

「ソフトウェアパターン」セッションテーマ紹介

鷲崎 弘宜† 細谷 竜一††

本稿では、ソフトウェアパターンの発展の経緯と現状を概観し、今後のソフトウェアパターンの取り組みの展望を得るために「ソフトウェアパターン」セッションがどのような議論を行うのかを説明する。

Session Topics of “Software Patterns”

HIRONORI WASHIZAKI† HOSOYA RYUICHI††

This paper outlines the progress of software patterns and explains what discussions will take place in the session “Software Patterns” in order to get insight into the directions of future software patterns efforts.

1. パターンランゲージ再び

ソフトウェアパターンに関する取り組みは、Christopher Alexander の建築学における 1970 年代後半の成果であるパターンランゲージ[1][2]に触発されて 1980 年代後半に米国で始まった。まず、OOPSLA’87 のワークショップにおいて、Kent Beck と Ward Cunningham がオブジェクト指向プログラミングにパターンランゲージの考え方を応用するアイデアを表明した[3]。以来、Hillside Group を代表とするソフトウェアパターンコミュニティが誕生し、Hillside Group 主催の会議 The Pattern Languages of Programs (PLoP) Conference を筆頭とした様々な活動が継続的に行われてきた。

初期のソフトウェアパターンコミュニティは、ソフトウェアアーキテクトのためのハンドブックの作成を目指していた。1995 年に GoF と呼ばれる四人の著者によって「Design Patterns」[4]が出版されたのを皮切りに、これまでにいくつかの有力なパターンカタログが出版された。

日本においては、ObjectWorld Tokyo’93 で Grady Booch がパターンについて言及して以来、1995 年に「Design Patterns」[4]の翻訳[5]が出版されるなどしてソフトウェアパターンへの関心が高まった。そして、JapanPLoP という日本版パターンコミュニティが結成され、また、その発展的解消と前後して情報処理学会ソフトウェア工学研究会の下でパターンワーキンググループが活動を開始するなどして、日本におけるパターンの普及について継続的に取り組まれてきた。

ソフトウェアパターンコミュニティの活動の成果として、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底上げされた。また、共有するに足ると考えられたノウハウや知識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。このように、先駆者たちが Alexander のパターンランゲージに触発された結果として、アーキテクトのためのハンドブック作成という初期の目的は達成されなかったものの、ソフトウェアに関する知識を書き表し、共有するためのソフトウェアパターンという方法は一定の効果を上げたと考えられる。

しかしながら、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一つも得られていない。パターンカタログは、強い関連性が認められる複数のパターンを集めたものであり、ソフトウェア設計プロセスの最初から最後までに設計者が迎えるであろう局面を次々と捉えて解決策を生成していくパターンランゲージとしての特性を欠く。Alexander の成果において個々のパターンは孤立した存在たりえず、必ずパターンランゲージの中の一員であることを考えると、ソフトウェアパターンのこれまでのあり方は本来的にパターンランゲージという軸を欠いている。

一つ一つのパターンの個別的な記述・利用には限界がある。一つのパターンを、他との関連抜きに記述するだけでは、幾らか気の利いたメモ書きとさしてかわらない。また、独立したパターンを単に蓄積していくと、ある局面で有用なパターンを、それを必要とするときに発見し取り出せなくなる。個別的に、系統立っていない方法でパターンを適用した結果得られるものは、必ずしも一貫性を持たず、全体としてそれがなぜそうになっているのかを筋道立てて説明できない可能性がある。すなわち、パターンという考え方を導入することによる固有の結果として何が良くなったのか、はっきりしないのである。

そこで、本セッションは、これまでの個々のパターンをいかに記述・利用するかという議論の一線を越え、ソフトウェアにおけるパターンランゲージとはいかにあるべきかを考える一つの起点となることを目指す。前提として次の二つの事柄を考慮している。まず、建築において

† 早稲田大学理工学部

School of Science and Engineering, Waseda University

†† 東芝ソリューション株式会社 SI 技術開発センター

Systems Integration Technology Center, Toshiba Solutions Corporation

Alexander のパターンランゲージが取り扱っている問題が、ソフトウェアの開発という所業における問題に置き換え得るものであるという信念を我々は持っている。さらに、パターンランゲージが建築問題の中にもたらしたものがソフトウェア開発問題の中にどのような姿となって見出せるのかについて、我々は依然として見解を得ていないという事実がある。

我々は本セッションを通じて、ソフトウェアパターンが当初約束したソフトウェア設計(あるいは開発)のパターンランゲージに向けて舵を切り、ソフトウェア開発に携わるより広い人々の層がソフトウェアパターンを活用できるようにすることを目指す。

2. 討論テーマ

本セッションでは上述した背景を踏まえて、ソフトウェアにおけるパターンランゲージのあり方を探ること、ソフトウェア開発活動とパターン活動の融合、そして組織とパターンの関係を柱とした議論を展開する。具体的には、以下のテーマを取り上げる。

2.1. パターンランゲージ体験と考察

「パターンランゲージとはこういうものか」ということを身近に理解し、「なるほど!」と言える体験を得るを試みる。身近なソフトウェアを題材に、設計者による設計の過程の各局面をパターンとして抽出し、参加者全員によってパターンランゲージとしてその場で「編み上げる」。こうした小セッションを通じて、パターンランゲージの成り立ちに対する洞察を得る。また、Alexander の理論に見られるパターンランゲージによる建築の経済性や建築物の質の問題を、ソフトウェア開発の経済性とソフトウェアの質の問題に置き換えて考察する。

2.2. パターンランゲージの工学的アプローチ

ソフトウェア開発において、パターンランゲージが機能する仕組みともたらす効果を工学的に明らかにする取り組みのあり方について議論する。具体的には、パターンワーキンググループにおける調査活動の成果の一部[6]を取り上げて、ソフトウェア工学に基づいた従来のソフトウェアパターンに関する研究成果から明らかとなった事柄と不明な事柄を整理し、将来の工学的な取り組みの可能性(もしくは限界)について考察する。例えば、パターンランゲージの考え方は本来的に、地域や組織・文化といった社会的・人間的要素に密接に関わっている。ソフトウェアにおけるそのような要素とパターンランゲージの関係を捉えるために、ソフトウェア工学にとどまらない学際的な取り組みが必要と考えられる。

2.3. 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の取り組みが、他のソフトウェアエンジニアリングの実践領域と積極的に結合していないことも、これまでのパターン活動に限界を与えてきた一因であろう。ここではソフトウェア開発プロセスの中でどのよ

うにソフトウェアパターンが作用するのか、また両者を融合するにはどうしたらいいかを議論する。また、パターンランゲージと要求獲得及び合意形成は深く関連する[7]。ここではまた、特に定量的あるいは形式的に捉えることの難しい要件の定義に焦点を当てて、パターンランゲージを用いた要件定義方法を考察する。

2.4. 組織活動とパターン

組織がその活動を通じて継続的に目的を達成するには、活動の過程において組織内で知識を共有し、新しい知識を生み出していくことが重要である。パターンは知識の共有を促進し、パターン同士の関連付け(あるいはランゲージ化)は新しい知識を組織に与える。この点においてパターンはナレッジマネジメントとゴールを同じくしている[8]。ここでは組織活動を通じて組織とパターンがどのように相互作用するのか、そしてナレッジマネジメントのプロセスとしてパターン活動を捉えることでどのような知見が得られるのかを議論する。

参考文献

- [1] Alexander, Christopher. *The Timeless Way of Building*. Oxford University Press, 1979.
- [2] Alexander, Christopher, et al. *A Pattern Language*. Oxford University Press, 1977.
- [3] Beck, Kent and Ward Cunningham. "Using Pattern Languages for Object-Oriented Programs." *Position statement for the OOPSLA'87 workshop on the Specification and Design for Object-Oriented Programming*, 1987. <http://c2.com/doc/oopsla87.html>
- [4] Gamma, Erich, et al. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [5] 本位田真一 and 吉田和樹(監訳). *オブジェクト指向による再利用のためのデザインパターン*. ソフトバンクパブリッシング, 1995.
- [6] 鷲崎弘宜 and 深澤良彰. ソフトウェアパターン研究の現在と未来. In *情報処理学会研究報告「ソフトウェア工学」*, Vol.2003, No.55 (SE-141), 2003.
- [7] 中埜博, 井上健, and 羽生田栄一. 要求獲得と合意形成のためのパターンランゲージ入門. In 野呂昌満, et al. *オブジェクト指向最前線<2003> 情報処理学会 OO2003 シンポジウム*. 近代科学社, 2003. <http://patterns-wg.fuka.info.waseda.ac.jp/event/oo2003/oo2003-tutorial-nakano.pdf>
- [8] 細谷竜一, 金澤典子, 沖田直幸 and 佃軍司. ナレッジマネジメントとしてのパターン活動. In 野呂昌満, et al. *オブジェクト指向最前線<2003> 情報処理学会 OO2003 シンポジウム*. 近代科学社, 2003. <http://patterns-wg.fuka.info.waseda.ac.jp/event/oo2003/oo2003-tutorial-hosoya.pdf>