

# アーキテクチャとパターン – セッションテーマ紹介

羽生田栄一<sup>†</sup> 鷲崎 弘 宜<sup>††</sup>

本稿では、ソフトウェア開発におけるアーキテクチャおよびパターンへの取り組みの展望を得るために「アーキテクチャとパターン」セッションがどのような議論を行うのかを説明する。

## Topics of the Session on Software Architecture and Patterns

EIITI HANYUDA<sup>†</sup> and HIRONORI WASHIZAKI<sup>††</sup>

This paper explains what discussions will take place in the session on software architecture and patterns to get insight into the directions of future software architecture/patterns efforts.

### 1. はじめに

ソフトウェアパターンとは、ソフトウェア開発における特定の文脈上で、繰り返し発生する問題と、問題に対する実証済みの解法として得られるソフトウェア構造、および、解法を導いたフォースなどを包括した知識を記述したものである。一方、アーキテクチャとは、機能要求や非機能要求に応じた決定により得られるシステムの基本的構造（の記述）である。アーキテクチャは既知のパターンを再利用して導出されたり、あるいは新たなパターンの雛形になるなど、両者は密接な関係にある。両者はそれぞれ活発に研究/実践されているが、近年の開発環境の急速な進化および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、両者の関係は未整理な部分も多い。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、アーキテクチャとパターンのそれぞれの課題や展望、両者の関係や周辺技術について幅広く議論する。

本稿では、ウィンターワークショップにおける類似セッションの議論の結果を概観する。続いて、本セッションにおける議論のテーマ案を紹介する。

### 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<sup>1)</sup>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

セッションを設置し、セッションにおける議論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<sup>2)</sup>。これは、パターン活動プロセス<sup>3)</sup>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<sup>4)</sup>。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<sup>5)</sup>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<sup>6)</sup>。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。

### 3. 発展経緯と議論テーマ

以上3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、他の周辺領域との関係については議論し尽くされていない。特に、ソフトウェアパターンがソフトウェアの構造上の決定指針を与えることを考えるとき、機能/非機能要求と構造設計の関係を扱うアーキテクチャ領域との関係について、さらなる議論が必要である。

#### 3.1 ソフトウェアアーキテクチャの経緯

アーキテクチャに関する議論は以前より存在し、古

<sup>†</sup> 株式会社豆蔵 Mamezou, Co., Ltd.

<sup>††</sup> 国立情報学研究所 National Institute of Informatics

くは 1960,70 年代において Dijkstra が階層構造の概念と有効性を, Parnas が情報隠蔽の概念と有効性をそれぞれ提案/指摘している。その後, 機能/非機能要求の複雑化/高度化/変化の早さや, 設計/実装技術環境の多様化/変化の早さ, および, それらに伴うソフトウェアシステムの複雑化/大規模化により, ソフトウェアの基本構造がもたらす支配的影響がよく知られるところとなり, 機能的特性と非機能的特性を考慮した基本構造の決定方法や, 特性との関係について様々な議論され今日に至る。

アーキテクチャの定義は様々であるが, 最大公約数的な定義は以下のようなものと考えられる。

- ソフトウェアの構成要素(サブシステムやコンポーネント), および, それらの間の関係の記述<sup>7)</sup>
- ソフトウェアの構成要素, および, それらの間の関係を設計/保守する際の原則や指針

### 3.2 ソフトウェアパターンの経緯

C. Alexander の建築学におけるパターンランゲージ<sup>8)</sup>に触発されて, 特にオブジェクト指向に基づく分析/設計に関するソフトウェア業界の知識がデザインパターンやアーキテクチャパターンとして底上げされた<sup>9)</sup>。また, 開発ノウハウをパターン形式で記述することも一般的になりつつある。

しかし, ソフトウェアパターンの定義は様々であり, パターンとそうでないものを分かつ条件も曖昧である。例えば Winn らは, パターンの多様さに起因してパターンの必要条件を見出すことは難しいことを指摘し, 必要条件の代わりに, 設計パターンが備えるべき以下の 9 つの本質的特性を提案している<sup>10)</sup>。

- ソフトウェア/ドメインの設計を暗示する
- 複数の抽象レベルにおける表現を結びつける
- 機能的特性と非機能的特性の両者を扱う
- 得られるソフトウェア設計中に痕跡が残る
- ホットスポットを捉える
- パターンランゲージの一部を形成する
- 繰り返し利用によって有効性が確認済み
- 特定のドメインに根ざしている
- 設計上の巨視的な方針を捉える

### 3.3 今回のテーマ案

上述の両定義例を比べて分かるように, 両者, 特に設計パターンとアーキテクチャは非常に密接な関係にある。そこで, アーキテクチャ/パターンそれぞれの特性や効果的運用方式に加えて, 例えば以下に挙げるような両者間の関係に関するテーマを議論したい。

- 品質特性を軸とした関係: パターンとアーキテクチャはどちらも非機能的特性上の影響を組み入れ

た記述であり, 特に品質特性を軸とした関係の整理が試みられている<sup>11)</sup>。従来の成果を踏まえつつ, 有効性や今後の展望を議論したい。

- 統合ライフサイクルプロセス: アーキテクチャ設計にあたり, 既知のアーキテクチャパターンを土台として参照することが多い。一方, 技術環境や社会要請の変化に伴い, 新種かつ類似の基本構造集合を一般化したアーキテクチャパターンが得られる。それらの統合プロセスを整理検討したい。
- 新世代開発手法/環境における関係: モデル駆動やアスペクト指向, Web2.0 などの新たな開発手法/技術環境におけるアーキテクチャとパターンの関係を展望したい。また, 形式手法のような最近実用フェーズに移りつつあるものも取り上げたい。

## 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。さらなる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アーキテクチャおよびパターン技術が活用されることを期待する。

## 参 考 文 献

- 1) <http://patterns-wg.fuka.info.waseda.ac.jp/>
- 2) 松下ほか: 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.
- 3) 鷲崎ほか: 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.
- 4) 紫合ほか: 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.
- 5) 羽生田 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, SRC, 2005.
- 6) 満田ほか: ウィンターワークショップ 2006・イン・鴨川参加報告, 情処研報 2006-SE-152, 2006.
- 7) M. Shaw and G. David: Software Architecture, Prentice Hall, 1996.
- 8) C. Alexander, et al.: A Pattern Language, Oxford University Press, 1977.
- 9) E. Gamma, et al.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.
- 10) T. Winn and P. Calder: Is This a Pattern?, IEEE Software, 19(1), 2002.
- 11) E. Folmer and J. Bosch: A Pattern Framework for Software Quality Assessment and Tradeoff Analysis, <http://www.eelke.com/research/literature/SQTRF.pdf>