
トピックス

第15回ソフトウェア工学国際会議 (ICSE15)

玉井 哲雄

1 はじめに

学会報告というのは、なかなか難しい。会議の全プログラムに出席することは、いくら勤勉な参加者でも、並行セッションが組まれているとすれば物理的に不可能だ。それにツール展示やら、並列ワークショップやら、チュートリアルやらが付随する。かりにすべてのプログラムに顔を出すことができたとしても、自分が興味を持ち理解できる分野は限られる。だから会議の全貌をつかむなどということは、一参加者としてはとてもできない相談である。

だいたい、会議全体についての評価や感想は、そういう限られた見聞のもとに自己の知識背景による偏見を加えた参加者たちが、さらにお互いに感想を交換する過程で作られていく。もちろん、意見交換する相手は全参加者のランダムサンプルというわけではなく、なんらかの立場を共有するバイアスのかかったグループである。以下の報告は、当然のことながら報告者の見聞と偏見と、話をかわした人からの影響に基づくものである。

第15回ソフトウェア工学国際会議 (The 15th International Conference on Software Engineering, 以下 ICSE, ICSE15 のように略す) は、1993 年 5 月 17 日- 21 日に米国メリーランド州ボルチモア市のストーファホテルで開催された。第1回の ICSE が 1975 年にサンフランシスコで開かれてから、すでに

18 年が経つことになる。当初は 1 年半に 1 回の開催間隔だったが、1987 年のモンタレイ会議以降年 1 回ペースとなり、開催月も最近では 5 月に定着した。

ICSE15 の大会委員長はメリーランド大学の V. Basili, プログラム委員長は R. DeMillo (パーデュー大学) と片山卓也 (北陸先端大) である。プログラム委員は、日本の二木厚吉 (北陸先端大), 磯田定宏 (NTT), 落水浩一郎 (北陸先端大) を含む 31 名である。開催地ボルチモアはメリーランド大学の地元であるだけでなく、ワシントン D C 周辺の政府関連組織、研究機関、企業が市の内外に散在し、それらに所属する研究者や実務者が会議の準備運営に当たった。たとえば、NASA ゴッダード宇宙センター、Mitre などの人たちである。

今回の論文募集や開催案内では、ICSE がソフトウェア工学分野の旗艦 (flagship) 会議であることを強調している。ソフトウェア工学も年を経るごとに細分化し、より特定のテーマのもとで個別に会議やワークショップなどが多く開かれているが、あくまでも全体の中心となるのはこの会議である、といいたいわけだ。裏を返せば、それだけ求心力を失いつつあるという危機感があるのだろう。開催地ボルチモアが元軍港で、再開発により観光地化されている港に Constellation という米国海軍最古の軍艦が停泊し、中を観光客に公開しているということも、旗艦という言葉遣いに込められてはいるのだろうが。

2 チュートリアル

最初の 2 日間はチュートリアルで、12 のプログラムがあった。そのうちの D. Parnas による “Functional Documentation for Computer Systems Engineering” と題する全日チュートリアルに出てみた。

Report on the 15th International Conference on Software Engineering.

Tetsuo Tamai, 筑波大学 経営システム科学, Graduate School of Systems Management, University of Tsukuba.

コンピュータソフトウェア, Vol.11, No.2(1994), pp.-, 1993 年月日受付.

Parnas の名前からかなり人気が集まるかと思っただが、参加者は 30 人ほどで意外に少ない。他のチュートリアルに出た人の話では、M. Deutsch の品質管理の話 (Applying Total Quality Management (TQM) to Software Systems) や P. Hausler と R. C. Linger によるクリーンルーム開発の話 (Cleanroom Software Engineering for Zero-Defect Software) に、多くの出席者があったという。文書化というテーマが地味な上に「関数的」という形容が敬遠された理由かも知れない。当人もそれを意識したのか、文書化とプログラムについてのジョークから話を始めた。あるソフトウェア会社で火災の避難訓練をやったが、誰も部屋から出て行こうとしない。そこで文書を書けといったら、あつという間になくなった。

Parnas のいう文書化とは、文書の標準化というような形式の話ではない。実際、Parnas が若い時に、所属する組織で規定されている文書標準の煩瑣な形式に反発して、標準には完全に従いながら中身がまったく無意味な文書をでっちあげ、システム企画書として提出したら通ってしまったという逸話があるそうだ。

対象とするシステムは、主としてリアルタイム系である。まず、観測変数と制御変数を区別して洗い出す。入力変数と出力変数ではなく観測変数と制御変数であるところが重要である。次にシステム (プロセス) の仕様を、動作のトレースとしてとらえる。ここで動作を関数適用と見なすところが「関数的」方法たるゆえんである。昔、Parnas は O 関数、 V 関数という概念を導入して、スタックなどの抽象データ型の仕様を鮮やかに記述してみせたが、そのスタックを例にとれば、push や pop という動作の列としてシステムの動的振る舞いを記述するわけである。そのトレースのうちで正準なものを見つける (スタックなら空スタックに 0 回以上の push を作用したもの)。その見つけ方は経験と直観によるという。一般のトレースと正準トレースとの等価性、不変的な性質などの議論・分析が可能である。

こういってしまえば、とくに目新しい考え方ではないことは明らかだろう。実際、記述力としては有限状態機械と同等だという。正準トレースが状態に対応する。だとすれば、一般の抽象データ型の仕様記述より

も記述力はむしろ小さくなっているのかも知れない。

実際、代数的方法の連中からはよく批判されるらしく、それへの対抗意識はかなりあるようだ。Parnas によれば、この方法のねらいは文書としての書きやすさと読みやすさという実践的なところにあるという。ほとんどすべての記述は表によってなされる。たとえば、原子炉の緊急停止のためのソフトウェアの文書化など、実際に使用した経験の裏打ちがあるところが強みのようだ。

チュートリアルは昼食付きで、知らない人と話をかわすよい機会になる。講師の Parnas と同じテーブルで隣り合わせに座ったので、話ができた。チュートリアルの内容に関する話もしたが、ほかに、奥さんが中国系で 10 歳のお嬢さんは英語と中国語の完全なバイリンガルであること、カナダに移ってもう 10 年以上になるが、そこが気に入って永住するつもりであること、日本からは何回か講演依頼の招待を受けたが、いつもたまたま都合が悪くまだ一度も行っていないこと、などを聞いた。

3 基調講演

5 月 19 日から本会議が始まる。プログラムは、1 つの基調講演、2 つの招待講演、42 の論文発表、5 つのパネル、2 つの最新技術サーベイ (招待講演)、3 つのワークショップ報告 (代表的な論文の発表) からなる。投稿論文数は 211 ということで、採録が 42 というのは例年の ICSE よりはかなり多めである。国別の採録数は、米国が 31 と圧倒的に多く、カナダ 6、ヨーロッパ 3、日本 1、オーストラリア 1 という内訳だった。日本の論文はわれわれのものだけで例年より寂しかったが、投稿数も少なかったようだ。

問題は参加者数である。チュートリアルの参加数が延べ 500 と主催者の予測より多く、財政的には大いに助かったそうだが、本会議の参加者の数は 600 で、期待を下回った。前はオーストラリアのメルボルンだったから、参加が 300 程度と少ないのはやむをえず、比較の対象とはならないとして、前々回のオースチンが 750 だったからかなりの落ち込みである。しかもさらに前のピッツバーグにおける第 11 回あたりまでは、参加者数は 1000 人を越えていた。たとえば、

1987年のモンタレイの会議(第9回)の時は、やはり会議報告を別のところに書いたので記憶にあるが、1200人の参加があった。今回はワシントンD.C.地域ということで、地元からの多くの参加が見込まれたのに、この結果である。ソフトウェア工学という分野全体の地盤沈下なのか、それとも冒頭に述べたようなICSEの求心力の減退なのか。

初日の基調講演は、Anita Jonesである。彼女はカーネギーメロン大学(CMU)でOSやプログラミング言語の研究をしていたが、その後バージニア大学に移り、つい最近、クリントン大統領から、Director of Defense Research Engineeringという職の指名を受けたという。つまり審査を通ればワシントン入りするというタイミングでの講演である。この基調講演の人選には、政治的な配慮もあるのかも知れない。

演題は「ソフトウェアアーキテクチャの円熟(The Maturing of Software Architectures)」である。このところCMUのM. ShawやD. Garlanなどを中心とする人々により、ソフトウェアアーキテクチャという概念が喧伝されている。実際、この2人によるArchitectures for Software Systemsと題するチュートリアルも今回のプログラムに含まれていた。これはソフトウェアシステムの大きな構造様式を指すもので、Jonesによる分類では、連合(federation)型、層(layer)型、職能(capability)型、階層(hierarchy)型、などが代表的なものだという。

このようなアーキテクチャの標準を定めること、図式化や記述方法を研究すること、参照モデルを作ることなどが、現在の課題であるとし、最後に80年代はプロセスが目立たれた時代だったが、90年代は再びプロダクトに回帰し、ソフトウェアアーキテクチャが円熟に向かう時代となるだろう、と結んだ。これはソフトウェアプロセスを研究しているかなり大きな研究グループには、挑戦的に聞こえる言辞だったろう。

スライドに有名な建築物の写真や絵をまじえるなど、よく準備された講演ではあった。しかし、建物の構造とソフトウェアのアーキテクチャの様式との間にそれほど関連性がなく、全体の主張にも強く共感を覚えたとはいえない。たとえば、アーキテクチャとデザインとは違うということを何度か強調していたが、

具体的にどう違うのか、ピンとこなかった。

講演後の質問に最初に立ったのがParnasである。まず、「円熟」とは何か、古くさくなることではないか、という軽い揶揄から始めて、今の話のどこに新しいところがあるのか、という質問というよりむしろ痛烈な批判を行った。Jonesがアーキテクチャのパターンはまだ整理されていないし、またそういう観点からの教育も行われていないと答えると、「あなたが教育していないだけのことだ」とあくまでも好戦的である。Parnasはこのような強烈な批判的質疑をすることで有名らしい。その場はBasiliが引き取って、なんとか次の質問者に移った。その後、質問に立ったのも、B. BoehmやL. Beladyなどさすがに名前がよく知られた人物ばかりだったが、Parnasの後ではみな常識的に聞こえ、余り印象に残っていない。

4 論文セッション

続いてセッション1A「形式的方法」を聞いた。最初はIBMのR. C. Lingerによるクリーンルーム・ソフトウェア開発の話(Cleanroom Software Engineering For Zero-Defect Software)である。クリーンルームとは、プログラムの開発者と検査者をまったく分離し、プログラムは自分のプログラムを一切コンピュータにかけない、テストはおろかコンパイルすらしない、一方テストは開発にはまったく関わらず、ただひたすらテストをする、という体制による開発を指す。技術的には、開発ではH. Millsによる形式的な方法(関数の機能を分解していくというもの)を用い、テストでは運用時のデータの分布を想定し、その分布に従う確率的なテストデータを用いた統計的方法を用いる。欠陥ゼロのソフトウェアを目指して、その目標に十分近づいたという話でかなり宣伝的な色彩も感じたが、同様のテーマのチュートリアルとともに聴衆の関心は高かった。

続くY. WangとD. L. Parnasによるソフトウェアモジュールの動作をトレースの書き換えによって分析する話(Simulating the Behavior of Software Modeules by Trace Rewriting)は、Parnasによるチュートリアルの続きのようなものだが、発表者Wangの英語が聞き取りにくいこともあって、技術的

な細部まではついていけなかった。最後の S. Gerhart による形式的な方法の産業への適用状況に関する報告 (Observations on Industrial Practice Using Formal Methods) は、Z, RAISE などの適用事例について面接調査を行った結果をまとめたものである。

この日の午後の 2 番目のセッション 2C 「要求工学」で、報告者は伊東暁人 (静岡大) と共著のソフトウェアプロセスの手戻りに関する実証的な研究 (Requirements and Design Change in Large-Scale Software Development: Analysis from the Viewpoint of Process Backtracking) について発表した。ユーザの要求の変化がソフトウェアプロセス上の手戻りとしてどう影響を受けるか、プロトタイプングがそれに対しどのような効果をもたらすか、といった問題について、2 つの大規模なソフトウェア開発プロジェクトを分析した結果に基づき論じたものである。発表後の質問はかなりあり、またその後、休憩時間や懇親パーティなどでも、いろいろな人から質問を受けたり感想を聞いたりした。とくに反響が強い発表であったというつもりでは決していない。反響としては平均的であろうが、このように少しでも関心を持つと積極的にコミュニケーションを求める姿勢が、とくに米国の研究者には強く、その点見習うべきであろうという感想を持った。

5 招待講演とパネル

2 日目の朝の招待講演は、モトローラ社の副社長の John Major (イギリス首相と同姓同名) による社内のソフトウェア品質改善運動の話 (Driving Software Excellence) であった。モトローラは半導体エレクトロニクス製造業だが、ソフトウェア開発の重要性が高まっている。現在ソフトウェア技術者は 350 人いるが、数年前までそのソフトウェア開発の体制は遅れており、品質は低かった。それに対しトップダウンに品質改善計画を作るとともに、TQC 的な小グループ活動によって動機付けを進め、かなりの成功を収めつつあるという内容である。話はじょうずで、スライドもよくできており、おもしろく聞いた。ただ、内容としてはよく聞くような話で、さほど目新しいものではないともいえる。

2 日目の午後のセッション 3E は、法的問題に関す

るパネルであった。座長は CACM などにもよく書いている P. Samuelson (ピッツバーグ大) の予定だったが、都合で別の人に代わった。ニューヨーク市大の人だったと思うが、名前を聞き漏らした。パネリストは S. Kahn と J. Band という弁護士で、Samuelson がパネルの設定をただけに、いずれも知的所有権に対する強い法的保護にはどちらかといえば反対の人たちである。最近出たソフトウェアの著作権に関する係争事件 (Computer Associates 事件) の判決に関する解釈と、その今後の影響が話題の中心だったが、この時点では事件そのものをよく知らず、あまりよく理解できなかったというのが正直なところである。また、弁護士の職業柄か、断定的な言い回しを避けるので、それも話をわかりにくくした。

3 日目の朝も招待講演がある。このところ恒例になっている 10 回前の ICSE の最優秀論文の表彰と、その表彰者による講演で、今年は第 5 回の ICSE から M. Weiser (現 Xerox PARC) のプログラムスライシングに関する論文が選ばれた。誰が考えたか知らないが、この 10 回前の会議の優秀論文を選ぶという着想は、なかなかおもしろい。開催時点で選考していたら違う結果になる可能性が高いが、その論文が出た後の影響力の大きさを考慮することにより、歴史的な評価が重ねられる。

Weiser の講演は、“Ubiquitous Computing and Strategies of Research” と題するものである。前半では、表彰の対象となったプログラムスライシングについて振り返った。そもそも Ph. D. の論文テーマとしてある題材を提出したが、審査で落とされたので、その代わりに急遽ひねり出したのがスライシング (切り出し) のアイデアだという。この論文の引用頻度を、この表彰に当たって自分で調べてみたとのことであるが、発表から数年間は引用回数が少ないのに、80 年代の後半から急に増えて、現在でもよく引かれるらしい。確かに、テストングや保守・リエンジニアリングの分野では、スライシングの概念を用いた研究がこの数年目につく。

Weiser はかなり率直で、スライシングの考え方を提唱し、学位論文としてある程度の成果をまとめた後、自身ではこの分野の発展にとくに大きな寄与はし

てこなかったこと、現在プログラムスライシングの定義には、Weiser が初めに与えたもののほかに後から別の人が与えたものがあり、そちらの方が主流になっていること、などを認めた。そして、自分は新しい着想を生み出すのは得意だが、それを地道に育てるのには余り向いていないとして、ここ数年提唱している現在の新しいテーマ、すなわち遍在的 (ubiquitous) コンピューティングの話に移るのである。

遍在的コンピューティングとは、いつでもどこにいてもコンピュータに自由にアクセスできる環境であり、しかも個々のコンピュータという機器を意識せず、そこら中にあるメモ用紙や電話や電卓のようなものと同じ感覚で接することのできる場である。その方向に向けて Xerox PARC で具体的に試作したものに、掌大で持ち歩き用の貼付式メモのイメージのパソコン、ノート大の通常のメモ用紙のイメージのパソコン、壁掛け式でグループで同時に見たり書いたりする黒板型の機器、などがある。通信には無線ネットワークを使う。従来のネットワーク技術と違うのは、固定されたノードのアドレスという概念が適合しないという点で、移動している個々の人を通信の受発信者として識別する必要があるところだということ。これらを使っている様子をビデオで見せたが、たとえば電話はある特定の電話器に定められた番号にかかってくるのではなく、その人の今いる場所にもっとも近い電話器にかかってくる、しかもその呼出音のパターンを人によって変えて、わかるようにしてあるところなどが、印象に残る。

遍在的コンピューティングの構想と今実験的にできているところとの差は大きいようだが、世の中で騒がれているマルチメディアだとか分散コンピューティングだとか MIT のメディアラボでやっていることとかとは、発想からして違うんだということを強調していた。

6 ICSE の今後

今後、ICSE はどうなるであろうか。初めに述べたように、ソフトウェア工学分野の学会は細分化が進んでいる。テーマでいえば保守、プロセス、テスト、などが独立の会議を持っている。

70 年代のソフトウェア工学の指導的な人物としては、E. Dijkstra, C. A. R. Hoare, D. Gries などがあげられるが、現在このような名前がソフトウェア工学と結びつけられることは少ないし、まして ICSE で顔を見ることもない。80 年代以降のソフトウェア工学は、大規模なシステムに対する管理的側面を重視したものとなっている。この流れへの反省という意味もあって、今年 (1993) 12 月に ACM SIGSOFT 主催の「ソフトウェア工学の基礎 (Foundations of Software Engineering)」というシンポジウムの第 1 回目が開かれる。報告者もそのプログラム委員になっているが、ICSE よりは理論に主眼をおいた会議で、Hoare を基調講演に招いており、技術論文も厳選してかなりレベルの高い会議になることが期待される。このシンポジウムは IEEE 主催の ICSE に対抗して作られたというより、補完的役割を果たすべきものだが、ソフトウェア工学全体という視点に立つと、やはり分散化を助長する傾向に沿ったものであることは否めない。

地域的な分散化も進んでいる。ヨーロッパのソフトウェア工学会議 (ESEC) は、毎年開催され活発のようである。オーストラリアも単独でソフトウェア工学会議を開催しているが、最近送られてきた今年のプログラムを見ると、なかなか充実しているように見える。アジアでも、1992 年から日本と韓国による合同のソフトウェア工学会議 (Joint Conference on Software Engineering) が始められ (日本側の主催は情報処理学会ソフトウェア工学研究会)、今年 (1993 年) も 11 月に九州で開催された。

このように見てくると、ICSE の求心力はますます弱まる可能性が感じられるかも知れない。しかし、分散化すればするほど、全体の状況をつかむための会議の存在理由が高まるともいえる。1994 年はイタリアのソレントで、また 1995 年は米国のワシントン州シアトルで、さらに 1996 年はドイツで開催されることがすでに決まっている。報告者は、シアトルの会議でツール展示委員長を H. Muller (ヴィクトリア大) と分担することになっている。主催側の一端を担うことになるからというわけではないが、やはり ICSE をより多くの人が集まる活気のある会議にすることは、十分意義のあることだろう。