



# SEAMAIL

Newsletter from Software Engineers Association

---

Vol. 14, Number **5** January, 2005

---

## 目 次

編集部から		1
インターネット時代のソフトウェア開発	筏井美枝子 ほか	2
SEA 名古屋, 月見ミーティング レポート	石川雅彦	4
いくつかの国際会議報告	岸田孝一	9
Proceedings for 1st Joint Workshop on Embedded Software		17
SD2005 参加者募集		49
SS2005 論文募集		53
SEA Forum January 参加者募集		56

---

# ソフトウェア技術者協会

## Software Engineers Association

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

その主な活動は、機関誌 SEAMAIL の発行、支部および研究分科会の運営、セミナー/ワークショップ/シンポジウムなどのイベントの開催、および内外の関係諸団体との交流です。発足当初約 200 人にすぎなかった会員数もその後増加し、現在、北は北海道から南は沖縄まで、400 余名を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 20 社を数えます。支部は、東京以外に、関西、横浜、名古屋、九州、広島、東北の各地で設立されており、その他の地域でも設立準備をしています。分科会は、東京、関西、名古屋で、それぞれいくつかが活動しており、その他の支部でも、月例会やフォーラムが定期的に開催されています。

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

代表幹事： 荒木啓二郎

常任幹事： 熊谷章 高橋光裕 田中一夫 玉井哲雄 中野秀男

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

事務局長： 岸田孝一

会計監事： 橋本勝 吉村成弘

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

支部世話人 関西支部：小林修 中野秀男 横山博司  
横浜支部：野中哲 藤野見延 北條正顕  
名古屋支部：石川雅彦 角谷裕司 野呂昌満  
九州支部：荒木啓二郎 武田淳男 平尾一浩  
広島支部：大場充 佐藤康臣 谷純一郎  
東北支部：布川博士 野村行憲

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

SEAMAIL Vol. 14, No. 5 2005年1月15日発行 編集人 岸田孝一  
発行人 ソフトウェア技術者協会 (SEA)  
〒160-0004 東京都新宿区四谷3-12 丸正ビル5F  
T: 03-3356-1077 F: 03-3356-1072 E-mail: sea@sea.or.jp URL: http://www.sea.jp  
印刷所 市田印刷株式会社 〒114-0014 東京都北区田端2-3-25  
定価 500円 (禁無断転載)

## 編集部から

☆

半年ぶり、例によって船便の SEAMAIL です。

☆☆

この号の巻頭には、SEA 名古屋支部からのミーティング・レポート2篇を載せました。報告をまとめてくださった筏井さん、石川さん、ありがとうございました。

☆☆☆

その他の支部や分科会も、それぞれ活発に活動されているようですが、その内容をぜひここで報告していただけると嬉しいです。

☆☆☆☆

原稿不足を補うために、編集責任者として、この夏以降自分が参加した国際会議やミーティングの報告をまとめてみました。何かの参考にしていただければ幸いです。

☆☆☆☆☆

この報告のなかでも触れましたが、11月に西安で開催された ISFST2004 の Post-Symposium Proceedings が、プログラム委員長の塩谷さんの手でまとまりましたので、この号に挟み込んでお届けします。

☆☆☆☆☆☆

最後に載せた英文の Workshop Report は、昨年の秋に韓国・ソウルで開かれた組み込みワークショップの記録(参加された方々の Position Statement と3つのグループ討論のまとめ)です。

☆☆☆☆☆☆☆

岸田孝一 @ SEA Office

## インターネット時代のソフトウェア開発

SEA名古屋支部 ミニセミナー報告

記録担当： 後井 美枝子

5月21日(金)に、名古屋市鶴舞の<つながれっと名古屋>セミナールームで、「インターネット時代のソフトウェア開発」をテーマに月例のミニセミナーが開催されました。

今回はパネルディスカッションということで、中野秀男(大阪市立大学)先生、岸田孝一(SRA-KTL)さん、田中一夫(日本フィッツ)さんのお三方をパネリストにお迎えし、中野先生からはインフラ・メディア技術からの視点で、岸田さんからは社会学・認知科学的な視点で、田中さんからはプロジェクト管理に基づく開発現場からの視点で、それぞれ、インターネット時代のソフトウェア開発を語っていただき、それらの多角的な視点をもとに、参加者全員でディスカッションをしました。

まず最初に、中野先生から、今日のパネルプレゼンテーションの一部を録画したものをインターネットを介して、動画として見せていただきました。目の前の中野先生に紹介されて、インターネットの向こうの双子の中野先生がプレゼンしているようなおかしな感じで、会場が思わず笑いに包まれました。中野先生からは、ネットワークが格段に早くなったことで、どこでも仕事のできる環境になったことや、動画系の様々な技術が普及し、リアルタイム中継や電子会議など多方面への利用が進んでいることなどを、実際にデモをしながらご説明いただき、先端技術が利用技術になるまでの期間がどんどん短くなっていることを、参加者一同実感しました。

続いて、岸田さんからは、様々な文化的・歴史的背景を持つ人間集団の活動を研究するための概念的枠組みとして「Activity Theory」の紹介がありました。この理論は、そもそもは、1920年代にソビエトの教育心理学で提起されたものが、1980年代になって社会学・経営学に応用され西欧からアメリカに普及し、1990年代になって、CSCWやCHIの世界で話題になったきたのだそうです。さらに、岸田さんからCSCW Magazineに掲載された興味深い論文が二つ紹介されました。一つは、J.Engestromによるもので、人間集団の活動を分析するツールとして、Activity Theoryに基づく三角形のフレームワークを提示した論文です。岸田さんは「このフレームワークの最大の欠点は、このフレームワークで分析したからといって、何がどう役にたつかわからないことです」と愉快そうに付け加えました。もう一つは、B.Nardi、S.Whittaker、H.Schwartzによるもので、人間同士のプライ

バートかつ恣意的なネットワークを NetWORK と定義し、NetWORK はダイナミックかつ複雑に 様相が変化するのでフォーマルなダイアグラムに表しにくいとしながらも、ヒューマンインターフェースや CSCW にとっての重要性を強く示唆した論文です。

最後は、田中さんから「でも、変わらないプロジェクト・マネージメント」という副題で、ソフトウェア開発現場の経験談をおりませながら、インターネット時代になっても、プロジェクト・マネージメントの世界は相変わらずスマートに行かないというお話しをしていただきました。たとえば、見積もりの誤りや、仕様が fix しないことが今も昔もプロジェクトの失敗の大きな原因であること、システムは大規模・複雑化しているが、納期は短くなる一方であること等があげられました。また、プロジェクト・マネージメントに役立つもので、ツールとして MS-Project や xDTS、教科書として PMBOK、環境としては仕事場の共有、Daylymeeting の実施、PMO などが紹介されました。特に、田中さんお薦めの課題管理ツール xDTS (ニルソフト) は実データを使ったデモによる紹介があり、参加者から高い関心が寄せられました。

インフラやメディアの技術が充実・高度化する中、インターネットの利用はすごいスピードで広がっています。その利用法も、よりダイナミックに、より複雑なものとなり、人間同士のコミュニケーションや仕事のスタイルも劇的に変化してきています。一方で、それらがダイナミックで複雑になればなるほど、仕事の管理(状態の把握やコントロール)はより一層難しくなり、以前の管理方法では立ち行かなくなってくるのは明らかです。ソフトウェア開発の世界も例外ではないはずなのですが、プロジェクト・マネージメントの世界にはさしたる変化がないとの発表に、時代の変化速度とのギャップが感じられました。管理技術が時代の変化にまだ追いついていないためなのか、はたまた、マネージメントそのものの特性/限界なのか？

26名の参加者による和やかなパネルディスカッションは、場の盛り上がりそのままに懇親会へと流れこみ、パネリストを囲んでにぎやかに遅くまで続いたの でした。

## SEA 名古屋、月見ミーティング レポート

石川 雅彦  
(SRA)

2004年10月27日(水)は名古屋の天気は快晴で、ほぼ完全な満月が見られました。この日に合わせて、SEA名古屋のオフラインミーティングを開催しました。参加対象は、kiui と呼ばれる、企画運営委員です。なぜ kiui と呼ばれるかという点、Kikaku Un-ei lin から適当に略してつけたからです。kiui は、社会人ソフトウェア技術者向けミニセミナーの常連を中心に、1996年くらいから企画運営を続けています。活動から8年もたつとさすがに、kiui スタッフのポジションや関心も変わります。kiui スタッフの関心を中心に単独主催セミナーを2-3か月に一度開催するのが従来のスタイルでしたが、2004年の活動を振り返ると単独開催は1回のみ。後は他組織のイベントに共催するという状況でした(表1)。また、一昨年、昨年と開催した夏の合宿は今年は開催なしでした。そこで、ひさしぶりに集まって近況を連絡しあうと共に、今どのようなことに関心があるか、その関心を基にこれからどのようなことをしていきたいか話し合うことになりました。

表1: SEA名古屋 2004年のアクティビティを書き出してみた

- 
- ・ 第4回オープンソースセミナー共催(1月)  
これでもいいのかインターネット ～spamとプライバシーの観点から～
  - ・ 第5回オープンソースセミナー共催(5月)  
オープンソースを活用した、組み込みマイコン開発
  - ・ 第6回オープンソースセミナー共催(9月)  
「オープンソース三つ巴」「Linux Zaurus/PDA事情」
  - ・ 第7回オープンソースセミナー共催(10月)  
「オープンプリンティング」
  - ・ 第8回オープンソースセミナー共催(11月)  
「SH-Linuxカーネルのリアルタイム割込み」
  - ・ GWのKIUI ミーティング
  - ・ SEA 単独主催セミナー(5月)  
タイトル: インターネット時代のソフトウェア開発  
パネラー: 岸田孝一(SRA-KTL)、中野秀男(大阪市立大学)、田中一夫(日本フィッツ)
- 

開催場所は名古屋市内本山周辺と決めました。本山は地下鉄東山線の駅でしたが、地下鉄

名城線の乗り換え駅になりました。

地下鉄名城線は従来、大曾根から南下し、金山駅で名古屋港行きと新端橋(あらたまぼし)行きに分岐していました。10月?日から新端橋～八事～本山～大曾根とつながって環状線になりました。金山～名古屋港は名古屋港線と名称が変わり、おたまじゃくしのしっぽのように丸い名城線に接続するようになりました。開通間もないうちに、新環状線に乗りたい、というのが場所選定動機となりました。

会場は、1階窓際で月が見えることが条件で選定。本山から山手通りを名古屋大学の方に上っていくと、ちょうど月を左にながめながら移動することができます。上り坂の途中で見つけたお店をのぞくと運良く1階窓際が空いていました。

集合時刻は18:00ということもあり、参加者はぼつぼつと集まってきました。参加表明は7名でしたが、1名が仕事都合で欠席。他6名全員が、19:00過ぎには揃いました。

ひさしぶりのオフ会ということもあり、最初は近況や雑談で始まりました。が、あまりにローカルなので割愛します。ここでは、SEAMAILにふさわしいと思える話題のみを抽出して取り上げます。

#### 民芸品のようなソフトウェア

工業製品ではなく工芸品のような、あるいは民芸品のようなソフトウェアをつくりたい、という話題が出ました。民芸品のようななどは具体的に何かは明確ではありませんが、大量生産を指向しないもの、規格を統一することは必ずしも要求されず、むしろある範囲のばらつきに価値があると解釈することもできます。

民芸品の例として「さるぼぼ」が挙げられました。(現在のさるぼぼが民芸品であるかについては異論も出ました。)

またミーティング後の情報交換で、私たちと同年代のソフトウェア技術者が、「業界で最後の竹細工職人を目指している」という発言を引用し、異なる場所で同じ方向を目指す人がいることをうかがわせました。

同じく、ミーティング後の情報交換で

- ・ 「民芸」ということばは、柳宗悦や浜田庄司、バーナードリーチといった人たちが、提唱したものだ

という指摘があり、指摘者の解釈によれば、

- ・ 民芸品とは、人間の日常の中で丁寧に作られ、しかも高い芸術性をもつが、奢った芸術ではなく、民衆の生活の中に溶け込み、時代を超えて、つつましく肅々と伝えられてきたものだ

とのことでした。

#### 浦河べてるの家

この話は、kiui スタッフの興味の方が、テクニカルやオペレーショナルな面だけでなく、社会や福祉の方向に向かっていることを示すトピックとして挙げます。

この話題の後、関心を持ったスタッフが情報源を調査し、提供しました。

[www.tokeidai.co.jp/beterunoie/](http://www.tokeidai.co.jp/beterunoie/)

[www.jinken.ne.jp/challenged/beteru/](http://www.jinken.ne.jp/challenged/beteru/)

<http://6104.teacup.com/bethel/bbs>

[www.yomiuri.co.jp/iryuu/renai/20040727sr11.htm](http://www.yomiuri.co.jp/iryuu/renai/20040727sr11.htm)

<http://www.amazon.co.jp/exec/obidos/ASIN/4260332104/250-4925998-0508212>

またこれに関して「べてるの家の「非」援助論」という本はお勧め、という意見が出ていました。話は更に、2003年の夏にSEA名古屋で講師をした、ITステップの山口つかさ氏が、視覚障害者のためのITのボランティアをしているという話も掘り起こされました。

#### 関西オープンソース 2004 参加の感想

2004年10月22日(金)23日(土)にかけて、関西オープンソース 2004 が開催されました。  
(<http://k-of.jp/>) kiui スタッフのうち、自主参加が2名いましたが、示し合わせてもいないのに、会場で偶然出会った話が披露されました。一人の方は、情報家電の講演が印象的だったと発言していました。

#### 今後のアクティビティについて

kiui は仕事とは違う集まりですが、そのような集まりは 鼓舞されたり触発される機会が大事だという感想を抱くミーティングでした。

日常の活動はお互いに忙しく、接点がありませんが、背景の違いを前提にして、意見や情報を交換する場が必要だと感じられました。それは、2-3時間のオフ会ではとても語りつくせないように思われました。

これは今回の参加者に共通するものがあつたようです。場所をあらためて時間をとって話す機会を計画することになりました。

場も人も異なる状況に身を置くことで、自分の考えに新たな発見が期待できるように思います。一言で言えば「ワークショップ」となるのでしょうか。kiui スタッフ、あるいはもう少し範囲を広げたワークショップを企画することで今回のミーティングの結論となりました。



## いくつかの国際会議の報告

岸田孝一

(SRA-KTL)

なんとなくあわただしかった2004年も終わろうとしている。あらかじめかなり覚悟はしていたのだが、予想通り（というか予想以上に）忙しい思いをさせられた。あらためてカレンダーを振りかえって見ると、夏以降の海外出張は次の通り：

7/6-12（上海）	Dync Project Meeting
8/9-15（フフホト）	ISFST2004 PC Meeting
10/13-16（上海）：	Pre-ICSE International Workshop
10/19-25（西安）：	ISFST2004
10/31-11/5（Newport Beach）	FSE2004
11/29-12/4（釜山）：	APSEC2004

この間に SEA, JASPIC, SMSG, Dync などの国内ミーティングがいくつかはさまっていたので、いそがしかったわけだ。11/22-28 には毎年友人たちと東京・京橋の画廊で開いている絵のグループ展もあった。

以下に、それぞれの海外ミーティングの簡単な報告をまとめておく。

### 1. Project Meeting（7/6-12, 上海）

Dync (Dynamic Community) というのは、文部科学省からの補助金をもらって昨年の秋から SRA-KTL が、阪大（井上研究室）・NAIST（松本研究室）・東大（中小路研究室）と一緒に進めている研究プロジェクトである。その内容については今年の SS2004 Proceedings に採録されている論文「知識共創のためのダイナミック・コミュニティ：理論・アーキテクチャ・応用」（葉・山本・岸田）を参照されたい。

関連した研究を行っている内外の研究者との交流を目的として、シンポジウム形式の会合を積極的に開いている。昨年の12月には UC Irvine, 今年1月には函館未来大学との Joint Workshop を開催した。今回の上海ミーティングは、プロジェクトの発案者である葉雲文 (Ye Yunwen) さんが母校である復旦大学をお借りして、同大学および華東師範大学・華東理工大学の研究者、そして上海在住の SEA 会員である鍾友良さん (Hypersoft Shanghai) を招いて1日のワークショップを開催し、そのあと1日はプロジェクト内部の技術ミーティングを行った。

いま中国では中央政府のお声がかかりで IT 産業振興を目的としたソフトウェア技術者の養成が進められており、全国32の拠点校に軟件学院（ソフトウェア専攻科）が開設されている。復旦大学軟件学院の所長は、むかし葉さんのクラスメートだった臧斌宇 (Zang Binyu) さんである。コ

ンパイラ作りや並列処理の専門家、大学を出たあと、上海にある米国メーカの研究所で仕事をしていたが、学長命令で大学に呼び戻された。以前一緒に食事をしたとき、「上からはブルーカラーのソフト技術者をたくさん育てろといわれているのだけれど、大いに疑問に思っている」といわれたので、「その通り、中国のソフトウェア産業を振興するためには100人のブルーカラーではなく少数のスーパー・プログラマを育てることが大切ではないか」とアドバイスしておいた。今回かれは会議出張で上海を留守にしていたので会えなかったが、葉さんの話では、復旦の軟件学院は他の大学とは異なる教育方針で少数精鋭を目指しているとのこと。何年かあとの成果に期待したい。

## 2. ISFST2004 PC Meeting (8/9-15 (フフホト))

昨年の SARS 騒ぎのため1年順延になった ISFST、PC Meeting も昨年予定していた内モンゴル自治区の首都フフホトでの開催となった。この街には、1999年の PC Meeting in 太原の後に訪問したことがある。5年前と比べて街の風貌はかなり変っていた。いくつもの高層ビルが立ち並び、さらに新しいオフィスビルやショッピングセンターが建てられつつある。

PC 会場のホテルは金歳大酒店 (Jin-Sui Hotel) という名前なのだが、裏手に市の税務署のビルが隣接していた。訊いてみると、このホテルは税務署の経営なのだという。金歳の「歳」は中国語では「税」と同じ発音だそう。いうなれば Golden Tax Hotel !?

PC メンバーの1人松原友夫さんの奥様が戦時中にお隣(といっても100KM以上離れた山西省・大同市に住んでおられたとかで、車をチャーターして日帰りで当時住んでおられた場所を見に行かれたのだが、ホテルに頼んでやってきた車は地元警察の公用車、ドライバーは制服姿の若いオマワリさん。途中渋滞した高速道路をサイレンを鳴らしながら突っ走ったそう。警察がそんなアルバイトをしてもよいのだろうか。「何でもありの国ですね」というのが松原婦人の感想。昔の家は取り壊されてそこに学校が建っていたけれど、街のたたずまいは昔と変らなかったとのこと。

1年の空白が影響したためか、今年は中国以外からの論文応募がやや少なかった。中国からの論文も以前より質が落ちたような感じがする。ここ数年、中国で開催される国際会議の数も増えてきたので、昔のような ISFST の希少価値も失われつつあるようだ。そろそろモデル・チェンジを考えなければいけないと感じた。

PC が終わったあとバスで山越えをして訪ねた大草原は中国国内の観光ブームで大きく様変わり。以前にはなかった観光用のパオ(ただしコンクリート製!)の団地が点在していて、昔の面影は失われてしまった。ただし、夜空の星と、大草原に上る朝日の景色は壮観。

## 3. Pre-ICSE International Workshop (10/13-16, 上海)

みなさんご存知のように ICSE2006 は上海での開催が決まっている。しかし、中国ではまだ ICSE の存在はそれほど知られていない。また、欧米の研究者たちの目からすると、中国におけるソフトウェア工学の研究・実践の状況がほとんどわからない。そこで、両者の相互理解を深め

ICSE2006 in 上海 を成功に導くために、今年そして来年にいくつかの Pre-Event を開催しようということが、General Chair の Lee Osterweil 先生と Local Arrangement 担当の Ju Dehua (居徳華) 先生の相談で決まった。

軟件工程国際合作研討会と名づけられた今回のワークショップはその第1弾である。上海市政府 (IT を担当する部局である信息化委員会 Shanghai Municipal Information Commission) がスポンサーになり、欧米から合計10人の研究者を招いて3トラック併設2日間のワークショップが開かれた。

Invited Speakers List (with presentation title) は次の通り：

(1) Process Track

Lee Osterweil: "Precise Process Definition"

Jacky Estublier: Processes, Interoperability, and Coordination

Ross Jeffery: "Software Process Improvement"

(2) Testing and Analysis Track

Mary Lou Soffa: "Integrating System Environments into Software Testing"

Lori Clarke: "Finite State Verification approach for Software Systems"

Mary Jean Harrold: "Testing and Maintenance of Software"

Testability and Reliability of Component-based Software

(3) Distributed, Component-based System Track

Alex Wolf: Publish/Subscribe Communication for Distributed Systems"

Jeff Kramer: "Model-based Design and Analysis of Concurrent and Distributed Programs"

Carlo Ghezzi: Evolution of Software Composition Mechanisms

一見してわかる通り、現在のソフトウェア工学の第一線を代表するメンバーである、これらの人たちの招聘費は全額上海市政府が負担した (中国の官はお金持ち！)。

会場の銀河賓館 (Galaxy Hotel) は、虹橋空港 (いまは主に国内線に使われている古い飛行場) から市内へ向かう直線道路の終端にあるホテルで、泊まるのは今度が初めてだった。SARS の休業期間中に内装を取り替えたとかで、部屋は4つ星らしくきれいだった。

ワークショップ前日の午後に、これからの中国におけるソフトウェア技術・産業を振興させるにはどうしたらよいかという主題のミーティングがあり、Osterweil, Soffa, Wolf の3先生とわたしが Speaker に指名されていたので、チェックインもそこそこに会議室へ直行。朱三元さん (もと上海ソフトウェアセンター所長) ほかに何人か旧知のひとたちの顔も見えた。

わたしはいつも引用しているスペインの哲学者オルテガの言葉を引き合いに出して自立した強い精神を持つエンジニアを育てることがまず必要だと話したのだが、いつもながら中国でのこうした会議では会場からの質問やコメントがあまり出ないので、どの程度こちらの意図が通じたかどうかはよくわからない。

翌日からのワークショップでは、Process Track で3人の招待講演と何人かの中国側の発表を

聴いた。

Osterweil さんの講演 "Precise Process Definition" は、ひとことでいえば「プロセス・プログラミングの現在」ともいうべき内容で、いまかれの所属する Massachusetts 大学 LASER グループ (Lab for Advanced Software Engineering Research) [\*] で開発しているモデル化言語 Little-JIL とそのインタープリタ Little-JIL を中心にした話だった。

[\*] <http://laser.cs.umass.edu/>

技術的な詳細は2年前に東京(日大文理学部のシンポジウム)で聴いたのとほぼ同じだが、ソフト開発以外の分野(e-コマースや医療)でのプロセス記述と分析のケース・スタディは耳新しく興味深いものだった。

プロセス・プログラミングの思想的背景を説明したスライドの1枚に CMM についての評価が、次のような辛口のコトバで書いてあった:

The Capability Maturity Model (CMM) is a Specific Approach to  
Software Process Improvement

It is a test plan for black box testing of process

Can't test quality into software process product either.

この点については会場からも質問があったが、Osterweil さんの意見は、「現在多くの企業が CMM にのめりこんでいるのは不幸な間違いだ」ということだった。"Software Process is Software, Too" というのが、かれの基本的な考えなので、ソフトウェアとしてのプロセスの品質をテストするには、CMM のようなマクロなブラックボックスのテスト計画を立てるだけでは無理で、プロセス支援技術やツールを考慮に入れたマイクロ・モデルを用いたホワイトボックスのアプローチが必要だという指摘だと思われる。

現在の日本でも、METI の意向を反映してか、CMM 支持者が多いように見えるが、いつか適当な機会に、SEA-SPIN あるいは JASPIC で、こうした批判的な意見を聴いてみるのもよいのでは、と感じた。

昼食休憩をはさんだ、午後のセッションではフランス・グルノーブル大学の Jacky Estublier さんのプレゼンがあった。SCM (ソフトウェア構成管理) のツールとして有名な Adele システムと、その後継システムである Apel の開発や応用をベースに、プロセス研究の過去・現在・未来を展望した話で、Workflow, EAI, BPM, Web Service, MDA/MDE などの話題までを取り込んでうまくまとめてあるのに感心した。フランス的知性のなせる業だろうか。ちなみに Adele グループの Web Page は: [http://www.adele.imag.fr/index\\_en.html](http://www.adele.imag.fr/index_en.html)

かれが使ったスライドの1枚に:

#### Beliefs

- (1) A product cannot be better than process by which it is produced  
CMM, RUP, etc
- (2) A company can improve its products, its productivity, and its benefits only by improving its process.

## SPI trend

## (3) A process is useful only if executable

## Software Process

という3つの信条のリストがあった。つまり、プロセスをどうとらえるかは「信仰上の問題」(!)なので、どれが正しいというのではなく、違いがあるということである。

かつて一連の ISPW (国際プロセス・ワークショップ) に参加していた人たちの多くは(3)の立場だったと思う。ツール開発者である Estubilier さんもその1人。CMM 発案者の Humphrey さんは少数派の(1)だった。

2日目朝の招待講演はオーストラリアの Ross Jeffery さん, University of New South Wales の先生なのだが、いまは政府の出資でできた NICTA という非営利組織で Empirical Software Engineering Research Project のリーダーに指名され、いまはそちらがメインの仕事になっているらしい: <http://nicta.com.au>

産学協同でプロセス・モデルにもとづくソフトウェア開発環境の仕事を手がけているという。具体的には、小規模(20~30人規模)のソフトウェア企業におけるプロセス改善を実際に行っていて、そうした規模の組織では、ソフトウェア・プロセスだけを改善してもだめで、それと密接に結びついたビジネス・マネジメント・プロセスの改善を一緒にやらなければいけない。そのための支援環境を開発し、データの収集と分析を行っているという話は、かなり具体的で説得力があった。

CMM についてはやはり批判的で、会場からの質問に答えて、「CMM が提唱されたのは、もう10年以上も前のことなのに、それ以降ソフトウェア・コミュニティに対して、なんら見るべき技術的貢献をしていないというのは、いかがなものか」と述べていた。

Small & Medium Size の企業が多いオーストラリアの場合には、SPI とビジネス・マネジメントの改善を一緒に扱えるシステムが必要だというのが、かれの主張。オーストラリアでは、その機能を持った SoftwareMark というシステムが広く使われているとのこと。Software Engineering Australia (略称: SEA) という会社が提供しているようだが、詳細は不明。

<http://www.seanational.com.au/index.jsp>

他の2つの Track を覗く余裕はなかったが、それぞれかなり充実した内容だったようだ。第一線のソフトウェア工学研究者の話がまとまって聴けるよい企画だと思ったので、あちこちに mail で PR したのだが、結局、一般参加の聴衆は中国の人たちだけで、日本や韓国からの参加がなかったのはちょっと残念だった。

ICSE2006 Pre-Event の第2弾としては、すでに会員 DM でもお知らせしたが、来年の5月末、ICSE 2005 in St. Louis が終わった次の週に、北京で Software Process に関する国際ワークショップが開催される。Program Co-Chair は、Osterweil さん, Barry Boehm さん, そして中国科学院・軟件研究所長の李明樹(Li Mingshu) さんの3人。PC には阪大の井上先生、NAIST の飯田先生、それにわたしも加わっている。日本からも多数の参加を期待したい。

#### 4. ISFST2004 (10/19-25, 西安)

上海ワークショップを終えて一旦東京に帰り、週末から月曜日にかけて雑用を片付けてまた中国へという忙しい旅になった。

西安訪問は一昨年の ISFST の Pre-Symposium Seminar 以来 2 年ぶり。街はさらに変貌していた。夜の空港に着いてタクシーで、市の郊外の大雁塔近くにあるホテルに向かった。玄奘三蔵ゆかりのこの古塔は 20 年前、初めての中国旅行のときに来たことがあるのだが、かつては廃墟に近かった風景がきらびやかな観光地に変っていて、塔も美しくライトアップされているのに驚いた。街がきれいになったのはうれしいが、しかし、途中の高速道路の側壁を彩る三色のネオンライトも含めて、大雁塔周辺の公園のデザインやライティングの色彩感覚は、折角の古都の風情を台無しにしているような感じで、あまりいただけない。

シンポジウムの主会場は 8 年前の ISFST1996 のときと同じ西北大学のサイエンス・ホール。ここは以前と変わっていないが、隣接するゲストハウスは新しくなっていて、パラレル・セッションにはその会議室が使われた。

1987 年秋、上海でのシンポジウムから始まった中国での国際会議も今年でもう 16 回を数える。日中シンポジウム 3 回(1987-90, 天安門事件で 1 回中止)、国際 CASE シンポジウム 5 回(1991-95)、国連大学 (UNU/IIST) の協賛を得て現在の ISFST になってから 8 回(1996-2004, SARS で一回中止)。PC Meeting の項で書いたように、そろそろ新機軸を考えるべき時期である。今回は、かつて北京(1991)およびアモイ(1997)の会議で試みて好評だったワーキング・グループ討論のセッション 3 つを会議初日の午前中に組み入れて、若干のモード変更を図った。それぞれ適当に盛り上がって成功だったと思う。

本会議オープニングの基調講演は、岡山の SS2004 にもお招きした韓国 POSTECH の K.C.Kang 先生。テーマはもちろん Product Line Software Engineering。これから何年かソフトウェア工学の中心的话题になるトピックだが、まだまだ研究先行で産業界における実践が遅れているのが現状だと思う。いま、Kang 先生が韓国や日本で進めておられる応用がうまくショーケースとして使えるような結果を生み出すことが望まれる

午後からの本会議は 3 つの並列トラックに分かれての論文発表。ここ数年、論文が採択されているのかかわらず、会議にはやってこない不心得者の数がふえてきた。アカデミック・ポイントを稼ぐためには、論文が Proceedings に載りさえすればよいと考える若い学生や研究者が多くなったらしい。こうした傾向は全世界共通のようで中国もその例に漏れない。ISFST では、そうした No-Show を防ぐために、前回から会議開催前に Proceedings を印刷することを止め、実際に会議で発表された論文だけを収録した Post-Symposium Proceedings を CD 媒体で作ることにした。

今回の会議の Proceedings CD は、この SEAMAIL に同封されているので、発表論文の詳細については、その CD をご覧いただきたい。

SEA が主催する会議は、ISFST もそうだし、国内の SS そのほかの会議もそうだが、Proceedings に論文を載せるのが目的ではなく、会議の場での論文発表とそれをきっかけとする

討論や交流を目的としている点が通常のアカデミック・コンファレンスとは異なるのだが、そのあたりの事情はなかなか一般には理解されず、論文集めやプログラム編成には苦勞が多い。

来年の ISFST は、ICSE2006 の Pre-Event の 1 つとして、上海での開催が予定されている。このへんで普通の国際会議とは一味違う特徴をだすために、論文発表を少なくして、いくつかのホット・トピックについてのグループ討論を主体にした形に変えるのも一案ではないかと考えているのだが、どうだろうか。これまでの経験からすると、中国の人たちは（韓国の人たちも）そうしたワークショップ形式の会議はまだ慣れていないような感じがするが、思い切って試してみたい。日本でも、1980 年代の初めに SEA の前身である旧ソフト協技術委員会がワークショップ・スタイルの合宿会議をはじめたときには、「なんで技術問題を討論するのに田舎のホテルや温泉に泊り込みで行く必要があるの？まさか遊びじゃないだろうね？」と、怪訝な顔をされたものだった。

西安での会議が終わったあと北京へ飛んで、病氣療養中の鐘錫昌 (Zhong Xichang) 先生を見舞った。教えられた 301 病院に行ってみると、正式名称は人民解放軍総医院。あとできいたら、中央政府で 2 番目に偉いランクの人がはいるところだそうだ。科学院ソフト工学センター所長というのはそうした地位なのだとあらためて感心した。

軍服姿の兵隊さんの門衛に特別治療棟の場所を教えてください、広い敷地の中まで歩く。建物の受付デスクにもやはり兵隊さん。来意を告げ、教えられた階段を上がって行くと、受付から連絡があったのだろう、奥様が廊下の外れまで迎えに出て来られていた。点滴の管はつけていたが、鐘先生は元気そうな様子だった。ちょうどアメリカ (ニューヨーク州立大) に留学中のお嬢さんと電話がかかっていたところだったので、東京で顔見知りだったお嬢さんともひさしぶりに日本語・英語とりまぜてお話しした。

去年の秋、中野先生が大阪で開かれているオープンソース・ソフトウェアのイベントに鐘先生をご招待して、本人も来る積りになっていたところへドクター・ストップ。それ以来ずっと入院生活だから、もう 1 年になる。科学院のオフィスの隣にあったアパートが道路拡張で取り壊されることになったので新しくアパートを買ったのだが、まだそこに足を踏み入れていないのだと苦笑されていた。治療の経過がよさそうなので、「来年には退院できるでしょう。お元気で」と別れを告げた。

##### 5. FSE2004 (10/31-11/5, Newport Beach)

Foundation of Software Engineering と名づけられたこの会議は ACM/SIGSOFT の年次大会で、偶数年は 11 月にアメリカで、奇数年は ESEC とジョイントで 9 月にヨーロッパで開催されることになっている。3 年前の 9 月、ウィーンでの会議は、会議初日が 9/11 のテロにぶつかって大騒ぎ、レセプションをそこそこに切り上げてホテルに帰り、BBC と CNN を交互に切り替えながら TV を観ていたことを思い出す。

Foundation というネーミングを見て、最初はさまざまな方法論の思想的「基礎」を議論する会議なのかと考えたのだが、それはこちらの勝手な思い込みで、発表論文のほとんどはかなり細

分化されたソフトウェア工学技法/ツールの詳細を論じたものが多い。わたしが今回参加した主な目的は、会議に併設して開かれる ICSE2006 準備委員会に出席すること、そして来年の SS2005 の基調講演者に予定している UC Irvine の Andre van der Hoek さんにお会いすることだったので、論文発表は個人的興味に合うものだけを適当に聴くだけにとどめた。

今回の会議場は、昨年の暮れに Dynv Project の UC Irvine Meeting のときに泊まったのと同じ Sheraton Newport Beach Hotel。チェックインのあと、ちょっと休憩して、やはり ICSE2006 準備委員会のために来られた JAIST の二木先生と、去年クリスマス・ポート・クルーズを見物にいった近くの島まで散歩して、イタリア料理を食べる。

3日間の会議の朝一番はそれぞれ招待講演。講演者とタイトルは次の通り：

Alex Wolf (University of Lugano)

“Is Security Engineering Really Just Good software Engineering?”

Joe Marks (MERL)

“The Usability Problem for Home Appliance: Engineers Caused It,  
Engineers Can Fix It”

Nancy Laveson (MIT)

“Making Embedded Software Reuse Practical and Safe”

Foundation というよりはむしろ Application of Software Engineering に近い内容のように感じた。

ちょうど会議初日が大統領選挙の投票日にあたっていて、TVは一晩中オハイオ州の開票速報について、同じようなコメントを繰り返していた。会議に集まっていた先生方はほとんどがケリー一支持で「なぜまたブッシュなの？」といささか諦め気味の表情を見せていたが、数日前に出されたピンらディンのメッセージが、ブッシュ再選を希望したかれの思惑通り、アメリカの一般大衆にインパクトを与えたのだろう。

## 6. APSEC2004 (11/29-12/4, 釜山)

この会議が韓国で開かれるのはこれで3回目になる。最初はまだ日韓ジョイントの会議だった(1992年のソウル)、パネルに招待されて参加したのが、わたしの初めての韓国訪問だった。そして参加国が拡大されて APSEC になってから第3回目(1996)がやはりソウルでの開催。あの時は故・坂本啓司さんや伊藤さん、中小路さんが一緒だった。

その後の3年間、わたしは、香港(第4回)、台北(第5回)、高松(第6回)と連続してこの会議に参加している。1997年の香港では、坂本さんほか SEA-SPIN のメンバーと一緒に、香港 Productivity Council とジョイントで併設のプロセス・ワークショップを開催、1999年の高松では、やはり中国・韓国の人たちを招いて国際ワークショップを開催した。

今回の釜山会議では、昨年11月ソウルでの組込みワークショップ、そして今年4月釜山での JCSE で K.C.Kang 先生や荒木先生たちと企画を話しあってきた PLSE に関する併設ワークショップの開催が実現した。



いざふたを開けてみると、日本からの Position Paper はかなり集まったが、韓国からの応募が予想より少なかった。これはあちらの経済事情のせいらしい。一時の危機的状況を IMF の支援で乗り切ったかに見える韓国経済だが、実態はまだかなり苦しく、IT 不況も底を打ってはいないように見える。某 S 社だけは元気がよいが、これにも裏があるらしい。今回の APSEC in 釜山の場合にも、S 社ははじめ協賛企業に名乗りを上げていたのだが、間際になってドタキャン。韓国の先生方もかなり頭に来たようだった。

ワークショップには S 者のエンジニアも何人が参加していた。そのうちの 1 人と夕食を一緒にしたとき、「どうなの？」と訊いてみた。「PLSE には個人的には興味があり、ぜひ現場に導入してみたいと思っているのだが、そのためにはいささか長期的な投資が必要。いまの現場にはそうしたことは許されそうもない。わたしの名刺の肩書きは Researcher となっているけれども、実際は研究職ではなく、現場のエンジニア。うちの会社には研究開発に投資をするという考えはないみたい。よその会社や大学にまかせておいて、実用的な成果が出そうになったらその人をスカウトしてくればよいというのが上の考え。困ったものです」。それを聴いていた韓国の先生いわく「早く君が出世してそうした企業風土を変えてくれよ」。日本の大企業も、バブル経済が崩壊して以降、多かれ少なかれ同じような傾向にあるというのがわたしの個人的感想。

今回の会議の招待講演者は、初日が Hassan Gomaa さん、3 日目が Mary Jean Harrold さん(2 人とも Georgia Tech) だった。タイトルはそれぞれ：

Gomaa: "Real-Time Design for Embedded Systems: From Centralized Control to Distributed Component-based Product Line"

Harrold: "Improving the Quality of deployed Systems through Remote Analysis, Monitoring, and Testing"

であった。講演のスライドは会議の Web Page (<http://se.kaist.ac.kr/apsec2004/>) に掲載されている。

Gomaa さんの講演は予想したとおりのチュートリアル風の話で、それほど印象に残らなかったが、Mary Jean Harrold さんの講演は内容が新鮮で興味深かった。かの女は、11月の上海ワークショップにも来ていたのだが、あのとき、かの女は Soffa さんや Clarke さんと一緒に別の Track (Testing and Analysis (このテーマはなぜか女性研究者が多い) に入っていたので、話をきくのは今回が初めてだった。

Georgia Tech のかの女の研究グループは、いま Gamma と名づけられたプロジェクトを進めていて、今回の講演もその内容紹介であった。プロジェクトの Web Page:

<http://www.cc.gatech.edu/aristotle/Research/Projects/gamma.html>

には、その概要が次のように述べられている：

The *Gamma* project of Georgia Tech proposes a new approach to improve software quality through continuous monitoring, analysis, and improvement of software systems after their deployment. The motivation for the project is the fundamental shift that we witnessed in the world of computing. We moved from a situation

characterized by a few software system running on mostly disconnected computers to a situation in which software is virtually everywhere, runs on mostly connected computers, and there is a large number of resources available to the single users.

Thank to this shift, we have the opportunity of leveraging field data and resources to improve the way in which we perform software engineering tasks.

Many software engineering tasks are limited by the fact of being performed in-house, on developers' platforms, and using developer's provided inputs that may not be representative of the way the software is used in the field. To overcome these limitations, we use field data and resources to augment software engineering tasks.

現在、われわれの周囲にある多くのソフトウェア・システムが、孤立した1台のマシンの上で稼動するのではなく、ネットワークでつながれた複数のマシン（コンピュータ、携帯電話、そのほか各種のデバイス）の上で分散して稼動していることは事実である。そしてそのシステムは周囲の条件の変化に応じて絶えず進化し続けている。そうしたシステム進化を遠隔モニタリングでフィールド・データを収集し、効率的な回帰テストを行ってシステム・メンテナンスがもたらす影響を分析するというアプローチは、まさしく王道を行くものといえるだろう。SMSG (ソフトウェア・メンテナンス研究会)で、今年はこの Gamma システムの行方をトレースしてみたいと考えている (<http://www.smsg.or.jp/>).

**Proceedings**  
*for*  
**1<sup>st</sup> Joint Workshop on Embedded Software**

**Nov. 19 ~ 21, 2003**  
**Seoul, Korea**

*Edited by*  
***Kouichi Kishida, Kyo Chul Kang, and Soo Dong Kim***

## **Workshop Organization**

- **General Co-Chair**
  - Kouichi Kishida, SRA-KTL, Japan
  - Kyo Chul Kang, POSTECH, Korea
- **Program Co-Chair**
  - Keijiro Araki, Kyushu University, Japan
  - Sungdeok Cha, KAIST, Korea
- **Organization Chair**
  - Byongju Choi, Ewha Women's University, Korea
- **Program Committee**
  - Takao Futagami, Toyo Corporation, Japan
  - Masahiko Ishikawa, SRA, Japan
  - Masao Ito, Nil Software, Japan
  - Keita Sato, Denso, Japan
  - Soo Yong Park, Sogang University, Korea
  - Kwangkeun Yi, Seoul National University, Korea
  - Jang Soo Lee, KAERI, Korea
  - Hoon Sun Chang, KOPEC, Korea
  - Jin Young Choi, Korea University, Korea
  - Soo Dong Kim, Soongsil University, Korea

## Table of Contents

1. Track A. Process and Project Management	
1.1 Position Statements	20
Kouichi Kishida, Koichi Sugahara, Noriko Wada	
1.2 Group Presentation	22
2. Track B. Testing and Quality Control	
2.1 Position Statements	27
Ikuko Suzuki, Masahiko Ishikawa, Takamasa Nara	
2.2 Group Presentation	28
3. Track C. Development Method and New Paradigm	
3.1 Position Statements	34
Sungwon Kang, Eunsook Cho, Soodong Kim, Sooyong Park, Takao Futagami, Masao Ito, Seiji Tokimatsu, Keijiro Araki, Norihiko Matsuda, Yoshikazu Hayashi, Akira Fukuda, Yoji Yamanoue, Hitoshi Yamane, Shigeru Hitomi	
3.2 Group Presentation	39

# 1 Track A. Process and Project Management

## 1.1 Position Statements

### 1.1.1 Kouichi Kishida

I have organized JASPIC (Japan SPI Consortium) 3 years ago. Now we have now 24 member companies, and a half of them are doing systems business. Generally speaking, those "embedded" companies are eager to do CMM-related SPI activities than other companies who are doing ordinary (rather business-oriented) software business.

One of the reasons is that software becomes a more important part of embedded system products. Those "embedded" companies have long experience in hardware manufacturing but do not know about the process of software development. So, they want to normalize software development process by using CMM or ISO standard as a guideline.

On the other hand, ordinary software houses are living in other business environment. Software systems which they are dealing with are "E-type" software by the definition of Prof. M.M.Lehman. "E" means "embedded", not in hardware, but in real world application process of radical change. So, their software development processes are difficult to normalize or control. Essential causes of the difficulty in front of them may not be solved by CMM or ISO.

At this workshop, I would like to discuss about the nature of embedded system development process, and to find out the proper direction of future improvement.

### 1.1.2 Koichi Sugahara

After developing various kinds of embedded software about 10 years, I became a SEPG staff within the company. At first, our activity was somehow ad-hoc style, and could not achieve any visible result. So, we decided to use CMM as a guideline for SPI activity.

- Lessons learned:
  - (1) Importance of project plan document to monitor the status of actual practice, especially to watch deviation from the original plan.
  - (2) Importance of visible presentation of QA result to the management.
- My observation about CMM:

(2) There may be several sub-step between level1 and 2 of CMM.

(3) We should prioritize various practices in level 2.

□ I'd like to propose following maturity model for embedded systems development:

○ Level 1.2: Bug management is OK.

○ Level 1.4: Requirement change management is OK.

○ Level 1.6: Schedule management based on WBS is OK.

○ Level 1.8: Quality assurance is OK.

○ Level 2.0: Quantitative risk management is OK.

### **1.1.3 Noriko Wada**

I will present the idea of evaluation model for SPI activity (5 evaluation parameters of organizational change). this talk is based upon her experience of SPI activity in Sony.

It shows a typical example of Japanese style "Bottom-UP" approach to SPI.

## 1.2 Group Presentation

### 1.2.1 Agenda

We had following 8 members:

- From Japan (3): Mr. Kouichi Kishida (SRA-KTL)  
Mr. Koichi Sugahara (Fuji Photo Film)  
Ms. Noriko Wada (Sony)
- From Korea (4): Prof. Doohwan Bae (KAIST)  
Prof. Hyuksoo Han (KSI)  
Dr. Hoonseon Chang (KOPEC)  
Mr. Seunghwa Yi (LG Innotek)

Unfortunately, most of Korean participants could not attend full time, because of some other business appointments.

We had following 4 presentations followed by question-and-answer

:

#### 1st Session (Afternoon of Dec. 20)

Process improvement activities from the level 1 to the level 2

by Mr. Koichi Sugahara

Exploring relationships between organization culture and SPI activities according to IDEAL model

by Ms. Noriko Wada

#### 2nd Session (Morning of Dec. 21)

Activities of SPIC Korea

by Prof. Doohwan Bae

Application of Embedded System in Nuclear Power Plant

by Dr. Hoonseon Chang (KOPEC)

## 1.2.2 Presentations

### 1.2.2.1 Mr. Koichi Sugahara

The title of Mr.Sugahara's presentation was: "Process improvement activities from the level 1 to the level 2".

Mr. Sugahara is now working as a full-time SEPG in Fuji Photo Film company after engaging embedded software development for the Mini-Lab systems more than 10 years. He presented his proposal of setting up several sub-steps in between level-1 and level-2 of CMM phases. He said that according to his own experience as SEPG, these sub-steps



will be useful for guiding SPI activities in embedded software development.

Proposed sub-steps from Level-1 to level-2 are as follows:

- Level 1.2: Bug management is OK.
- Level 1.4: Requirement change management is OK.
- Level 1.6: Schedule management based on WBS is OK.
- Level 1.8: Quality assurance is OK.
- Level 2.0: Quantitative risk management is OK.

The purpose of this proposal is to prioritize various KPAs in the course of SPI from level-1 and level-2. In such a sense, the groupmembers agreed that it will be useful somehow.

#### 1.2.2.2 Ms. Noriko Wada

The title of Ms.Wada's presentation was: "Exploring relationships between organization culture and SPI activities according to IDEAL model".

Sony started a company-wide project called SW21 to evaluate and improve software management practices according to the Software CMM and IDEAL model. She is a member of corporate level staff team to support several software development organizations within the company.

In SW21 project, 4 organizations were selected with the goal of achieving CMM level-2 within 2 years. Some of them achieved the goal and some failed. Ms. Wada and her colleagues made an analysis of reasons of success and failure. The question they had was: " Are there any elements to the success of SPI activities other than the tasks described in IDEAL model?"

So, they made evaluation of the progres of SPI activities according to the following hypothetical equation:

$$\text{Organizational change} = f(\text{Environment [Internal, External], Activities for change, Ability to accept changes, Previous condition})$$

For actual evaluation, they slightly modified the equation as follows:

$$\text{Phase achievement score} = f(\text{Environment, Specific Practice activities, General Practice activities, Previous phase scores})$$

After careful analysis of 4 organizations results, they observed several interesting

phenomena, and derived a number of hypothetical knowledge about the relationship between organizational culture and SPI activity progress (For the detail, see the slides in appendix).

The group members deeply impressed the data shown and are looking forward further survey in future.

### 1.2.2.3 Prof. Doo Hwan Bae

Prof. Bae made an informal explanation about Korea SPIC (Software Process Improvement Center), which is one of 32 research centers under control of ITRC.

In SPIC, there are 4 projects are running:

PSEE: Porcess-oriented Software Enggineering Environment

SQA: Software Quality Assurance

CMM Supporting Process (Requirement, SCM, etc)

CMM Management Process (Metrics, Measurement, etc)

As for PSEE, the goal of the project is to produce an integrated process-oriented CASE tool environment, which will be adaptable to the needs of each customer, in 3 years.

In porcess area, SPIC has already produced the guideline for CMM/CMMI implementation for small/medium companies. It is a practical guide for implementing KPAs towards level 2.

The other important mission od SPIC is to train process experts through seminar or workshop in cooperation with KSI. Also, teach process technologies to graduate students in university.

It seems to be a good idea to provide support tool environment with management guideline. The action of Korean government is quicker than that of Japanese government, and also little more integrated.

### 1.2.2.4 Dr. Hoon Seon Chang

The title of Dr. Hoon Seon Chang was: "Application of Embedded System in Nuclear Power Plant".

He explained the situation of software development and maintenance projects for nuclear power plants in his company.

The projects are those of analog systems to digital systems. In the case of nuclear power plants, safety is much more important than efficiency. So, to apply new development methods, you must be very cautious.

Also there are a number of strict regulations and standards to follow. Development life-cycle is classic waterfall model with strict verification and validation by third party.

Systems are safety critical, and the delay of development causes heavy cost problems in users side. It looks like yet another embedded systems with very unique characteristics .

### 1.3. Group Coordinator's Summary (Kouichi Kishida/SRA-KTL)

In Japan, SEA-SPIN was kicked off in 1995. Its major activity is information exchange via ML, regular off-line meetings, and workshops on special topics. Now we have about 120 ML members, and more than 50 among them are from embedded systems community.

Also, a non-profit organization called JASPIC (Japan SPI Consortium) was established 3 years ago. Now It has 24 member companies, and half of them are manufacturing embedded systems.

The reason why?

There is an essential difference between hardware manufacturing process and software development process. Embedded systems people are experts in hardware process, but they are novice in software process.

Nowadays the role of software components in embedded system products became much more important, and the size and complexity of those software has been rapidly increasing. So, embedded people eagerly want to learn from software community's best practices.

In the paper "Why the Vasa Sank?" (IEEE Software, March-April, 2003), Professor Dick Fairly pointed out following 10 problems in process management:

- Excessive schedule pressure
- Frequently hanging needs
- Lack of technical specification
- Lack of a documented project plan
- Excessive innovation
- Secondary innovation
- Requirements creep
- Lack of scientific methods
- Ignoring obvious
- Unethical behavior

To determine which one is most critical in the case of embedded systems development, we will need to do more careful analysis of various case studies. When we perform such an analysis, we should be careful about the meaning of "embedded".

In the famous theory of software evolution dynamics, Professor M.M.Lehman (Imperial College, UK) notified that there are 2 types of software around us: S-type and E-Type. S-type is the software which we can specify precisely in advance, for example mathematical calculation subroutines, etc. namely, "S" means "Specifiable". On the other hand, E-type software is "Embedded" in real world application process. Specification changes continuously, because real world is changing continuously.

We are now dealing with software system which is embedded in some hardware. They have mixed characteristics of S- and E-type. Some components are S-type, but system as a whole is E-type. Mixing rate varies depending upon application domain: home electronics, industrial control system, nuclear power plants, etc.

Also, when we discuss about process, we should be careful about different process spaces. There is a hierarchical nesting structure of processes:

- Program execution process (inside hardware)
- System development process (outside hardware)
  - Technical aspects
  - Management aspects
- System operation process (in application process)
- System evolution process (in social process)

And there are complex relationships among various process components. In a simple sense:

- Relationship between hardware and software in program execution process space
- Relationship between system (hardware + software) and human project team in software development/maintenance process space
- Relationship between system and society (users/market) in system evolution process space

So, when we discuss about process problems and process improvement, at first we should orientate ourselves properly in which we are in. The four presentations in our group contain many interesting issues in various process spaces. We will be able to get some useful knowledge by analyzing them from proper perspective. It is the task of discussions in future workshops.

## **2 Track B. Testing and Quality Control**

### **2.1 Position Statements**

#### **2.1.1 Ikuko Suzuki**

I am now involved in various embedded system development such as mobile phone, etc. In the past, I have developed CAD/CAE systems and also DBMS.

The size of software embedded in various home appliance products has been increased drastically in recent years. And quality assurance of those software becomes more difficult, because of the shortening of development time caused by rapid change in market needs.

Generally speaking, the ratio of testing in total development activity is increasing. Testing can not assure the quality, but it is still very important central activity in QA.

I believe that high quality design is essential to plan and perform effective and complete testing. At this workshop, I would like to discuss this relationship between design and testing with Korean experts.

#### **2.1.2 Masahiko Ishikawa**

There is a strong trend of using open-source software (OSS) in embedded systems development. But, it is not easy.

To utilize OSS effectively, users need to have fairly high technical knowledge and experience. because OSS are provided to public with no customer support not likely as commercial software products.

I think that following two issues are important for the use of OSS in embedded systems development: 1) How to downsize OSS in order to fit into the limited memory space of embedded system products. 2) How to conduct quality assurance activity. It may be rather difficult because various customization are made to OSS source code.

Yet another issue is the handling of multibyte characters. It is of the weak points o some of some kind of OSS. I would like to discuss these issues at the workshop and to share common critical mind with Korean experts.

### 2.1.3 Takamasa Nara

I have been working in QA division in the company and been conducting testing and/or quality assurance activities for business application systems.

My major concerns are: software reliability, development process, testing, project management, etc.

I am interested to find out what is common and what is different between business systems and embedded systems. Then there will be a possibility of technology transfer.

## 2.2 Group Presentation

Korean participants in the group 2 on test and quality control include Sung-Deok Cha (KAIST, coordinator), Hoon-Sun Chang (KOPEC), Kee-Choon Kwon (KAERI), Jang-Soo Lee (KAERI), Han-Seong Son (KAERI), Sang-Sup Song (ADD), Jong-In Lee (KARI). We all have common interest in that our primary interest lies in formal methods and safety-critical embedded software application. For example, most of us are involved with development and quality assurance aspect of the KNICS project whose goal is to develop our own software-controlled nuclear power plant instrumentation and control system. S.S. Song, a manager at ADD, works on quality assurance of software for military applications. Jong-In Lee is a chief architect of KOMPSAT software. KOMPSAT is a satellite being developed in Korea, and needless to say, the system is safety-critical and software-intensive. When addressing quality control aspect of safety-critical software such as KNICS, safety analysis is also included because it is an essential ingredient to the quality assurance program and government approval process.

Basic approach to testing and quality assurance is fundamentally based on the notion of formal specification and verification. For example, in the KNICS project, we're specifying software requirements using notation called NuSCR which we developed based on SCR (software cost reduction) project at NRL in the U.S. While basically adopting tabular notation –called either structured decision table or AND-OR table in literature – NuSCR incorporates graphical and timed automata to visually specify requirements related to system states and transitions. From NuSCR specification, we have developed rules to automatically generate design in function block diagram (FBD) which is a programming language for programmable logic controllers (PLCs). Since design in FBD is mechanically compiled by a CASE tool such as the Space toolset developed by Siemens, FBD should be regarded as implementation language.

Our research interest lies in how we can effectively (and whenever possible automatically) perform various safety analysis including fault tree analysis and generate test cases. For example, we are also involved in research where test cases are automatically generated from Statecharts specification or NuSCR specification. We use, like other researchers, model checkers to generate test cases according to the selected criteria. Fault tree analysis based on NuSCR and FBD is also the subject of current research in progress.

Despite the use of formal methods and model checking techniques to aid testing and quality assurance tasks, we still see opportunities for further research on the following topics. Let me, on behalf of the group, briefly summarize them below and we look forward to discussing these and other technical issues with distinguished Japanese participants to the workshop.

- ❑ Testing of hardware/software interaction. When using embedded software in real-world applications, it is known that unknown and subtle errors may occur due to unanalyzed pattern of software and hardware interaction. How one can effectively test such interaction is, in our opinion, an important research task.
- ❑ Modeling of environment in which embedded software/system is to operate. Analysis of embedded software in isolation, without proper consideration of environmental factors, is at best ideal. Often, such naive testing is outright unrealistic. How one can effectively include environment into testing and quality assurance process/task is one of our research interest.
- ❑ Quality measurement of test cases. Using model checker and well-defined coverage criteria, one can generate test cases. How useful are they? Some in research community speculate that taking whatever test cases generated by a model checker (e.g., SMV) based on the criteria is not that useful in that they are not that useful in detecting potential flaws. There could be many counterexamples that may violate the property we are interested in. How do we (or Can we) influence/control model checker so that "quality test cases" are picked up?
- ❑ Code-based verification. Research on code verification and model checking is active in research community, and there are useful tools such as Bandera toolset and Verisoft. Some are used in real-world projects with success. Are they applicable to safety-critical software such as KNICS? Can they be applied "as is"?

Group on testing and quality control for embedded systems first noted that there are several types of embedded software and that different approaches are needed to perform the most effective testing and quality assurance tasks. As shown below, after lengthy discussion, the group agreed to review the following three groups separately, identify

characteristics that influence our recommended choice to testing and quality assurance, and compare differences between them. They are:

- A market-driven environment in which requirements for embedded software are relatively stable. For example, software embedded in vehicle (e.g., cruise control, ABS control, etc) are obvious candidates. Product life-time is relatively extended (especially when compared to software embedded in consumer electronics).
- A market-driven environment in which requirements change frequently. It is no surprise that typical life-time of a cellular phone nowadays are less than a few months and that companies introduce several dozens of new models every year to stay competitive. While much of basic software embedded in such cellular phones is undoubtedly reused, size of software being developed new for each model is expected to be significant. Testing of such software must be completed in a short period of time (even when measured by man-months allocated to the task), and different approaches are needed when compared to quality assurance task for software embedded in a vehicle.
- The third, yet completely different environment from the viewpoint of quality control for embedded software, is the environment in which software perform safety-critical tasks and quality assurance task is closely related to government regulations and approval process. Software controlling emergency shutdown system for nuclear power plants is an obvious example. In fact, in Korea, the Wolsung nuclear power plant has software-implemented shutdown system which is in operation, and the next generation system for nuclear power plant is in development using formal methods, and many of the participants in the group actively contribute to the project.

For three different types of embedded software, following identification of key characteristics that strongly influence desirable approaches to testing and quality assurance, the group began discussion on the recommended practices on software development practice as well as quality assurance. Results of the group discussion are summarized below:

	Market Driven - Stable Requirements	Market Driven - Frequently Changing Requirements	Regulated - Stable Requirements
<b>General Features</b>			
Safety Analysis	Safety analysis is optional but recommended	No safety analysis seems necessary	Safety analysis mandatory to obtain operating license
Controlled complexity	Desirable	May not be desirable/essential	Absolutely essential



Market/Duplication	Many(100,000+)	A Lot(1,000,000+)	Several(10+ or 1+)
<b>Development Process</b>			
Req./Design (How much extensively apply formal methods for QC/Testing?)	Formal methods may be additionally required according to the criticality (ex. ABS, Airbag Control)	UML based method is recommendable. Japan has an experience that Quality Function Deployment/Taguchi methods are applied for the development of Home Applications, Navigation Systems, etc.	Highly recommended, but to what degree should we use the formal methods?
Extreme programming?	Unlikely to be acceptable	For small projects (less than 10-20 members), XP seems desirable. But for large projects, see below	Highly unlikely to be acceptable, but some practices are routinely applied (ex. Test-first)
<b>Testing</b>			
Qualification of test cases (How many known defects can we reveal through the test cases? How can we justify the efficiency of the test?)	See Figures 1 and 2 attached below (The figure is supplied by Mr. Takamasa Nara and contains Japanese text.)		
Test process (Which test activities should we perform? How can you allocate testing budget?)	Integration/System test (including h/w and s/w integration test) is more important than unit test.	Integration/System test (including h/w and s/w integration test) is more important than unit test. Standardized test completion guideline should be customized by each company	Integration/System test (including h/w and s/w integration test) is more important than unit test. Unit test is also mandatory.
Independence of V&V/Testing	Highly recommended (In practice, our understanding is that IV&V is not common)	Independent testing is common, but independent verification is not a common practice	Required in the regulation

Inspection, or typical QA activities

The group noted that inspection is such a basic task for QA process and there are obvious no differences on when inspection processes are in place.

The group also noted that extreme programming practice is unlikely to be useful in “hardware-intensive embedded software system” because requirements do not change frequently and development in isolation is difficult

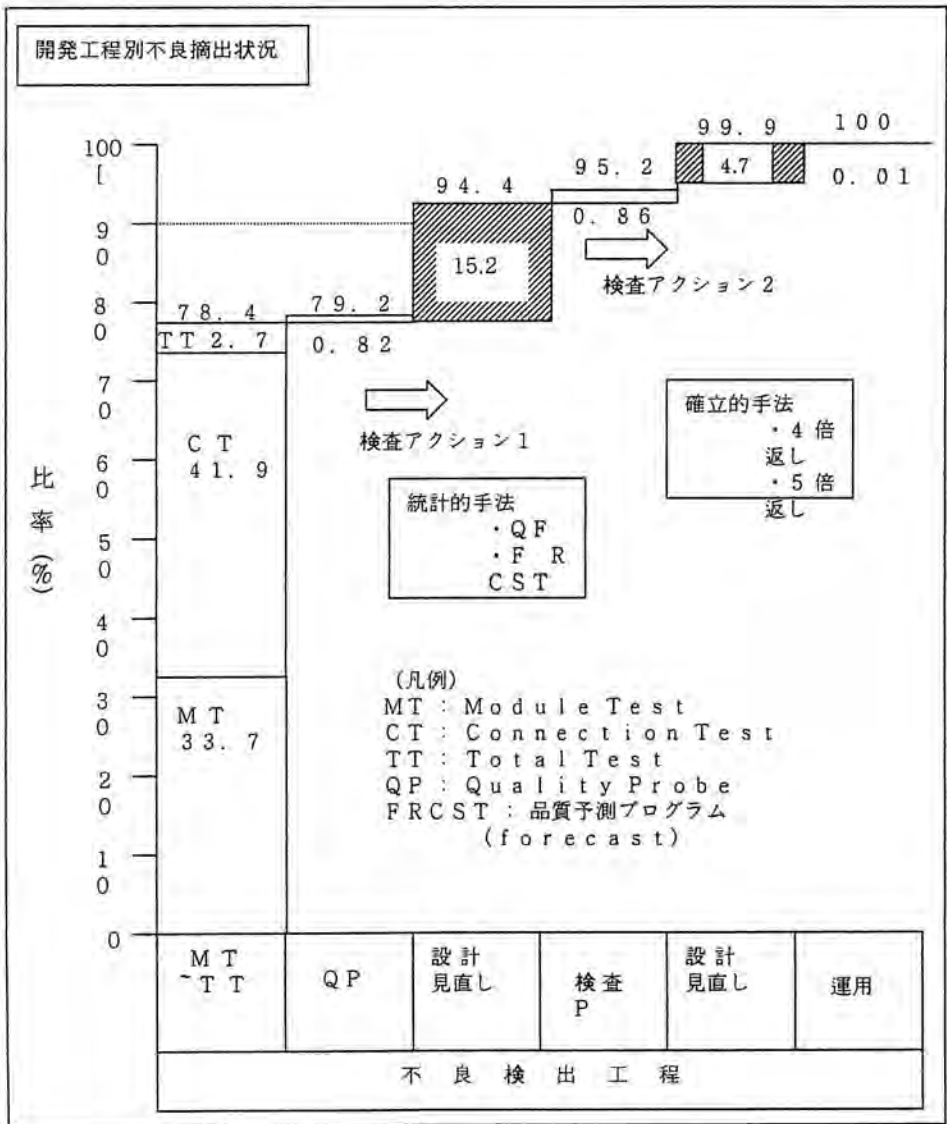


Figure 2-1

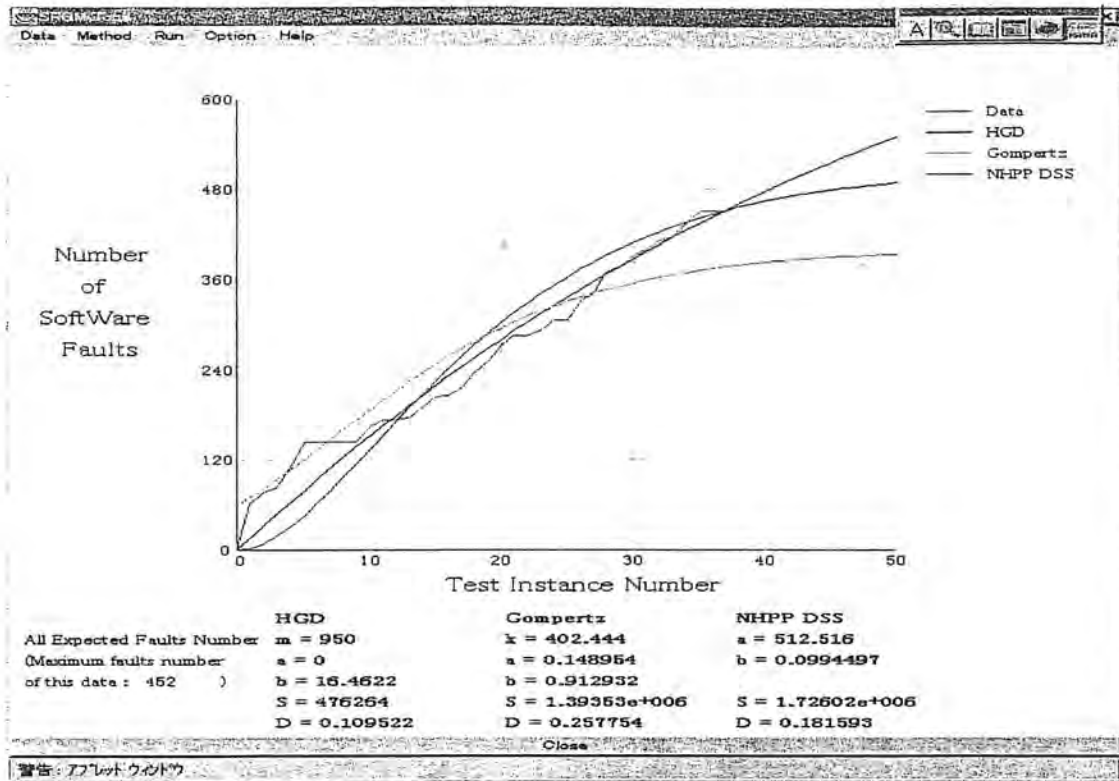


Figure 2-2

### **3 Track C. Development Methods and New Paradigms**

#### **3.1 Position Statements**

##### **3.1.1 Sungwon Kang, Architectural Issues in Pervasive Computing**

In the pervasive computing environment, computer software should be designed such that it can continuously attend human users' needs and adjust themselves to serve human needs in an efficient way. Given that software is for change, the more so in the pervasive computing environment, we are interested in answering the following questions:

What are the requirements for the architecture of such software? What architectural aspects should embedded software have in common in pervasive computing environment? Should individual software be built with a certain architecture that reflects those characteristics? Or would it be better if we have some sort of middleware that take care of the characteristics? How soon are we going to be rewarded by adopting the new approach to architecture? What is the status of the current need for it?

##### **3.1.2 Eun Sook Cho, Adopting UML for Modeling Embedded Software**

The growing dominance of software in embedded system development requires a careful look at the latest modeling technique for software specification and design. UML extension for embedded software modeling is one of those efforts. First of all, I would like to show existing modeling techniques or methodologies for embedded software development.

Second, I would like to present the needs of and the capabilities and lacks of current UML for embedded software design. Also, I would like to discuss the elements of UML extension for embedded software design.

##### **3.1.3 Soo Dong Kim, Methodology for Develop Embedded Software**

Embedded software development has a set of characteristics that are distinguished from traditional software development approach. In embedded software development, the hardware and software are co-designed. There is diversity on hardware and software for embedded system including device heterogeneity. As the market grows, the competitiveness is largely driven by the time-to-market. To shorten the time-to-market, the concept of intellectual property (IP) is introduced as a reusable asset.

Meanwhile, the component-based development (CBD) and product-line engineering (PLE) are two emerging software engineering approaches, and they are widely accepted in software industry. These two principles are known to be effective in reusing design and reducing development effort.

By considering the characteristics of embedded software, I realize that the key mechanisms of CBD and PLE can be well applied to the development of embedded software. Moreover, it should be feasible to define a commercial level methodology to develop embedded software that is based on CBD and PLE. I would like to sketch how the general principles of CBD and PLE can be utilized in embedded software development

#### **3.1.4 Sooyong Park**

Embedded software environments are getting more complicated nowadays due to increased demands on software functionalities and more interactions among systems. For example, more than 80% of automobile control unit's functionalities are done by software and its role is increasing. Furthermore, that software are interacting with others such as maintenance management system and networking system. That leads more expected or unexpected environments and functional changes during run time. To deal with these changes, software needs to be self adaptable during run time.

The self adaptable software evaluates its own behavior and changes its behavior when the evaluation indicates that it is not accomplish what the software intended to do, or when better functionality or performance is possible.

Development of self adaptable software raises a lot of research issues including

- How can we get information from running software?
- How can we make a judgment what went wrong?
- How can we adapt?
- How can we verify the result of adaptation?

Through this workshop, I would like to share current trends in self adaptable software research and discuss how we can accomplish that goal by utilizing or extending our research activities. For example, we can discuss extension of software product line approach, software agent technology, or adopting formal method techniques. Also, I

would like to discuss proper application domain that we can put our research effort to prove our concepts.

### **3.1.5 Takao Futagami**

There are many development methods for software development which has good engineering background. Some of them had applied to embedded software development so far. Each development method has specific point and can be applied to specific domain. UML and OO is one of them which some people expects to be dominant in near future.

I like to discuss how to exploit UML concept and idea in small scale embedded software development. I am glad to show how we are incubating embedded engineers with UML idea in Japan. This was presented in Embedded Software Symposium in Oct 2003.

**\*\* Report of UML education for embedded engineers \*\***

Large scale UML driven embedded application development education program was completed in the year 2002 and 2003. More than 100 trainees received education and learned how to use UML and how to apply it to implementation. This experience introducing us some interesting perspective for education and result through modeling contest.

### **3.1.6 Masao Ito**

Describing "time" in real-time system design, and some possibilities of tool support. There are many ways to describe the time factor in real-time system design (i.e. temporal logic, model checking). But in practice, it is hard to apply those approach, because those systems are running on a special environment in term of the resource constraints.

We present a way of simple description of time to handle this situation, and propose a tool implementation in order to bridge the gap between theory and practice.

### **3.1.7 Seiji Tokimatsu**

In developing prototype software from a formal specification, we have to get an executable code that matches the specification. We choose Erlang for writing the executable code. Erlang has several effective features for an embedded system. For

example, real-time, fault tolerant and so on. Although it is a functional language, practical.

We research developing an embedded system using the functional and practical language Erlang. In this workshop, we present a code generator from VDM++ to Erlang.

### **3.1.8 Keijiro Araki**

We intend to propose a systematic and effective way to develop embedded software systems based on formal methods. It will support to model and describe domains concerning embedded system development. Especially, we hope to apply formal methods in describe domain models, propose architectures for embedded software systems, provide effective ways to analyze and detect logical errors and feature interactions at the earlier stages in development.

We will briefly introduce several formal approaches and tools, and discuss about the possibility to apply them to embedded software systems.

### **3.1.9 Noribiko Matsuda**

Our organization is developing Factory Automation Components. These components are used in factory for many years. So, quality is 1st concerning point. And additionally, considering current business condition, quick development become important matter.

To solve these problems, I am very interest in new paradigm PLSE. I am glad if I get some practice of PLSE adaptation in this workshop.

### **3.1.10 Yoshikazu Hayashi**

I want to discuss Product Line Engineering with other participants and, through the discussion, want to emphasize the importance of forming and maintaining a community. Reuse is a key in engineering productivity. For example, Omron Corp., a Japanese leading embedded systems company, found productivity of the designers became four times as much over a five year time of period by promoting reuse while quality of the final systems got better[1].

The concept of Product Line Engineering is a relatively new but prospective (See, for example [2]). It is a more realistic method for promoting reuse. Today's engineering projects are getting more and more tight-budgeted and short-termed. Without proper efforts for increasing overall productivity, engineering organizations will be left behind

and bound to disappoint their customers. That is even more true in embedded software system engineering, where, because of global competition, systems are much more complex and the pressure to not allow after-release deficiency is much higher.

As I discuss on above topics, I want to make the following point. While the workshop participants may gather primarily for knowledge exchange, the true value of workshops of this kind is to share the sense of "oneness." Knowledge by itself is powerless unless those who try to use it feel connected with their goals, beliefs, and community. Communication is no mere knowledge transfer but finding someone who shares your sense.

#### References

- [1] Sakamoto, K. Kishida, K., and Nakakoji, K., "Cultural Adaptation of the CMM: A Case Study of a Software Engineering Process Group in a Japanese Manufacturing Factory," Software Process, John Wiley & Sons, West Sussex, England, 1996, pp. 137-154
- [2] Kang, K. C., Lee, J. J., and Donohoe, P., "Feature-Oriented Product Line Engineering," IEEE Software July/August 2003 issue, pp. 58-65

#### **3.1.11 Name: Akira Fukuda**

Software development consists of many stages such as from marketing research and formal descriptions to actual software development and implementing it. Each of them has been developed individually and separately. It seems for me that there are no methods that connect these stages; there are no seamless methods. I think that proposing and developing seamless methods and methodologies is important for embedded systems

#### **3.1.12 Yoji Yamanoue**

The Web services architecture is a new paradigm being expected to play a prominent role in developing next generation distributed systems and embedded software systems. We research current cases of Web services and address dependability in the Web services architecture. In particular, we apply formal method for Web services.

#### **3.1.13 Hitoshi Yamane**

I am working as a software engineer for development of plant production lines control systems. I want to discuss Product Line Software Engineering and its adaptation to my problems.



### 3.1.14 Shigeru Hitomi

I coordinate teams for real time software engineering for Industrial Automation domain. One of my recent interests is how we can revolutionize our systems development. Product Line Software Engineering is a good candidate. I would like to seek discussing colleagues whom I can exchange views and opinions with on control software systems development. And I appreciate it if I could also discuss management issue regarding introducing a new paradigm into organizations which are soaked in old development style.

## 3.2 Group Presentation

### 3.2.1 Topic #1. Product Line Technology

- How FODA and Kobra need to support embedded software
  - FODA is mainly used for domain analysis.
  - It is suggested that combining key constructs and elements of various methods is a feasible approach, rather than using a single method. (Prof. Kang)
  - Candidate Methods
    - Kobra, FODACOM, FORM, FAST
- Model Variation Points
  - how to identify variation points
  - how to specify variation points (reqs, arch, comp.)
  - Mechanisms to implement variation points and their quality properties
    - Maintainability, Reliability
  - Performance Issue with Implementing Variation Points
  - Handling evolution of variation points
  - how to trace between variation points
  - how to model domain knowledge
    - what knowledge we need to model
- Goal-Driven Modeling in Embedded Systems is essential.
  - Separate specification from implementation.
- Followed by Feature-Oriented Analysis
- Each feature will be realized in terms of objects and functions.
- Life-cycle Models

- Is the incremental model appropriate?
- proactive vs. reactive product line
  - Pros and Cons for each approach
- Configuration Management
  - How can we handle asset evolution and application specific changes?
    - More stakeholders for PLE including hardware participants
- Product line tools that handle variation points and support reuse (patterns, frameworks, styles, components)
  - requirements
  - architectures
  - designs
  - code
  - configuration management
  - A tool supporting process enforcement/management

### 3.2.2 Topic #2. Product Line Adoption

- Technology adoption issues
  - How to transfer technology from its creators/enhancers to users (engineers)?
  - How to make engineers who are knowledgeable of the target product domain, the technology, and good system architectures?
  - How to ensure the engineers know the basics
    - OO in general, modeling skills, domain analysis, etc.
  - Adaptation of the concept to the running environment with limited resources
  - Lack of CASE tools that are aware of the PLSE process
- Suggestions
  - 2.1.1
    - Tutorial, Workshop,
    - Special Interest group,
    - Collection of PLSE-related materials, translated in the local language (with the copyright holders' permission, of course!)
    - Case study, successful show case
  - 2.1.2
    - 2.1.1 + Mentor

- 2.1.3
  - Tutorial
  - Reading
  - Group learning
- 2.1.4
  - Providing development/maintenance tools that interface the higher-level design and the actual programs
- Process preparation
  - 2.2.1 Management Process
    - Organization-wide configuration management → complex because of more stake holders
    - Core assets, products, change control, etc
    - Development plans → concurrent H/W, Core assets S/W, Product S/W
  - 2.2.2 Engineering Process
    - Technology adoption process
    - Core assets/products development process
  - 2.2.3 Marketing Process
    - Marketing division and engineering divisions hand in hand
- Suggestions
  - 2.2.1, 2.2.2 SPI (Agile? Light-weight?)
  - 2.2.3 Use a Feature Model as a communication means between engineering staffs and management/marketing
- 2.3 Organizational Management
  - 2.3.1 Business case
    - ROI
  - 2.3.2 Organizational change
    - Structure
  - 2.3.3 Senior managers' commitment for organizational support
- Suggestion
  - 2.3.1, 2.3.2, 2.3.3 Success stories of practical adoption from companies

### **3.2.3 Topic #3. Utilizing UML for Developing Embedded Software**

- Motivation for UML Profile for PLE

UML is the most common notational convention for modeling software systems. However, embedded software development has a set of characteristics that are distinguished from traditional software development approach. Also, product-line engineering has elements and constructs that are not directly supported by UML. Hence, it is necessary to define an extended set of UML-based notational convention and semantics.

□ Representing Functional View

Use case model can be used to capture the primary system functionality of SOI. For embedded systems, some functionality is provided by hardware elements, which should also be captured in the functional model. Hence, in the extended use-case model, a use-case can represent either hardware or software functionality. Consequently, we need to classify and distinguish two types of functionalities by using UML extension mechanisms.

In the extended use case model, secondary use cases that are invoked by the primary use cases through `<<include>>` and `<<extend>>` relationship. A thorough identification of the primary and secondary use cases will result in the complete identification of all the functionalities that should be provided by the target embedded system.

□ Representing Structural View

Class diagram captures the structural element of the SOI. In the extend UML profile, both hardware and software elements are modeled as objects. A set of stereotypes and tagged values can be pre-defined to ease the process to model embedded system requirements. Relationships such as association, composition, dependency and inheritance should also be applied between hardware object and software object.

With the component diagram, the unit of 'package' will be useful to model a composite container of embedded system. As a grouping unit, the container typically contains a set of related hardware objects and software objects.

Deployment diagram can be used to denote the physical characteristics of deployed elements of embedded system. A node would consist of packages, and the connection among nodes would be a bus or a hardware-level inter-process channel.

□ Representing Dynamic (Behavior) View

The dynamic view of the embedded system models the interaction among diverse elements which can be hardware or software objects. Sequence diagram can be drawn with hardware objects as well as software objects in order to show the interaction among the various objects. The UML profile should

provide ways to differentiate the interaction between software elements and the interaction between software object and hardware object. For the interaction between software elements, object-oriented message is the most common form of interaction. For the interaction between software object and hardware object, there can be several different forms of interactions. Hence, with the extended mechanisms of UML profile, we should be able to express the correct form of interaction.

□ Representing Variability, Variation Points and Variants

Product-line engineering is mainly for inter-organizational reuse, rather than intra-organizational/project reuse. Hence, it is essential to model the commonality and variability (C&V) the target domain. UML does not provide constructs to express the C&V. The UML profile should include constructs to denote the variation points and variants.

Also, the components with variability in a product-line need to be customized for each product-line member. Hence, it is essential to provide the customization mechanisms and notations.

□ Representing Extra-Functional View

For embedded systems, extra-functional requirement is as important as the functional requirement. For example, most of the embedded systems require the hard or soft real time processing. This relates to the performance requirement, more specifically response time. Hence, a design model with the UML profile should denote the extra-functional requirement.

### 3.2.4 Topic #4. Self-Adaptive/Context aware software

□ Motivation

- Embedded software environments are getting more complicated nowadays due to increased demands on software functionalities and more interactions among systems. For example, more than 80% of automobile control unit's functionalities are done by software and its role is increasing. Furthermore, that software are interacting with others such as maintenance management system and networking system. That leads more expected or unexpected environments and functional changes during run time. To deal with these changes, software needs to be adaptable during run time.

□ Definition of Self-Adaptive Software

- (Self) adaptive software evaluates and modifies its own behavior (by itself) in response to changes in its operating environment by end-user input, external hardware devices and sensor, or program instrumentation to accomplishing what the software is intended to do, or when better functionality or performance is possible.

- ❑ Development of self adaptive software raises a lot of research issues including
  - How can we get information from running software?
  - How can we make a judgment what went wrong?
  - How can we adapt?
  - How can we verify the result of adaptation?
  - Web services for embedded software
- ❑ This is not only issue of web services, it is an issue of internet and embedded systems. Traditionally, embedded systems are stand-alone systems but they are connected through internet and, furthermore, web.
- ❑ Issues
  - How to confirm consistency or trust other services on the web?
  - How to meet the timing constraints?
  - Do we need any other mark up language for web based embedded software?
  - What are the most applicable domains for web based embedded software?

### **3.2.5 Topic #5. Web Services for Embedded Software**

- ❑ Can we apply Web Services to Embedded Systems? And is it effective?
  - This is the prime issue.
- ❑ How can we apply Service Oriented Architecture (SOA) to embedded systems?
- ❑ The Web Services are unreasonable for strict real-time and consistency systems, because of using Internet and parsing XML.
- ❑ It is a very difficult trade-off
- ❑ How can we specify specifications and verify coordination between each service?
- ❑ Which measures and/or tools are useful for Web Services development?

### **3.2.6 Topic #6. Verification & Validation**

- ❑ How can we handle non functional requirements?
  - Real Time, Reliability, Performance, Safety, Security ..
- ❑ We will overcome cross compile Environments/How can we Simulate the real environments
  - Almost impossible to test without hardware porting
- ❑ How can we make systematic testing process in embedded system?

- Organizing the independent SQA Team
- How can we set up the software quality standard?
  - Finding a software quality characters & metrics
- How can we make a software checklist for formal inspection in embedded system?
- Dynamic configuration ability requires codification and run-time application of design knowledge
  - How to identify context changes
  - How to find the best method to address the context changes
  - How to seamlessly reconfigure the system
- Verification of safety/security critical systems
  - Formalization and verification of assets with built-in variation points
  - Verification of applications with instantiated variation points
  - Identification of feature interaction problems
  - Verification of applications that are dynamically reconfigured
- Agent technology/ AI technologies

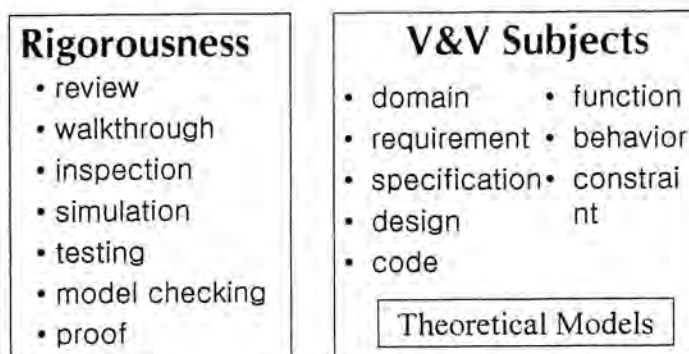


Figure 3-1. Agent Technology and AI technology

### 3.2.7 Topic #7. Characteristics of Embedded Software

- Embedded system usually consists of embedded and real-time.
  - They are different.
- Real-time requirement relates to the natural low / phenomena.
- Embedded constrain of programming is tight then assumption usually used in software development may not appropriate.
  - Constraints, Limited Resource

- Software
  - Many devices which should be controlled (I/O) exist.
  - Time (hard or soft real-time) description is essential.
  - Design of task is hard work
    - To some extent, trial and error
- Developing
  - Hard to debug
  - Small number of utility library
  - Usually, the functionality of OS is poor and instable.
  - Have to know about many I/O protocols
- To embed executable program and information construct into small memory box.
  - Let' s make virtual memory system.
- To embed execution functionality into narrow time zone
  - Human doesn' t recognize delay
- To embed all additional capability to fix hardware problem into project
  - Replace it.
- Strong preference to decrease resource rather than increasing good understanding
  - Continuous effort to separate domains
- operation to the hardware is mostly considered in designer' s mind
  - Human / actor is center of development.
- If system does not satisfy the requirement, penalty is high and rise safety problem.
- Complexity in function and in size
- Timeliness
  - Even with infinitely fast computers, embedded software would still have to deal with time because the physical processes, with which it interacts, evolve over time
- Concurrency
  - Embedded systems rarely interact with only a single physical process.
  - They must simultaneously react to stimulus from a network and from a variety of sensors, and at the same time, retain timely control over actuators.



- Devices operate in parallel.
- Liveness
  - In embedded systems, programs must not terminate or block waiting for events that will never occur.
  - In embedded computing, terminating programs are defective.
  - It needs high reliability and safety.
- Interfaces
  - Most widely applied component technology based on interfaces is procedures.
  - Procedures are finite computations that take pre-defined arguments and produce final results.
  - But, Procedures are poor match for many embedded system problems.
  - For embedded software to benefit from a component technology, that component technology will have to include dynamic properties in interface definitions.
- Heterogeneity
  - Embedded systems mix heterogeneous event handling styles.
  - They interact with events occurring irregularly in time(alarms, user commands, sensor triggers, etc.) and regularly in time(sampled sensor data and actuator control signals).
  - Facilities to interact with special purpose hardware.
- Reactivity
  - Reactive systems are those that react continuously to their environment at the speed of environment.
  - Reactive systems have real-time constraints, and are frequently safety-critical to the point that failures could result in loss of human life.
  - Reactive systems typically do not terminate.
- Low manufacturing cost
- Application and Domain Specific
- Distributed
- Operate in harsh environment
- Must meet high integrity quality attribute requirements
- Small size and weight

- Power concern

### 3.2.8 Topic #8. Future Collaboration

- Establishing a research community with participants
- Candidate Names
  - Product Line Engineering for Embedded Software Development
  - Software Engineering for Embedded Software
- Activities
  - Workshops
  - Web site for the community
  - News, Q&A, BBS, etc.
  - Exchange of Research Results and Technical Materials
  - Seeking Research Fund
  - Extended Mailing List
  - Jointly publishing books and training materials.
- Action list
  - Setting up a web site (Prof. Kang, Mr. Futagami)
    - Q&A Groups
  - Maintaining a mailing list (Prof. Kang)
  - Repository of Technical Materials (Prof. Kim)
  - Jointly publishing books and training materials.
  - Organizing 1st Workshop on Product Line Engineering
    - Can be combined with other event such as APSEC.

## DW2005

---

### デザインワークショップ DW2005

「組み込みソフトウェア仕様検証技術の最新動向 -- PROMELA/SPINを用いて」

主催:ソフトウェア技術者協会

異なるプロセスがあり、それぞれが通信することで構成されるシステムを如何に設計するかは、昔から議論されてきた話題です。但し、かつては一部の人たちの知識や経験だったものが、現在では多くのソフトウェア設計者が身に付けておくべき技術となりました。組み込みシステムや、分散システムが一般化したことが理由として挙げられます。

今回は、SPINというLTLとCSPIに従ったモデル検査が可能なツールを利用することで、日頃の我々の設計作業の中で生かし、より安全で信頼性の高いシステムを作るための手段と方法について議論したいと思います。

課題となる問題を用意して、実際に仕様を記述し、SPINで仕様のチェックするという実習を行います。できるかぎり、SPINのインストールされたPCをご持参下さい。

日時: 2005年 2月 24日 - 26日(2泊3日)土曜日昼解散

場所: ウェルハートピア熱海(厚生年金ハートピア熱海) [http://www.kjp.or.jp/hp\\_99/](http://www.kjp.or.jp/hp_99/)

定員: 15名

参加費:

学生	25000円
SEA会員	30000円
SEA賛助会員	35000円
一般	40000円

申し込み方法

以下の申し込み書に記入の上、emailで sea@sea.or.jp までお送りください。折り返し、請求書をお送りしますので、1ヶ月以内に指定の銀行まで振り込みをお願いします。

【申込書】デザインワークショップ DW2005参加申込書

氏名:

ふりがな:

種別:

SEA 会員 (No.            )

SEA シルバー会員 (No.            )

所属会社(学校)名:

部門&役職:

住所

〒            -

TEL(内線):

FAX:

E-MAIL:

---

なお、課題を事前に(お好みの方法で)記述するようにして下さい。

## DW2005のための問題(山崎利治さん作成)

## ご利用に際して

あなたの物置番号は125です。いま扉は閉まっています(施錠灯が点いています)。

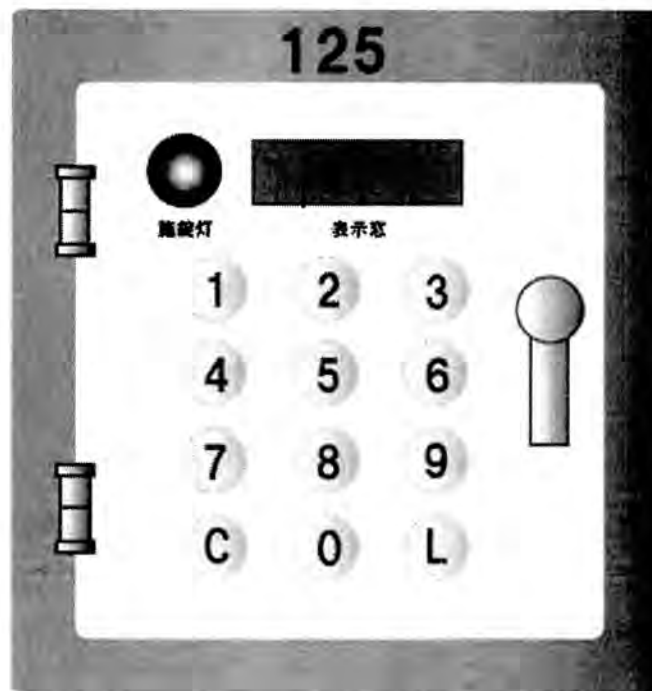


図1. 物置

物置番号の書かれた物置の扉にはパネルがあり、そこに、施錠灯、0から9までの押ボタン、C,Lの押ボタン、10進数字4桁の液晶表示窓、90度回転する取っ手が付いています。

## 扉の開けかた

- (1) 数字ボタンを1234(施錠鍵といいます)と左から順に押してください。
- (2) 押した数字を表示窓で確認してください。
- (3) それが1234であれば施錠灯が消えます(開錠)。
- (4) そこで扉の取手を時計回りに90度廻して手前に引けば扉が開きます。
- (5) 数字ボタンを押し間違えたときは、Cボタンをおして、始からやり直してください(Cボタンを押すと表示窓の数字が消えます)。

## 扉の閉めかたと施錠

- (1) 扉を閉め取手を逆時計回りに90度廻します(機械的な施錠)。
- (2) Lボタンを押します(電子的な施錠)。
- (3) ここで施錠灯が点きます(直前に扉を開けたときのあるいは、次項によって変更した鍵による施錠)。
- (4) その後は開錠手続きをしない限り、取手を廻して引いても扉は開かないはずです。

## 鍵の変えかた

施錠灯が消えているときに、つぎのようにして好みの鍵にすることができます。

- (1) 数字ボタンを4回押す。この4数字が新しい施錠鍵になりますので必ず表示窓で確認の上ご記憶ください。
- (2) ついでLボタンを押します。そこで鍵が登録され、表示窓の数字が消えます。

以上

## 論文/報告募集

### ソフトウェア・シンポジウム2005

<http://ss2005.jaist.ac.jp/>

2005年6月8日(水)～10日(金)

富山国際会議場 <http://www.ticc.co.jp/>

主催: ソフトウェア技術者協会

協賛(予定含む): 日本ソフトウェア科学会, 情報処理学会, 電子情報通信学会ソフトウェアサイエンス研究会,

情報サービス産業協会, 富山県情報産業協会, 富山県総合情報センター

後援: 富山県, 富山市

ソフトウェア・シンポジウムは、ソフトウェア技術に関わるさまざまな人々(技術者, 研究者, 教育者, 企業人, 学生, 等々)が一同に集い、発表や議論を通じて互いの経験や研究成果を共有することを目的に、毎年日本各所で開催されています。初めて1980年に開催されたこのシンポジウムも、来年で25回目(四半世紀)を迎えることとなり、開催会場を「立山あおぐ特等席」@富山市に用意することができました。

今回は「節目となる四半世紀」をテーマに、例年どおりの研究論文/経験論文/事例報告の他にショートペーパーという投稿枠も設け、ソフトウェア技術の未来を予見させるような論文や報告を期待しています。未来を予見するとは、過去の実績を踏まえた上で大胆な一歩を踏み出すことです。ダイナミックな技術変化を積極的に捕らえ、皆さんと共に明日のソフトウェア技術の姿を議論出来ればと考えております。

ここにソフトウェア・シンポジウムの原点となる第1回目のProceedings巻頭の言葉を引用し、論文募集のメッセージとさせて頂きたいと思ひます。皆さまの積極的なご応募を心よりお待ちしております。

「(前略)学問的なそれ等の成果を、現実の仕事に如何に導入するか、そしてその結果が再び学問的な場でいかに採り上げられるかという事も欠くことの出来ない巡路である。ソフト協の今回行うシンポジウムは学会の二番煎じではない、学会の縮刷版でもない、全く別の視点からの問題の議論こそ、このシンポジウムに期待する処である。会員各位の御賛同を願うこと切である。」(社)ソフトウェア産業振興協会会長 服部 正(1980年12月当時)

### 主要スケジュール

- 投稿締切: 2005年2月28日(締切は決して延長されません)
- 採否通知: 2005年3月28日
- 最終原稿締切: 2005年4月25日

### 対象分野

募集する論文/報告として以下の分野を想定していますが、ソフトウェア技術に関連していれば他の分野でも構いません。

■ 要求工学  
■ セキュリティ

■ 品質管理/標準化  
■ 開発ツール/環境

■ プログラミング技法  
■ 分析/設計/開発方法論

- |                     |                   |                 |
|---------------------|-------------------|-----------------|
| ■ Web情報システム/Webサービス | ■ ビジネスプロセス/ドメイン分析 | ■ 構成管理          |
| ■ 保守/リエンジニアリング      | ■ 再利用技術/コンポーネント   | ■ オブジェクト指向技術    |
| ■ 組み込みシステム          | ■ オープンソース開発       | ■ テスト技法         |
| ■ メトリクス             | ■ 開発プロセス/プロセス改善   | ■ 形式手法・システム検証技法 |
| ■ 教育支援技術            | ■ HW/SWコデザイン      |                 |

## 投稿要領

研究論文、経験論文、ショートペーパー、事例報告の4種類を募集します。ショートペーパーは萌芽的な研究や遂行中の実践・事例の報告を目的としたものです。下記URLの要領で作成し、カバーシートと合わせて、電子メールで [ss2005-submit@jaist.ac.jp](mailto:ss2005-submit@jaist.ac.jp) 宛てに2005年2月28日(必着)までにお送りください。投稿を確認するメールが返信されます。返信がない場合には [ss2005-inq@jaist.ac.jp](mailto:ss2005-inq@jaist.ac.jp) までお問合せください。

<http://ss2005.jaist.ac.jp/submit/>

投稿された論文/報告はプログラム委員会で査読を行い、採否を決定します。採否審査結果は3月28日までに応募者全員に電子メールで通知します。また、採録された論文/報告について、研究論文・経験論文から最優秀論文を、全登壇者から最優秀発表をそれぞれ選定し、シンポジウム最終日に表彰します。

応募論文/報告は未発表のものに限ります。他への重複投稿もご遠慮ください。論文は原則として日本語で執筆して下さい。例外として英文も受け付けますが、当日の発表は日本語に限ります。採録された論文/報告は、論文集(紙、CD-ROM)に掲載されるほか、インターネットで公開される場合があります。

## 研究論文

新規性や独創性のある理論、手法、ツールなどを提案する論文です。A4版5ページ以上10ページ以内とします。研究論文は研究の新規性や独自性を重視して審査します。当日は発表に20分、質疑討論に10分が割り当てられます。

## 経験論文

既知の理論、手法、ツールなどの適用経験を元に得られた、その利点、欠点、導入障壁などの知見を述べる論文です。ソフトウェア技術に関する団体や活動の運営経験を元に得られた知見を述べる論文も歓迎します。A4版5ページ以上10ページ以内とします。経験論文は知見の有用性を重視して審査します。当日の時間枠は研究論文と同様です。

## ショートペーパー

斬新なアイデアによる進展中の研究や、小規模な適用経験や運営経験などをA4版4ページ以内で述べる論文です。研究と経験の区別を設けて通常の論文と同様に審査しますが、加えて萌芽性や速報性が求められます。当日は発表に10分、質疑討論に10分が割り当てられます。



## 事例報告

経験論文と同じく適用経験や運営経験による知見を報告するものです。文書としてまとめられない、あるいは公開できない知見を報告いただくために、要旨(A4版1ページ)と発表スライド原稿(10枚程度)だけで審査します。スライド原稿に要点などを注記していただくと、査読者の理解が深まります。経験論文と同様に知見の有用性を重視して審査します。当日は発表に15分、質疑討論に5分が割り当てられます。

## シンポジウムスタッフ

### 実行委員長

近藤 康二(ソニー)

### 実行副委員長

田中 一夫(JFITS)

### プログラム委員長

菅原 耕一(富士フイルムソフトウェア), 藤枝 和宏(JAIST)

### プログラム副委員長

鈴木 郁子(シャープ)

### プログラム委員

荒木 啓二郎(九州大), 飯田 周作(専修大), 伊藤 昌夫(NIL), 今泉 貴史(千葉大), 臼井 義比古(環境システム研究所), 小笠原 秀人(東芝), 落水 浩一郎(JAIST), 川井 奈央(IPA), 菊池 豊(高知工科大), 蔵川 圭(NAIST), 小池 太(日立), 権藤 克彦(東工大), 阪井 誠(SRA-KTLEASE), 酒匂 寛(Designers' Den), 佐原 伸(JFITS), 新谷 勝利(IPA), 鈴木 裕信(鈴木裕信事務所), 高井 正三(富山大), 富永 和人(東京工科大), 富山 宏之(名古屋大), 豊永 正人(日立ソフト), 中野 秀男(大阪市大), 西 康晴(電通大), 野呂 昌満(南山大), 畑田 稔(富山県立大), 林 香(SRA), 東田 雅宏(PFU), 藤野 晃延(In Arcadia), 二上 貴夫(東陽テクニカ), 古木 良子(オープンテクノロジーズ), 堀田 文明(JAIST), 松下 誠(阪大), 松田 充弘(日立国際電気), 水野 修(阪大), 山本 晋一郎(愛知県立大), 鷲崎 弘宜(NII)

### ローカルアレンジメント

新森 昭宏(INTEC Web and Genome Informatics)

### 事務局

岸田 孝一(SRA-KTL), 中島 千代子(SEA)

### 問合せ先

[ss2005-inq@jaist.ac.jp](mailto:ss2005-inq@jaist.ac.jp)

## SEA FORUM (January, 2005)

## 第13回 SEA新春教育フォーラム

教育工学の現状を切る ～教育は理論と実践を尊重しているか～

## 参加者募集

SEA恒例の新春教育Forumを下記の要領で開催します。SEAの教育分科会（SIGEDU）では、過去18年間教育工学について多くの議論を繰り返して、効果的な教育を効率的に実践するための情報交換をしています。

2005年は終戦後60年ですが、日本の教員免許試験に出題される教育原理は戦前のものが大半です。比較的新しい教育工学を導入している組織であっても、バブル経済中頃の30年前ぐらいの状況にとどまっています。教育分野の古さやブームを、理論と実践の両方を重視する観点で議論します。

ふるってご参加ください。

## \*\*\*\*\* 開催要領 \*\*\*\*\*

- 日時：2005年1月28日(金) 13:30～17:00
- 場所：東京都南部労政会館（東京・品川区大崎1-11-1 ゲートシティ大崎 ウエストタワー2F）  
地図：<http://www.sea.jp/Maps/NanbuRousei.html>
- 実施内容：

パネラ	森和夫	東京農工大学:「職業としての大学教員を分析する」
	新井吾朗	職業能力開発総合大学校:「公共職業能力開発を切る」
	中原淳	メディア教育開発センター:「企業内人材開発HRを切る」
	池田真司	リコーテクノシステムズ:「教育体系開発ID手法を切る」
	甲圭太	長岡技術科学大学:「計算機援用教育CBTを切る」
コーディネータ	米島博司	NECテレネットワークス

都合により、パネラやパネラのスピーチ内容の変更があります。

## 4. スケジュール：

13:00～13:30	受付
13:30～14:45	パネラ紹介と各パネラのポジションスピーチ
14:45～15:00	コーヒーブレイク（ご歓談の時間）
15:00～16:45	全体討論
16:45～17:00	エンディング

詳細については、実行委員長・君島浩 ([mso-ced@ic.jmsdf.go.jp](mailto:mso-ced@ic.jmsdf.go.jp)) までお問い合わせください。

- 参加費：SEA正会員・賛助会員および学生：1,000円、一般 3,000円
- 申込み方法：下の申込用紙に必要事項をご記入の上、SEA事務局までE-Mailでお申込みください。なお、参加費は当日会場受付にてお支払いください（領収書を発行します）。申込受付後のキャンセルは原則としてお断りします。定員は45名(先着順)

申込み宛先：ソフトウェア技術者協会(SEA)

E-Mail: [sea@sea.or.jp](mailto:sea@sea.or.jp) Fax: 03-3356-1072 URL: <http://www.sea.jp>

## SEA-Forum (Jan. 2005) 参加申込

氏名(ふりがな)： \_\_\_\_\_ ( \_\_\_\_\_ )

会社名： \_\_\_\_\_

部門・役職： \_\_\_\_\_

住所：(〒 \_\_\_\_\_ ) \_\_\_\_\_

Tel: \_\_\_\_\_ Fax: \_\_\_\_\_

E-Mail: \_\_\_\_\_

種別(該当欄にチェック):  正会員 (No. \_\_\_\_\_)  賛助会員 (No. \_\_\_\_\_)  一般  学生



**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