

第5回アジア太平洋ソフトウェア工学国際会議 (APSEC'98) 報告

Wu Wenxin[†]

白銀 純子[‡]

[†] 東京工業大学 情報理工学研究科 計算工学専攻

〒152-8552 東京都目黒区大岡山 2-12-1

E-mail : wenxin@cs.titech.ac.jp

[‡] 早稲田大学 理工学研究科 情報科学専攻

〒169-8555 東京都新宿区大久保 3-4-1

E-mail : junko@fuka.info.waseda.ac.jp

あらまし 第5回アジア・太平洋ソフトウェア工学国際会議 (APSEC'98) は1998年12月1日から12月4日に渡って台湾の台北で開催された。本稿では、その会議の概略を述べる。

キーワード APSEC'98, 報告書

Report on the Fifth Asia-Pacific Software Engineering Conference(APSEC'98)

Wenxin Wu[†]

Junko Shirokane[‡]

[†] Department of Computer Science, Tokyo Institute of Technology

2-12-1 Oookayama Meguro-ku, Tokyo 152-8552, Japan

E-mail : wenxin@cs.titech.ac.jp

[‡] Department of Information & Computer Science, Waseda University

3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

E-mail : junko@fuka.info.waseda.ac.jp

Abstract The Fifth Asia-Pacific Software Engineering Conference was held in Taipei, Taiwan, from December 1th through 4th, 1998. In this article, we briefly summarize this conference.

Key words APSEC'98, Report

1 はじめに

第5回アジア・太平洋ソフトウェア工学国際会議 (Asia Pacific Software Engineering Conference, 略称 APSEC) は1998年12月1日から12月4日に渡って台湾の台北で開催された。オーガナイザは Academia Sinica と Institute for Information Industry である。台湾の Ministry of Economic Affairs などがスポンサーとなっていた。前回の APSEC は、1997年12月2日～12月5日に香港で開催された。今回は、情報処理学会ソフトウェア工学研究会主催で高松で開催される予定である。

APSEC はアジア・太平洋ソフトウェア地域を主対象とするソフトウェア工学に関する国際会議で、毎年12月に開催されている。日本、韓国、オーストラリア、香港、台湾、シンガポールが幹事国・地域となっている。参加者はアジア・太平洋地域以外にヨーロッパや北米などからも参加が数多くある。

今回の参加者は195名であった。図1から今回の参加者数は過去最高であることがわかる。参加者の内訳は図2の通りで、主な参加者は、台湾、日本、韓国がそれぞれ134名、20名、18名であった。その中では、主催の台湾からの参加者が全体の3分の2も占めていることから、本会議の現地での関心度の高さが伺える。

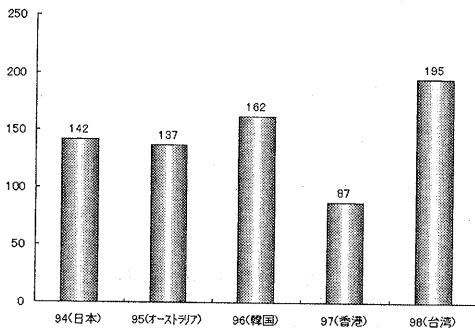


図1: 過去の APSEC の参加人数

2 会議内容

今回の APSEC は、3件の基調講演、2つのチュートリアル、44件の論文発表と5つのパネル討論か

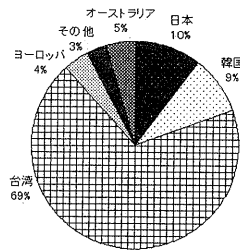


図2: 参加者の国別内訳

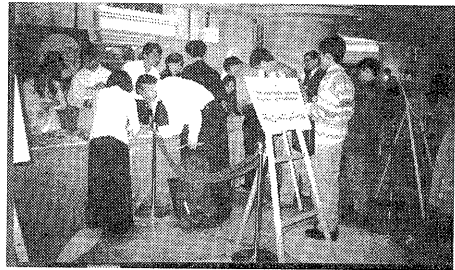


図3: Registration(APSEC'98 ホームページより)

ら構成されている。本会議は、2日～4日の3日間に、台湾の台北市の Grand Hyatt ホテルで行われた。また、チュートリアルは本会議の前日である12月1日に行われた。

2.1 基調講演

1. Dr. Raymond T.Yeh による講演

本会議の初日には、Academia Sinica の会長による Opening Speech に続いて FunSoft の社長を勤めている Dr. Raymond T.Yeh による “Rethinking Software Engineering”

の基調講演が行われた。Dr. Yeh は市場変化の速さ及びビジネスマンと開発者の間のギャップが、ソフトウェア開発が直面している中でもっとも厳しい問題の1つであり、ソフトウェア開発の新たなチャレンジは、ビジネスプロセスの変化への適応と生産時間の徹底的な削減であることを指摘した。産業界での実際



図 4: オープニング

の経験に基づき、ソフトウェア工学を再考する必要がある。講演では、ソフトウェア工学の要素と考えるソフトウェア開発プロセス、開発組織のカルチャ(自己管理, チームワーク etc.), インフラ, 特に管理のインフラについての認識を話した。そのうち、開発プロセスとビジネスプロセス間のギャップの縮小を図るために、それぞれのモデリングを示した。講演の中、ソフトウェア工学のフレームワークを企業の情報技術インフラまで拡張し、“プロセス”を中心とする企業管理を行おうとする考えは興味深かった。

2. Prof. Tetsuo Tamai による講演

本会議の二日目には、東京大学の玉井先生による

“Multiple Views of Software Engineering”

の講演なされた。ソフトウェアが大規模且つ複雑になるにつれ、ソフトウェア工学を捉える視点はもはや単純なものではなくなってきた。講演者は、アカデミック vs. 産業, 理学 vs. 工学, 地域 vs. グローバルなど様々な視点からのソフトウェアの捉え方を説明し、視点の多様性への認識の必要性、異なる複数の視点でシステムを捉えることの重要性を唱えた。最後に、ソフトウェア工学において、形式手法、発展、コンポーネントウェアなど有望な分野についての提示もあった。

3. Prof. EC.Kohli による講演

本会議の最終日には、近年ソフトウェア開発

分野において、著しい発展を見せているインドの Tata Consultancy Services の Ec.Kohli 氏による

“Software Industry Development in India-Its past and future”

が講演され、インドにおける状況の紹介がなされた。

2.2 論文発表

93 件の論文応募があり、その中から 44 件の論文が採用され、発表がなされた。分野はオブジェクト指向技術, 形式手法, テストなど広範囲にわたっており、ソフトウェア工学の分野をほぼカバーしていた。論文数は主催国の台湾と日本が同じ 20 件でトップを占め、第二位は韓国の 19 件であった。発表された論文の中ではアジア・太平洋以外の国や区域からも 6 件ほどあり、世界中にまたがっていることがわかる。図 5 に国別論文の投稿数および発表数の詳細を示す。産業界から論文発表は 3 件で、例年と同程度であり、今後さらに多くなることを期待したい。

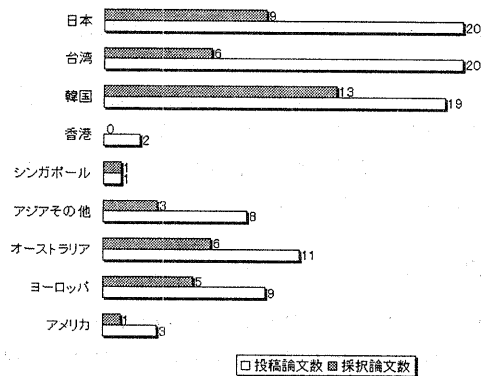


図 5: 国別論文の投稿数および発表数

発表は質疑応答を含めて 30 分であり、1 セッションが 3 件ないし 4 件の発表があった。以下に示す合計 12 セッションが 3 つ並列に開催された。

Session1A Architecture, Framework & Components I

Session1B Formal Methods

Session1C Metrics & Quality Assurance

Session2A Architecture, Framework & Components II

Session2B Software Design Methodology

Session2C Validation and Verification

Session3A UML

Session3B Software Development Environment

Session3C Validation and Verification

Session4A Object-Oriented Techniques I

Session4B Distributed and Mobile Systems

Session5A Object-Oriented Techniques II

発表は台北 Grand Hyatt ホテルの3階にある会議室で行われた。各部屋は50席程度用意されていた。発表論文の内容は形式手法とオブジェクト指向技術、中でもUML関連技術に関するものが多かった。過去の論文の内容に比べ、ソフトウェアプロセスに関する論文は少なくなってきた。また、今年からは“アーキテクチャ、フレームワーク & コンポーネント”というセッションが新たに設けられ、しかも2セッションあったことから、アーキテクチャやフレームワークに関連する研究が益々注目されてきていることがわかる。

ここで、筆者らが参加したいいくつかのセッションについて概説する。

■ Formal Methods

1B-1 “How to Reconcile Formal Specifications and Automatic Programming: The Descartes System”

J. -Y. Lucas, J. -L. Dormoy, B. Ginoux, C. Jimenez-Dominguez, and L. Pierre

プログラム自動生成のための形式記述言語 Descartes を提案した。Descartes は集合論、一階述語論理などに基づいた形式言語である。発表では、Descartes 言語とそれによる仕様記述から自動的に手続き型言語を生成する手法を木構造を扱う例題で紹介した。それに、Descartes を実際のアプリケーションに応用した経験を紹介した。

1B-2 “Integration of Software Analysis & Design Methods with Formal Description Techniques”

S. Sureerat and M. Saeki

形式的な仕様記述を行うために、形式記述言語と仕様化設計の方法論を統合する手法を提案した。その統合メカニズムは方法論によって記述された仕様書を形式記述言語による記述へ変換する規則に基づいたものである。その変換規則は方法論のメタモデルから形式仕様記述のメタモデルへの変換を規定している。例として、OMT法のメタモデルからLOTOSのメタモデルへの変換規則を示した。

1B-4 “A Refinement Calculus for the Development of Real-Time Systems”

Z. Chen, A. Cau, H. Zedan, X. Liu, and H. Yang

リアルタイムシステム開発のための、形式仕様をオブジェクトに変換する calculus を提案した。具体的には、HRT-HOOD というオブジェクト指向設計法により要求を分析し、階層的に分割する。分割された各々の要求を形式言語 TAM によって記述を行う。開発した calculus を使って、形式記述を具体的なオブジェクトへ変換する。変換されたオブジェクト記述を結合することにより、システムを構築できる。発表者はケーススタディを通してこの手法のリアルタイムシステム開発への適用を示した。

■ Validation and Verification I

2C-1 “High Performance Technique for Ultrareliable Execution of Tasks Under Both Hardware and Software Faults”

O.A. Abulnaja, S.H. Hosseini, and K. Vairavan

Dynamic Group Maximum Matching (DGMM) アルゴリズムに基づいたソフトウェアとハードウェアの両方におけるフォールトをオンラインに診断するアプローチ (IFT) を提案した。また、IFT のスケジューリングアルゴリズムをも提案した。

2C-2 “Specification-Based Class Testing with ClassBench”

L. Murray, J. McDonald, and P. Strooper
仕様に基づいたテスト手法とクラスに基づいたテスト手法を結合する方式を提案した。具体的には、形式的なオブジェクト指向仕様(ここでは、Object-Zによる仕様)から Finite State Machines(FSMs)を生成し、それで ClassBench のテストフレームワークを用いて生成された FSMs をテストグラフに変換する。そのグラフを用いてテストケースを生成し、クラスの実装をテストする。発表者は2つの例題を用いてその変換手法について詳しく説明した。

2C-3 “Mutation-Based Inter-Class Testing”

H. Yoon, B. Choi, and J. -O. Jeon

継承関係もしくは集約関係を持つクラス間のパブリックメソッドの相互関係をテストする mutation-based inter-class テスト技術を提案した。具体的には、まずクラスのパブリックメソッド間の相互作用を分類するために、Inheritance-Call グラフを生成して、テスト項目を定める。そして、状態に基づいた mutation testing criterion をクラスの挙動を表わす状態遷移図に適用することによって、テストケースを求める。

2C-4 “A Technique for Analyzing and Testing Object-Oriented Software Using Coloured Petri Nets”

H. Watanabe, H. Tokunaka, W. Wu, and M. Saeki

カラーベトリネットのシミュレーション、解析理論を用いてオブジェクト指向ソフトウェアの解析とテストを行う手法を提案した。例として UML で記述された設計仕様を CPN への変換、および変換されたネットに対する解析、テストとシミュレーションを示した。

■ Object-Orient Techniques I

4A-1 “Interpreting the Object Constraint Language”

A. Hamie, J. Howse, and S. Kent

Object Constraint Language(OCL) は UML ダイアグラムだけでは記述できない制約を記述する言語である。OCL に形式言語 Larch に

よって意味論を与える手法を提案した。その手法により、OCL の曖昧性を検証し、OCL の改良を図る。

4A-2 “Unification and Consistency Verification of Object-Oriented Analysis Models”

T. Aoki and T. Katayama

OMT のようなオブジェクト指向方法論に従った設計を行う際に使われるモデル: オブジェクトモデル、動的モデル及び機能モデルを統合するメカニズムを提案し、それらの間の一貫性を検証する手法を提案した。

4A-3 “Change Impact Analysis for a Class Hierarchy”

Y. K. Jang, H. S. Chae, Y. R. Kwon, and D. H. Bae

テストしたプログラムに変更があった場合の再テストを効率的に行うために、クラスの継承における変更が及ぼす影響を解析するメカニズムを提案した。このメカニズムは “Class firewall method” に基づいたものである。

■ Object-Orient Techniques II

5A-1 “The CACTUS Object Server: Design and Implementation Issues”

I.-H. Meng, W.-P. Yang, S.-M. Lin, M.-A. Chen, S.-K. Chen and J.-C. Dai

CACTUS という、分散処理能力を持ったオブジェクト・ストレージ・システムについての提案を行なった。CACTUS がインストールされたワークステーションは、ネットワーク上で対等な関係となる。また CACTUS は、オブジェクトのローカル処理とリモート処理の両方を扱うことができる。

5A-2 “Constructing a Toolset for Software Maintenance with OOAG”

C.-H. Hu, J.-T. Yang, F.-J. Wang and W. C. Chu

OOAG (object-oriented attribute grammar) という、保守のためのツールセット構築に有用なモデルを提案した。OOAG は、model-view-shape (MVS) アプリケーションフレームワークと、AG++ から成り立っている。OOAG

は、OO と AG の双方の利点を保つことができる。

5A-3 “A Difference-Based Version Model for OODBMS”

K.-F. Jea, H.-B. Feng, Y.-R. Yau, S.-K. Chen and J.-C. Dai

アプリケーション内の、新しいオブジェクトは、既存のものとは大部分が似通っているため、通常、既存のものから作成される。この論文では、オブジェクト指向データベースシステムのための、差分に基づく version モデルを提案した。オブジェクトは、その親からの差分のみが蓄積される。

5A-4 “Method of User-Customizable GUI Generation and Its Evaluation”

J. Shirogane and Y. Fukazawa

複雑な GUI アプリケーション開発の効率化のために、開発手法を提案した。ダイアログにアプリケーションの制御の流れを記述し、GUI についての情報を持たせて、アプリケーションの骨組みを自動生成する。また、生成されたアプリケーションの外観は、既存のツールを用いて修正を行う。

2.3 パネル

パネル討論は、本会議中の毎日の最終のセッションに行われ、初日と二日目は2つパネルが並列に行う形であったが、最終日はパネルと論文発表が並列に行われた。以下に示す合計5つのパネルが行われ、その内容は充実していた。

1. Session 1A How can a small software company survive in Asia

パネルチェア: C.N.Liu(Century Development Corp., Taiwan)

2. Session 1B How can we conduct research in software engineering

パネルチェア: Franck Xia(University of Macao, Macao)

3. Session 2A How can we solve Y2K problem in time

パネルチェア: Karl Reed(La Trobe University, Australia)

4. Session 2B Challenge of Human Resource Shortage

パネルチェア: Doo-Hwan Bae (KAIST, Korea)

5. Session 3A Object-Oriented Computing Architecture

パネルチェア: Mikiko Aoyama(Niigata Institute of Technology and Fujitsu Limited, Japan)

以下、最終日のパネル討論について報告する。このパネルは分散オブジェクト指向アーキテクチャをテーマに活発に討論が行われた。

● パネルチェア: 青山 幹雄 (新潟工科大学)

● パネリスト:

Robert Lai (International Software Process Constellation, U.S.A.)

Melfyn Lloyd (The University of Queensland, Australia)

Ren-Dar Yang (Institute for Information Industry, Taiwan)

各パネリストの主張は以下の通りである。

青山: 分散オブジェクト環境、ソフトウェアアーキテクチャの現状のオーバービューを行い、SoftwareCALS などにおける分散オブジェクト指向開発実証実験の経験を述べた。

Lai: アーキテクチャにおけるスコープの重要性、プロセスガイドラインなどを主張した。

Lloyd: コンポーネントとそのインターオペラビリティの問題を述べた。

Yang: 1996年から5年計画で始めているコンポーネントベースのフレームワーク開発プロジェクトでの経験を述べた。

分散オブジェクト指向アーキテクチャがテーマであったが、コンポーネントやフレームワークの話題が中心となった。アプリケーションを開発する上で基盤となる分散オブジェクト環境だけではなく、フレームワークやコンポーネントが重要になっていることと、アジア諸国でもこのような先進的技術への取り組みが行われていることが明らかとなった。

2.4 チュートリアル

本会議の前日に、以下の2つのチュートリアル (Half day) が行われた。

- “Component Based Business Application Development”, Deming Cha, SAP
- “Java in Enterprise”, Edward.Joe, Sun Micro systems Taiwan Limited

2つのチュートリアル合計70人近くが参加していたという盛況であった。

3 その他のイベント

Banquet が本会議二日目に行われた。台湾の民族踊りなども披露され、好評であった。また、会議に平行して、現地のローカルイベントとして以下のようなものがあった。

- Information Month
- Chang Dai-chien and Pablo Picasso: Joint exhibition of works by the modern masters of East and West

4 雑感

1. セッションの分け方

本会議のセッションのタイトルと発表論文との内容が一致していなものもいくつかみられた。セッションタイトルでどのセッションの聴講をするかを定める参加者も多いと思われるので、論文の内容も一見しただけでわかるように、アブストラクトや査読者の推薦文などをプログラムにつけるなどの工夫が考えられる。

2. 参加者

APSEC'98 の参加者数は過去最高の195名にも達した。しかし、台湾を除くほかのアジア国は61名しかいなかったのは、経済不況が原因であると思われる。参加者のうち、特に大学院の学生と思われる若い人が多かったように思った。

3. 主催側の意気込み

台湾の空港についたとき、まずは APSEC'98

の大きな旗が目に入り、本会議の開催地での関心の高さと主催側の意気込みを感じさせられた。

また、会場が非常に豪華であり、コーヒープレークではケーキやフルーツも用意されていた。食事はバイキングスタイルで、種類が豊富で味も良かった。

そのほか、すべての会場に3~5名の学生スタッフが配置され、レストランでも常に3,4名以上の会議スタッフがいて、トラブルをすぐ解決してくれたスタッフの方々の熱心さに感動した。

4. 台湾の総選挙

台湾の総選挙の投票は APSEC の次の日に行われたため、我々は遅く台湾人の政治への情熱を知った。まず、町中至るところに宣伝用の旗が重なって挙げられている。夜市場では、選挙後援会の人たちが、支持する候補者を宣伝するチラシや雑誌のほかに、宣伝用の CD-ROM も配っていた。投票の前日であった12月4日の夜には、遅くまで応援しに集まった人たちが賑わい、人々の興奮した大きな声以外、ラッパー、爆竹、花火の音も耳に響いていた。その風景は本当に驚くものだった。さすが、80%という高い投票率を誇る台湾だと思った。

5 おわりに

本稿では、第5回アジア太平洋ソフトウェア工学国際会議の内容を概説した。会議に関する詳細な情報は次の Web Page から入手可能である。

<http://www.selab.org.tw/APSEC98>

次回は、日本が主催国となり、高松で12月7-10に開かれる。次回の会議はたくさんの論文投稿や参加を期待し、さらに盛況で充実したものになることを期待したい。

謝辞

この報告書をまとめるにあたり、多くの御指導を頂いた青山幹雄教授(新潟工科大学)、深澤良彰教授(早稲田大学)、佐伯元司助教授(東京工業大学)に深く感謝致します。