

# 第13回アジア太平洋ソフトウェア工学国際会議 (APSEC2006) 参加報告

井垣 宏<sup>†</sup>, 横森 励士<sup>†</sup>, 川口 真司<sup>††</sup>, 花川 典子<sup>‡</sup>

<sup>†</sup> 南山大学 <sup>††</sup> 奈良先端科学技術大学院大学 <sup>‡</sup> 阪南大学

2006年12月にインドのバンガロールにて開催された第13回ソフトウェア工学の国際会議(13th Asia Pacific Software Engineering Conference<sup>1)</sup>:APSEC2006)に参加したので, 取り上げられた主な内容を紹介する. 会議の傾向として, オブジェクト指向/サービス指向のソフトウェアアーキテクチャに関する取り組み, および実証的ソフトウェア工学, モデル駆動開発手法に関する取り組みが多く見られ, 両分野への取り組みの活発化を伺えた. 会議には約200名の参加があり, アジア太平洋地域の研究者がソフトウェア工学の最先端の取り組みについて議論し交流する良い機会となった.

## Report on the 13th Asia Pacific Software Engineering Conference (APSEC2006)

Hiroshi IGAKI<sup>†</sup> Reishi YOKOMORI<sup>†</sup> Shinji KAWAGUCHI<sup>††</sup> Noriko HANAKAWA<sup>‡</sup>

<sup>†</sup> Nanzan University <sup>††</sup> Nara Institute of Science and Technology <sup>‡</sup> Hannan University

This paper reports major topics of the 13th Asia Pacific Software Engineering Conference at Dec. 2006 in India. There were many presentations on modeling, architecture and testing. Moreover, we saw the wide spread of model based application development(including MDD) and empirical software engineering.

### 1 はじめに

本稿では, 2006年12月にインドのバンガロールにて開催された第13回アジア太平洋ソフトウェア工学国際会議(13th Asia Pacific Software Engineering Conference<sup>1)</sup>:APSEC2006)において取り上げられた話題を紹介する. 同会議に参加した各メンバそれぞれの立場からの紹介を通じて, 執筆時点における最先端のソフトウェア工学研究のアジア太平洋地域における傾向の一部を概観する.

APSECは, アジア太平洋地域におけるソフトウェア工学の分野を扱う国際会議として広く認知されている. 今回のAPSEC2006は, IEEE Computer Society 他の主催により, インドのバンガロールのRoyal Orchid Hotelにて12月6日から9日まで(うち本会議は6日から8日まで)の日程で開催された. 参加者数は総数で約220人程度であり, ソフトウェア工学の最近のトレンド・取り組みについて積極的に議論を行う良い機会となった.

本稿では以降において, APSEC2006に参加した

筆者らにより, 本会議および併設イベント等で取り上げられた話題のうちで主要なものを紹介する.

### 2 プログラム概要

#### 2.1 本会議の構成

本会議3日間にそれぞれに3件の基調講演があり, それ以外は3並列トラックにより研究論文発表とチュートリアル, パネル討論などが行われた. 基調講演を行ったメイン会場と80名程度収容できる2会議室にて研究論文発表およびチュートリアルが行われ, パネル討論については, 昼食時の会場が屋外であったため, その横にブースを区切る形で行われた. 尚, 29カ国から230件の投稿論文があり, うち100件近くはインドからの投稿であった. インドからAPSECへの投稿は去年27件, それ以前は10件未満であり, 地元インドのバンガロールでの開催に伴い飛躍的に増加した.

- 研究論文: 17セッションに分かれ, フルペーパー59件の発表があった. 採択率は約26%であり, 各発表後には活発な質疑応答があった.



Fig. 1 ポスターセッションの様子

セッションテーマはコンポーネント指向開発やオブジェクト指向技術、要件獲得、サービス指向アーキテクチャ、保守、プロダクトライン、SE教育など幅広く設定されていた。また、コンポーネント指向は3セッション、エンピリカルソフトウェア工学は2セッション設定されており、最近の研究動向を伺うことができた。国別の採択論文の内訳としては、インド27%、日本17%、オーストラリア12%、他にアメリカ、韓国が5%で続いている。

- チュートリアル：サービス指向アーキテクチャとオープンソース開発プロセスの2テーマでのミニチュートリアルと、Vivek Bhatnagar氏によるエンタープライズアーキテクチャのテーマで実行可能性のあるソフトウェア要求とその設計方法に関する半日チュートリアルが開催された。
- パネル討論：ソフトウェア開発プロジェクトの複雑さを管理するテーマとソフトウェアプロセスでの改革のテーマにてパネル討論が行われた。
- ポスターセッション：昼食・コーヒーブレイク会場に隣接する場所で、Microsoft、Infosys等の企業におけるソフトウェア工学関連の研究の紹介がポスターによって行われた(図1)。
- 企業訪問：バンガロールのエレクトリックシティにあるInfosys Technologies社とMind Tree Consulting社を訪問する機会が与えられた。これらの企業はキャンパスと呼ばれる広大な敷地に、衛生通信を装備したTV会議シ

ステムを持つ多言語対応会議室などの充実したオフィス設備や、ミニゴルフコースやアスレチックジム等のレクリエーション施設をひとつの敷地にもつ巨大企業であり、それぞれの企業がひとつのCityを形成していた。広大さと最新設備の数々、さらにそこに働く人の数に圧倒された見学であった。



Fig. 2 バンケットの様子

## 2.2 併設イベント+ソーシャルイベント

ソーシャルイベントとして、本会議前日の夜にレセプション、本会議2日目の夜にバンケットが開催された。筆者らが到着したのは本会議前日の深夜であったためレセプションには参加していないが、バンケットは多数の参加者が交流し歓談する良い機会であった。バンケットは昼食と同様にバイキング形式の立食パーティの形式で、今回の開催に貢献した関係者への表彰も同時に行われた。バンケットの合間には、インドの舞踊学校の学生による踊りが披露された(図2)。踊りの最中に装置の故障で音楽演奏が止まるなどのハプニングはあったが、参加者がカメラやビデオで撮影している姿があちこちで見られ、好評であった。

また、本会議3日目の全プログラムの終了後の午後には、Infosysのバンガロール本社(バンガロールキャンパス)の会社見学が開催された。Infosys社は、この10年でインドで最も成長した企業の一つで、インドで一番最初にNASDAQに上場した企業として知られている。

Infosys本社は会場となるホテルから1時間ほどのところにあり、約40人ほどが参加した。本社の広大な敷地内では、48の建物に15000人の従業員



Fig. 4 パネル基調講演の様子



Fig. 3 Infosys 社内風景：ゴルフのグリーンもあり，週末に練習する人もいるようだ。

が働いている。敷地内は、本社の外のインドらしい風景とは全く異なり、文字通りアメリカの大学のキャンパスを連想させるような開放感を感じる内容であった（図 3）。

時差のある環境を生かし、自分たちのシステムを必要とする顧客にいかにかデリバリーしていくかについてノウハウと自信をもっている Infosys 社のグローバリティが感じられる会社見学であった。

他に、最終日にはバンガロールのシティツアーも開催され、インドの伝統的なパレスや、豪華な市庁舎などを見学した。

### 3 本会議の話題

#### 3.1 基調講演

基調講演は、一日目はカーネギーメロン大学の Kurt Wallau 氏、二日目に Microsoft Research India の Sriram Rajamani 氏がそれぞれコンポーネント指向開発とソフトウェアの自動化について講演を行った。最終日のパネル基調講演では TCS India の Gautam Shroff 氏、IBM の Anand Vaishampayan 氏、Infosys の Subu Goparaju 氏、Philips の Erik Kaashoek 氏らがそれぞれの立場でこれからのソフトウェア工学に関するチャレンジについて発表を行った。どの発表も実際の開発のケーススタディなどを織り交ぜた非常に説得力のある問題提起となっており、その後の議論も大いに盛り上がった。基調講演の様子を図 4 に示すとともに、以降では各基調講演を紹介する。

### 3.1.1 Beyond Vacuity in Component-Based Software (Engineering), Kurt Wallnau

カーネギーメロン大学の Kurt Wallau 氏よりコンポーネント指向ソフトウェア開発が抱える問題点と、それらを解決するアプローチについての講演があった。コンポーネント指向という言葉が使われはじめて長い時間がたつが、現在、その意味するところは非常に曖昧なものとなっており、発話者によってその意味がさまざまに変わりうるのが実情である。

氏が提唱している Predictable Assembly from Certifiable Components (PACC) では、コンポーネントは以下の 2 つの性質を備えていなければならないとしている。

- 堅牢性やパフォーマンスなどの品質が保証されていること
- 振るまいが予測可能であること

これらの基準を満たすコンポーネントを組みあわせることで始めて、コンポーネント指向が目標とする高性能かつ高い信頼性を持つソフトウェアを納期通りに開発することが可能になるとしている。

講演では、PACC を実現するための要素技術として、コンポーネントが備えるべき標準的なインタフェースを定義した Pin 技術や、それらのコンポーネントを組みあわせるためのアーキテクチャ記述言語 Construction and Composition Language (CCL)、それぞれのコンポーネントの性能からシステム全体のパフォーマンスを自動的に類推する Lambda-star 手法をはじめとする技術についても、それぞれ紹介と考察が行われた。

興味を持たれた方は PACC のホームページ<sup>3)</sup>も参照されたい。

### 3.1.2 Automatic property checking for software: past, present and future, Sriram Rajamani

二日目の基調講演では Microsoft Research India の Dr. Sriram Rajamani が Automatic property checking for software: past, present and future のタイトルで、ソフトウェアのモデル検査についての講演が行われた。モデル検査は Unix の utility である Lint から始まり、現在では、到達しない状

態を検知するような仕様記述言語を含むツールや可読性や拡張性、完全性、操作性などのエンジニアリングトレードオフ考慮したツールへ発展している。また、反例駆動型詳細化の技術を用いた C 言語プログラム静的解析ツール SLAM<sup>4)</sup> について具体的な例にて解説した。さらに、モデル検査に関して静的なプログラム解析ツールは同時実行のコンカレントプログラムの解析や、コンカレントエラーの検出が今後重要なポイントになるだろうと述べた。

### 3.1.3 Keynote Panel on Near Term Challenges in Software Engineering

3 日目のパネル基調講演では、Infosys の Subu Goparaju 氏、Philips の Erik Kaashoek 氏、TCS India の Gautam Shroff 氏、IBM の Anand Vaishampayan 氏らが順番にソフトウェア工学の近い将来におけるチャレンジについて講演を行った。

最初のパネリストである Infosys の Goparaju 氏は、ますます複雑になりつつあるソフトウェアシステムの開発とその管理がチャレンジであると述べた。氏によると、現在のソフトウェアサービス産業は、開発規模開発に必要な技術の複雑化、継続的な生産性増加への要求、開発チームの分散と能力/人員不足およびそれに付随する技術共有の困難さといった問題に直面している。これらの問題に対処するための提案として、講演ではエンジニアリングとメンテナンスの各プロセスにおける改善について述べられた。エンジニアリングプロセスでは、ビジネスプロセスモデリング、モデル駆動設計およびテスト、早期の validation のためのシミュレーションと自動化による技術の単純化、プロセスとプロダクトを正確に見積もるためのメトリクスの策定といった点が改善のためのチャレンジとして説明された。また、メンテナンスプロセスでは、大規模なソースコードを迅速に理解するための方法論や、コード内の変更点の把握、システムの健全さの定量的な見積もり、継続的なシステムの健全さの維持といった点が必要とされているという説明があった。

二人目のパネリストである Philips Consumer Electronics の Erik Kaashoek 氏は Consumer Electronics の観点から見たソフトウェアとそのチャレンジについての講演を行った。氏によると、Consumer Electronics の分野では、ハードウェアから

ソフトウェアに開発の軸が移行しつつあり、それに伴い開発対象となるソフトウェアの複雑度や規模が増大しつつある。しかしながら、Consumer Electronics におけるソフトウェアの開発はハードウェアの影響を受けることが多いため、ビジネスアプリケーションのようなソフトウェアのコンポーネント化が難しい。そこで、現状の、そしてこれからのチャレンジとして、分散開発のためのハードウェアのコンポーネント化、ソフトウェアコンポーネントの標準化、およびそれらの結合が考えられている。

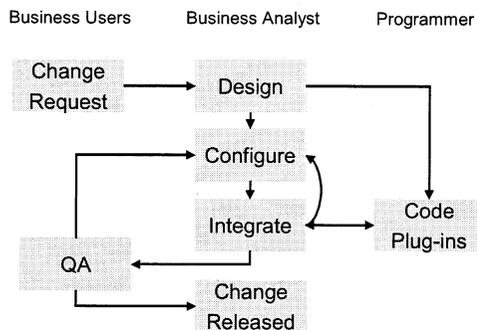


Fig. 5 Gautam Shroff 氏の主張する Dev2.0 における開発プロセス

三人目のパネリストの Tata Consultancy Services の Gautam Shroff 氏は Enterprise Software を対象とし、ビジネスのスピードはそのまま経済的なパフォーマンスに影響を与えるという MIT の報告をもとに、SOA(Service Oriented Architecture) や Web2.0 を用いた迅速な大規模ソフトウェア開発がこれからのチャレンジであると述べた。氏はそのような開発形態を Dev2.0 と呼び、図 5 に示したようなプログラマより上流のアナリストやユーザといったステークホルダーによるコンポーネント開発が求められるようになると主張している。

最後のパネリストである IBM Global Business Services の Anand Vaishampayan 氏はまず、昨今のソフトウェアアーキテクチャパラダイムは EAI から SOA におけるサービスへと移行しつつあると述べた。氏の講演では、このようなサービスを中心にしたアプリケーション開発が進むにつれ、機能/非機能両方の要求変化に対応するための impact

Table 1 研究論文の傾向

分野	件数
テスト/解析	14
開発手法/プロセス	13
実証	9
アーキテクチャ	6
検証	6
保守	4
要求工学	3
教育	3

analysis や Defect free software 実現のための品質評価がますます重要になってくることがチャレンジとして述べられていた。

### 3.2 研究発表

採択された研究論文の分野内訳を表 1 に示す。セッション別でいうと、APSEC2005 との主な違いは、Education, Empirical Software Engineering, Maintenance のセッションが新たに増えたこと、Component-Based Software Engineering が 3 倍に増えたこと、Agent-Based Software Engineering, Aspect-Oriented Software Technology が無くなったことなどがあげられる。教育、ソフトウェア開発現場における開発手法や実証的なデータの利用等、より実的なソフトウェア工学を扱う分野へと推移しつつある流れを示しているのではないかと考えられる。

この変化は、具体的な開発プロセスやモデルを利用したテスト/解析を対象とした研究や実証的なデータを利用した研究論文が増加する傾向にあるということを示している表 1 からよく分かる。

以降において筆者らが聴講したセッションとそのセッションにおける研究論文発表の幾つかを取り上げる。

#### 3.2.1 EMPIRICAL SOFTWARE ENGINEERING Session

このセッションは、実際の開発に関するデータを利用し、パターンの抽出やコンポーネントの再利用性の評価などを行う発表が行われた。ここではその中から特に “Usage Patterns of the Java Standard API” というタイトルで Homan Ma, Robert Amor and Evan Tempero らが発表した研究について紹介する。

この論文は、筆者が用意したオープンソースソフ

トウェアの集合をコーパスとして、Java 標準 API の利用パターンを調査し、得られた結果を考察したものである。結果の要約を以下にまとめる。

- 用意したコーパスでは、Java 標準 API に関して、API パッケージの約 9 割、約 50% の API クラス、約 21% の API メソッドを利用して
- java.lang パッケージが一番よく利用され、以下 java.util, java.io と続く。
- String クラスが一番よく利用されている。int 型は String クラスと同じくらい、boolean はその約 4 分の 1 ほどの利用回数がある。
- String クラスの equals メソッドが最もよく利用されている。StringBuffer.append(), String.length(), Iterator.next() と続く。
- フレームワーク中のクラス内のメソッドを普通直接呼び出したりはしない。そのため利用されないメソッドは、使われないクラスのメソッドと、フレームワーク中のクラス内のメソッドが大部分を占める。
- たくさんの API に埋もれてしまうことで、java.nio や java.\*.spi のような最近追加された API が広く認知されていない可能性がある。
- JavaAPI 自身は、API パッケージの約 96%、約 73% の API クラス、約 73% の API メソッドを利用している。利用していないメソッドはフレームワーク中のクラス内のメソッドが大部分を占める。
- インフラに相当する部分が JavaAPI に組み込まれており、肥大化の原因となっている。それらを JavaAPI から分離したほうが、開発者のためになるであろう。

### 3.2.2 SE Education Session

教育に関する本セッションは、昨年時の APSEC2005 では無かったセッションの一つであるが、このセッションのテーマは IT 企業主導で発展したインドのソフトウェア工学の特徴をよく表していると思われる。特に、“Security: Bridging the Academic-Industry Gap Using a

Case Study” のタイトルにて紹介された Microsoft India Development Center と大学が提携して行われた Security に関する教育に関する発表が興味深かった。

ソフトウェア開発におけるセキュリティの重要性の認識は産業界と大学の教育の間では大きな差がある。そこで産業界での実例を構造的に分解し、セキュリティの設計からコーディングまでの教育の方法論を示し、実際に大学教育にて実施した。その結果、学生のセキュリティに関する設計やコーディングの問題意識が向上したという報告であった。Philips Research India の Sinha 氏は質疑応答にて企業の実例による教育を絶賛し、このような教育がインド中に広がることを望んでいると述べていたことが印象的であった。

### 3.2.3 MAINTENANCE/REVERSE ENGINEERING Session

MAINTENANCE/REVERSE ENGINEERING セッションでは、ソースコードの再利用性に着目したパターン抽出の方法、リファクタリングの新しい方法の提案や、実際の開発データを利用した外注ソフトウェアの管理方法に関する発表があった。ここでは、“Genericity - a “Missing in Action” Key to Software Simplification and Reuse” というタイトルの Stan Jarzabek 氏による発表について紹介する。

プログラム中には同様の構造が何度も現れる箇所が見られる。そのような部分はソフトウェアの重要なコンセプトを表していることも多い。しかし、重複が多いくほど複雑度やコード行数の増加を招き、また再利用が困難になる。このような汎用的コンセプトの記述方法として、STL はある一定の成功を収めている。同様の手法をより設計レベルの類似性など、より広い対象に適用できないだろうか。

そのような記述を可能にする仕組みとして、本論文では汎用的メタプログラム記述言語 XVCL(XML-Based Variant Configuration Language) を提案している。XVCL では、まず汎用的テンプレートを記述し、別途テンプレートに挿入する文字列を与えることで具体的なインスタンスを生成する。テンプレートやそのパラメータには任意の言語を記述することが可能となっている。

論文では XVCL と Java 言語を組みあわせてい

くつかのプログラムを記述した事例が述べられている。J2EE アプリケーションへの適用例では、さまざまな EntityBean が共通して持つ同様の操作 (add, delete, update など) を XVCL を用いた汎用的な記述に置きかえを行っている。それらの結果、プログラム中の重複を取りのぞき、行数の削減に成功している。また汎用的に利用できるデザインを別途記述することによって、デザインの再利用を支援することを可能にしている。

### 3.3 ワークショップ

本会議に併設して開催されたワークショップは以下の3件であった。

1. 1st International Workshop on Service and Process Oriented Software Engineering
2. Workshop on Software Product Development Challenges
3. International Workshop on Software Competency Building and Education.

これらのワークショップは、本会議終了後の翌日12月9日に開催された。筆者らは日程の都合上参加できなかったが、それぞれのテーマごとに活発な議論が行われた模様である。

## 4 所感

**井垣:** 初めて訪れるインドにおける初めて参加する APSEC と、初めてが重なった今回の会議であった。インド、特にバンガロールでは IT 関連の発展が著しいと以前より聞いていたため、インドにおける APSEC もさぞや盛り上がるだろうと事前に想像していたが、実際は想像以上だった。どのセッションにおいても研究発表に関する議論が大いに盛り上がり、筆者自身も発表者として実に有意義な知見を得ることができた。また、ソーシャルイベントとして参加した Infosys への会社見学では、社員規模が数十万人という桁違いの大きさに圧倒された。一方で、欧米風の会社敷地内と、工事中であつたり余り舗装されていなかったりする敷地外との格差の激しさもインドの一つの特徴なのだろうかと感じるところがあった。今回の APSEC では、実証的ソフトウェア工学やモデル駆動開発を始めとするオブジェクト指向の概念等のより実際的な分野への意識が高まりつつあるのを感じた。

このこともインドを含むアジア地域におけるソフトウェア工学産業の昨今の盛り上がりに関連があるのではないかと考えている。

**横森:** インドでの初めての APSEC ということもあり、インドの IT に関する情熱とソフトウェア工学に対する期待を十分に感じ取れる大会であった。大会の運営は大きな混乱もなく非常に整然として行われ、大会として成功していたと思う。参加者として、関係者の尽力に感謝したい。具体的な研究発表については、コンポーネントベースのアプローチとエンピリカルソフトウェア工学を題材とした研究発表が非常に多く（自分もその中の一人であるが）、自分にとって非常に興味深いものが多かった。今回の APSEC 参加は有意義で今後の自分の研究に大きくプラスに作用するものであったと感じている。ただ、自分はインド社会の混沌さには対応できず、オーバースタッキングで非常にストレスのたまる思いをしたり、帰国直前に体調を崩して帰りの飛行機で地獄の思いをするなど、大会参加以外については残念ながらインドでいい思い出を残せなかった。

**川口:** 本会議は Asia-Pacific の名が示す通り、アジア・環太平洋圏からの参加者が多数を占めており、欧米圏からの参加者が多数を占めることが多い他の国際会議とはまた違った趣きがあった。研究発表においても、さまざまな地域からの多様な発表があり、ICSE や FSE とは異なる、ソフトウェア工学全般を扱うもう一つの会議としての APSEC の存在価値を示していたように思う。また、今回はインドでの開催ということで、会議全体に非常におおらかな雰囲気が漂っており、いたるところでインドらしい気風を垣間見ることができた。

**花川:** バンガロールは4年ぶりの訪問であった。空港を一步出たときの喧噪は相変わらずで、マンパワーに任せられた秩序に乏しい混沌とした状況はむしろ懐かしさを感じた。しかし、バンガロール市内に近づくとも明らかに4年前とは様子が異なっていた。まずは交通事情である。4年前は我々のチャータした新車の4輪駆動車がかなり目立つ存在であったが、現在ではほとんど違和感がない。さらに、道路にわがもの顔で横たわっていた牛（ヒンズー教では神様）がもう見られないこと。少々寂しい思いがしたが、バンガロールの発展の勢いが牛を街から追いやってしまったのだろう。その発展の勢

いは APSEC2006 の各セッションの議論でも感じることができた。発表が終了するとともに質問が飛び交い、さらに発表者を差し置いて、聴衆者同士で議論を展開する場面を何度も見かけた。バンガロールを始めとするインドの発展の勢いは、閉塞感のある日本社会で生活する私にとってある種の羨望と懐かしさを感じることができた。

## 5 おわりに

今回のインド、バンガロールでの APSEC2006 では、発表における実務者/研究者間の活発な議論と交流があったのは勿論のこと、企業と大学の共同研究による発表や、Empirical Engineering に代表される実際の開発データを利用した研究の増加等、ソフトウェア開発の現場においてソフトウェア工学の重要性が増しつつあることを十分に感じさせる会議であった。

今後、APSEC2007<sup>2)</sup> は 2007 年 12 月 5 日～7 日、日本・名古屋で開催される予定である。Abstract Submission が 6 月 25 日、Paper Submission が 7 月 2 日の締め切りとなっているので、本稿発表時点では研究論文の投稿まではまだ余裕がある。APSEC2006 では、インドからの投稿が 100 本以上、全体の投稿数の 46% を占めていた。同様に日本からの投稿と発表、参加が増加することを切に期待したい。

## 参考文献

- 1) Asia-Pacific Software Engineering Conference 2006(APSEC2006), <http://www.cse.iitk.ac.in/apsec06/>
- 2) Asia-Pacific Software Engineering Conference 2007(APSEC2007), <http://www.apsec-conferences.org>
- 3) Predictable Assembly from Certifiable Components (PACC), <http://www.sei.cmu.edu/pacc/>
- 4) SLAM Project, <http://research.microsoft.com/slam/>