

第 18 回アジア太平洋ソフトウェア工学 国際会議(APSEC2011) 参加報告

大森 隆行^{†1} 大山 勝徳^{†2}
林 晋平^{†3} 青山 幹雄^{†4}

本稿では、2011年12月5日から8日まで、ベトナムのホーチミン市にて開催された第18回アジア太平洋ソフトウェア工学国際会議（APSEC2011）について紹介する。

A Report on the 18th Asia-Pacific Software Engineering Conference (APSEC2011)

Takayuki Omori^{†1} Katsunori Oyama^{†2}
Shinpei Hayashi^{†3} Mikio Aoyama^{†4}

This paper gives our views on the 18th Asia-Pacific Software Engineering Conference (APSEC2011) held in Ho Chi Minh City, Vietnam on December 5-8, 2011.

1. はじめに

本稿では、2011年12月5日から8日にかけて、ベトナムのホーチミン市 Rex Hotel において開催された、第18回アジア太平洋ソフトウェア工学国際会議(APSEC2011)

^{†1} 立命館大学
Ritsumeikan University

^{†2} 日本大学
Nihon University

^{†3} 東京工業大学
Tokyo Institute of Technology

^{†4} 南山大学
Nanzan University

において取り上げられた話題を会議の様子を交えながら紹介する。本稿では、今後の APSEC をはじめとするソフトウェア工学関連の国際会議への活発な参加を促すことを目的に、アジア太平洋地域における研究動向やホットな話題を紹介する。

APSEC（アプセック）は、1994年に東京で開催されて以来、ソフトウェア工学全般を扱うアジア太平洋地域における国際会議として広く認知されている¹⁾。APSEC2011では、ベトナム外の22の国と地域から96名が集まり、ベトナムからの参加者98名と合わせて、194名の参加があった。日本からの参加者は18名であり、韓国からの27名の参加者に次いで多かった。この他の主な参加国は、中国12名、フランス7名、オーストラリア、ドイツ、台湾がそれぞれ3名であった。

本年度の開催地であるベトナムは隣国である中国やタイと同様、アジア圏の中で高い経済成長率を見せている。ホーチミン市では、高層ビルやショッピングセンターの建設ラッシュを見ることができる。町中には親日的な人が多い。1日中賑やかであり、活気に満ちあふれている。

今年のテーマは「Software Engineering and Emerging Technologies」であり、会議では、ソフトウェア工学分野に関するさまざまな取り組みについて、活発な議論が行われた。本会議では、連日、質の高い基調講演、研究発表、ポスターやデモがあった。基調講演には Barry Boehm²⁾や Leon J. Osterweil³⁾が参加している。また、本会議の前後にチュートリアルやワークショップ等が併設された。



図1 会場の Rex Hotel

本稿の構成を述べる。2章で会議の内容を概説する。3章では、各テクニカルセッション、チュートリアル、キーノートの内容を紹介する。4章では、ソーシャルイベントについて紹介する。5章では、著者らの APSEC2011 に関する所感を述べる。最後に、6章でまとめを述べる。

2. 会議の概要

2.1. 論文投稿と採択件数

論文投稿件数は 178 件で、採択件数は 53 件（採択率 29.8%）であった。

今年の APSEC では、最近のソフトウェア工学の研究において、複数の国にまたがる国際化が進展していることに合わせて、投稿・採択論文の全著者の分布を集計している。表 1 に主な国・地域の投稿・採択状況を示す。投稿した著者は 39 か国 455 名にわたり、アジア太平洋地域が 305 名（67.0%）、ヨーロッパが 101 名（22.2%）、アメリカが 18 名（4.0%）、アフリカが 31 名（6.8%）であった。アジアでは中国からの投稿がここ数年最多となっている。一方、インドなどからも投稿が増えている。例年、多数の投稿がある韓国は、参加者は最多であったが、投稿は例外的に少なかった。また、例年、ヨーロッパからの投稿が少なからずある。今年は、歴史的なつながりからか、フランスからの投稿が多かった。

表 1 主な国・地域の投稿・採択状況

国・地域	著者数	投稿件数	採択件数	採択率
オーストラリア	18	8.2	3.0	0.37
中国	94	35.5	10.8	0.31
フィンランド	12	3.8	1.0	0.26
フランス	28	9.5	4.4	0.46
インド	20	10.7	1.7	0.16
日本	63	20.0	11.5	0.58
韓国	24	7.7	0.7	0.09
ニュージーランド	12	4.0	2.0	0.50
パキスタン	13	7.0	1.0	0.14
スウェーデン	20	1.8	2.0	0.26
台湾	14	4.0	2.0	0.50
イギリス	9	3.5	0.3	0.07
米国	8	2.8	0.5	0.18
ベトナム	10	4.3	0.3	0.08

2.2. 会議の構成

会議ではさまざまなセッションが催された。会場の様子を図 2 に示す。

初日（12月5日）は、併設ワークショップである The 4th International Conference on Theories and Applications of Computer Science (ICTACS2011) と、青山によるチュートリアルが並列で行われた。また、夕方には Welcome reception が開催された。

2日目（12月6日）は、オープニングセレモニーの後、Boehm氏による基調講演が行われた。その後は、3セッション並列で、合計6つのセッションが開催された。

3日目（12月7日）は、Osterweil氏による基調講演の後、3セッション並列で、合計6つのセッションが開催された。また、午前中には、併設イベントである Industry Talks が行われ、ベトナム人を中心とした発表が行われた。夕方には、バンケットが開催された。

4日目（12月8日）は、2つのセッションと Boehm氏によるチュートリアルが並列で行われた。その後、昼頃にクロージングセレモニーが行われ、会議は閉会した。

また、会議とは別に、local attractions として、ベトナム戦争時に使用されたトンネルで有名なクチへの訪問やホーチミン市内観光、メコンデルタへの小旅行などが提供されていたようである。



図 2 会場の様子

3. セッション紹介

本章では、APSEC2011における基調講演、チュートリアル、テクニカルセッションについて紹介する。なお、基調講演とチュートリアルのプレゼンテーションはAPSEC2011のWebページ⁴⁾で公開されている。

3.1. 基調講演

(1) "Skating to Where the Puck is Going: Future Software Engineering Opportunities and Challenges", Barry Boehm (University of Southern California):

ソフトウェア工学の現状を踏まえ、今後、ソフトウェア工学がどこに向かうのか、どう進歩するのかを示唆するスピーチであった⁵⁾(図3)。冒頭で8つのトレンドと2つのwild-card trendsを挙げ、それぞれについて詳説する形で話が進められた。

ソフトウェア工学の現状を示す8つのトレンドを次に示す。

- 1) 素早い変化
- 2) 臨界状態 (criticality) と信頼性 (dependability)
- 3) 複雑さの増大
- 4) COTS やオープンソースの成長
- 5) スマートなシステムと膨大なデータからのマイニング



図3 講演を行う Barry Boehm 名誉教授

- 6) ユーザ進化とエンドバリューフォーカス
 - 7) 計算量の大きさ, マルチコアチップ
 - 8) システム工学とソフトウェア工学の統合
- これらの話題の要約は次の通りである。

近年、グローバルな連携や競争によって、技術や市場環境の変化が早まっている。このような状況では、これまでに増して、機敏さ、継続的な学習が必要となる。大きなシステム開発には、機敏で計画に基づいたプロセス (agile/plan-driven process) が必要である。また、ソフトウェアは、ますます製品やサービスの成功に不可欠となっている。しかしながら、ソフトウェアの信頼性は、ベンダにとっては、必ずしも優先事項ではない。Boehm氏は、このような状況では、2025年までに「9/11」が再来するという警告を行った。ここでの「9/11」は、甚大な被害をもたらすソフトウェア欠陥を意味している。これを回避するためには、市場の要求、より強力な製品の保証 (warranty) が有効に働くと考えられる。

さらに、ソフトウェアに関する現状の問題として、構成要素が互いに強く連結しているストープパイプシステム、レガシーシステムに関する問題が指摘された。また、商用既製品 (COTS)、オープンソース、クラウドコンピューティングの拡大、メガセンサーによる利用できる情報の増大、ユーザ指向の開発手法の広がりなどの現状と、それらが抱える問題点が概説された。

8つめのトレンドに関しては、システムエンジニアリングを無視しては、良いソフトウェア工学とならない。また、クリティカルな成功要因を無視しては、良いシステムエンジニアリングとならないと述べられた。

講演の終盤では、2つのwild-card trendsとして、自律的ソフトウェア、生物学とコンピューティングの融合について述べた。ここでは、ロボット労働力に触れ、人間の欠点を補うことができる可能性を持っているが、人間の優位性を失わせるという側面もあることや、いまだ、自律的システムの導入をいつ、いかに行うべきかは不透明であること、生物とコンピュータのミスマッチ等の問題が示された。最後に、ソフトウェア工学教育の重要性を強調して、講演は終了した。

(2) "The Promise of Process Definitions that are Rigorous, Broad, and Clear", Leon J. Osterweil (University of Massachusetts):

Osterweil氏はプロセスをマクロプロセスとマイクロプロセスに分け、マイクロプロセスとして有用なLittle-JILプロセス言語とその事例について解説と議論を行った(図4)。彼の言うプロセスはソフトウェア開発のプロセスだけでなく、医療やビジネスなどの広い分野に適用できる汎用プロセスである。

従来の考え方をすれば、プロセスは全てワークフローで大雑把に書けるかもしれない。しかし、ワークフローでは厳密性もコンピュータが活用できる実行可能性にも乏

しい。例えば、誰が行うのか、例外や並列性をどうするのかについては記述しづらい。Osterweil氏はワークフローに置き換わるものとして Little-JIL プロセス言語を紹介し、階層的なツリー形状で「ステップ」を要素とするマイクロプロセスの書き方を述べていた。

紹介された適用事例の中で興味深いのは、医療・介護の分野において輸血ミスに導かれる原因をフォルトツリー分析で追跡するマイクロプロセスである。フォルトツリーは、マイクロプロセスから自動で生成できると述べられている。この点については、多くのミッションクリティカル分野にインパクトがあるので、興味を持って質疑を行った聴講者も最も多かったように思われる。

質疑応答の中には、「論文が reject されました。どうすれば私の論文は accept されますか？コツを教えてください。」と直球質問をする方がいた。質問者の研究分野もまたプロセス関連なのかもしれない。この分野の研究で手法に関するデータを評価して売り込むことは容易ではないし、ソフトウェア工学研究者の一般の関心から外れる可能性もある。少しでも Osterweil 氏のコツを知って帰りたいという意欲が感じられた。accept されるのを待つより、積極的に答えを得る姿勢のある参加者もいたようである。



図4 講演を行う Leon J. Osterweil 教授

3.2. チュートリアル

(1) "Requirements Engineering Based on REBOK (Requirements Engineering Body Of Knowledge)", Mikio Aoyama (Nanzan University):

青山らが策定に携わっている要求工学の基礎知識体系 REBOK に関するチュートリアルが行われた。50名を超える参加者があり、会場は満席に近い盛況であった。

チュートリアルでは、まず要求工学 (RE) の重要性について述べられた。まず、正しい要求がなければ、あらゆる開発が失敗に終わると述べられた。さらに、開発が成功するかどうかに対して、プロジェクト管理や人間の組織構成等の要因と比較して、RE が最も影響を与えることが示された。

SWEBOK 等、ソフトウェア工学に関する他の知識体系は存在するが、RE をカバーする体系は REBOK 以前には存在していなかった。REBOK は、ソフトウェアへの要求、システムへの要求、ビジネス・製品への要求をカバーしている。REBOK のステークホルダは、RE に関わる人、つまり、ソフトウェアやシステムのベンダ、開発者だけでなく、顧客、ユーザまでも含む。



図5 チュートリアルの様子

REBOK には、8つの知識領域 (KA: knowledge area) が含まれる。その8つとは、(1) requirements engineering fundamentals, (2) requirements engineering process, (3) requirements elicitation, (4) requirements analysis, (5) requirements specification, (6) requirements verification, validation, and evaluation, (7) requirements planning and management, (8) practical consideration である。

チュートリアルのスライドは APSEC2011 のサイト⁴⁾にて公開されているので、そちらも参照されたい。また、REBOK についての書籍「要求工学知識体系」⁶⁾ が発行されている。この書籍は、表紙に REBOK のシンボルであるツインエンジェルが描かれている。

(2) "A Process Decision Framework: Incremental Commitment Spiral Model", Barry Boehm (University of Southern California):

Boehm 氏の最近の成果 (ICSM: インクリメンタル・コミットメント・スパイラルモデル)⁵⁾を中心にプロセス決定の仕方に関するチュートリアルが行われた。本セッションは ICSM のより詳しい解説と議論の場となっている。

最初に、Boehm 氏は ICSM 開発の動機を明かしている。スパイラルモデルの提唱者として、ある日、質問者に「スパイラルモデルのどこにリスク分析がありますか？」と質問された。そのことをきっかけにして Boehm 氏の手でスパイラルモデルの拡張がなされている。現在進行形で開発されているプロセスである。ICM (インクリメンタル・コミットメントモデル) と呼ばれていたが、2010年頃に ICSM とした経緯がある。

チュートリアルの中で Boehm 氏は、ICSM のリスク分析をギャンブルに例え、一度きりのルーレットゲームと少しずつ賭けの選択のできるポーカーやブラックジャックの比較を行った。スパイラルモデルの各イテレーションで開発チームの状況を見て、少しずつリスク分析と契約を行う方法を紹介した。最後に彼の ICSM を実践したケーススタディが報告された。

パラレルセッションのため人数は分散するにも関わらず、メインホールの半分を埋める参加が見られた。チュートリアル中の質疑応答は限られるが、休憩時においても Boehm 氏は積極的な質問と議論を歓迎していた。セッションは和やかに終了していたが、プロセスに関する生の意見をぶつけるにはこれ以上無い場であると思う。

3.3. テクニカルセッション

ここでは、著者らが参加したテクニカルセッションにおける発表の一部について紹介する。

(2C) Maintenance and Evolution

"Toward Inconsistency Awareness in Collaborative Software Development", P.T.T. Huyen, et al.:

ソフトウェア開発の成果物を統合する作業で矛盾に気付く場合がある。特に、開発作業中の変更が原因となり、成果物が統合できずに問題になることは多い。協調開発の支援技術をテーマとする研究には、編集権限ロックを設けて未然に矛盾の発生を防ぐ方法や変更後の成果物を分析して矛盾を発見する方法に注目するアプローチが多い。

本研究で特に目を引くのは、変更後の成果物を分析するのではなく、変更作業中の潜在的な矛盾を見つける際にワークフローの時間順序を解析する点である。主に、ワークフローの時間順序から矛盾パターンを分類して、矛盾発見の形式化を行っている。このテーマに関連する別の論文で、さらに、UML 要素の依存関係から矛盾を視覚化するツールの開発を報告している。開発者間のコミュニケーションを支援する技術の進化の可能性を感じさせられる発表であった。

(3C) Metrics and Measurement

本セッションは、2件が No Show であり、2件の発表が行われた。

"A Case Study of Measuring Degeneration of Software Architectures from a Defect Perspective", Z. Li, et al.:

システムの進化に伴って、ソフトウェア構造の劣化が発生し、当初のモジュール化が徐々に破壊される。発表では、当初は分割されていたモジュールが進化につれて1つにまとまってしまう様子が散布図を用いて視覚的に説明された。本研究では、4つの question と、それに関する6つのメトリクスが提案されている。商用コンパイラの2つのバージョンに関してこれらのメトリクスの計測を行い、劣化が発生していることが示された。知見として、複数コンポーネントの欠陥 (MCD) の修正のために必要なコード変更の量が後のバージョンの方が増加していること、劣化の際には "hotspot" と呼ばれる少数のコンポーネントが他のコンポーネントより劣化に寄与していることが示された。

"CMCD: Count Matrix Based Code Clone Detection", Y. Yuan, et al.:

コードクローンは、ソフトウェアの保守性に悪影響を与える。本研究では、特に検出が難しいクローンである「複数のコード断片が同じ計算を行うが、それらが異なる構文で実装されている箇所」に着目している。本手法は、構文の記述順に依存しない count matrix に基づいて類似度を計測するため、ステートメントの順番が入れ替わっているような場合でも、効果的にクローンを発見することができる。本手法を適用した

実験では、学生の作成したプログラムや JDK のソースコードにおいて多くのクローンが発見された。

(4C) Program Analysis

本セッションは、3 件の発表が行われた。ここでは、そのうちの 2 件を紹介する。

"Recommending Proper API Code Examples for Documentation Purpose", L.W. Mar et al.:

API 利用法のドキュメンテーションのための半自動手法 PropER-Doc を提案している。利用者がドキュメンテーション対象の API (型) を指定すると、まず PropER-Doc はコード検索エンジンから該当 API の利用コードを検索する。次に、利用コードを、該当 API がどういった他の API と組み合わせて用いられているかにより分類し、順序付けて利用者に提示する。評価では、Eclipse JDT の ASTParser, SearchEngine の 2API に対して適用し、良好な結果が得られたことが示されている。

"Understanding Source Code Differences by Separating Refactoring Effects", S. Thangthumachit et al.:

ソースコード差分からリファクタリング操作の影響を分離する手法を提案している。リファクタリング等の振舞いを変えない変更とそれ以外の変更が混在した差分を理解することは困難であるため、どういったリファクタリングが行われたかをあらかじめ抽出しておき、旧版のプログラムにこれらを適用したあとの版を用いて差分を計算することにより、リファクタリングの影響を除去した差分を得る。実験により、抽出したリファクタリングの適用が問題なく行えること、リファクタリングの影響を除外した差分が理解に貢献すること等を確認している。

4. ソーシャルイベント

会期中のソーシャルイベントとして、次の 2 つが開催された。

(1) Welcome Reception

12 月 5 日のワークショップ、チュートリアルを終了後、会場近くの Nostalgic Club と呼ばれるフランススタイルの建物で、立食パーティがあった (図 6(a))。

(2) Banquet

12 月 7 日に会議会場の Rex Hotel の屋上レストランを会場として開催された (図 6(b))。ベトナムの民族音楽や踊りなども披露された。また、Best Paper Award と Outstanding Paper Award の表彰、あわせて、会議の運営にあたった関係者へのねぎらいの紹介もあった。表彰された論文は以下の通りである。

Best Paper Award:

M. Aoyama and H. Tanabe: A Design Methodology for Real-Time Distributed Software Architecture Based on the Behavioral Properties and Its Application to Advanced Automotive Software.

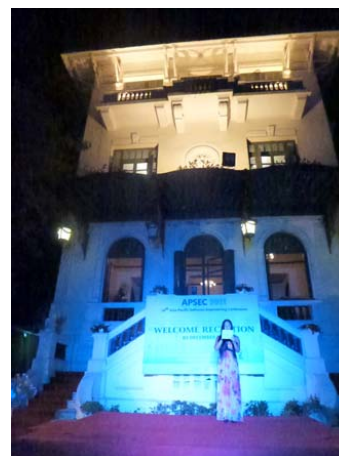
Outstanding Paper Award:

C.-H. Hsueh, Y.-P. Cheng and W.-C. Pan: Intrusive Test Automation with Failed Test Case Clustering.

E. Escott, P. Strooper, J. Steel and P. King: Integrating Model-Based Testing in Model-Driven Web Engineering.

5. 所感

大森：私としては、2005 年台北、2009 年マレーシアに続き、3 回目の APSEC 参加であった。開催国がベトナムということもあり、ベトナム人の参加が多く、ソフトウェア工学研究のベトナムでのアクティビティの高まりを感じることができた。また、一方で、日本人の参加者が多く、アジア太平洋圏での日本のプレゼンスはまだ健在であると感じた。全体的に No Show が多かったことは残念である。会議の運営に関しては、登録サイトの開設が遅れるなどの問題はあったものの、全体的には、大きな問題なく運営がなされていたように感じた。今後とも、学生や若手研究者の発表の場として APSEC が活用されれば良いと思う。



(a) レセプションスピーチ



(b) Banquet でのパフォーマンス

図 6 レセプションとバンケットの様子

大山: 国際色豊かなソフトウェア工学研究者の参加があり、盛況であった。アジアに限らず、欧米からの参加もあり、オープンなコミュニティのイメージがあった。その場で知り合う各国の発表者達の間からは、APSEC 参加者にはアジアで研究発表できるメジャーな国際会議、又は、研究者の輪を作る場という点で共通認識があることが伝わってくる。私自身は初めての APSEC 参加である。アジア各地の輪を広げる機会として、今後の盛り上がりを期待するとともに、少しでも役立てればと思う。

林: 名古屋 (2007)、北京 (2008)、シドニー (2010) に続き 4 回目の APSEC 参加だった。日本から多くの発表があったこと、アジア太平洋圏のみならず世界各国の研究者による発表も多く見られたことは良い点として印象に残った。一方で、多くの No Show があったこと、質疑に活気のないセッションがあったことは残念に感じた。今後も多くの学生や若手研究者の登竜門としてこの会議が機能し続け、アジア太平洋圏での国際会議発表・開催経験の充実化に貢献していくことを願う。

青山: 今回の APSEC はベトナムで初めて開催された。ベトナムの主催関係者もこのような規模の国際会議を運営することは初めてということで、まごついたこともあったようであるが、多くの学生ボランティアを含めて熱心に運営されている姿勢に好感を抱いた参加者が多かったようである。

Boehm と Osterweil という、本分野で最も著名な研究者の 2 人を基調講演に招くなど、内容面でも質が高かったと言える。



(a) 学部の講義の様子

(b) 大学院の講義の様子

図 7 University of Science の様子

会議終了の翌日、主催した University of Science (<http://hcmus.edu.vn/>) の情報工学部 (FIT: The Faculty of Information Technology) を訪問する機会を得た。この学部には 6 学科あり、ソフトウェア工学科 (Dep. of Software Engineering) が最大とのこと。教員 45 名、学部生数は~200 名/年、大学院生数は学部全体で~100 名/年を擁する大規模な学科である。講義は早朝 7 時からで、大学院生の講義は新しい校舎で空調の効いた立派な部屋であったが学部生の部屋は古い校舎で窓を開けていた (図 7)。学部生の講義も英語で行われているものもあった。学生達が眼を輝かせて話を聞いてくれた。ベトナムは日本からのソフトウェア開発のアウトソース先として注目されているが、将来、多くの技術者、研究者を生み出す可能性を感じた。

6. おわりに

本稿では、ベトナムで開催された APSEC2011 について概観した。

次回の APSEC2012 は香港で、2012 年 12 月 4~7 日の予定で開催される。主要な情報は APSEC2012 の Web ページ⁷⁾にて公開されている (2012 年 2 月現在)。

参考文献

- 1) 川口 真司, 柿元 健, 大森 隆行, 王 秋時, 南川 恭洋, 坂本 一憲, 野田 訓広, 福島 義彦: 第 16 回アジア太平洋ソフトウェア工学国際会議 (APSEC2009) 参加報告, 情報処理学会研究報告, Vol. 2010-SE-167, No. 22, pp. 1-8 (2010).
- 2) Selby, R. W. (ed.): Software Engineering: Barry Boehm's Lifetime Contributions to Software Development, Management, and Research, IEEE Computer Society/Wiley (2007).
- 3) Tarr, P. L. and Wolf, A. L. (eds.): Engineering of Software: The Continuing Contributions of Leon J. Osterweil, Springer (2011).
- 4) APSEC 2011: <http://www.apsec2011.org/>.
- 5) Boehm, B.: Some Future Software Engineering Opportunities and Challenges, The Future of Software Engineering, pp. 1-32, Springer (2011).
- 6) REBOK 企画 WG: 要求工学知識体系 第 1 版, 近代科学社 (2011).
- 7) APSEC 2012: <http://apsec2012.comp.polyu.edu.hk/>.