

ウィンターワークショップ2006・イン・鴨川参加報告

満田成紀^{†1} 沢田篤史^{†2} 鷺崎弘宜^{†3}
山本里枝子^{†4} 吉田 敦^{†1}
門田暁人^{†5} 野中 誠^{†6}

2006年1月に開催されたウィンターワークショップ2006・イン・鴨川の概要と、ワークショップで議論が行われた五つの討論テーマ(組込みソフトウェア、ソフトウェアパターン、サービス指向、ソースプログラムの書換えとその基盤技術、実証的アプローチ)について報告する。

Report on Winter Workshop 2006 in Kamogawa

NARUKI MITSUDA,^{†1} ATSUSHI SAWADA,^{†2} HIRONORI WASHIZAKI,^{†3}
RIEKO YAMAMOTO,^{†4} ATSUSHI YOSHIDA,^{†1} AKITO MONDEN^{†5}
and MAKOTO NONAKA^{†6}

This paper describes an overview of Winter Workshop 2006 in Kamogawa, January 2006, and reports on five discussion themes: Embedded software, Software pattern, Service oriented architecture, Source program transformation and its base technology, and Empirical approach.

1. はじめに

ソフトウェア工学研究会では、1997年1月以来、参加者同士の議論を中心としたワークショップを定期的に開催している。そこでは、その時々に応じて、ソフトウェア工学分野において注目を集めている話題をワークショップテーマとして取り上げ、テーマを絞った集中的な議論を行っている。

2006年は1月26日から二日間にわたり、千葉県鴨川市にある東洋大学鴨川セミナーハウスで開催された。昨年から継続して設定された「組込みソフトウェア」「ソフトウェアパターン」「サービス指向」の3テーマに加え、新たに「ソースプログラムの書換えとその基盤技術」と「実証的アプローチ」の2テーマが設定された。各テーマそれぞれ、当該分野に対する造詣の深い討論リーダーのもと、現状の認識をはじめとして、今何を行うべきか、そのために解決すべき課題は何か、今後どのような方向へ進めるべきか、といった視点で

討論を行った。

今回初めての試みとして、テーマ同士の関係性についてより具体的に議論することを目的に、二つの交流セッションを企画した。「組込みソフトウェアのパターン」「BPMのための実証的アプローチ」という横断的なテーマを設定し、参加者同士が各々の立場から期待や可能性について意見を交換した。

以下、五つの討論テーマと交流セッションについて議論の内容を報告する。

2. 組込みソフトウェア

本ウィンターワークショップでの組込みソフトウェアセッションは、今回で六回目の開催となる。今回は狭い意味の組込み技術だけにとらわれず、ソフトウェア工学およびその周辺技術を視野に入れた横断的な議論により、「組込みソフトウェア工学」の確立に向けた検討を行うことを大目的に掲げて参加者を募集した。その結果、次の7名の参加者(うち大学7、企業1)が討論に加わった。

青木 利晃 (北陸先端科学技術大学院大学)

紫合 治 (東京電機大学)

野中 誠 (東洋大学)

東 篤子 (南山大学)

沢田 篤史 (京都大学/情報通信研究機構)

†1 和歌山大学

†2 京都大学

†3 国立情報学研究所

†4 富士通研究所

†5 奈良先端科学技術大学院大学

†6 東洋大学

渡辺 晴美 (東海大学)

久保秋 真 (テクノロジックアート)

セッションでは、まず参加者全員がそれぞれのポジションを発表し、議論を行った。この過程で、組込みソフトウェアにおいて重要視されるべき性質として、(1) 環境 (外界) に対するリアクティブ性、(2) ソフトウェアが組込まれる製品に対するビジネス要求、(3) ハードウェアも含めた製品開発のプロセスとマネジメント、などが挙げられたが、今回は (1) のリアクティブ性に関するテーマを議論の中心とした。これは、先に掲げた本セッションの目的に対し、後二者のテーマでは工学的技術についての議論になりにくいという判断による。

何らかの製品に組込まれて利用されるソフトウェアは、製品と実世界との直接的な関わりを扱うことができなければならない。つまり、誰がどう使うか分からない条件のもとで、あらゆる入力に反応することが求められる。この性質を実現するために、組込みソフトウェアは、その内部に外部環境を反映した動作や構造を持ち、外部環境をシミュレートすることができなければならない。これをここではリアクティブ性にとらえている。以下、各参加者のポジションをこの観点から振り返ってみる。

状態遷移図のモデル検査を有効に機能させるには、モデルの意味論とイベント送受仕様とを明確にする必要がある、という青木氏の主張には、システムと外部とのやりとりの仕様、つまり環境モデルの構築が重要であることが示されている。また、組込みシステムの設計をシステムが間接的に制御する実体の分析に基づいて行う、という柴合氏の方法論は、外部環境を広くモデル化する必要があるという主張を示したものである。野中氏のポジションでは、要求分析・製品計画段階から、環境に起因する要求をとらえる、つまり、ソフトウェア要求の出处として不特定多数の利用者が様々な行動を行う外部環境は重要であることが報告された。また、沢田は、家電連携サービスの構築支援の一環として、様々な領域の機器仕様の用語集を作成していることを報告した。これは家庭の構成要素としての家電仕様を通じてホームネットワーク環境のモデル化によるサービスの設計支援を実現するものである。多元的アスペクトにより車載システムの部品間の関係を抽出し、構造設計を支援するという東氏のポジションでは、部品間関係の抽出に用いるアスペクトが車載システムのおかれた環境の要求であり、結果抽出された部品群や個々のプロダクトの構造は、環境要求を反

映した外部環境のモデルであるといえる。

ロボットコンテストについての話題もポジションとして提供された。ロボットコンテストの教育効果を高めるには、外部との間である程度のボリュームと密度をもったやりとりを行うシステムの開発を行う必要があるという渡辺氏のポジションは、外部環境の分析やモデリングが組込み開発では重要であるという主張につながる。久保秋氏は、MDD ロボットチャレンジのモデル審査経験を通じて、組込みソフトウェアのためのモデリングについて考察した。良いプロダクトを開発するためには、制御対象やシステム自身の物理モデルを綿密な分析に基づいて構築する必要があるという点であり、この主張も外部環境のモデリングが重要であるという点につながる。

このように、本セッションでは、外部環境のモデリング、外部環境のシミュレーション、リアクティブ性に主な焦点をあてて組込みソフトウェア工学技術を議論した。利用者や実環境と直接的な接触のある組込みソフトウェアでは、このリアクティブ性を考慮することが特に重要であり、外界のモデル化の困難さが組込み開発における課題につながっているともいえる。ここで得られた問題意識や今後の検討課題は、ESS2006、MDD ロボットチャレンジ 2006 を等を通じて継続的に議論していきたい。

3. ソフトウェアパターン

3.1 目的と経緯

本セッションでは、ソフトウェア開発において繰り返し発生する問題と解決策・フォース等をまとめたソフトウェアパターンの特性、および、パターンの運用・周辺技術の特性について議論した。参加者は、驚崎弘宜 (国立情報学研究所) を討論リーダとして、下滝亜里 (大阪産業大学大学院)、羽生田栄一 (豆蔵)、河合昭男 (オブジェクトデザイン研究所)、久保淳人 (早稲田大学大学院)、中山弘之 (早稲田大学大学院) の6名であった。また、一部の議論について1名のオブザーバが参加した。

ソフトウェア工学研究会パターンワーキンググループは、これまでウィンターワークショップに毎年パターンセッションを設置し、継続的にパターンおよびパターン技術の本質を明らかにしてきた。経緯を以下に示す。

- 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¹⁾。これは、パターン活動プロセス²⁾を構成する利用活動の評価・適用工程に

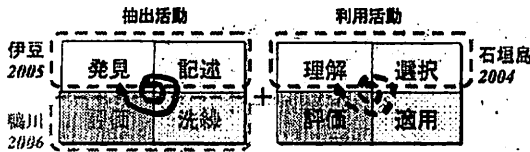


図1 パターン活動プロセスとワークショップテーマ

ついて得られた知見である(図1参照)。

- 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なことや、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³⁾。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。

以上の成果により抽出・利用活動の前半部について主要な特性が明らかになったものの、後半部について依然として不明であった。そこで本セッションでは、ライターズワークショップの実施により抽出活動の洗練・評価工程の特性を明らかにすることを試みた。また、ポジションペーパー発表を起点とした議論を通じて、効果的なパターン技術や周辺技術のあり方を明らかにすることを試みた。

3.2 議論成果

(1) ライターズワークショップ

ライターズワークショップとは、新規に提案されたパターンを複数名で推敲・洗練する活動⁴⁾であり、その形式は集団による詩作活動に類似する。本セッションでは2つのパターンランゲージを取り上げて以下の成果を得た。

- 「業務システム開発における要求獲得パターンランゲージ」(提案: 鷲崎ら)は、顧客から要求を円滑に獲得する活動を支援するパターン集合からなる。ライターズワークショップにより、抜けている新規パターンの追加、および、不適切なパターンの削除を行い、より有効性を高めた(図2参照)。
- 「モノ・コト分析パターン言語」(河合)は、適切なドメインモデルを効率よく作成するためのパターン集合からなる。ワークショップにより、より適切なパターン名への修正などを行い、有効性を高めた。
- パターンランゲージが備えるべき特性として、扱う範囲設定が妥当なこと、および、パターン間の関連が明示されていることを明らかにした。

(2) パターン関連技術の特性

ポジションペーパーを基に、パターン活動支援技術や

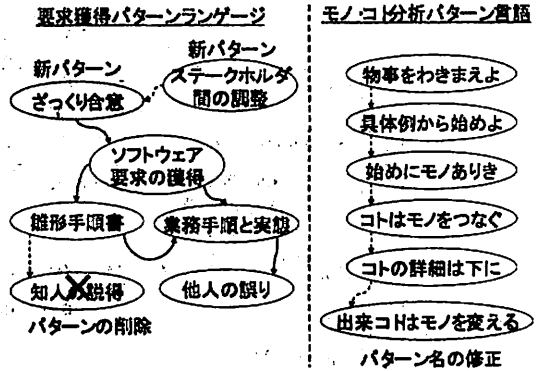


図2 ライターズワークショップ成果(各楕円はパターン)

周辺技術の特性を議論して以下のように明らかにした。

- 「非対称関連によるソフトウェアパターンの粒度測定法」(発表: 久保ら): パターン間の非対称関連から各パターンの粒度を測定する手法が提案された。今後の展望・コミュニティへの要請として、パターンの粒度の意味や性質の明確にする必要があることを明らかにした。
- 「検索のためのソフトウェアパターン重要度とその正規化」(中山ら): パターン間の関連に基づいて各パターンの重要度を正規化して算出し、検索に活用する手法が提案された。展望・要請として、パターンの(検索などを活用した)学習過程を明確にする必要があることを明らかにした。
- 「ベンチマークとしてのデザインパターンの使用に向けて」(下滝): 設計手法や実装言語の比較評価のためのベンチマークとしてデザインパターンを捉える考え方が提案された。展望・要請として、各パターンにおける実践的サンプルの明示が必要なことを明らかにした。
- 「ビジネスアーキテクチャの(4+1)×1ビュー」(羽生田): 情報システムが扱う業務や組織のアーキテクチャを多次元で捉える枠組みが提案された。展望・要請として、多様なパターンの、工程以外の軸による分類の必要性を明らかにした。

3.3 現状と将来の展望

成果の全ては⁵⁾において公開している。以上の議論を通じて、パターンについての現状として、デザインパターンは十分に普及しているものの、上流工程における経験のパターン化は未成熟であることを再確認した。

続いて、課題を以下のように明らかとした。

- 工学への道筋: パターンのリファレンス実装が必要
- パターン化の促進: 豊富な適用経験を分析する枠

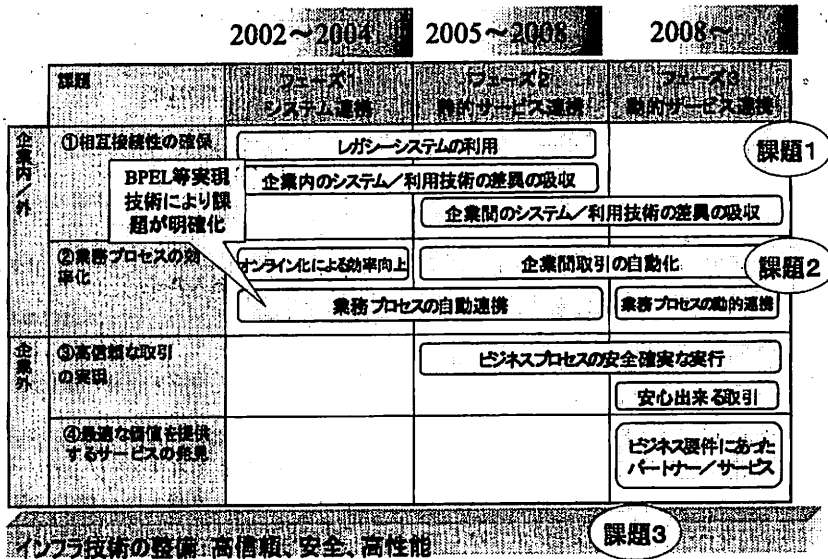


図3 サービス関連技術の現状と課題 (2006年版)

組みとパターン化を奨励する取り組みが必要

- 領域の拡充: 組み込み等のパターン発掘不十分な領域について開発上制約の名付けと整理が必要

今後は、これらの課題について、パターンWG内外における議論と実践を通じて取り組み、開発・組織活動におけるパターンの活用と支援技術の発展を目指す。

4. サービス指向

4.1 背景と討論テーマ

サービス指向技術は、実装を隠蔽した“サービス”を組み合わせることで変化に強い柔軟なシステムを構築するITアーキテクチャを提供する。当研究会では、2002年のウィンターワークショップから継続して、Webサービスやサービス指向に関する議論を重ねてきた。

これまでの議論と同様に、今回も幅広い討論テーマを抽出するよう、分散システム統合を実現する技術に関して幅広く論文を募集した。

本セッションには以下の4種類の技術に分類される7件の論文が提出された。

- サービス指向開発における開発支援技術
 - 業務プロセスからの開発自動化
 - セキュリティ設定の自動化
 - 複合サービスの生成
- サービス実行環境
 - フューチャの再構成
- 動的なサービス連携技術
 - 価値に基づくサービス選択

- Webアプリケーションの開発事例、Webアプリケーションの性能

議論テーマとして、まず上記の分類の観点で技術課題と取り組みを議論した。また、実装技術であるWebサービスの実現の現状、残課題、新たな課題を議論することで、今後取り組むべき技術課題を明確にした。参加者は大学6名、企業3名の計9名で、実践面・学際面の両面から議論できた。

4.2 討論の成果

(1) 技術の現状

Webサービスは2002年当時、公開・登録・検索する規約(UDDI: Universal Description, Discovery and Integration)、インタフェース仕様の規約(WSDL: Web Services Description Language)、XMLに基づくトランスポート独立なプロトコルSOAP(Simple Object Access Protocol)が整備されたことで、インターネット上の次世代の分散コンピューティング環境として注目された。そこからの変化も踏まえて、現状と課題を整理したのが図3である。現在は公開UDDIの閉鎖など、利用ストーリーの変化したが、SOAP、WSDLは基盤技術として根付いた。それを背景に、技術は企業内連携、セキュリティまでを実現した状態で、フェーズ2に入っている。具体的な実現時期を定義できたことも成果と考える。

(2) 課題

以下が課題として明確になった。今後、技術とビジネスの2方向からのアプローチが必要である。

- (1) 非機能要件：現状は Transaction など主要なものも未だに作業中。ベンダはそのサポートに、ユーザは利用のための設定に、多大な工数が必要。
- (2) トップダウンの難しさ：ビジネスの状況に応じて、トップダウンに「業務プロセス」を定義しないと SOA のメリットが活きないが、現状は企業トップからのアプローチが困難。
- (3) 高信頼なインフラ技術：従来技術と同様に性能等、固有な技術にサービスの動的検索等がある。

5. ソースプログラムの書換えとその基盤技術

5.1 背景と目的

ソフトウェアの開発支援として、単にプログラムを解析するだけでなく、リファクタリングといった、解析結果に基づいたソースプログラムの書換えが支援に含まれつつある。また、ソースプログラムを XML へ変換することで、XML の書換え技術と連携させる試みも広く行われるようになってきている。しかし、それぞれの研究者が独自に書換えを試みたり、応用を考えたりしている状況であり、研究者間での知見や課題を共有する機会は少ない。そこで、本セッションは、それぞれの研究者が持つ技術や問題意識などを紹介してもらおうと同時に、解決すべき課題や今後の発展の方向について議論を行なった。

本討論グループには、9 件の論文が提出され、参加者は 10 名であった。論文は以下の通り 5 種類に分類でき、書換えに関わる様々な技術や問題点について広範に議論することができた。

- 事例紹介 (2 件)
- 書換えパターン記述 (2 件)
- ソースプログラムの変更履歴の応用 (2 件)
- ソフトウェアリポジトリのあり方 (2 件)
- 書換えの難しさの分析 (1 件)

5.2 討論の結果

(1) 書換えの目的

書換えの目的は何かといった観点で議論を行い、「下流工程にだけに留まらずに、上流への反映が必要」、「意味的に非等価な変換の支援すべき」といった意見が出された。本来、上流における分析・設計を十分に行なっていれば、下流は単に仕様に従ってプログラムの記述や改変を行う。しかし、現実には、実際にプログラムとして動かしてみ初めてわかることもあり、下流から上流への流れも存在する。例えば、アスペクト指向におけるソースプログラムからのコンサーンの抽出が該当する。ソースプログラムに対する書換えも、

単にソースプログラムだけを変換して下流に留まるのではなく、書換えることで上流のプロダクトやプロセスへフィードバックが起すなど、広い視野で書換えの活用を考えるべきである。

また、書換えを考える場合には、意味的に等価な変換を考えることが多い。例えば、リファクタリングは適用の前後で意味を変えないのが前提となっている。しかし、プログラミングにおける知的創造的活動を支援するためには、非等価な変換についても対象を広げるべきである。

(2) 書換えの難しさについて

プログラムの解析は論理的に綺麗な範囲に留まっている限り、現実のソースプログラムに適用することが難しく、その結果、解析精度が低くなる。解析が困難である以上、解析結果に基づく書換えはさらに困難になる。そのような状況の中で、書換えを支援していくためにはどうすれば良いのか議論を行い、ツールの仮定や限界を明示すること、全自動の書換えを諦めること、テストの自動化技術でカバーすること、書換え機能を小さく制限することなどが意見として出された。

(3) ツールの記述指針の必要性

ツールの書き方に関して、やみくもにツールを書くのではなく、指針あるいはスタイルが必要であるとの意見が出た。特に、宣言的記述 (パターン記述) と手続き的記述 (メタプログラミング) の使い分けや組み合わせ方を具体化していくことが必要である。AWK のようなツールが必要であろうとの声が多かった。

(4) ツール構築環境

ツール構築の経験者から、「常に正しい解析結果と不完全なソースプログラムの解析が必要」との意見が出された。また、XML を使う機会が増えているが、XML の有利な点として、「抽象構文木より柔軟性の高いデータ表現」が挙げられた。その他、書換えを記述は、XSLT では難しく DOM の利用が必要であることや、専用の API が必要であり、API の仕様は環境を提供する人のセンスが問われるといった意見も出された。今後の課題としては、RSS などウェブの既存技術を応用した分散環境も必要である。

(5) 今後の発展方法

今後は、ツールや環境の軽量化を考える必要がある。書換えの基盤となる解析技術については、精度の向上を無闇に求めず、ある程度の割り切りが求められるとの意見が出た。また、精度を切替えることで、ツールの利用者に選択を与えることも必要である。

同様に、書換えについても完璧さは求めず、エラーが出ることを容認すべきである。その代わり、エラー

を発見したり、あとで替换を元に戻せるようなセイフティネットが必要であるとの意見が出た。

6. 実証的アプローチ

6.1 背景と討論テーマ

ソフトウェアの大規模化、多機能化への要求が増大する一方、市場競争の原理により開発期間の短縮、及び、高信頼性の確保への圧力がますます強まっている。このような厳しい状況の中、ソフトウェア開発プロジェクトを成功に導くためには、プロジェクト内で起こっていることを定量的に評価し、現状を的確に把握し、問題点の検出、現場へのフィードバック、データの蓄積と再利用を行うことが必須となる。

本セッションは、このような、ソフトウェア開発に関するデータを計測、分析し、改善を行うという実証的ソフトウェア工学に関する話題を広く募集し、産学双方の立場から現状の課題と今後の展望について議論した。参加者は、佐藤慎一(NTT データ)、大杉直樹(奈良先端大)、門田暁人(奈良先端大)、阿萬裕久(愛媛大)、戸田航史(奈良先端大)、松村知子(奈良先端大)、肥後芳樹(大阪大)の7名であった。

討論は、参加者の持ち寄ったポジションペーパーに基づいて、「計測・分析のゴール」「データ計測あり方」「データ分析のあり方」という3つの観点から行われた。

6.2 討論の成果

(1) 計測・分析のゴール

データの多くは人手により計測されるため、質の高いデータを集めるには、データ計測者をモチベートするゴールが必要である。ゴールの設定は、データ計測者の持つ課題や問題意識に基づいて設定することが望ましい。なお、開発者(計測者)、管理者、経営者それぞれのゴールは互いに異なるため、注意が必要である。

解決すべき課題(ゴール)と、計測データの関係は、明確になっている必要がある。一つの方法として、GQM(Goal Question Metric)アプローチが利用できる。

(2) データ計測のあり方

コンテキスト(開発プロセス)を考慮した計測が重要である。ソフトウェア開発は非常に多様であるから、あらゆる開発プロジェクトのデータを収集したとしても、分析が困難となる場合が多い。開発プロセスに関して何らかの前提を決めて、その前提に合うプロジェクトのみから計測することが望ましい。これはCMMIの考え方と同じである。

効果的な計測を行うには、ユーザ(発注側企業)からの働きかけも重要である。データ計測を行うには追

加のコストを要するため、余裕のない開発現場では計測が満足に行われない場合がある。そのような場合においても、発注側からの条件としてデータ計測が義務付けられていれば、計測は行われる。

いくつかのデータは、開発支援ツールから自動収集できる。例えば、構成管理ツールや、障害管理ツールである。ただし、質の高いデータを得るためには、これらのツールの運用ルールを明確に決めておく必要がある。例えば、障害報告をクローズする際には、対応するソースコードの名前を必ず入れる。構成管理ツールには、1日1回必ずチェックインする。ファイルの命名規則を決めておく、などである。

(3) データ分析のあり方

データ分析に先立ち、データを整理する(クレンジングする)ことは極めて重要である。信頼性の低いデータを除外したり、データの定義をそろえたり、特異点となるデータを除去したりといった作業である。データのクレンジングを行わずに、やみくもに統計手法を用いることは慎むべきである。

データ計測のみならず、分析においても、開発コンテキストを考慮することが必要である。例えば、規模(人員数)が異なると、開発プロセスも大きく異なる。分析目的に応じて、どのコンテキストの開発データに着目すべきかを見極める必要がある。

分析者の興味と開発現場の興味は異なる場合があるため、分析の視点は、現場へのフィードバックを考慮して行う必要がある。例えば、コードクローンの分析において、現象面から定量的な分析を行うことも重要であるが、何が良くて何が悪いのかについて、現場の開発者に指針を与えるような結果も必要である。

7. 交流セッション

7.1 組込みソフトウェアのパターン

本セッションでは、「組込みソフトウェアのパターン」をテーマに、組込みソフトウェアとソフトウェアパターンの2つの討論グループに、ソースプログラムの替换えとその基盤技術討論グループからの参加者を交えて、自由に議論を行った。

組込みソフトウェア討論グループからは、プロダクトの側面に限定した場合、組込みソフトウェア固有の設計方式などベストプラクティスは存在するであろうが、パターンとして体系的に整理できる段階にないのではとの、やや否定的な意見も出された。これに対して、ソフトウェアパターン討論グループおよび参加者から、議論の対象を限定し過ぎずにパターン化できる部分を考えるべき、パターンとして記述してはじめて

議論や評価が実施できるなど、前向きに取り組むことの必要性が主張された。また、組込みソフトウェア開発における外界モデリングは業務系システム開発におけるビジネスパターンと類似性があるのではないかと、組込みソフトウェア開発ならではの厳しい制約のなかでどのようなパターンが適用されているのか調査することは有意義であろう、などの意見が出た。組込みソフトウェアパターンの記述・発見・適用実験など、産業界から事例を集める以外にも、ロボットチャレンジという研究会主体で実施できる組込みソフトウェア開発実践の場を活用する方法も考えられよう。

交流セッション自体が初めての試みということもあり、やや成り行き任せの議論になった側面も否めないが、組込みソフトウェアのパターン化に向けた建設的な議論を行うことができた。より効果的な議論を行うためには、例えば、ロボットチャレンジのモデルを題材にしてパターン記述した結果について議論するなど、事前の準備が求められる。

7.2 BPMのための実証的アプローチ

サービス指向アーキテクチャを積極的に活用する技術としてBPM (Business Process Management) が取り上げられている。また、実証的アプローチではソフトウェアプロセスを対象に研究が行われているものが多い。そこで「プロセスつながり」という企画側の勝手な思い込みにより、「BPMのための実証的アプローチ」というテーマで交流セッションが行われた。サービス指向と実証的アプローチの二つの討論グループを中心に、他グループからの参加者も交えて意見交換がなされた。

まず、サービス指向グループからBPMの紹介が行われた。その中で、BPMの中心にあるモニタリング技術と業務プロセス管理への応用方法が説明され、さらに、実証的アプローチへ期待することとして、モニタリング情報に対する統計的な処理技術の確立について述べられた。実証的アプローチグループからは、研究コンセプトの説明の後、ソフトウェアプロセスを対象とした研究成果の一部が紹介された。BPMへの応用については、業務プロセスとソフトウェアプロセスの違いはあるが、人によるプロセスを対象とした研究における問題点をいくつか共有できるのではないかと意見が述べられた。その後、人によるプロセスの管理における現実的な問題がいくつか会場から紹介され、その解決方法についての意見が交わされた。

8. おわりに

本ワークショップに参加し活発な討論を行っていた

だいた皆様に感謝する。また、ワークショップを通して寄せられた意見は、ソフトウェア工学研究会の運営委員会等で紹介するなど、今後のワークショップ実施に反映されるようにしたいと考えている。

参考文献

- 1) 松下誠 他: ウィンターワークショップ・イン・石垣島参加報告, 情処研報, 2004-SE-145, 2004.
- 2) 鷲崎弘宜 他: ソフトウェアパターン研究の発展経緯と最近の動向, 情処研報, 2004-SE-147, 2005.
- 3) 柴合治 他: ウィンターワークショップ 2005・イン・伊豆参加報告, 情処研報, 2005-SE-148, 2005.
- 4) 羽生田栄一 監修, パターンワーキンググループ 著: ソフトウェアパターン入門, ソフトリサーチセンター, 2005.
- 5) <http://patterns-wg.fuka.info.waseda.ac.jp/>