

## APSEC2001 参加報告

青 木 利 晃<sup>†, ‡</sup>

APSEC (Asia-Pacific Software Engineering Conference) は、アジア太平洋地域で開催されるソフトウェア工学に関する国際会議で、年々着実にその知名度及び貢献度をあげている。その APSEC が、2001 年 12 月 4 日から 7 日までの 4 日間、中国の特別行政区であるマカオで行われ、筆者も参加した。本稿では、APSEC の内容と参加した感想について報告する。

### Report on APSEC2001

TOSHIAKI AOKI<sup>†</sup>

APSEC, Asia-Pacific Software Engineering Conference, is an annual international conference of software engineering which is held in asia-pacific area. APSEC was held in Macau SAR last year and I attended it. In this article, I report the details of APSEC and my impressions.

#### 1. はじめに

APSEC (Asia-Pacific Software Engineering Conference) は、アジア太平洋地域で開催されるソフトウェア工学に関する国際会議で、年々着実にその知名度及び貢献度をあげている。その APSEC が、2001 年 12 月 4 日から 7 日までの 4 日間、中国の特別行政区であるマカオで行われた。今回で、第 8 回目の開催である。

筆者は、今回で 2 回目の参加である。前回は、1998 年に台湾で開催された第 5 回目の会議であった。2 回の参加だけの感想であるが、APSEC は研究報告だけでなく各国のソフトウェア工学の研究者交流や様々なプロジェクトの相談の場としても重要な役割を演じているように見受けられる。以下では、筆者が APSEC2001 の内容と参加した感想について報告する。

#### 2. 会 場

マカオは、1999 年 12 月 20 日にそれまで統治してきたポルトガルから中国に返還され、「中華人民共和国マカオ特別行政区」となった。日本からの参加の場合ビザの取得は必要ない。マカオには国際空港はあるが日本からの直行便はなく、ほとんどの参加者は香港

経由で行ったようである。香港とマカオの間は、ターボジェットと呼ばれる高速船が 24 時間 15~30 分間隔で運行されており、約 1 時間で行き来することができる。マカオは、マカオ半島、タイバ島、コロアネ島の 3 つの地域から構成されており、APSEC2001 の会場となったマカオ大学はタイバ島にある。マカオ大学は、山の中腹から頂上までを効率的に利用して校舎が建てられており、その中の International Library<sup>1</sup>にある 2 つの部屋でセッションが行われた。1 つは 200~300 名ほどの収容能力を持つホールで、もう 1 つは、100 名ほどの席を持つステージ付きの部屋であった。なかなか綺麗な建物と部屋であった。大学に建物の工事は付きものであるが、APSEC 開催期間中にも建物の工事が行われており、発表が聞きにくいことがあったのが残念だった。

主なホテルから会場まではバスによる送迎があり、また、主催者に宿泊予定ホテルを前もって知らせておくと、Proceedings やプログラム等がホテルの部屋まで届けられた。手の込んだ気配りである。

#### 3. イベント

例年、APSEC では本会議と共に数件のワークショップやチュートリアルが開催されるようだが、今回はその両方とも行われなかった。内部事情は良くわからないが、個人的にはこれらのイベントは魅力的なので是非開催してほしい。

今回の APSEC の social event は 4 つあった。1

<sup>†</sup> 北陸先端科学技術大学院大学

Japan Advanced Institute of Science and Technology

<sup>‡</sup> 科学技術振興事業団 さきがけ研究 21

PRESTO, Japan Science and Technology Corporation



図1 マカオ大学 International Library



図3 国連大学



図2 会場



図5 Conference Dinner

つ目は、初日に行われたマカオ政府主催の Welcome Banquet である。Welcome Reception は、通常、簡単に行われるが、この Welcome Banquet は、そのような位置付けにもかかわらず盛大に行われた。残りの3つは、3日目に集中していた。後でプログラムを示すが、3日目は、午前中に Keynote と Technical Session が行われ、午後からは social event のみだった。マカオには国連大学 (UNU/IIST, United Nations University/International Institute of Software Technology)<sup>3</sup>があるが、その国連大学が3日目の昼食を用意してくれた。国連大学はマカオ半島に位置し、その建物は目立たないが、国連の旗が掲げられているのが目を引く。また、建物に面している道路は毎年開催される F1(Formula-1) のコースの一部になっている。国連大学で昼食をとった後、マカオ市街のツアーに出た。ツアーでの移動は2回ほどバスを利用したが、マカオの市街地には小さな路地が多く観光スポットも近接しているため、ほとんどが徒歩であった。また、参加者が多かったため、バスの待ち合わせや徒歩の移動に時間がかかり、結構疲れた。ツアー終了後、

マカオで一番目立つ Lisboa ホテルでの Conference Dinner が催された。Conference Dinner では、通常、伝統舞踊等のパフォーマンスが行われるが、今回は料理が出されただけであった。そのため、知合い同士の近況、共同研究や研究の方向性の話で盛り上がったようであったが、少し賑やかさに欠けた感じがした。

#### 4. プログラム

プログラムとセッションタイトルを以下に示す。  
12/4

- Keynote I: Model Checking Value-passing Processes
- Session 1A: Formal Methods I
- Session 1B: Tools and Environments
- Session 2A: Reactive and Real-time Embedded Systems
- Session 2B: System Modeling and Design Methodology
- Session 3A: Software Architecture and Framework



図4 参加者

- Session 3B: Requirements Engineering
- 12/5
- Keynote II: Formalizing Models and Meta-models for System Development
  - Session 4A: Formal Methods II
  - Session 4B: Software Management and Planning
  - Session 5A: Software Testing Techniques
  - Session 5B: Object-Oriented Analysis and Design
  - Session 6A: Software Maintenance
  - Session 6B: Formal Methods III
- 12/6
- Keynote III: Orchestrating Computations on the World-Wide-Web
  - Session 7A: Formal Methods IV
  - Session 7B: Software Architecture and Framework
- 12/7
- Keynote IV: Cost Analysis of Games Using Program Logic
  - Session 8A: Formal Methods V
  - Session 8B: Reengineering and Reverse En-

gineering

- Session 9A: Component-based Design Techniques/Empirical Studies
- Session 9B: Formal Methods VI

プログラムを見て、まず、気づくことは、Formal MethodsがI~VIまで、6つもセッションが組まれていることである。たしかに、全体的に形式的な話題は多かった。しかしながら、個々のFormal Methodsのセッションの内容を見ると、ハードウェア記述言語、コンポーネント、Web技術に分類できるものが全部、Formal Methodsのセッションに含まれている。筆者は、主に、Formal Methodsのセッションに出席していたが、内容の近いものを集めたセッションもあったが、ほとんどのセッションはばらばらの内容であった。そのため、研究動向が掴みづかった。プログラム構成は、採択された論文から推測されるソフトウェア工学の研究動向に沿ったものにすべきであり、そのためには、一般的な分類は避けるべきだと思う。

次に、今回のAPSECでは開催期間中の最初のセッションにKeynoteがあった。Keynoteには多くの出席者があり、それに続くセッションにも好影響を与えていたように思う。また、朝一番にKeynoteを聞くと、頭が活性化され、他のセッションが聞きやすかつ

たように感じた。

今回のプログラムにはパネル討論が無いことも気づく。良く言えば王道を行ったプログラムではあるが、悪く言えば、単調なプログラムである。正直なところ、Keynote と論文のプレゼンテーションのみを聞いていたので、最後のほうになると聞いているのが少し辛くなった。議論中心のパネル討論は白熱して Conference が盛り上がるので、是非、プログラムに組み込んでいただきたい。

## 5. トピックス

### 5.1 Keynote

Keynote は 4 件あったが、いずれも朝一番のにもかかわらず大勢の出席者があり、盛況であった。

1 つ目の Model Checking Value-passing Processes は、モデル検査の基礎技術に関する講演であった。通常のモデル検査手法では並行動作する有限状態モデルの集合の性質を網羅的に検査するが、この講演では値の受渡しを含む有限状態モデル(プロセス)の集合の検査を行う手法について述べられた。2 つ目の Formalizing Models and Meta-models for System Development では、異なる view で記述されたシステムの一貫性を保証する手法について述べられた。3 つ目の Orchestrating Computations on the World-Wide-Web では、インターネット上に分散しているサービス資源を簡単に結合するための手法について述べられた。4 つ目の Cost Analysis of Games, Using Program Logic では、Markov Decision Process を  $\mu$  計算で表現する手法について述べられた。この手法が Cost Analysis of Games とどのように関係するのかは、残念ながら良くわからなかった。

この中で、筆者が興味を持ったのは 2 つ目の講演である。筆者も以前から、複数の側面から記述したシステムの一貫性を保証するための手法について考えている。UML で開発対象システムの記述を複数の側面で記述する際、それらの複数の側面の記述は完全に独立ではなく、重なる部分がある。その重なった部分は、それぞれの側面からの記述の間で矛盾したことを書いている可能性があり、大規模システムになる程その可能性は大きい。彼らの手法では、それぞれの view と関連づけられる meta-view を導入し、この問題を解決することを試みている。meta-view と view は、view の記述の仕方と実際に記述した view の関係であり、UML の meta-model と個々の対象システムの記述のようなものである。そして、このような meta-view 上で一貫性の記述を行う。さらに、彼らは、このよう

道具立てを形式的に定義し、一貫性の検証を計算機により自動的に行うことを目指している。筆者も同様な手法が有効だと感じており、同様な研究を行っている。上記したように、複数の側面からの記述の間には重なる部分がある。UML にはダイアグラム間の一貫性について定義されておらず、また、一貫性の定義はシステム開発毎に異なる。よって、このようなメタモデリングにより一貫性を柔軟に定義して、自動的に検査する手法は現実的かつ有効であるものと思われる。

### 5.2 Technical Session

APSEC2002 の会議録によると、今回は、28 カ国から 145 件の投稿があり、うち、44 件を regular paper として、12 件を short paper として採録したようである。倍率は約 2.5 倍である。採録された論文の傾向としては、形式的手法が多く、また、その中でもモデル検査法 (Model checking) に関するものが多かった。

近年、モデル検査法等の軽量フォーマルメソッドが注目されており、様々な自動検証ツールが使用されはじめている。そのため、軽量フォーマルメソッドの改善や適用に関する話題が特に多かった。その 1 つである、The Steam Boiler in a Unified Theory of Z and CSP について以下で紹介する。

Z 記法はデータの観点からシステムの仕様を記述するのに適しているが、並行性などの振る舞いに関する仕様には適していない。一方、CSP のようなプロセス代数は、このような並行性に関する仕様を記述するのに適している。そこで、著者らは Z と CSP を組み合わせた Circus という言語を提案している。この論文では、この Circus を用いてソフトウェア工学の代表的な問題として知られている steam-boiler 問題を記述し、検証を行っている。Z 記法で記述した仕様を検証するためシステムとしては Z/EVES という定理証明システムが知られている。また、CSP により記述された仕様は FDR というモデル検査器により自動検証することができる。この論文では、Circus の CSP に対応する部分を FDR で、Z に対応する部分を Z/EVES で検証することを提案している。実際の検証は、現在、行っている最中のようなのである。

この論文に書かれているように、定理証明システムとモデル検査手法を併用したハイブリッド手法はこれから有望であるものと思われる。定理証明システムは強力な検証システムであるが、検証の際、人間との対話的な推論を行う。このため、検証のコストが非常に高くなる。一方、モデル検査法では、自動的に検証を行うためコストは低い。検証対象が限定されてしまう。そこで、自動的に検証できるところはモデル検査

器で、そうでないところは定理証明システムで検証を行うというアプローチは自然であり、かつ、現実的である。

また、発表者同士で議論が盛り上がったセッションもあった。そのセッションには以下の2つの論文に関する発表が含まれていた。

- Operational Semantics for Verilog
- Deriving Operational Semantics from Denotational Semantics for Verilog

どちらの発表もハードウェア記述言語 Verilog の操作的意味論に関するものであった。前者の論文は、Verilog のほとんどすべての概念を含んだ言語に操作的意味論をつけるもので、後者の論文は、Verilog の一部を対象として表示的意味論と操作的意味論の同値性の証明を目的としたものである。仕様の形式化を行う時には良くあることだが、曖昧な部分を独自の解釈で厳密に定義しなければならない。具体的な議論の内容は忘れてしまったが、これらの著者は、ハードウェアに置いた独自の解釈が異なっており、そのため議論が白熱していたように記憶している。どちらの研究も、彼らが主張する操作的意味論を用いて Verilog 記述のシミュレータの作成や事例への適用を行ってはいなかった。実際にそれらが行われていると、もっと具体的でおもしろい議論になったのかもしれない。

## 6. おわりに

今回の APSEC は、国連大学が関わっていることもあるかもしれないが、形式的な話題が多かった。筆者は形式的な話題が好きなので十分に楽しめた。また、日本では、最近、ソフトウェアの誤りによる社会システムへの影響や製品のリコール等の話題が絶えない。このようなソフトウェア開発の現状に対して、様々な投げかけをした学会になったのではなからうか。

次回は 2002 年 12 月 4~6 まで、オーストラリアゴールドコーストにおいて APSEC2002 が予定されている。論文の投稿締め切りは 2002 年 7 月 1 日である。詳細は<sup>1)</sup>に示されているので、興味のある方は参照していただきたい。

## 参 考 文 献

- 1) 9th Asia-Pacific Software Engineering Conference Home Page:  
<http://www.apsec2002.acs.org.au/>
- 2) 8th Asia-Pacific Software Engineering Conference Home Page:  
<http://www.iist.unu.edu/apsec2001/>