

ウィンターワークショップ・イン・石垣島参加報告

松下 誠^{†1} 平山 雅之^{†2} 青木 利晃^{†3}
廣田 豊彦^{†4} 原 裕貴^{†5} 権藤 克彦^{†6}
鷺崎 弘宜^{†7} 細谷 竜一^{†8}

2004年1月に開催された、ウィンターワークショップ・イン・石垣島の概要と、ワークショップで議論が行われた討論テーマについて報告する。

Report on Winter Workshop in Ishigakijima

MAKOTO MATSUSHITA,^{†1} MASAYUKI HIRAYAMA,^{†2}
TOSHIAKI AOKI,^{†3} HIROTAKA HARA,^{†5} KATSUHIKO GONDOW,^{†3}
HIRONORI WASHIZAKI^{†7} and RYUICHI HOSOYA^{†8}

This paper describes about overview and discussion themes on the winter workshop in Ishigakijima, January 2004.

1. はじめに

ソフトウェア工学研究会では、1997年より、年1回の割合でワークショップを開催している。毎回、ワークショップでは事前に討論テーマを設定した上で参加者を募集し、集中的な議論を行う場として活用されている。

今回のワークショップは2004年1月29日から2日間にわたり、沖縄県石垣島にある石垣島全日空ホテルにおいて開催された。設定したテーマは、組み込みシステム、要求工学、Web サービス技術、新世代ソフトウェア開発ツール、ソフトウェアパターンという5つである。各テーマは近年注目されている分野であり、今後の発展も期待されているものである。今回の参加者はあわせて34名であった。

ワークショップの進行は以下の通りである。まず、参加者全員が集合して今回の討論テーマについて確認を行った後、ひき続いて各テーマごとにわかれて議論を行った。最後に、再度全員で集合して、各テーマごとに討論のまとめを発表し、質疑を行った。

以下、各討論テーマごとに、本ワークショップにお

ける議論の内容等について述べる。

2. 組み込みシステム

組み込みソフトウェアセッションでは、まず、参加者のポジションについて発表し、それぞれについて議論を行った。参加者の興味は、領域で分けると、「車載システム」と「家電」、技術で分けると、「設計技術」と「テスト/検証」であった。個々の内容については、ワークショップの論文集を参照していただきたい。次に、組み込みソフトウェアの設計技術に注目して議論を行った。

組み込みシステムの分野では、従来から、実時間性、メモリ制約などのハードウェア制約の充足性が主な議論の対象であった。近年、組み込みシステムの大規模化、多様化に伴い、組み込みシステムの定義や議論の対象が変化してきた。しかしながら、組み込みシステムと呼ばれるものは多種多様であり、通常のシステムやソフトウェアとの境界が曖昧になってきたため、明確にそれら特定することが困難なのが現状である。参加者による議論の結果、従来からこの分野で議論されてきた実時間性やメモリ制約などの実装技術より、それらにより抽象的な観点からとらえる設計段階の技術が重要であるという結論に達した。主な理由は、第一に、実装技術は従来から十分に議論されており、かつ、それらは、現在でも適用可能であること、第二に、実装技術はそれぞれの現場で培われたものがあり、それらを変更することは困難であること、第三に、現在の組み込みシステムの問題点の根源の1つはその大規模化であ

†1 大阪大学

†2 東芝

†3 北陸先端科学技術大学院大学

†4 九州産業大学

†5 富士通研究所

†6 東京工業大学

†7 国立情報学研究所

†8 東芝ソリューション

り、それを管理・整理するためには、実装より抽象的な視点、すなわち、設計段階の決定のインパクトが大きいこと、などである。

では、組み込みソフトウェアの設計とは何であるか?、この疑問に対して、多数が納得する明確な答えを我々は持っていない。抽象的ではあるが、この分野において、注力すべき対象や研究の方向性を決定する重要な点である。答えを探る糸口は、一般のソフトウェア設計技術を、いかに組み込みソフトウェア開発に適用するか、その際、どのような困難があるか、について考えることである。そこでワークショップでは、まず、注目すべき設計技術について列挙した。その結果、MDA(Model Driven Architecture)、モデル検査、テスト、アーキテクチャ記述・評価、再利用が挙げられた。次に、これらの技術を用いた組み込みソフトウェア開発のストーリーを作成した。具体的には、それらの適用により得られたアーティファクトや結果の組み合わせ方、適用される順序、役割分担、問題点/改良点などである。この一連の議論では、「組み込みソフトウェアの設計とは何であるか?」の答えには到達できなかった。しかしながら、組み込みソフトウェアの設計の観点から、それぞれの参加者の研究課題や仕事内容において注目すべきいくらかの点が明確になった。次回以降では、この議論に基づいた具体的な事例が出てくるものと期待している。

「組み込みソフトウェアとは何であるか?」や「組み込みソフトウェアと一般のソフトウェアの違い」について質問されることが多い。今回のワークショップでは、組み込みソフトウェアの設計について焦点を当てて議論を行ったが、明確な答えに到達していない。最初にも述べたが、組み込みシステム、組み込みソフトウェアについて長い議論がされてきたが、現在、我々が対象にすべきものは、従来から議論されつづけて来たものとは異なる。では、現在の組み込みソフトウェアについて明らかにする必要があるが、ウィンターワークショップでは、今回を含め、組み込みソフトウェアに注目したセッションを3回開催して来たが、それで結論の出るものではなかった。現在の状況では、組み込みソフトウェアの定義や分類を行えるほど明らかで安定した分野では無くなってきた、ということである。

過去3回のウィンターワークショップにおいて、1回目は、現在、組み込みシステムと呼ばれるものの列挙、技術的なものに限らず、ざっくりばらんな困っていること、課題について意見を交換した。2回目では、技術的側面、ビジネス/政治的な側面から課題を整理した。今回は、技術的側面の中でも設計に関する技術に議論の対象を絞った。そこで、次は、現在の組み込みソフトウェア開発の事例に関して議論をすべきである。このセッションは、組み込みソフトウェアワーキンググループの活動の一貫として行っているため、今年度のワーキンググループの活動として設計事例に関する紹介や

議論を行い、それらを踏まえて、次回のウィンターワークショップで、より具体的な整理や検討を行いたい。

3. 要求工学

3.1 参加者

要求工学セッションには、以下の8名が参加した(順不同)。中野武司(明治大)、廣田豊彦(九州産業大)、佐伯元司(東工大)、大西淳(立命館大)、新原敦介(東工大)、山田宏之(愛媛大)、中谷多哉子(エス・ラゲーン)、妻木俊彦(日本ユニシス)。

ソフトウェア工学研究会要求工学WGでは、ウィンターワークショップとは別に年2回ワークショップを単独開催している。今回は研究会主催のワークショップであり、オープンに参加を呼びかけたが、結果的にはWGのメンバとその関係者(学生)のみの参加となった。

3.2 討論内容

昨年(2010年)のウィンターワークショップ¹⁾に引き続き、要求品質とソフトウェア品質のマトリックスを討論した。このテーマは当初、昨年(2009年)のウィンターワークショップで完結する予定であったが、議論が白熱し、その後の2回の単独開催のワークショップ(5月都城、10月蔵王)でも完結せず、今回のウィンターワークショップに至った。

「要求仕様書の品質からソフトウェア製品の品質を予測する」ということを目指して、要求仕様の品質項目とソフトウェア製品の品質項目との間の相関関係を議論してきた。ここで、要求仕様の品質としては、IEEE830-1998に定義されている品質特性を用い、ソフトウェア製品の品質としてISO/IEC 9126において定義されている品質(副)特性を用いることとした。また、ソフトウェア製品は、要求仕様書とおり正しく開発されていることを前提としている。

一昨年末に、表1の中身が空白になっているものを、ウィンターワークショップ・イン・神戸の参加予定者10名に送付し、中身を事前に埋めてもらった。10名の返答を、(3点)、(2点)、(1点)、×(0点)で点数化し、平均値を計算した。その平均値を参照しながら、議論を重ね、○、△、□、×を定めた。

状況に応じて、参加者各自の回答も参照し、なぜそう考えたのかについて活発な議論がなされた。議論の結果、新たな分類(○:品質を計測しやすくなる)が登場したり、結論が一つにまとまらなかったものもあった。一つにまとまらなかったものは、意見の対立というよりは、「どちらもありうる」という雰囲気であった。

今回のワークショップでは、ソフトウェア製品の品質特性「保守性」の副特性である「安定性」と「試験性」に対して、要求仕様の8個の品質特性がどのように関わっているのか、一つずつ順に議論した。ソフトウェア製品の品質特性「移植性」に関しては、結果的にはす

表 1 要求仕様の品質とソフトウェア製品の品質

ソフトウェア製品の品質	機能性				信頼性			使用性			効率性			保守性			移植性				
	合目的性	正確性	相互運用性	標準適合性	セキュリティ	成熟性	障害許容性	回復性	理解性	習得性	運用性	時間効率性	資源効率性	解析性	変更性	安定性	試験性	環境適用性	設置性	規格適用性	置換性
要求仕様の品質																					
正当性	○	○	△	△	○	×	×	×	※	※	※	×	×	△	△	△	△	×	×	×	×
非あいまい性	○	◎	△	△	○	△	△	△	×	×	×	△	△	※	△	○	◎	×	×	×	×
完全性	○	◎	△	△	○	○	○	○	○	○	○	○	○	×	△	○	×	×	×	×	×
一貫性	△	△	△	△	○	△	△	△	△	△	△	△	△	※	○	○	×	×	×	×	×
ランク付け	◎	×	×	×	×	×	×	×	×	×	×	×	×	△	▲	×	△	×	×	×	×
検証可能性	△	○	△	△	△	△	△	△	△	△	△	△	△	○	●	△	◎	×	×	×	×
変更可能性	□	□	□	□	□	□	□	□	×	×	×	□	□	△	●	×	×	×	×	×	×
追跡可能性	□	□	□	□	□	×	×	×	△	△	△	×	×	◎	●	◎	◎	×	×	×	×

◎:強い相関がある ○:相関がある △:弱い相関がある □:品質を計測しやすくなる
 ×:相関はない ●:条件付○ ▲:条件付△ ※:○と×に意見が割れた

べて×（相関はない）となったが、特に副特性「環境適用性」については詳細な議論がなされた。

3.3 成 果

昨年のウィンターワークショップから今回まで1年間議論を継続し、表1に示す結果がまとまった。

今回のウィンターワークショップに引き続いて、本年4月に単独の要求工学ワークショップを開催した。そこでの議論の結果、今後はソフトウェア一般ではなく、特徴毎にソフトウェアを分類し、分類ごとに品質特性の議論をすること、また、議論を抽象的なものに終わらせないために、要求工学WGのメンバが具体的な事例を持ち寄り、それに基づいて議論を進めることとなった。

4. Web サービス技術

4.1 はじめに

Web サービス技術は、次世代のシステム間連携の標準技術として、セキュリティやトランザクションなどの基盤技術としての標準化が進むとともに、amazon.com など一部では実ビジネスでの利用も始まっている。

Web サービス技術の本質的なメリットは実装の抽象度を一段高めた点にあると考える。すなわちSOAP/WSDLといった標準化により、システム間のインタフェースをプラットフォーム非依存な形で抽象化することができるようになり、それによりシステム間の疎結合な連携を実現できるようになった点である。

しかしその一方、Web サービスが当初から掲げていたサービスの動的結合については、まだ実用化には遠い状況にあり、また Web サービス技術によりシステム開発そのものが劇的に改善するまでには至っていない。本セッションでは、Web サービス技術の実用化

に向けて、技術マップの再整理とシステム開発技術への影響について議論を行った。

4.2 参加者とポジションペーパー

参加者からは以下のような発表が行われた。

(1) フォーム自動記入のためのサービス統合方式 藤原（秋田大学）

フォーム自動記入のために構築した Web サービス / Web アプリケーションのラッピングによる動的サービス統合の実現方式と現状の Web サービスの課題について議論する

(2) Web サービス連携におけるシステム開発の自動化について 吉田、坂田（NTT データ）

SOAP/WSDL などの Web サービス技術とサービス指向アーキテクチャは、Web サービス連携という新たなシステム開発手法を可能にする。我々は Web サービス連携におけるシステム開発の開発プロセスについて、それを迅速化することを目的とし、開発工程の自動化を可能とする開発支援環境を提案する

(3) サービスブローカによる Web サービスの動的組み合わせと課題 青山（南山大学）

Web サービスの動的組み合わせのパターンと課題について整理し、サービスブローカによる仲介モデルを提案し、その実現、課題について議論する

(4) Web サービスによる動的結合技術の実現とシステム開発プロセスの変革 原（富士通研究所）

Web サービスが次に目指す技術として動的結合とシステム開発プロセスの変革について、特に B2B や ERP 等標準化が進んでいる領域での事例をベースとして、議論する

4.3 議論の要旨

ポジションペーパーの発表を踏まえて、「Web サービスの技術マップの整理」に焦点をあてて作業および議論を行った。

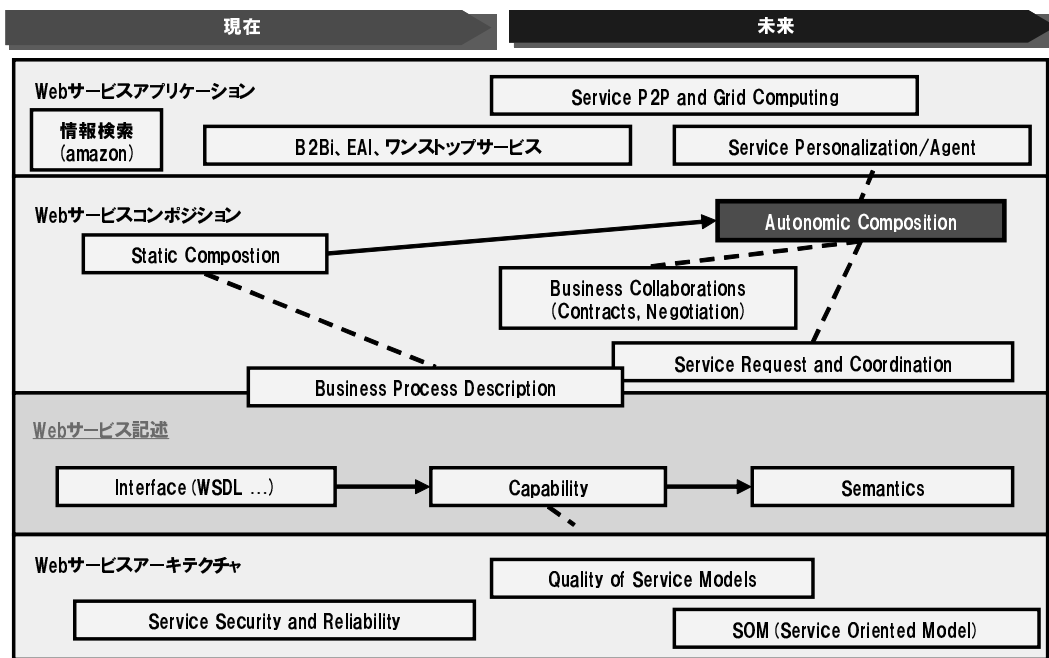


図 1 Web サービス技術の今後の方向性

Web サービスが目指す目標としては以下の 3 つがある。

- (1) プラットフォーム・実装への非依存性．これについては SOAP による相互接続検証などにより現在でもかなり実現されている．
- (2) 抽象化．従来はたとえば Java 言語のようにコンピューティングレベルでの抽象化であったが，Web サービスではコンテンツ（サービスの意味）レベルで抽象化を行う．
- (3) 動的なサービスの実行（late binding）．実行時でないと最適なサービスを選べない場合に有効であるが，まだ実用化の段階にはない．

Web サービスについては仕様の標準化のマップはあるが，このような目標に対する技術の方向性がならずも明確でない．すなわち現在の仕様策定は，分散オブジェクト技術の焼き直し，すでに実現可能なことを XML で定義しなおしているものが多く，必ずしももとの Web サービスの理想に近づく動きにはなっていない．

そこで，以下のような最新のカンファレンスのトピックなどを参考に，今後の方向性の検討を行った．

ICSOC 2003 (International Conference Of Service Oriented Computing)

ICWS 2004 (International Conference Of Web Services)

これについてまとめたものを図 1 に示す．

アーキテクチャについては，サービスの信用問題な

どに対応する Web サービスだからこそそのアーキテクチャが必要となる

Web サービス記述は自動利用を可能にするため，より豊富な情報を記述する方法が必要となる．オントロジそのものも大切だが，その決め方を決めることが重要である．

Web サービスコンポジションについては，サービスの連携を事前に決める静的なものから，要件にしたがい自律的に連携するものへと進化し，その際には，信頼できるコンポジションができるかが重要となる．コンポジションの記述方式（今の BPEL では役不足）やコンポジションの検証方式を確立することが必要である．

Web サービスのアプリケーションは，単純なスタートレスサービスからコンポジションサービス，さらにはユーザ指向サービスへと進化していく．

Web サービス技術により開発方法は，従来のウォーターフォール型から CBD(Component Based Development) と進化しており，さらに実行時ごとに，その時もっとも利用価値のあるサービスを用いて，システムを完成させる方法としての Service Oriented Development へと進化していくことが期待される．

5. 新世代ソフトウェア開発ツール

5.1 はじめに

近年，ソフトウェア開発のためのツールの重要性が再認識されつつある．例えば，大学など研究機関から

は、自らの研究成果を実用的なツールとして公開する動きがある。また、Eclipse に代表されるように、既存の公開されたツールプラットフォーム上で別のツールを作成する、ということも広く行われている。一方、オープンソース開発では、CVS やバグ報告ツールといった、従来からある基本的なツールを有機的に運用することによって、大きな成果をあげる事例も報告されている。これらは、ソフトウェア開発のためのツールについて、その機能や役割、設計方法や利用形態などの点で新しい動きが生まれてきていることを示している。

そこで本セッションでは、まず参加者全員がポジションペーパーについて発表を行った上で、これら新しいソフトウェアツールに関する話題を取りあげ、情報交換を行うと共に、現状の問題点や将来の方向性について議論を行った。

5.2 主な議論

短い時間の中で様々な議論を行ったが、ここでは全体会議で報告した主な議論だけを述べる。

5.3 eclipse について

eclipse は企業主導だが、eclipse のような成果を大学側から出せるのか？ という疑問がまず出された。eclipse は (やや) 枯れた技術を多大なコスト (お金と人) で集大成したものであり、それは必ずしも大学の役目ではないという意見や、eclipse の UI 統合やデータ統合は十分ではないので、貢献できる余地はある、という意見が出された。

5.3.1 大学でのツール開発のあり方

大学のツール開発では、学生が流動的なので継続的なツール開発が難しいことや、ツールをまじめに保守すると論文生産性が落ちるという問題を議論した。産学連携やインターンシップの一つとして、学生のコーディング進捗を企業が監督してもらうというアイデアが出された。また、現状では大学間の共同ツール開発が少ないので、WIDE 合宿にならない、ツール研究者たちも合宿すべきであるという意見も出された。

5.3.2 上流と下流のギャップについて

現状では、ツールは上流と下流が分離しており、垂直方向の追跡性が弱いという問題点が議題になった。MDA などの上流からのアプローチと XP などの下流からのアプローチがあるがどちらも直接ギャップを埋めることはできない、問題が難しいので追跡性は必ずしも機械処理可能でなくてもよいのではないかと、フレームワークや方法論を固定するとつながりやすい、しかし歴史を積み重ねる必要があるなどの意見が出された。

5.3.3 研究の新規性について

例えば、XML を使ったツール開発は old wine in new bottle、つまりうわべは新しいがテクニックは昔と一緒にのではないかという疑問や、ツールについて新しいアイデアはないのかという疑問が全体会議で出

された。これに対しては、ツールの古いアイデアに対して、実用化が不十分な部分や、実用化されていても非常に開発コストが高い場合が多いので、XML を用いて低コストで実用レベルのツールを研究開発することは意味があるという答えがなされた。

5.4 まとめ

ツールの研究開発の問題は、保守コストの問題や人的要因など多岐に渡るため、議論がやや発散傾向にあったことは否めないが、ツール開発の研究者が集まり、最新のツール研究の動向やお互いの研究について、議論できたことは非常に意義のあることである。

残念ながら、将来の方向性としてきれいな切り口を出せるところまでは議論が進まなかったが、実用的なツールを開発しつつ、大学としての特色を出していく、という努力が必要なことは間違いないだろう。

6. ソフトウェアパターン

6.1 目的と参加者

建築におけるパタン・ランゲージに触発される形で、様々なソフトウェアパターンが発見され用いられてきた。しかし、ソフトウェア設計のための包括的なパターンランゲージを志向した成果は、今にいたるまで事実上存在しない。この問題意識を踏まえて本トラックでは、ソフトウェアにおけるパターンランゲージのあり方について議論した。参加者は、鷺崎弘宣 (国立情報学研究所) と細谷竜一 (東芝ソリューション) を討論リーダーとして、落水浩一郎 (北陸先端科学技術大学院大学)、河合昭男 (オブジェクトデザイン研究所)、佃軍治 (日立製作所)、久保淳人 (早稲田大学) の 6 名であった。ソフトウェア工学研究会パターンワーキンググループ (以降、WG) の参加者と非参加者が合わさった構成となった。

6.2 議論の内容と成果

まず、パターンランゲージの成り立ちを参加者全員で体験した。具体的には、『パタン・ランゲージによる住まいづくり』(中埜博著、井上書院、1988 年) をテキストとして、パターンランゲージが解決する建築の問題を、ソフトウェア開発の問題に置き換えることを試みた。さらに、Web アプリケーションのユーザインタフェース (UI) 設計を題材として、要求獲得においてパターンランゲージを活用する様子をシミュレーションした。

続いて、ポジションペーパーを基に、「パターンランゲージの工学的アプローチ」「ソフトウェア開発プロセスとパターン」「組織活動とパターン」の 3 つのテーマを設定し、ポジションペーパー発表を起点として集中的に議論した。

(1) パターンランゲージ体験

上記『パタン・ランゲージによる住まいづくり』に見出せる建築問題のソフトウェア問題への置き換えを

試みた結果、「家相」がアーキテクチャに相当するといった対応関係を確認した。また、建築のパターン・ランゲージにおけるストーリー仕立ての開発や合意形成プロセスが、ソフトウェア開発においても必要であるとの結論に至った。

続いて、UI 設計を題材としたパターンランゲージの編み上げ体験を行った。体験では、非機能要件を列挙した後で、対応するパターンを、UI 憲法（『EXOS 統一操作環境の実現』（河合昭夫著、UNISYS 技報、1991 年 11 月））から選び対応付けることによって、非機能要件から生じる設計上の制約を解消することを試みた。成果の一部を表 2 に示す。得られたパターンと解決の集合は、パターンランゲージの雛型になっていると考えられる。

この体験から、次のような結論を得た。まず、用意されたパターンランゲージに基づいて、状況に特化したパターンランゲージを編み上げることが可能である。さらに、顧客が提示していなかった（あるいは顧客が当初思いつかなかった）非機能要件を、用意されたパターンランゲージから逆に推測して提案し、顧客満足度を向上させることができる。ただし、既存のパターンだけでは不十分なことがあり、開発経験などに基づいて、その場で新たにパターンを作る能力も必要である。また、UI は視覚的イメージを共有できるため、パターンランゲージを作成しやすいことが分かった。

表 2 非機能要件と UI 憲法の対応付け（一部）

非機能要件	UI 憲法の条文 (パターン)	条例 (パターンが与える解決)
反応の遅さを感じさせない	必ず即座に反応	クライアント側での処理機能
取り消しができる	逆操作を許す	Undo 機能
中断・再開できる	(該当なし)	一時保存機能

(2) パターンランゲージの工学的アプローチ

鷲崎らの報告『ソフトウェアパターンランゲージ工学に向けて』に基づき、これまでのソフトウェアパターンへの工学的取り組みのほとんどが、デザインパターンの扱い一辺倒な現状を確認した。パターン形式がもたらす効果に関する調査や、認知科学・言語理論といった他の学問領域との学際的な取り組みが、今後の重点的に取り組むべき課題であるとの結論に達した。

パターン形式とパターンの関係については、久保らによる『ソフトウェアパターンの自動的な体系化の試み』の発展が期待される。この試みでは、テキスト処理によってパターン文書のパターン形式を自動的に判別し、類似度に基づいたパターン間の関連付けを試みている。

(3) ソフトウェア開発プロセスとパターン

ソフトウェアパターン技術が、ソフトウェア工学体系において他の技術と結合しておらず、効果を発揮できていない現状を確認した。その解決にあたり、ま

ず、既存のソフトウェア開発方法論/プロセスと、ソフトウェアパターンの統合が求められる。その試みとして、落水は、ロールの対応関係に基づいて、RUP に Coplien らの組織パターンを組み込むことを行っている（『ソフトウェア開発方法論とソフトウェアパターン』）。また、要求工学の観点からパターンランゲージがもたらす役割と効果を明らかにすることが重要である。河合の報告『パターン言語を非機能要件定義に』に基づき、非機能要件の獲得と記述にパターンランゲージが有効に機能するとの結論を得た。非機能要件は、定義するものではなく、パターンランゲージを用いて解消するものである。

(4) 組織活動とパターン

これまでのパターン活動が、パターンを書くことのみ目を奪われている状況を確認した。組織活動においてパターンを生かすための方策として、佃は、ナレッジマネジメントにおける知識運用サイクルが、パターンの運用についても重要であることを指摘した（『パターン活動とナレッジマネジメント』）。この指摘に基づき、パターンは書いたら終わりではなく、日常的な活動とする必要性を確認した。また、パターン運用の全体的な枠組みが必要であることを確認し、その土台として、細谷が提案するパターンエンジン（『知識創造装置パターンエンジン』）を利用可能であるとの結論に達した。パターンエンジンは、パターンを知識の媒体として捉え、知識の発生や共有・進化といったナレッジマネジメントの過程をモデル化した仮定の装置である。

6.3 パターン宣言と今後の展開

議論の成果の全ては、WG の Web サイト <<http://patterns-wg.fuka.info.waseda.ac.jp/>> より公開している。我々は会議の終わりに、「パターンはソフトウェア技術者・組織の資質を磨くために不可欠である」と宣言した。この宣言は、パターンがコミュニケーションの質の向上と、個人の良識を養うことに有効であるとの信念に基づく。この宣言を実行に移すために取り組むべき方策 A~C と、WG を主体とした取り組みを以下に挙げる。

(A) 有用なパターンの提示: UI に関するパターンランゲージの事例研究と、他のパターンランゲージ化に適した問題領域の探索が必要である。これまでに、WG の勉強会を起点として、Web の UI デザインに関するパターンランゲージの事例調査を開始している。

(B) パターン運用の枠組みの確立: 実行可能なパターン活動支援技術を確立・普及させる必要がある。また、開発プロセスとパターンランゲージのさらなる融合が求められる。これまでに、WG の Web サイト上でパターンエンジンを活用したパターン運用に関するパターンランゲージ『パターンライフサイクルマネジメント』を公開し、その議論と洗練を開始している。

(C) パターンに関する基礎理論の探求: 前述の重点

課題を解決することによって、パターンランゲージ工学の確立が求められる。これまでに、WG 研究タスクにおいて、研究動向の調査や、パターン関連手法とパターン活動の対応付けを行っている。

今後は、各方策への取り組みの成果を広く公開・共有し、WG 内外での議論と実践を通じて、ソフトウェア開発活動・組織活動におけるパターンランゲージの活用を目指していく。

7. おわりに

今回は場所が石垣島と少々遠い場所であったため、参加者が少なめであり残念であった。しかし、少人数だったことがかえってまとまって議論を行うには好都合だったように思われる。今回は、過去数回の反省を踏まえ、討論時間をおよそ半日延長したが、時間的な余裕をもちながら議論を行うことができ、たいへん有意義であった。

最後に、本ワークショップに参加し活発な討論を行ってくださった参加者の皆様に感謝する。なお、ワークショップをはじめ、他のソフトウェア工学研究会主催行事については、研究会ホームページを参照していただきたい。

参 考 文 献

- 1) ウィンターワークショップ・イン・神戸 論文集
・ 情報処理学会, 2003.