

ウインターワークショップ 2010・イン・倉敷 開催報告

中鉢欣秀^{†1} 川口真司^{†2} 肥後芳樹^{†3} 中村匡秀^{†4}
羽生田栄一^{†5} 鷺崎弘宜^{†6} 土屋達弘^{†7} 石黒正揮^{†8}
梅村晃広^{†9} 森崎修司^{†10} 岸知二^{†11}

2010年1月21日、22日の2日間に倉敷市芸文館（岡山県倉敷市）にて開催したウインターワークショップ 2010・イン・倉敷（WW2010）の概要について報告する

Report on Winter Workshop 2010 in Kurashiki

Yoshihide Chubachi^{†1}, Shinji Kawaguchi^{†2}, Yoshiki Higo^{†3},
Masahide Nakamura^{†4}, Eiiti Hanyuda^{†5},
Hironori Washizaki^{†6}, Tatsuhiro Tsuchiya^{†7},
Masaki Ishiguro^{†8}, Akihiro Umemura^{†9}, Shuji Morisaki^{†10}
and Tomoji Kishi^{†11}

This paper reports on “Winter Workshop 2010 in Kurashiki (WW2010)” held at Kurashiki Geibunkan in Kurashiki City, Okayama Prefecture from January 21th to the 22th.

1. はじめに

情報処理学会ソフトウェア工学研究会では、テーマを絞った集中的な議論の場を提供するため、ワークショップを1997年より毎年開催している。本年度も表1に示す7つのテーマについて討論を行った。

表1 セッションのテーマとリーダー

テーマ	リーダー
プログラム解析「プログラム解析技術の普及に向けて」	川口真司（奈良先端大） 肥後芳樹（大阪大）
要求工学「要求工学」	中鉢欣秀（産技大）
サービス指向「SOAにおけるサービス品質」	中村匡秀（神戸大）
アーキテクチャとパターン「ソフトウェアの設計とパターン・アジャイル開発」	羽生田栄一（豆蔵） 鷺崎弘宜（早稲田大）
ソフトウェア開発マネジメント「ソフトウェア計測とその活用」	森崎修司（奈良先端大）
形式手法「形式手法：技術と応用」	土屋達弘（大阪大） 石黒正揮（三菱総合研究所） 梅村晃広（NTTデータ）
ソフトウェア工学研究の評価「ソフトウェア工学研究の評価」	岸知二（早稲田大）

^{†1} 産業技術大学院大学

Advanced Institute of Information Technology

^{†2} 奈良先端科学技術大学院大学

Nara Institute of Science and Technology

^{†3} 大阪大学

Osaka University

^{†4} 神戸大学大学院工学研究科

Graduate School of Engineering, Kobe University

^{†5} 株式会社豆蔵

Mamezou, Co., Ltd

^{†6} 早稲田大学

国立情報学研究所 GRACE センター

Waseda University

National Institute of Informatics GRACE Center

^{†7} 大阪大学

Osaka University

^{†8} 株式会社三菱総合研究所

Mitsubishi Research Institute, Inc

^{†9} 株式会社 NTT データ

NTT Data Corporation

^{†10} 奈良先端科学技術大学院大学 情報科学研究科

Graduate School of Information Science, Nara Institute of Science and Technology

^{†11} 早稲田大学

Waseda University

それぞれの分野に造詣の深い討論リーダーを中心として、現状の認識、そして解決すべき課題は何か、我々はこれから何を行うべきか、といった視点で議論した。

今回は、2010年1月21日(木)、22日(金)の2日間の日程で、倉敷芸文館と倉敷アイビースクエアを会場として開催した。今回は予想を超えて昨年を上回る90名の参加があり、大変盛況であった。参加者の内訳は、ソフトウェア工学研究会登録会員41名、情報処理学会会員18名、学生22名、非会員11名、であった。また、このうち26名は企業からの参加(社会人学生を含む)であった。

参加者は上述の7つの検討グループに分かれ、それぞれ検討リーダーを中心として、おのおの大学・企業の立場から研究活動や技術課題について議論を行った。

以下、2~8節では各討論グループで行われた議論の内容について述べる^{a)}。9節で運営に関する課題等について述べ、10節でまとめを行う。

2. プログラム解析「プログラム解析技術の普及に向けて」

2.1 はじめに

「プログラム解析」セッションでは、例年プログラム解析技術を用いたさまざまなプログラム開発および保守支援に関して議論していた。特に、近年では、ソースコード以外の成果物に対する解析やソフトウェアリポジトリに蓄積されている履歴を解析対象とすることに関心が高まっている。

今回のワークショップでは、プログラム解析技術の普及に関して議論したい。ソフトウェアが社会のあらゆる場面で利用されており、より高い信頼性が求められている。それに伴い、ソフトウェア開発企業では、品質保証のために開発したソフトウェアをソフトウェア工学的な手法を用いて調査・管理することが増えてきている。また、産学連携がさかに行われており、大学等の研究機関で開発された技術を開発現場において積極的に生かす動きがある。本セッションでは、このような動向を支援するために、技術的な側面と非技術的な側面の両面から議論を行う。

2.2 プログラム解析技術の普及に向けて

我々は、プログラム解析技術の普及に向けて、プログラム解析技術は下記の技術的な側面を持つことが重要であると考えている。

2.2.1 スケーラビリティ (実規模ソフトウェアへの適用可能性)

実際の開発プロセスにおいて解析技術を適用する場合、現実的な規模の計算機資源を用いて解析が実用的な時間で終わらなければならない。解析結果を待つために、本来のしなければならない作業ができなくなるとは本末転倒である。そのため、スケーラビリティは解析技術の重要な属性である。NP 完全などの非常に高い計算コ

a) 2~8節の内容は本ワークショップ論文集に掲載の、各セッションの紹介原稿に基づく。ページ数の都合上、一部を割愛している。また参考文献は省略したので、詳細は論文集を参照されたい。

ストを必要とする技術を利用する場合は、利用範囲を限定する、または、近似アルゴリズムを用いる、などの工夫が必要となってくる。

2.2.2 現場で利用しているツールへの組み込みやすさ

現場で利用している開発ツールで簡単に解析手法が使えることが普及に当たって重要である。例えば、Visual Studio を用いて開発しているプロジェクトでは、Eclipse のプラグインとしてのみ提供されているプログラム解析技術の導入には二の足を踏むであろう。簡単に利用できるプログラム解析技術(とその実装)は、普及しやすい側面を持つといえる。

2.2.3 開発者の手を煩わせない(半)自動的情報収集

開発者は日々の作業に追われているため、プログラム解析のために新たに情報収集作業を行うことには消極的である。解析に必要な情報を集める過程において開発者に負担をかけないほど、利便性は高いといえる。完全に自動で収集することが難しい場合は、一日一回10分間、ウィザード形式で漏れなく情報を収集するなどの工夫も必要だろう。そのため、自動的に必要な情報を収集可能なプログラム解析技術は、実用的であるといえる。

2.2.4 十分な信頼性

実際の開発現場でプログラム解析技術を利用するためには、その信頼性が十分でなければならない。例えば、再現率が90%の手法があった場合、学術的には良い評価を受けるだろうが、実際の開発現場では、10%の確率で見逃してしまう、とのことで利用できないかもしれない。実際の開発現場の適用状況に応じて、十分な信頼性を持つことがプログラム解析技術として重要である。

2.2.5 既存の開発プロセスとの親和性

開発プロセスに組み込みやすい解析技術でなければ、利用は難しい。また、その開発プロジェクトでは通常生成しないような成果物を入力として必要とする解析技術は、その解析技術を用いるために新たな成果物を生成するための手間が必要になり、効率的にソフトウェア開発を支援できているとはいえない。通常プロジェクトにおいて生成される成果物のみで利用可能な解析技術、また「現場で利用しているツールへの組み込みやすさ」とも関連するが、そのプロジェクトで利用されているツールに組み込んで利用可能な解析技術が、実用的であるといえる。

2.2.6 開発手法の配布・周知プロセス

素晴らしい解析技術を提案と実装を行っても、それを現場の開発者が知らなければ利用してもらえない。大学等の研究機関で開発した技術をいかに素早く、そして広く産業界に周知するのには、プログラム解析技術の普及に当たって大きな課題である。現状では、モチベーションの高い開発者が自主的に学会論文誌に目を通したり、オープンソースソフトウェアに関心のある開発者がオープンソースプロジェクトのウェブページで情報収集を行ったりと、開発者の個々の行動に大きく依存している。そうで

はなく、開発プロジェクトとしていかにそのような情報をシステムティックに取り込むか、また大学側としてはいかに開発した技術を周知するのかを議論しなければならない。

3. 要求工学「要求工学」

3.1 はじめに

要求工学とは情報処理システムに対する要求の獲得、記述、仕様化、評価などのソフトウェア開発プロセスにおけるいわゆる上流工程を取り扱う学問領域である。近年、情報処理システムに対する要求は高度で複雑なものになっている。一般的に、情報システムの開発において上流工程での失敗はそのままシステム開発全体の失敗につながるとされる。これを防ぐためにも、要求工学に関する研究が必要である。

3.2 今回の議論

本年度の投稿件数は 10 件である。以下、各論文のタイトルとアブストラクトを紹介する。

3.2.1 要求仕様の構成変更支援

ソフトウェア要求仕様の機能要求には、利用者のクラス、フィーチャー、入力データに基づくなどさまざまな構成法があり、それぞれ記述の観点が異なるため、記載される情報や書き易さや読み易さが異なる。手作業で構成を変更することは容易ではなく、本稿では特定の構成による要求仕様を異なる構成の要求仕様に自動的に変換する手法を提案する。

3.2.2 ユーザ満足度を損ねずに省エネルギーを目指す要求適正化手法

環境への配慮等から、低消費エネルギー化が求められている。しかし現状では、システム開発の上流工程からの統括的な取組みは十分とはいえず、ユーザが求める以上の機能や性能が作りこまれる半面、必要以上のエネルギー消費を行う製品の開発も少なくない。我々はユーザ満足度を損なわずに低消費エネルギー化を行うための要求適正化手法の研究開発を行っている。本文書では手法の概略と、それに基づいて実施したケーススタディについて報告する。

3.2.3 授業科目の要求分析

大学においてシステム開発演習を実施しているが、時間や回数が限定されており、実施すべき内容を絞り込まなければならない。今回、製造業で定評のある QFD（品質機能展開）を流用して、演習の要求分析を行う。

3.2.4 システムの利用率は要求分析の対象では？

電子政府の電子申請システムの利用率の悪さが数年前にも今も問題になっている。利用者視点を忘れ、多額の税金を投入し、電子化実施率を競ってきた結果と言えるが、そもそもシステムの利用率予測は要求分析の対象ではないのかという疑問がある。

3.2.5 属性つきゴールグラフにおけるゴールの品質特性

この論文では、属性つきゴール指向要求分析法(AGORA)において、ゴールや枝に付加された属性値を活用し、ゴールの品質特性を評価する手法を提案する。

3.2.6 ウィンターワークショップポジションペーパー

近年要求獲得に際して、非機能要求の獲得の議論が活発になってきている。システム運用においても、非機能要求で求められる運用性を中心に、要求獲得の段階で要求を的確に抽出し、システム運用段階での運用のし易さや、信頼性、効率性、セキュリティ等の適切な運用がなされることが求められている。本ポジションペーパーでは、要求工学の知見に基づき、システム運用の立場から、如何に非機能要求を獲得していくかについて議論する。

3.2.7 ユーザ特性の分析によるアクセシビリティ要求獲得支援

アクセシビリティの高いソフトウェアを開発するために、アクセシビリティガイドラインが多く企業の企業や団体で策定されている。しかし、ガイドラインの項目は数多く存在し、各項目への対応の難易度も様々である。中には、開発段階の早期に対応を検討しなければ満たすことが難しい項目も存在する。従って、ソフトウェアに対するアクセシビリティの要求を、開発のできるだけ早い段階で抽出しておくことが重要である。本研究では、ソフトウェアに対するアクセシビリティ要求の漏れを低減することを目的として、ユーザの特性を分析することにより、アクセシビリティ要求を抽出する手法について提案する。

3.2.8 プロブレムフレームにおける要求から仕様への変換

プロブレムフレームにおいて、ドメインプロパティを固定させると、要求を変化させることにより、その要求を満たすマシンの仕様が変化する。ここでは、要求を安定状態に於けるドメインの組の取りうる状態によって規定するものとし、それから仕様を自動生成する方式について述べる。

3.2.9 PRINCE モデルに基づく計画的な要求獲得に向けて

ソフトウェアの開発プロジェクトを成功に導く技術的課題は多い。PRINCE モデルは、過去のプロジェクトで行われた要求獲得プロセスから得られた知見に基づいて、要求獲得計画を立てるためのガイドを提供する。本稿では、要求獲得計画を立案するために、要求の変わりやすさと要求獲得計画との関係について議論する。

3.2.10 コンテキストモデル

車載電子システムの開発プロセス再構築を目指して、要求開発の成果物やプロセス定義を行う上で、コンテキストの表現方法や抽出方法に深い関心がある。

3.3 おわりに

以上見てきたとおり、昨年に引き続き本年も要求工学領域における様々なトピックスを取り上げた多様な議論が展開された。

要求工学は情報システムの開発における主要な研究課題であり、この領域の発展は

情報システム産業界にも幅広く寄与する。

4. サービス指向「SOAにおけるサービス品質」

4.1 はじめに

ウィンターワークショップ 2010・イン・倉敷のサービス指向セッションでは、サービス指向アーキテクチャ(SOA)におけるサービス品質に焦点を当て「良い SOA サービスとは何か」を多角的に議論すべく論文募集を行った。

サービスが氾濫する世の中で、「良いサービス」を様々な観点から見極める品質属性が必要である。上記の背景を踏まえて、本年度のサービス指向セッションでは、SOA サービスの品質を性質づける様々な要因や属性について議論を行うことにした。キーワードとしては、「品質属性、プロダクトメトリクス、プロセスメトリクス、SLA、ポリシー、評価法」等が挙げられるがこれらに限定するものではない。構想段階の研究アイデアや開発秘話などインフォーマルな(だからこそ核心をついた)発表も含め、幅広く募集を行った。

4.2 2. IEICE/SIGSC の協賛

今回は、電子情報通信学会・サービスコンピューティング時限研究専門委員会(IEICE/SIGSC)の協力を得て、IPJS と IEICE の両学会をまたいだサービス指向コミュニティ形成を狙った。IEICE/SIGSC は、NiCT 村上陽平氏を委員長、NII 石川冬樹先生を副委員長とし、サービスコンピューティングを専門とする時限研究会である。2009 年に発足した新しい研究会であり、2009 年 7 月に第 1 回研究会を実施したばかりである。このたびウィンターワークショップへの協賛をお願いしたところ、快諾いただいた。SIGSC としては第 2 回の会合となる。

4.3 採録論文

募集期間中に 17 件のポジションペーパーが寄せられ、全て採録した。カテゴリ別に紹介する(敬称略)。

4.3.1 クラウドコンピューティングにおけるサービス品質

クラウドは、様々な計算資源(SW,HW,DB, サーバ,etc) をネットワーク側で用意し、それらを利用者が必要な時に必要な分だけサービスとして提供する新しい ICT の利用形態である。クラウド時代に重要なサービス品質とは何か、どのような課題や応用があるのかを議論すべく、以下の 4 つの論文が寄せられた。

- クラウドコンピューティングのサービス品質モデルと課題: 青山幹雄(南山大)
- クラウド時代の SOA の役割: 小林透(NTT)
- 教育学習支援のためのクラウド型 CLE アーキテクチャの検討: 梶田将司(名大)
- ユビキタスクラウドによる適応型ユビキタスサービスの提案と評価: 江上公一(神戸大)

4.3.2 サービス品質を考慮したSOA システム設計, リエン지니어リング

一般のソフトウェアと同様、サービスの機能的な品質はサービス開発の上流工程(要求・設計) で作りこまれるべきである。ただし、一般のソフトウェアと SOA サービスとの間には、その利用方法や配置方法、運用方法において大きな差がある。したがって、SOA サービスならではの設計や再利用の手法、ノウハウが存在し然るべきである。以下の論文は品質を意識したサービス設計、リエン지니어リング手法を議論している。

- サービスの機能設計・品質設計方法の提案: 細野繁(NEC)
- ゴール指向要求モデルに基づくサービス設計の提案: 茅竹(筑波大)
- マイニング技術に基づく既存システムの最適化に向けて: 佐藤宏之(NTT)
- サービス設計における付随機能の考慮による QoS の向上: 高橋竜一(早大)

4.3.3 サービス検索, 選択, マッシュアップにおけるサービス品質

将来的に無数の SOA サービスが利用可能になると、どのサービスを選ぶべきか、どのサービスとどのサービスを組み合わせて使うべきかを利用側が自ら判断する必要がある。利用者にとっては、いつでも「品質の良い」サービスを選べることに越したことは無いが、問題はそう簡単ではない。以下の 4 論文は、サービスを検索、選択、マッシュアップ(合成) する際に考慮すべきサービス品質について、様々な角度から議論を行っている。

- サービス品質に関する多様な想定に応じた選択・適応手法の活用に向けて: 石川冬樹(NII)
- 組み合わせ QoS を考慮した自動サービス合成に関する研究: 渡辺敦(早大)
- SMuP:センササービスのマッシュアップを実現するサービス指向基盤: 坂本寛幸(神戸大)
- Notification 型 Web サービス特有の QoS によるサービス選択に向けてのアプローチ: 片淵聡(早大)
- テスト実行に基づく REST 形式 Web サービスの高精度な検索: 重井康宏(早大)

4.3.4 サービス品質に基づくサービス制御・運用

サービスを配備(デプロイ) した後も、ソフトウェアやハードウェアの故障、例外の発生、アクセス集中、トラヒックの急増、顧客要求の変化、セキュリティポリシーの変更等に伴うサービス品質の低下が懸念される。以下の 4 論文は、運用時に発生する様々な状況に応じて、柔軟なサービス制御を行うことで、サービス品質を向上させるアプローチを議論している。

- SOA におけるトランザクション: 齋藤伸也(オージス総研)
- QoS-Based Dynamic Service Deployment: 林冬恵(NiCT)
- 言語グリッドにおけるポリシー制御: 村上陽平(NiCT)
- サービススーパビジョンによる複合 Web サービスの QoS 制御: 田仲正弘(NiCT)

4.4 おわりに

今回 17 件もの多くのポジションペーパーが寄せられ、どれも SOA における重要なトピックを議論するものである。国内のサービス指向コミュニティが少しずつ形成されているのを実感しており、参加・協力していただいた全ての方に感謝したい。この状況に満足せず、今後もウィンターワークショップをはじめ、様々な研究集会に積極的に参加し、サービス指向のコミュニティの発展に努力していきたい。

5. アーキテクチャとパターン「ソフトウェアの設計とパターン・アジャイル開発」

5.1 はじめに

パターンとは、特定の文脈上で頻出の問題と解決を一般化した記述であり、ソフトウェアの設計や、プロセス・組織など様々な面で創造性を持って決定指針を与える。例えばソフトウェアの設計において、扱う要求や制約の類似性に伴って繰り返し共通に出現する構造をデザインパターンとして再利用することで、新たな設計を効率よく一貫性のある形で進めることができる。

設計、パターン、アジャイル開発は、対象の構造（設計）を徐々に生み出す過程ならびにその結果という面において共通性を持つが、近年のさらなるソフトウェア開発形態や社会環境の変化により、それぞれ議論し尽くされておらず、また、関係も依然として未整理な部分があると考えられる。そこで本セッションでは、参加者のポジションペーパー発表を起点として各領域に関連する経験や提案を概観し、続いて、それぞれの課題や展望、関係や周辺について幅広く議論する。

本稿では以降において、これまでの議論経緯と本セッションにおける議論のテーマ案を紹介する。

5.2 これまでの議論

情報処理学会ソフトウェア工学研究会では、2003 年にパターンワーキンググループが結成され、以降、パターンやパターン活用支援技術に関する議論を重ねてきた。ウィンターワークショップにおいては、2004 年よりワーキンググループが中心となって「パターン」セッションおよび「アーキテクチャとパターン」セッションを設置し、各議論の成果を公開してきた。これまでの成果を以下にまとめる。

- 石垣島 2004: ソフトウェア要求獲得を実験し、建築でのパターンランゲージがもたらす要求獲得支援効果が、ソフトウェア開発についても得られることを明らかにした。これは、パターン活動プロセスを構成する利用活動の評価/適用工程について得られた知見である。
- 伊豆 2005: 参加者の経験からパターンを発掘した結果、パターンの必要条件として対立するフォースの組が不可欠ことや、マイニング手法によってパターン

ランゲージへの発展を支援できる可能性を明らかにした。これらは、パターン活動プロセスを構成する抽出活動の発見/記述工程について得られた知見である。

- 鴨川 2006: ライターズワークショップの実施により、パターンランゲージが備えるべき特性として、範囲設定の妥当さ、および、パターン間の関連の明確さを明らかにした。さらに、パターン活動支援技術の特性を幾つか明らかとした。
- 那覇 2007: 種々の要素技術の討議の上で、ソフトウェア開発におけるアーキテクチャとパターンの関係を整理することを試みた。特に、アーキテクチャ設計に注目した場合に、同関係の 1 つの見方としてパターンがコンテキストと実現手段を結びつけるものであり、かつ、プロセスであると位置づけ可能なことを整理した。
- 道後 2008: 各技術に関連する経験や提案を概観し、続いて、アーキテクチャ進化とパターンの課題や展望、関係について幅広く議論するとともに、アーキテクチャ進化のメタモデル記述実験やプレゼンパターンのパターンランゲージ化実験を行い、進化やプレゼンパターンの可能性を確認した。
- 宮崎 2009: ソフトウェア開発におけるパターン、リファクタリング、アーキテクチャの課題や展望、関係について幅広く議論するとともに、アーキテクチャとパターン・実現手段および要求との関係をメタモデルとして記述する実験を行い、そのメタモデルを通じて、アーキテクチャ設計においてパターンが要求を含む文脈と実現手段を結びつけることを改めて確認した。

5.3 議論テーマ

以上 6 年間の継続的議論は主にパターンの抽出/利用の観点から実施されたものであり、2007-2009 年の議論についてもパターンと設計、ならびにアジャイル開発の関係について議論し尽くされていない。そこで、各ポジションペーパー発表を起点として、特に下記を議論のテーマとして検討する。

- 原理や原則: Alexander の原則との関係、設計原則、その他の普遍的な原理など
- 非機能要求や「無名の質」: 設計における品質や制約とパターン、アジャイル開発プロセスの関係など
- ユーザ参加やワークショップのプロセスとパターン: パターンマイニングワークショップ、パターンダンスの効果と拡張など
- 表現とモデリング: パターン記述形式、設計のモデリングとパターンの関係など

さらに上記に加えて、2010 年 3 月に本セッション参加者を中心としてアジア圏におけるパターン会議 AsianPLoP の開催を予定しており、PLoP 会議の運営やパターンコミュニティとの関わりや、アジャイル・パターンムーブメントの展望についても合わせて議論したい。

5.4 おわりに

本セッションにおける議論の成果は、パターンワーキンググループの Web サイトやソフトウェア工学研究会を通じて公開し、さらなる議論へと繋げていく予定である。今後、さらなる議論、研究および実践を通して、得られた知見が検証され、ソフトウェア開発活動および組織活動一般においてパターンを軸とした設計やアーキテクチャ、アジャイル開発が活用されることを期待する。

6. ソフトウェア開発マネジメント「ソフトウェア計測とその活用」

6.1 はじめに

ソフトウェアの浸透が社会的インフラストラクチャーにまで広がり、ソフトウェアの信頼性は高まるばかりである。また、製品やサービスの価値の支配的要因をソフトウェアで実現される機能が担っていたり、製品やサービスの市場への投入に際して、ソフトウェアの開発期間がクリティカルパスとなっていることも少なくない。ソフトウェア開発が予定どおり進行することに対する要求は極めて大きくなっているといえる。

PMBOK, CMMI, ISO9000など、プロジェクト管理やプロセス改善、品質管理の枠組みが目ざされている。多くのソフトウェア開発組織がこれらの枠組みを導入しつつある。これらの枠組みはいずれも、プロセスやプロダクトの計測による現状把握・改善を要求している。

これら計測は、リスクの早期発見や状況判断等、円滑な進行を助けるばかりでなく、開発が適切に実施されたことを示すエビデンスにもなり得る。本セッションでは研究者、実務者、発注者、受注者等、立場に限らず、計測の方法、計測結果の利用方法、計測にかかる労力、事例等、多面的な議論を展開したい。

6.2 討論テーマ

6.2.1 対象

本セッションでの計測対象はプロダクト、プロセス、プロジェクトとする。プロジェクトの中には開発メンバが含まれ、メンバの生体情報も含まれるものとする。これらいずれかから取得、計測した情報をもとに状況把握、対策を検討し、プロジェクトの計画、管理を円滑に進めていくために役立つ。

6.2.2 フェーズ

本セッションで議論される計測とその活用は、主に以下の3つのフェーズを明らかにしつつ議論を進めたい。

- データ取得、計測 データを集める時期である。可能なものについては、集める方法についても簡単に議論したい。また、可能であれば計測項目の設定とその動機やノイズ値除去についても議論したい。
- 予測、検討、活用 得られたデータを分析、利用する時期である。定式化や形式化の場合、起きている問題と定式か、形式化の対応を議論したい。また、どのようにプロジェクト

の計画や進行に活かすかを議論したい。

- 改善、振り返り 得られたデータを利用して、予測、検討、活用した後、今後のデータ取得、計測や予測、検討、活用の改善や振り返りについて議論したい。
たとえば、予測の場合、予測精度の向上に向けた指針やより低コストでのデータ取得、計測に向けた議論である。

6.2.3 計測データ

本節では、本セッションの研究テーマを理解しようとする読者や本セッションの参加者の理解を助けるために、以下のような計測、取得データを例示する。ただし、これ以外の計測データが対象にしないという意図ではない。つまり、本セッションで対象とする計測データの種別を制限するものではない。

- プロジェクト属性データ 複数の開発プロジェクトの種別、規模、品質、コスト、期間等のメトリクス値を計測したものである。
- 品質データ 品質データはプロジェクト属性データと類似している。サブシステムや機能毎に、レビューや試験で検出された不具合の件数を記録したり、サブシステムや機能の規模、レビューや試験の規模、を記録したものである。

6.3 おわりに

ソフトウェア計測とその利活用により、より円滑なソフトウェア開発、また、適正な開発を実施したことを示すエビデンスとなることが期待される。本セッションの議論が、産業界にも役立つものとなることを願う。

本セッションの議論は公開する等して、引き続き検討していきたい。

なお、本稿の一部は、文部科学省「次世代 IT 基盤構築のための研究開発」の委託に基づいて行われた研究を含む。

7. 形式手法「形式手法：技術と応用」

7.1 はじめに

モデル検査を始めとする形式手法は技術的に成熟期を迎え、産業界における注目も高まっている。一方で、実際の適用において形式手法技術を有効に利用するためには、個々の技術固有の性質を理解し、問題にあった適切な手法を選択しなければならず、また、種々のチューニングやモデルの抽象化など、克服すべき課題も多い。

このような技術の成熟と適用に関する実用上の要請を背景として、今回のワークショップでは「形式手法：技術と応用」と題し、特に実用上の問題点に焦点を当てて、形式手法について議論する。

7.2 採択論文

本セッションでは最終的に 14 編のポジションペーパーが採択された。以下に、採択論文で取り上げられている形式手法と、手法の適用対象をまとめる。

7.2.1 形式手法, ツール

ここでは, 14 編のポジションペーパーで扱われている形式手法, および, 形式手法を実装したツールを挙げる. 分かりやすいよう, モデル検査技術と仕様記述技術に大別している. ただし, 仕様記述技術でもモデル検査器同様の自動検証が可能なのは多く, 逆に, モデル検査ツールへの入力モデルは形式仕様記述とも考えられるため, これはあくまでも大まかな区別であることに注意されたい.

7.2.2 適用対象

今回採択された論文では, 一般的な適用対象を想定した形式手法自体について論じたものは少数で, 多くは何らかの具体的な文脈における手法の利用について議論している.

対象の多くは組込みシステムの範疇に入るものである. この分野は, 2008 年のワークショップでも, 「組込みシステムのためのモデリング技術」と題したセッションが設けられるなど, 従来より形式的な手法の適用が図られてきた分野である. 一方で, Ajax アプリケーションやビジネスプロセス記述など, 従来にない新しい対象が形式手法の射程に入ってきたことは, 注目すべき動向と思われる.

7.3 議論のテーマ

今回のセッションでは, 通常の技術的な情報の交換だけでなく, 経験上得られた文書化困難な勘所や, 形式手法導入に係わる教育や組織上の工夫といった公表される機会の稀なテーマについても, 議論を深められることを期待している.

7.4 おわりに

本セッションでは, 形式手法の普及とより効果的な利用を目指し, 上述の内容を中心に議論を深めたい. 本セッションは, システム開発プロセスでの実用経験を背景に持つ企業人と, 研究教育機関の研究者が会し議論できる希有な機会であり, 議論を通じた課題と情報の共有によって, この分野の発展に寄与できることを期待している.

8. ソフトウェア工学研究の評価「ソフトウェア工学研究の評価」

8.1 はじめに

ソフトウェア工学研究の妥当性をどのように確認し主張するかは難しい問題である. しかしながら, 自らの研究がソフトウェア工学上の問題解決にどの程度貢献し, 既存の理論や技術と比べてどのような優位性や課題があるのかを評価できない限り, 研究を進めることはできない. また論文を書く際には, 提案する内容の価値を主張し査読者に納められないと採録されない. もちろん研究者自身だけでなく, 他の研究成果を理解しようとする立場にある人は, それぞれの立場からその研究を評価する必要がある. 本ワークショップでは, こうしたソフトウェア工学評価に関して様々な観点から検討を加え, ソフトウェア工学の評価のあり方を模索する.

8.2 研究や評価の多様性

ソフトウェア工学研究の評価を難しくしているひとつの理由に, ソフトウェア工学研究や評価の持つ多様性が挙げられる. 以下それらのいくつかについて考える.

8.2.1 研究領域の多様性

ソフトウェア工学の研究は多様な研究領域にわたっており, それぞれが異なった特性や背景を持っている. そのためソフトウェア工学としてひとくくりにできない点にひとつの難しさがある. そもそもソフトウェア工学がどれだけの領域を含むものなのかについても, 明確な境界が定義されているわけではない. 科学的なスタンスを重視する研究領域もあれば, より工学的な側面, あるいは経験的な側面を重視する研究領域もある. そうした研究領域ごとに評価の方法が異なり, またよく行われる評価の常とう手段も異なる. こうした多様性が, 評価の議論を難しくしている. このことは, 同じ論文であっても査読者の持つ研究領域や問題意識によって, 評価が大きく分かれる危険性をもたらす.

8.2.2 応用局面の多様性

ソフトウェア工学研究はソフトウェアやソフトウェア開発上の問題を解決するために行われる. しかしながらソフトウェアの世界とひとくちにいっても, 極めて多様であり, その研究がどういう問題のセグメントに対してどのような貢献をするものであるかというスタンスを明確にしなければ, 研究評価が困難になる. さらに, ターゲットとして設定したセグメントがあまりに特殊な個別事情に支配されているものだと, その設定そのものの妥当性を主張することはできない. 実際に検討するターゲットは具体的なインスタンスであっても, そこに内在する普遍性や共有すべき問題を明らかにしなければ, そもそも評価すること自体の意味合いが薄れてしまう.

8.2.3 評価観点の多様性

評価を行う際の立場や観点も多様である. 例えば理論的に正当性を証明したりシミュレーションによって妥当性を確認したりするなど, その理論や技術の世界の評価観点で評価する立場がある. 一方, 実際の開発に適用することでその有効性や妥当性などを評価するという立場もある. また同じ立場でもその評価の観点は多様である.

前者の場合は, その世界での評価観点が, ソフトウェア工学的に妥当なものであることが重要であるし, 後者の場合は, その理論や技術が実際の開発上の成功とどのような因果関係を持つのか納得性のある議論をする必要がある. こうした評価観点は, つまりはなぜ評価を行うかという評価の目的につながる問題であり, 評価を行う状況や, 論文投稿先のコミュニティの問題意識などに即した評価観点の設定が重要となる.

8.3 評価の枠組み

こうした多様性の中で研究評価を行うためには, その評価の枠組みを明確にすることが重要である. 以下, それに関わるいくつかの点について指摘する.

8.3.1 問題の構造化

ソフトウェア工学研究の大目的が、ソフトウェア開発をより価値に結び付く活動へと高める、といったことであるとすると、個々の研究をその大目標の達成度合いに直接照らして評価することは極めて困難である。したがって、個々の研究領域がそうした大目的とどのような関わりを持っており、そのためにその研究領域ではどのような問題設定を行い、それをどのような基準で評価しようとしているのか、そうした問題構造を明確にすることが重要でとなる。

もちろんこうした大目的は一般に曖昧で様々な捉え方があるため、8.2.2 で指摘したターゲットセグメントの明確化の議論と同様に、妥当な目標を設定し、納得性の高い問題設定をする必要がある。一般に、個々の理論や技術と最終的な大目的との間の因果関係は複雑であり、また様々な要因が関係するために、特定の問題構造のみが唯一正しいものであると主張することは困難である。したがって、設定した問題やその評価基準が妥当であることを納得してもらえることが重要である。問題の構造化は個々の研究分野にとっての最重要課題のひとつと考える。

8.3.2 評価パターンの確立

研究の活性化のためには、個々の研究分野での問題設定や評価基準にもとづき、基本的な評価パターンを確立することが有用と考える。特定の研究分野での同様な研究は、同一の枠組みで評価することによって、それぞれの比較が容易になる。ソフトウェア工学の研究は必ずしも定量的な尺度だけで評価できない場合も多いだけに、こうした一定の評価パターンの共有は研究者自身にとっても、他の研究成果を理解しようとする人にとっても有益であると考えられる。

8.3.3 問題設定の共有

問題設定や評価基準が妥当なものであるかどうかの主張や議論は大変であるため、そうした問題設定や評価基準を個々の研究者が独自に設定することは困難かつ混乱を招く。したがって、それらを相互に議論・共有し、その共有された枠組みで研究を評価することが、建設的な方法である。

例えば欧米では研究コミュニティができると、ワークショップなどの場を通じて議論を重ね、そうした過程で問題設定や評価基準の共有などが行われていく。しかしながら日本の研究者は欧米の研究者とこうした議論をする機会が一般に少なく、結果として国際的に理解されづらい問題設定を独自に行い、評価されづらい状況を作ってしまう危険性を持っている。

8.4 おわりに

評価が難しいとは、評価の目的や観点が不明確な場合、具体的な評価の方法がはっきりしない場合、あるいはその方法にそった評価の実施が困難な場合など、様々な状況がありうる。

目的や観点に関しては前述したように多様性があり、その研究コミュニティとして

の共有が必要であろう。また論文を通すという観点からは、欧米のコミュニティでの議論を理解しておくことが重要である。方法に関しては評価パターンの整備などが有用であろう。実施については、例えばすべての論文が実開発に適用したエビデンスを求められては機能しない。評価パターンの提示する評価に要する時間や工数は、現実的なものでなければならない。

研究の評価という議論が必要なことは、ある意味ソフトウェア工学という領域が未成熟であるからだという指摘もある。しかしながらコンピュータが使われだして約 60 年、ソフトウェア工学が出現して 40 年という期間、情報社会は急激な発展や変化をしてきたことを考えると、それを安易に批判はできない。むしろその間に生み出された数多くのソフトウェア工学の研究成果をポジティブに受け取るとともに、ソフトウェア工学を今後さらに健全に発展させるために、この評価議論があると考えたい。

9. 運営に関する課題

本年度のワークショップを計画するにあたり、昨年度に比較してほぼ 2 週間早いペースで進行させて頂いた。結果的に、若干のスケジュールのブレはあったものの、余裕を持って準備を進めることができた。

ただし、本年度は前年度を上回る 90 名の参加があったことから、参加者数の増大に伴う各種の作業量も増大した。現在、実行委員長の業務はほぼすべて手作業で行っている。メールをベースにした参加者とのやり取りは相当な作業量であった。今後、何らかの形で支援システムを構築すること検討するべきだ。

また、準備の早い段階で実行委員（ローカル担当）や前任者と顔合わせを行い、打ち合わせを行っておく必要性を感じた。今回はメールによる打ち合わせのみであったため、細かい情報の伝達がうまくいかなかったこともあったと感じている。

その他、申し込み時に宿泊に関する情報も送ってもらうようにする、セッションの紹介原稿を早めに依頼する、原稿のフォーマットについてより徹底する、といったマイナーな改善点もあるので、次年度の担当者になるべく経験を引き継ぎたい。

10. おわりに

以上述べてきたとおり、本ワークショップは全体として成功裏に終わったことを報告する。本ワークショップをきっかけとして、今後ともますますソフトウェア工学の研究が発展することを切に願う。

謝辞

この場を借りて、不慣れな実行委員長を支えて頂いた元実行委員長を始め、参加者、全ての関係者の皆様に深く感謝致します。