなテニスクラブの会員管理でデータフロー図を書くと、すぐ100個くらいエラーが出てくる。

それは皆構文エラーで、それはやっぱり見つけてくれるとありがたいです。

構文エラーを自分で100個見つけるというのはしんどいですね。

峰尾:

あといいのは、ヨーロッパの人達はワークショップ をやっているみたいです。

書き方にいろいろなパターンがありますから、その パターンでの仕様を書いて、同じ課題を別ので書いて と、それを合宿していろいろやっているみたい。

皆さんだいたい先生がいなくて困っているところが 多いと思うので、フォーマルメソッドに関してそうい うことやるのも、いいのじゃないか。

中来田:

他に質問はありませんか。

会場B:

安全性の関係で研究しているもので、フォーマルメ ソッドが、開発されたソフトウェアの検証といいます か安全性について、どういうことができるかというこ とに関心があるのですが、その点についてご意見をう かがえたら。

二木:

ョーロッパも盛んではあるけど、多分本当に技術を もっているのはアメリカじゃないか。つまりDoD関係 の。

TestingとかVerificationのワークショップはこっちに比べればだいぶやってますし。

そういうときに今使われているのは旧来型のフロイド・ホーアタイプの、先ほどの山崎さんのお話しになった事前事後条件をpredicateで書いて、それからプログラムの教科書になっていますから大学で習われた方もいるかもしれないですが、つまりVerification Conditionという一階述語論理の式を、わっと大きなのを出してですね、それをやみくもにAutomatic Verificationのテクニックを使うのかしれませんけ

ど、それで証明するというのがだいたい使われている技術らしいです。

ただそれじゃ限界があるというのがわかってきていますので、それに対してオブジェクトというか抽象データ型的なもののレベルでライブラリ的にモジュール化して証明する。

先程峰尾さんが、Bとおっしゃってましたけど、B なんかはそういう検証を少し意識しているということを聞いてますけど、自分では調べてないです。

そういうことからいうと、フォーマルスペシフィケーションランゲッジで書くというのはそういうことの出発点です。

日本でやるとするとたぶん通信系の、通信が強いですからね。実際通信の人達はやってらっしゃるみたいです。あとトランスポーテーション関係、交通関係、 鉄道。

あと私が知っていて、アカデミアというか研究としてよくやられているのは、ハードウェア・ヴェリフィケーション。

OBJという言語があるのですが、それをちょっと拡張して、VLSIチップの仕様を書けるようにして、それをverifyする。

組み合わせ回路的なもののverificationは大体いけるらしいです。

ですから入られるとすると、そこら辺じゃないかと思います。

実際こういったものを使ってメリットのあることを 仕事にしている人は、是非使っていただきたいという か、そういうのが本当の気持ちです。

そうじゃないとpayしないですね。よさそうだから やってみたといっても、それで証明できて、完全性が 検証されたっていえば、それは商売というか全体の経 済活動として、地位が確立されたわけですから、そう いうふうにやるのでなければだめかなという気が私は してます。

山崎:

安全性と言われたのは、かなり狭い意味ですか。

例えばよいことは必ず起こるとか、悪いことは決し て起こらないなんて。

時制や様相を入れたような場面での用語としてお使 いになったのですか。

会場B:

私は鉄道関係ですが、駅の信号制御の関係で、駅の信号というのはリレーを使った装置で制御していますが、それを今電子機械で置き換えるというか、その検証ということで。

早い話が例えば安全性という、列車がぶつからない とか危険な状態が起きないことを、仕様をフォーマル に記述して、プログラムを開発したときに証明という か、そういうようなことを。

今まで信号のあれはノウハウがあるわけですが、それは必ずしも形式的な意味で保証された方法論というわけではなく、そういうことを形式的に保証したいという要望が段々高まっている感じです。

二木:

それは非常にいい例だと思います。かなり使えるも のがある。

つまりテンポラリティーもそんなに複雑じゃないですし、かなり組み合わせ的なことができると思いますから、ある程度形式的なもの書かれれば、かなりいけるんじゃないでしょうか。

ただ先程話題になっていましたが、ドメインのノレッジとツールとどうなるかという問題が、非常にシビアに現われますから、使ったからと油断していると本当に人死んじゃいますから。

そういうものでないと思いますが、私、ツールとしてはかなりあると思います。そういうドメインの問題でしたら。

佐原:

彼が言っていることをやるために、もっとフォーマルなメソッドを最初からきちんと作って、でそのとき 当然今の質問にあったようなことも後から出てくる。

山崎:

実はこの前、泰山のシンポジウムでも特別講義でビョルナー先生がそんな話をされました。要するに、ドメインに一番従属した部分の理論を作れというんですね。

二木:

そこまでラディカルにやるんだったら、研究としてはそこら辺からがチャレンジの面白いところです。

例えば鉄道に対するシステムにノウハウ持っている 方がかなりおいでなら、ミドルから攻めてもかなりい けるとは思います。

具体的にどうするかは、ディシジョンは、いろいろ あります。

かなり本当の列車の操作のモデルから作ってやるのか、それともある性質を証明するためだけにツールとして使うかですね。

会場B

現状だと、あるツール的なモデルというか、外国で 研究されたのがあって、それを使って。

最初はそんなに現実に近いあれて証明できるとはこちらもちょっとそこまで大それたことは考えてないですけど。

二木:

中国の鉄道省と組んでやっているのはあります。ですから幾つかそういう例はあると思います。

意欲があれば実際自分で始められるというのも十分 可能だと思います。是非やっていただきたいと思いま す。

中来田:

他に質問は。

会場C:

今通信系でフォーマルメソッドを使われた話が出ましたが、例えばプロトコルを決めましょうという話に使ったらいいのか、それとも通信だと24時間基本的に動くとか、交換機というか、それの状態遷移といいますか、OSの部分に使った方がいいとおっしゃったのかどうか。

二木:

私が言ったのはプロトコルの話です。

OSIのモデルで7層モデルというのがありますが、国際的にもプロトコルをちゃんと決めて、それを皆で使ってうまくやろうという話があって、日本は一応リーディングカントリーという、OSIに関してはちゃんとやるよという国に位置づけられているわけです。

そういう標準を作るときにパフォーマンス・スペックというのは決定的に重要で、これは位置づけを説明するにはいいんですけども。

つまり標準というのは、文化的な背景が違う世界の 何十ヵ国の人が読んで理解できないと、特にシステム インターフェース、プロトコルみたいなものというの は役に立たないわけです。

今までOSIとか、CCITTでも英語なりフランス語なりで書いていいよということになっていた。それをこういったフォーマルスペシフィケーション言語で書こうというアクティビティがあって、ちょっと話したLotosとか、そういうものを意識した。

今ちょうどそういうステージにあるが、プロトコル をきっちり定義してみて、次に何ができるのというと きには、やっぱりテストとか実際のプロトコル・エン ティティというか、プロトコルを実行しているマシン ですね、ソフトウェアは。ある種のOSみたいなものだ と思っていい。

そういったものをテストをするとか検証するとか 言ってますが、そんなに完全解には至ってません。

まだ一生懸命、OSIの7層のあそこのことを書いて、 自然語から形式的な仕様に移そうかとしているとこ ろ。ただそれに非常に抵抗がある。

私の言ったコンテキストはプロトコルのそういうアクティビティですが、そこら辺のアクティビティはご存じなんでしょうか。

会場C:

SDLとか使ってやりましょうという話があるのですが、現実問題としては、今動いている交換機が10年前とかという。

ただ将来的な話で言うと確かに外国とつなぐ場合と かは、当然話を詰めなきゃいけない。通じないといけ ない、通信できないといけないので。

二木:

確かに今ある分野じゃなしに、新しい分野じゃない となかなかこういう新しい技術というのは、育たな い。

中来田:

他の質問は。

会場D:

先ほどRAISEツールという話が出ましたが、それは 二木さんが言われたように、自然言語というのを フォーマルメソッドにあたるようなものに変換するの をRAISEツールというのでしょうか。

峰尾:

いろんなツールありますけどみんな同じで、決められた言語で、ひとつひとつ言語で書いたやつを入れるわけで、COBOLならCOBOLで書いたやつを入れるとコンパイルしてくれますね、シンタックスチェックもしてくれます。まずその辺ですね。

それから重要な問題で、ワードプロセッサとしての 機能もあります。複雑な、数式的なのがありますか ら、そういうようなものは記述できるということに なっています。

最もベーシックな段階としては、構文に従って入れれば、そのシンタックスチェックをしてくれる。その 辺です。

佐原:

自然言語で書くんじゃなくて、問題領域をRAISEという言語で書くのをワープロ的に支援してくれて、構文チェックしてくれて、一部言語のジェネレーションもしてくれる。そういうツール。

峰尾:

あとグラフィカルにそれをサポートするようなもの もあるようですが、RAISEツールにそれはありません。

会場D:

そういう意味でフォーマルメソッドは、そんなに主流にはならないにしても必要だろうって言われているのは、確かにそうなのかという気があったのですが、CASEという意味では、自然言語からそこに移るのにそれほどまだ強力なCASEのツールみたいなものはないのか。

峰尾:

ええ、とにかく記述は全部人間がやるわけですか ち。自然言語を自動変換するというようなことはあり ません。

山崎:

でもそういうのを書こうと思うと自然言語で書くのは、至難のわざでしょう。

自然言語はちょっと使えないね、ということから仕 様記述言語ができたわけですから。

会場D:

ツールという意味に少しこだわっているのですが。 どうも話としては、とにかく人間がある程度考えて、 平面に書いていくということですね。

例えばコンピュータが会話してくれて、先ほどクラシカルにと言われてましたが、例えば「これは何ですか」とかごく単純な話で、あいまいさを指摘してくれるようなツールというか、もう少し人間的なツールというのは、まだあまりないですか。

山崎:

ええ、ありませんですね。

二木:

今の質問に関する感想ですが、CASEというのをどういうふうに捉えるかとかあると思います。

AIベースとか、つまりインタラクティブに人間と会話しながら人間の方が教育されているとか、そういうのはAIの人なんかが研究しているわけですね。

そういうアプローチと比べて、CASEをそういう方向 に発展させて、将来的には計算機と会話しているうち に何かできるようになる、という研究している人もい ますが。

フォーマルメソッドはそれとはちょっと両極をなすような考え方で、先ほど動かないように書いた方がいいというのはヨーロッパの偉い先生なんか非常に言ってたりするのですが、つまり動くような言語で書くのは邪道である、そういうのに象徴されるように、フォーマルメソッドの一番のコアは何かというとやっぱり、モデルです。

数学的なモデルと言うべきですか、数学的にかなり きっちりしたモデルに基づいていろんなものをやる。 言語とかいうのはそのサポートなわけです。

そうするとCASEで計算機を、ちょっと誤解を恐れずに非常に大胆な言い方をすると、何をやってるのかわからないけど計算機を利用してます、ということじゃなしに、むしろ人間の側が積極的にそういうモデルを考えてこうか、そういうアプローチ、メソッドです。

ですからそこら辺はかなりフィロソフィー的に違うかもしれません。

佐原:

あとCASEベンダーから言うとですね、まだそういう レベルまで、フォーマルメソッドもそうですし、オブ ジェクト指向とかそうだし、構造化分析ですらそこを ダイレクトに支援するCASEだけで精一杯で、人間側に 教えるとか、そういった仕掛けを入れるのは当然難し い。

とにかく今は物理的に、力仕事的に全然そっちまで 手が回らない状況はあると思います。

二木:

それとこれは何度もこういう場で話題になりますが、どちらがいいかという問題ですね。

例えば先ほど言った鉄道の信号機とか、非常にセイフティクリティカルなシステムや技術に、自分がその動きをちゃんとわかってないようなインタラクティブツールを使ってできたからって、それで走らせる気になるかというと、やっぱりちょっと難しい問題があると思います。

ですからそこら辺は基本的なアプローチの違いで、 非常にセイフティクリティカルってことになれば、い わゆるAI的にということではなしに、フォーマルメ ソッド的なモデルからがっちり組み上げていくという とらえ型も、あるのじゃないかなと私は思ってます。 そこら辺は、自分がアプライしなければいけないドメインといいますか、それの特質をよく考えてからでないと、いろんなもの適用すると間違うのじゃないか。

特にセイフティクリティカルというのは地味ですけど重要な分野ですから、多分これからどんどん重要になると思うので、そういうところにどういうテクノジーで攻めていくかというのは、むしろ本当はこういうふうにいろいろ評論している立場ではなしに、やられる方がそういうことはきっちり認識していてもらわないと、大きな問題じゃないかなと思います。

中来田:

例えば鉄道とか原子炉というものをモデリングする というと、そのもの自体をよく分からないければいけ ないと思います。

ところがそれ自体を勉強しているのは、1週間とかじゃなくて数年というので、ある程度そこのプロフェッショナルな能力まで持っていかないと、モデリング化するまでたどり着かない。

そうすると反対に、一般の人間が、自分の知っている専門分野をフォーマルメソッドで記述するのをもっと簡単にできるようなものを、提供していくのも僕たちの仕事になるのか。専門分野の人が使えるものを、使いやすいものを提供するべきなのか。

二木:

例えば今話題になっているセイフティクリティカル というか、人命に係わるシステム、ハイセキュリティ なシステムを作る技術というのは、SE側の責任だと思 います。ドメインノレッジでは解決できない。

ドメインは基本的なデザインデシジョン関しては提供するでしょうけど、そこに関してはその分野に特化したSEを育てないと、ドメインの方にそれを全部任せるのはソフトウェア技術者としては無責任じゃないですかね。

会場E:

我々がこういう会議を持って、SEとして技術を高めようと言っている中で今の話を聞いてますと、SEとドメインの人の協力体系みたいなことが一番最初の現実的な話じゃないかと思います。

実際のモデル作りをするのにあたって、SE側として こういう体系を作ればいいとか、そういう話があるわ けです。

ドメインを作っていく過程で、ただのプログラマで 終わりたくないというのが我々の気持ちですし、かと いってドメインのことまで足突っ込むかといったら突っ込めないという立場にある。

そういうときになるとコミュニケーションという話が出てくると思います。

私もマルチメディア関係やってまして、ああじゃないこうじゃなかったとか、そういう話が出てきますが、そういうコミュニケーションを規制していく方法とかも、人間の要素が、たぶんフォーマルメソッドという形以外のどんな言語使っても出てくると思います。

そこら辺をやるために、前向きにどの辺を攻めたら いいのか。

こういう問題をこういう形で片づけたらいいとか、 こういうサンプルプログラムはこういうふうに作った 方がいいとか、そういうふうなものを具体的に片付け ていく方が、逆に今言われている技術的な小手先の話 よりも重要なことじゃないかなと思います。

そこら辺どういうふうに。

佐原:

技術的な問題の方も結構大きな問題だろう。

会場E:

それはわかってます。ただ技術的な問題というのは、我々が知識を積み重ねていけばできる話ですが、 今言った問題というのは、積み重ねるだけで本当にで きるのかというところもある思います。

先ほど言っていたように焦らないでやっていきま しょうって話は技術的な話は解決できますが、逆にド メインを抽出していくSEという、

中来田:

それは今回の話とは違い、商売の話に近いと思います。 エンジニアがお金を払ってくれる人との話。

今まではエンジニアのしゃべっている言葉をあなたたちは勉強してきなさいという世界でやってたのですが、そうはやっていけなくなってきて、お客さんに対して、お客さんの言葉で話すようなエンジニアの能力が必要になってきたという話で、それが出来る会社は伸びているという世界が現実です。

ですからその意味でSEはもっとエンドユーザーとかドメインとかに近づいて会話しなさいという話で、今回のとはたぶん違う話だと思う。

はい、どうぞ。

中谷:

私同感です。全然話は違わないです。先ほどから二 木先生おっしゃってますけれども、根本的な問題とい うのは、例えばフォーマルメソッド使おうとオブジェクト指向使おうと、構造化を使おうと、我々にとっては道具であって、何を考えるときにそれを使っていくかということだと思います。

先程の話で図書館問題の例がありましたが、自然言語で書いてあるときにそれはもう気が付かなければいけないとか、ドメインを理解していないとモデルを作れないということになってくると思います。

そうするといかなる技術を利用するにしても、やは りその一番ドメインに近いところに、ユーザーに近い ところにSEがコミュニケーションとれるかというとこ ろが一番重要だと思います。

コミュニケーションがとれて初めて、コミュニケーションとれたものを実現するために、どういうものを 道具として使っていくと最適なシステムが作れるかと いうところに。

ですから私は今回の話と全然離れているのではなく て、一番大事なことだという気がするんです。

佐原:

それはビューとしては違うでしょう。実際には重要な話ですが、視点がいろいろあって、フォーマルメソッドをどういうふうに普及するかとか、そういう話には絡みますけど。

会場E:

私が言いたいのは、この場でやられているビューという形では少しはずれていますが、でも先ほどからモデル化することが重要だということが出てますし、その手法として何があるかという話になれば、やはりこれは避けて通れない問題だと。

山崎:

結局その対象分野の課題を理解できなければ、何に もできないですね。だからどうしたらできるようにな るかという話だと思います。

得意な分野を作ればいいことだと思います。

基本的にはできたソフトウェアを見ると、きっとある型があると思います。

例えばアプリケーションを自分でどうモデル化した か、要するにどういう理論を作ったかです。これはそ の分野の専門家ではできない仕事ですね。

例えば、有限要素法という応力解析の手法があります。 飛行機の翼の解析あたりから発達したものだけ ど、バックグラウンドは考えてみれば変分法だとか で、しっかり数学的な理論として解析法を作りあげることができる。

それで、それを使ったアプリケーション・パッケージとしてNASTRANというのがあるのですが、そのNASTRANをどう作ったかに意味があります。

ある日本の企業で金魚のウンコみたいにたくさん、 有限要素法に基づく応力解析コードを作ったところが あります。

要するに近似要素の形に基づいてコードを1個作っ ちゃうから、五万とできるんですね。

ところがNASTRANは、膨大なコードにはなっているけれども、同じような課題に対するコードが五万とできたりはしない。

それはやっぱりソフトウェアエンジニアがいたから NASTRANができたのです。

ところがその有限要素法とか、応力解析の専門家が 一生懸命書くと、努力は多とするけれども、金魚の ウンコみたいにできてて後で困る。

結局捨ててNASTRANにしたというような話がある。

医者の診断システムを作ろうと思うと医者から聞き 出す能力がないといかんわけです。いち早く、できた システムを想定できないとだめです。

自分の身にあった分野でそういうことができるよう にするのだと思います。

例えば事務計算3年やってれば、大体どういう格好 になるかというのがわかって聞き出せると思います ね。

さっき鉄道の信号システムとおっしゃってましたが、例えば鉄道でなくて自動車の信号システムがあります。どうしたら道路に車がスムースに流れる信号の付け方があるかという話があります。

でもそのプログラム書く時に走行理論だの、あるい は待ち行列の専門家にならなくても、ソフトウェア エンジニアが寄与できると思います。

それはどういうことなのか。

自動車の流れについての理論ができていて、それに ついてはその分野の専門家に依存せざるをえないで しょう。

だけどその信号システムというプログラムを作るためには我々の出番があって、できた挙げ句はこうですよ、こういうことになるというイメージを持つとそのシステムが作れるようになる。

結局、青、黄、赤のサイクリックな流れを作り出す だけ。そういう赤、黄、青の3つの点滅するものが あって、基本的にはそれを制御するだけの話と思えば、ソフトとしてはわかるわけです。

そういうのがぱっと、ちょっと話を聞いてできなければならないわけです。

それは豊富に経験したからといって、できることじゃないかもしれません。

きっと外延から内包をとるような能力かと思います けれども。

何かいろんな現象を見て、それを説明する理論を作 る能力がないとだめですね。

そういう理論を作る能力はどこから生まれるかとい うのは、素直にはわからない。

たぶんパックグラウンドだとかによって決まることじゃないかと思います。

会場E:

ドメインと我々SEという間では、SEの方がある程度 ドメインに近寄った特化が必要だろうというのが今の 意見だと思います。それは確かに必要なことですし私 も賛成します。

ただドメインに対してある程度互いに解決しやすい 方に寄っていくだろうと思います。そういう手法が必 要になるという話です。

私がひっかかっていたのは、SEというのはドメインに対してどう対応するか。SEはプログラムを作ることに能力はあるが、先方の持っている知識を吸収していかないといけないし、コミュニケーションのやり方の手法も、個々SEにかかわっているかもしれませんけど、ドメインに関しては考え方が全く違う人達が多いだろうと。

その形への対応の仕方みたいな、というのがこれからの課題じゃないかなという気がする。

佐原:

先ほどから中谷さんと二人の意見を聞いて、何かおかしい。ユーザーという別ものがあって、それに近寄ってコミュニケーションとらないといけないという。

私の専門としている分野では、証券会社とか事務処 理とか、ドメインエキスパートはいない。行ってみた らいない。

証券会社だとしたら証券会社の業務を知らない。業 務って、伝票発行の仕方とか知ってますが。 だけど 本来そのモデル、証券会社でどういうことをどうやっ て、どう儲ければいいか。

例えば株の制度とかどうすればいいかということ を、少なくとも全体にわたって知っている人はいない。

会場E:

ドメインというのは交通システムから証券、事務処理、科学技術計算というものまでいろいろ範囲があるかと思います。

それを一長一短で全部処理しようというのは多分難 しい話だと思います。

佐原:

ドメインの知識を持ったユーザーがいるというのでなく、自分が入っていってその中にドメイン知識を作ればいいだけじゃないですか。

会場E:

一人でやっている分には構わないですが、やはりソフトウェア自体は、複数の人間が作っていくのでそこら辺は重要かと私は思いますが。

中来田:

あと残り4、5分で、最後にしたいのですが。

会場F:

峰尾さんの、一種のブームに終わらせたくないということに、非常にひっかかる。

基本的には一般の人間はツールとして捉えるもので すから。

ツールを提供する側とされる側とあって、やはりされる側ってのは次これですよって言われれば、ああそうですかって一生懸命勉強する。次はこれですよって言われればまた勉強する。

そうすると一種の疑心暗鬼といいますか。

実際非常に変な話ですが、片やプレゼンテーション の名士である山崎さん、岸田さんも、一切フリーラン スどころかワープロひとつ使わないでやっている。

我々は開発していて、ああですこうですと言われる と、次はこれを勉強しなければいけないってまた少し 経つとそれがですね。

佐原さんみたく、どんどんそれを物にしてこなせる 人はいいのだけど、大半の人間はフローチャート一回 書いたら、次のものは入らない人間です。そうする と、これ本気でやっていいのかというか、段々もう疑 心暗鬼になってくる。

ですから新しい人間に対して、今度はこれでやりましょう、教育しましょうというのは非常にいいのですが、他のものが頭に入っている人間に、それを変えてこうしなさと言うのは大変努力がいるわけです。

それをしてまでやる一つのテーマなのかどうかというのを、さっきの皆さんの話から私は。

そうであれば、やらなくちゃいけないと思いますけど、その辺がちょっと気になりますが。

中来田:

一言ずつ。

佐原:

コンサルティングしていろいろ見ていますが、実際 に見たシステムは、ほとんど目茶苦茶なのです。

その目茶苦茶な結果何になるかというと、すごいお 金が多分かかっている。 あらゆるユーザーのソフト ウェアを維持したり、改良したりに莫大なお金をかけ ている。

その莫大なお金は、例えばフローチャートしか書けないプログラマーがいて、それは新しいことを覚えられないからだというのは、おかしいなという気がする。だから変えないと、気づかないのではないかという気がしてます。

私はどんどん適用できると言いますが、大して適用 してなくて、何か道具を使うと、大体今まで使った ツールは、全部一つのプロジェクトで元が取れる、メ ソッドとかですね。全部一つのプロジェクトで元が取 れてる。それは、どなたがやっても。

二つ目のプロジェクトでないととれない人もいるかもしれないが、そんなに大変なことじゃないと思う。

多分実際にやってないから、大変だ大変だって言い ながら実際にやらないから、利益が得られないのでは ないか。

二木:

我々SEの基本的な素養とか、プロフェッショナルと の差別化ができないというのがかなり大きな問題じゃ ないかと思います。

そういうことからすると、フォーマルメソッドは峰 尾さんがおっしゃったとおり、そう流行る要素とか は。つまり比較的モデルを学習するとか、やり方自体 を学習するようなことなので、ツールを使うことは あっても、ツールだけ使ったからってフォーマルメ ソッドというのはそのやり方を実施しているとは思い ません。

そういう意味では、例えばどういうツールが売れた かという入り方はしないようなものだと思います。

そのかわりSEの、他の専門家と差別するための、職業としての一番基本の技術というものを考えていったら、別に誰が言われなくてもこういう種類のことは教育に入ってくる。そういう種類のものだと思います。

他のものとはそういう意味で性格が違い、峰尾さんが言いえて妙のとおり、そんなに流行らないけどそんなにすぐなくならない、というのが私の見方です。

中来田:

時間も来ました。はい、最後。

峰尾:

フォーマルメソッドのプロモーターって、ユーザー の立場で、僕もそうですけど、ものすごいギャップが あると思います。

プロモートする人は立派な人達だから、はい、これが仕様ですよって、さらさらって定義されるわけですけど、それを勉強している方というのは、やはり結構しんどいと思います。

実際にいろんな勉強が必要で、ヨーロッパのエスプリプロジェクトの報告を見ましたら、やっている立派な人達は大体マスターを出ている。専門のマスターを出ている。半分以上そうです。

決してフォーマルメソッドだけやっているわけじゃなくて、いろんな勉強をやっていると思います。それやった上でフォーマルメソッドは役に立つというか、そういうものだと思うので、だからやっぱり息長くやらないとだめじゃないかな、という気がします。

その穴を埋めていくには我々も、ユーザーにもどうやって提示したらわかりやすくなるかとか。

ビジュアルなものとの組み合わせだとか、いろんな 工夫の余地があると思うので、そういうものをいろい ろ、しこしことやっていくしかないのではないかな。

中来田:

どうもありがとうございました。これで終わりたいと思います。

(終わり)

ソフトウェア・シンポジウム 294

1994年6月15日(水)~17日(金)

於: 金森ホール (北海道函館市)

開催案内および参加者募集

主 催 ソフトウェア技術者協会



協 賛

日本ソフトウェア科学会 (JSSST) 情報処理学会 (IPSJ) 情報サービス産業協会 (JISA) 北海道ソフトウェア協会 (HSA)

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, さまざまな場で活躍している技術者, 管理者および研究者が一堂に会し, ソフトウェア技術に関する多面的な経験や知識を交流するユニークで貴重な場として開催されてきたこのシンポジウムも, 今回で14回目を迎え, 初めて本州を離れ, 津軽海峡を越えて, 函館で開催します.

ここ数回は、ダブルトラックでの会議がつづきましたが、今回は、会場の関係もあって、単一トラックで、招待講演x1、論文発表x5(17篇)、パネル討論x2というセッション構成で、プログラムを編成しました、ツール展示はなくし、その代わりに夜のパネル (IEEE Software 誌で話題を呼んだエッセイ "Software Remmingineering" に関するディベイト)を入れて、より充実した意見の交流が図れるように、と意図しました。

招待講演は、情報処理振興事業協会 (IPA) の棟上昭男さんに、「日本のソフトウェア産業の将来を考える」というタイトルで、鋭い問題提起をお願いしてあります。また、クロージングバネルでは、「ソフトウェアプロセスの国際標準化動向」と題して、最近とみに注目を集めつつあるISO-9000 や SPICE といった話題を中心に、最新状況の報告も含めホットな討論をたたかわせたいと考えています。

多数の方々の積極的な参加をお待ちします.

シンポジウム・スタッフ

実行委員長: 杉田義明 (日本 NCD)

プログラム委員長:玉井哲雄(筑波大学) 渡邉雄一(アスキー)

プログラム委員 (50 音順):

青山幹雄(富士通) 荒木啓二郎 (NAIST) 伊藤昌夫 (MASC) 大塚理恵 (RSK) 大場充 (広島市大) 兼子毅 (東京大) 久野靖 (筑波大) 岸田孝一 (SRA) 熊谷章 (PFU) 元治景朝 (さくら KCS) 佐伯元司 (東京工大) 坂本啓司 (オムロン) 佐藤千明 (長野県協同電算) 塩谷和範 (SRA) 武田淳男 (安川電機) 高橋光裕 (電力中研) 中野秀男 (大阪大) 布川博士 (東北大) 野口正浩 (新日本製鐵) 野呂昌満 (南山大学) 藤野晃延 (FXIS) 二木厚吉 (JAIST) 細谷僚一 (NTT) 堀江進 (NES) 松本健一 (NAIST) 山崎利治 (フリー)

ソフトウェア・シンポジウム′94 プログラム(予定)

***** 6月15日(水) *****

13:00 - 14:00 受付

14:00 - 14:30 オープニング 司会: 杉田 義明 (日本 NCD)

14:30 - 16:00 招待講演 司会: 玉井哲雄(筑波大学大学院)

日本のソフトウェア産業の将来を考える 棟上 昭男 (情報処理振興事業協会)

16:00 - 16:30 <休憩>

16:30 - 18:30 Session 1: CASE 司会: 野口正浩(新日本製鐵)

クラスライブラリにおける設計ノウハウの共有と再利用 佐藤 啓太,藤野 晃延(富士ゼロックス情報システム)

Tcl/Tk を用いた PCTE アプリケーションの構築

星 孝哲 (SRA)

スキーマ定義を利用した PCTE 上のオブジェクト指向環境 小室 睦 (日立ソフトウェアエンジニアリング)

統合環境に於けるツールの在り方と統合のためのツールの分析及び設計法に関する考察 伊藤 昌夫 (MHI エアロスペースシステムズ)

18:30 - 21:00 <情報交換パーティ>

***** 6月16日(木) *****

09:00 - 10:30 Session 2: プロセス 司会: 大塚 理恵 (RSK)

People Meet Process: 開発プロセスとの出会いと対峙

青山 幹雄 (富士通)

現状プロセスの徹底分析によるソフトウェア開発のプロセス改善 田中 敏文, 千頭 良造, 加地 覚 (オムロン), 北田 泰久 (オムロンソフトウェア)

ソフトウェア開発プロセス計画策定時の会話モデルとその実装方式

石若 通利,元治 景朝 (さくらケーシーエス),荻原 剛志 (奈良先端科学技術大学院大学),井上 克郎 (大阪大学)

10:30 - 11:00 <休憩>

11:00 - 12:30 Session 3: 品質·見積り 司会: 武田 淳男 (安川電機)

ソフトウェア品質の内部特性の階層化モデルの提案 三宅 武司,西山 茂,古山 恒実 (日本電信電話)

機能量にもとづく情報システムの見積りと評価

高橋 光裕, 菱谷 淳 (電力中央研究所)

T型マトリックスを活用したソフトウェア障害の分析 岡田 俊明 (富士ゼロックス)

12:30 - 14:00 <昼食休憩>

14:00 - 16:00 Session 4: 仕様 司会: 佐伯元司(東京工業大学)

仕様の分類によるマイコン組込型システムの要求仕様の構造化について 小松 幸広, 佐藤 正道, 丹羽 徹, 片山 泰司, 林 誠 (オムロン), 阿部 恵三, 大家 雅宏 (オムロンソフトウェア)

Human-Oriented な形式的仕様記述法

古木 良子, 荒木 啓二郎 (奈良先端科学技術大学院大学)

OBJによるZ仕様の検証の試み

谷津 弘一,本間 毅寛,中川中(情報処理振興事業協会)

マイコンソフト開発における設計手法の提案

小幡健司,相阪泰之,飯田薫,山田豊,尾本林貞(ダイキン工業)

ソフトウェア・シンポジウム′94 プログラム (予定)

16:00 - 16:30	<休憩>								
16:30 - 18:00	Session 5: 人間的要因 司会: 布川 博士(東北大学)								
	ACM カリキュラムに従ったプログラミング入門講座の事例								
	君島浩(富士通)								
	要求変更および保守要求要因の分析による利用者の開発参画に関する考察 中谷 多哉子 (富士ゼロックス情報システム)								
	CACE: A New Paradigm for CASE that Supports Collaborative Evolution of Software Artifact								
	Brent Reeves(SRA), Kumiyo Nakakoji(Univ. Colorado at Boulder)								
18:00 - 18:30	<休憩>								
18:30 - 20:30	イブニング・パネル 司会 : 塩谷 和範 (SRA)								
	Software Lemmingineering!?								
	パネル・メンバ 未定								
	***** 6月17日(金) *****								
09:30 - 12:00	クロージング・パネル 司会 : 大場 充 (広島市立大学)								
	これからのソフトウェア・プロセス・マネジメント								
	− ISO-9000, CMM, SPICE 等の標準化動向とその影響 −								
	パネル・メンバ 未定								
12:00 - 12:30	クロージング 司会:渡邉 雄一(アスキー) 最優秀論文賞発表/表彰								
	次回開催予告								
	A Lo#								
	参加費								
会員:	参加費 38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円								
会員:	38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円								
	38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円 参加申込み要領								
	38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円 参加申込み要領:必要事項をご記入の上,郵便または Fax または E-Mail で								
以下の申込書に	38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円 参加申込み要領 :必要事項をご記入の上, 郵便または Fax または E-Mail で 〒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 事務局								
以下の申込書にまでお送り下さ	38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円 参加申込み要領:必要事項をご記入の上, 郵便または Fax または E-Mail で 〒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 事務局 TEL: 03-3356-1077, FAX: 03-3356-1072, E-Mail: sea@sea.or.jp								
以下の申込書にまでお送り下さ	38,000円(SEA および協賛団体会員) 一般: 48,000円 学校関係者: 25,000円 学生: 5,000円								
以下の申込書にまでお送り下さ	参加申込み要領 - 必要事項をご記入の上, 郵便または Fax または E-Mail で 〒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 事務局 TEL: 03-3356-1077, FAX: 03-3356-1072, E-Mail: sea@sea.or.jp ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・								
以下の申込書にまでお送り下さ	参加申込み要領 必要事項をご記入の上,郵便またはFax またはE-Mailで 〒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事務局 TEL: 03-3356-1077, FAX: 03-3356-1072, E-Mail: sea@sea.or.jp い. 折り返し受付票および請求書をお送りいたします。 ***********************************								
以下の申込書に までお送り下さ 、 氏名: 種別: □ SEA	***********************************								
以下の申込書に までお送り下さ 、 氏名: 種別:□SEA 参加費:	***********************************								
以下の申込書に までお送り下さ までお送り下さ 、 氏名: 種別: SEA 参加費: 会社(学校)名:	***********************************								
以下の申込書に までお送り下さ までお送り下さ 、 氏名: 種別: □ SEA 参加費: 会社(学校)名: 部門:	***********************************								
以下の申込書に までお送り下さ までお送り下さ 、 氏名: SEA 参加費: Mean 会社(学校)名: 部門: CEM 住所: (〒	***********************************								
以下の申込書に までお送り下さ までお送り下さ 、 氏名: 種別: □ SEA 参加費: 会社 (学校)名: 部門: 住所: (〒 Tel:(***********************************								

航空券およびホテルの案内

ソフトウェア・シンポジウム'94 参加者の便宜のために,近畿日本ツーリスト(東京・錦糸町支店)の御協力により,往復の航空と宿泊をセットした格安出張プランを用意しました(通常料金と比較して,5~9,000 円ほど安くなります).

ホテルのご案内 (Check In: 6/15, Check Out: 6/17 または 6/18)

- ☆ 函館国際ホテル (港のそば、シンポジウム会場に最も近い)
- ☆ フィットネスホテル 330 函館 (JR 函館駅前, プールやジムの設備あり)
- ☆ ホテル法華クラブ函館 (史跡五稜郭の近く, 会場までは市電利用)

フライトのご案内(予定時刻)

往路: 6月15日 ANA851(羽田 7:30→函館 8:45) または ANA853(羽田 9:00→函館 10:15)

復路:6月17日 ANA862(函館 16:00 → 羽田 17:15) または ANA864(函館 18:00 → 羽田 19:15)

復路: 6月18日 ANA858(函館 13:30 → 羽田 14:45)

料金表	6/15 発 Air 2 泊 3 日		6/15 発 Air 3 泊 4 日		6/15 発 Hotel のみ 2 泊 3 日コース		
部屋の種別	シングル	ツイン	シングル	ツイン	シングル	ツイン	
函館国際ホテル	63,000	60,000	77,000	72,500	28,000	25,000	
フィットネスホテル 330 函館	56,000	53,000	66,000	62,500	20,500	19,000	
法華クラブ函館	54,000	53,000	64,000	62,000	20,600	18,000	

上記費用は往復航空運賃 (Air & Hotel コースの場合), 朝食付宿泊費 (税・サービス) を含みます. ただし, 航空運賃は, 申込人員が20名様に満たない場合は, 片道あたり2,000円のアップになりますのでご了承下さい.

ご希望の方は下の申込所の該当欄に〇印を入れ、必要事項を記入して、FAX にて下記までお送りください。 申込み締切は4月28日(木)です。なお、これらのコース以外のご相談にものりますのでお問い合わせください。

> 近畿日本ツーリスト 錦糸町支店 (担当:清水,斉藤,松浦) TEL: 03-3632-5321 FAX: 03-3632-4364

取消料: 5/24 まで無料, 6/7 まで旅行代金の 10%, 6/13 まで 20%, 6/14 まで 30%, 6/15 まで 50%.

ソフトウェア・シンポシウム'94 旅行手配申込書

		申込日:	月	日
申込者氏名 (ふりがな):()			
性別: □男 □女 年齢:才				
会社名:				
所 属:				
住 所:				
Tel: Fax:		_		
ツインの場合同室希望者名:				
申込みコース:				
□ Air & Hotel 2 泊 3 日 □ Air & Hotel 3 泊 4 日 □ Hotel のみ 2 泊 3	日			
希望のホテルとタイプ				
□ 函館国際ホテル □ フィットネスホテル 330 函館 □ ホテル法華 2 □ シングル □ ツイン (同室者指定) □ ツイン (同室者はだれでもよ		ブ函館		
希望フライト				
往路:6月15日 □ ANA851(羽田発 7:30) □ ANA853(羽田発 9:00) 復路:6月17日 □ ANA862(函館発 16:00) □ ANA864(函館発 18:00) 復路:6月18日 □ ANA858(函館発 13:30)				

CALL FOR PAPERS

First World Congress for Software Quality

- Software Quality Through Worldwidwe Teamwork -

June 20-23, 1995 Fairmont Hotel San Francisco, California, U.S.A.

Organized by:

American Society for Quality Control (ASQC)
European Organization for Quality (EOQ)
Union of Japanese Scientist and Engineers (JUSE)

For the first time, three major software quality organizations, the American Society for Quality Control(ASQC) - Software Division, the European Organization for Quality (EOQ) -Software Committee and the Software Production Control Board of the Union of Japanese Scientist and Engineers (JUSE), will be joining together for the First World Congress for Software Quality (1WCSQ).

The role of applying quality technology in the development of complex software products and systems is increasing more now than in any other time since the first line of code was written.

Software is used in all walks of life. Almost everything you encounter has software involved. As software quality professionals, we must assure that software is of the best quality available. In many cases, human life is dependent on software. There is no room for mistakes. We must also realize that we each have methods and processes that work. We must work together as professionals to assure the software is the best possible. This teamwork comes form sharing of ideas, research, and experiences.

1WCSQ intends to be the premiere world wide forum for exploring the science and practice of applying quality technologies to software engineering disciplines. 1WCSQ is seeking system and software engineers, managers, software quality professionals, customers, users, producers, and others interested in the application of software quality technology to join us in this historic event by presenting their own ideas, research, and experiences related to:

International Software Quality Standards
Teamwork in Multi - Cultural Environments
Software Quality from the Customer's View
Metrics and Data Collection/Analysis
Software Process Improvements & Application
Communicating with/Users World Wide
International Standards = World Wide Standards?
Barriers to Software Quality
Legal Issues in Software Quality
Results of Using Software Quality Principles
Competition and Teamwork for the Benefit of the Customer
Other Related Topics

INSTRCTIONS TO PROSPECTIVE AUTHORS

Prospective authors are invited to submit their full paper reporting their research or experiences to the Program Committee member in their respective region. The submitted paper needs to follow these listed guidelines:

- 1. The paper must be previously unpublished.
- 2. The paper must be written and presented in English.
- 3. Papers must be submitted via diskette or email and hardcopy.
- 4. The paper must be no greater than ten pages in lenght.
- 5. The title of the paper, author's name, affiliation (company, university, etc.) and country must be printed at the top of the first page.
- Single space typing shoule be used for the text and double space typing should be used to indicate paragraphs and subtitles/headings.
- 7. An abstract shoule be included at the beginning of the paper.

- 8. All references used must be numerically listed at the end of the paper. When references are made in the text, please indicate the reference number between slashes, e.g./1/.
- 9. Illustrations should be inserted as soon as possible when mentioned in the text.

All papers will be reviewed by the Program Committee. The Program Committee reserves the right to request to request modifications to accepted papers before publication. All accepted papers will be published in the congress proceedings and the authors will be expected to present their paper at the congress.

Authors are requested to submit along with their paper, a brief biography that includes their address, phone, and fax number.

CONGRESS COMMITTEE

General Chair: John E. Lowe (SOS)

3718 Ackerman Blyd. Kettering, Ohio 45429. Tel:(513) 429 - 6458 Fax (513) 429 - 6368

Program Ccommittee

Japan/Asian Region: Yoshinori IIZUKA (University of Tokyo)

Hongo, Bunkyo-ku, Tokyo 113, JAPAN. Tel: +81-3-3812-2111 ext. 7300 Fax: +81-3-3816-1760

American Region (North and Ssoth America): Doug Hoffman (Software Quality Methods)

1449 San Tomas Aquion Rd San Jose, California 95130. Tel/Fax: (408) 984 - 7560

European Region: Nicholas Ashley (BRAMEUR Ltd.)

Clark House, Kings Road, Fleet, Hampshire GU13 9AD. Tel: 0252 812 252 Fax: 0252 815 702

Middle East Region: Geula Nissimov (Eibit Computers Ltd.)

P.O.Box. 5390 < Haifa 31053, Israel. Tel: 04-316481 Fax: 972-4-550002

Tutorial Chair: Rusty Perkins

836 Oak Ridge, League City. Texas 77573. Tel:(713) 335 - 6861 Fax:(713) 335 - 6677

WCSQ STEERING COMMITTEE:

James Hemsley(Chair), Micheal Smith(ASQC-Software Division), Yoshinori IIZUKA(JUSE-SPC)

Ayatomo KANNO(JUSE-SPC), Francois de Nazelle(EOQ-Software Committee), John E. Lowe(Congress Director)

IMPORTANT DATES FOR PAPERS:

Full Papers due - July 1. 1994

Acceptance notification - December 1. 1994

Final paper due - February 1. 1995

CALL FOR TUTORIALS

On June 19. 1995, we will have a full day of tutorials. 1WCSQ is asking for tutorial proposals for subjects to be covered by internationally known experts. Please send any suggestions to the Tutorial Chair.

IMPORTANT DATES FOR TUTORIALS:

Suggestions for tutorials - July 1. 1994

Selection notification - December 1. 1994

Material due - February 1. 1995

INVITATION FOR EXHIBITORS

Vendors of software quality tools and/or companies providing services in support of Software quality are invited to exhibit their products during the congress. Interested vendors can contact the General Chair for information concerning the Congress Exhibit.

CALL FOR VOLUNTEERS

If you are interested in working on the Congress Committee or helping during the congress, please contact the General Chair. .sp .3

AMERICAN SOCIETY FOR QUALITY CONTROL

611 East Winsconsin Avenue, P.O.BOX 3005, Milwaukee, Wisconsin 53210-3005

本件問い合わせ先: 財団法人 日本科学技術連盟 第2事業部 (仲谷, 三田, 中島)

〒151 東京都渋谷区千駄ヶ谷 5-10-11 Tel: 03(5379)1222 Fax: 03(3225)1814

SEA 1993 年~1994 年のイベント (実績と予定)

1993	5/28 5/28 6/9 ~ 11 7/16 9/1 ~ 4 9/14 10/13 ~ 16 10/26 10/28 ~ 30 11/12 11/25 ~ 26 1/27 1/27 ~ 29 3/24	Seminar (技術の国際化・標準化) Forum (業界リストラクチャリング) Software Symposium '93 Seminar & Forum (ダウンサイジング時代の技術者教育) 第 11 回 夏のプログラミング・ワークショップ Seminar & Forum (インターネット) 泰安国際 CASE シンポジウム (TICS'93) Seminar & Forum (ISO 9000) 第 7 回 教育ワークショップ Seminar & Forum (CASE, OO, Formal Approach) 第 14 回ソフトウェア信頼性シンポジウム Seminar & Forum (技術者教育) SIGENV ワークショップ Seminar & Forum (SPICE) ************************************	東東仙東金東泰東福東奈東長東京京台京沢京安京山京良京野京
	4/21 5/26 6/2 ~ 4 6/15-17 9/21-24 10/27-29	Seminar & Forum(最新インターネット事情) Seminar & Forum(ビジュアル開発環境) 第 6 回 テクニカル・マネジメント・ワークショップ Software Symposium'94 第 12 回 夏のプログラミング・ワークショップ 第 8 回 教育ワークショップ	東京京田館岡路四国

入会申し込み先 〒 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 入会申込書 (正会員:入会金 3,000 円, 年会費 7,000 円) 94-03

氏 名			(ふりかん	₹)
生年月日	日:19年_	月	目	性別(男	女)	血液型	(A	ОВ	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:										
資料送付先 & 連	絡先(どちらかに	チェック)[□勤務先	□自宅						
	cm. 7 A ± 17 ==	/## TL A D .	~ ^ ** .		- /-		-			
	SEA 入会申込書	(質叨云貝:	平云實 1	□ 100,000 i	一, 10	口でも可) 5	04-03		
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:										
)									
TEL:		(内線_)	FAX: _						
申込口数:	П									



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