SEMAT メソッドアーキテクチャによる ソフトウェアパターンの共通表現

鷲崎 弘宜†

アブストラクト
種々のソフトウェアパターンについて、利用局面や種類によらずに統一的かつ適用・再利用に適したモデル化の方法として、SEMAT（Software Engineering Method and Theory）のメソッドアーキテクチャおよびエッセンス上で表現する方法を検討する。

Common Pattern Representations by SEMAT Method Architecture

Hironori Washizaki†

Abstract
This paper discusses common representations of software patterns based on SEMAT Method Architecture and Essence.

1. はじめに

ソフトウェアの特定の文脈で繰り返される問題と解決をまとめるソフトウェアパターン（以降、パターン）は従来、コミュニティにおいて概ね共通合意された形式に従った半構造文書として定義され、公開洗練され、用いられている。しかしながら、大量のパターン集合の関連付けや整理、検索、さらには計算機による機械的な適用・再利用、拡張といった処理を効率的かつ正確に実施するためには半構造文書は必ずしも記述形式として適さないため、デザインパターンを中心に様々なモデル化・形式化の試みがなされている[1-3]。

従来のモデル化手法は、（モデル化の考え方として自然なことであるが）特定のパターンや目的に特化しており、様々なパターンの種類や目的・利用局面に対応しない。しかし、実際のソフトウェアの開発や運用においては様々なパターンの利用局面が存在し、様々なパターンを扱う。そこで、統一的かつ適用・再利用に適したモデル化の方法があれば、組織あるいはコミュニティにおいて種々のパターン群を統一的なモデル化および比較整理の結果として一貫したライブラリとして蓄え、一貫した活用が期待できる。

異なる種々のソフトウェア開発手法やプロセスを共通基盤上で表現することで比較や拡張および実際の開発・運用における適用・進捗管理に用いる考え方として、メソッドアーキテクチャおよびその基盤となるエッセンスが SEMAT（Software Engineering Method and Theory）Initiative において提案されている[4-5]。ただし、メソッドアーキテクチャにおけるソフトウェアパターンの比較整理と同時に再利用・適用に適した統一的な表現方法は明らかとなっていない。そこで本稿では以降において、エッセンスメソッドアーキテクチャを概観したうえで、問題解決としてのパターンの表現方法を検討する。

2. メソッドアーキテクチャ

SEMAT Initiative は、理論に裏打ちされた共通基盤上でのソフトウェアエンジニアリングの再建や共通理解をビジョンとして掲げる運動である。その日本支部は 2013 年 4 月に筆者を代表として設立されている。

共通基盤としての「エッセンス」を含むメソッドアーキテクチャの全体を図 1 に示す。メソッドアーキテクチャでは、ソフトウェア開発や運用において状態（進捗）管理すべき本質的な要素（例えば要求やソフトウェアシステム、仕事など）を「アルファ」と呼び、それらの間の関係を「カーネル」として定めている。ここで、個々のアルファが状態を持つ点が特徴的であり、「状態指向」な手法と捉えることができる。共通基盤としてのカーネルや、図形式のドメイン特化言語を用いて、種々のプラクティスを記述および合成できる。カーネルおよびドメイン特化言語はまとめて「エッセンス」と呼ばれる。最終的に個々の開発手法やプロセスは、プラクティスの集合から構成される。

†早稲田大学グローバルソフトウェアエンジニアリング研究所
Global Software Engineering Laboratory, Waseda University

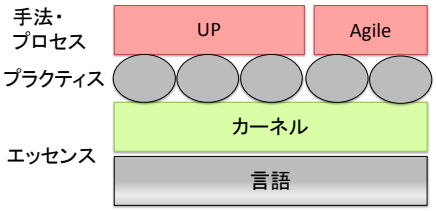


図 1: SEMAT におけるメソッドアーキテクチャの概要

3. メソッドアーキテクチャにおけるパターンの表現

SEMAT エッセンスにおいてパターンは、意味的には「プラクティスにおける構造の記述」および「制約の少ない追加ガイダンスを定義するもの」と定義され、構造的にはカーネル上のあらゆる要素（例えばアルファやアクティビティ）と関係を持ちうるものとして位置付けられている。プラクティスは「特定目的のための繰り返し可能なアプローチ」と定義されている。

上述の標準的な定義は純粋に頻出の構造という意味でのパターンであり、必ずしも問題解決としてのパターンを意味していない。従って、パターンの記述における自由度が極めて高く、半構造文書形式と同様に利用局面や種類によらない統一的な扱いは難しい。

そこで筆者は、問題解決としてのパターンに共通する特徴に着目して、エッセンス上の要素の構成に幾らか制約を持たせたモデル化を提案する。そのようなパターンは、問題解決を通じた（開始・初期）文脈から結果文脈への状態遷移系とみなすことができる[6]。筆者らはこれまで、そのような状態遷移系とみなすモデル化により多数のパターン群の自動的な関連付けに成功している。本稿では、同様のモデル化にあたりエッセンス上の要素を用いることで、パターンを一貫した形で表現しながら、ソフトウェア開発・運用上の手法やプロセス、活動と容易に接続・組み合わせ可能とすることを目指す。

問題解決（すなわち状態遷移系）としてのパターンをエッセンスで表現する場合の一案を図 2 に示す。図 2 において、エッセンスにおけるパターン関連を用いて、パターンが扱うアルファ、活動、成果物を明示している。パターンの問題解決において影響のある文脈の表現にはアルファを用い、解決前後の状態を異なるアルファの状態として表現する。解決の表現には活動を用いる。なお図 2 の図形要素は全て、エッセンスのドメイン特化型言語に従っており、エッセンスに準拠した開発環境や手法における扱いを可能とする。

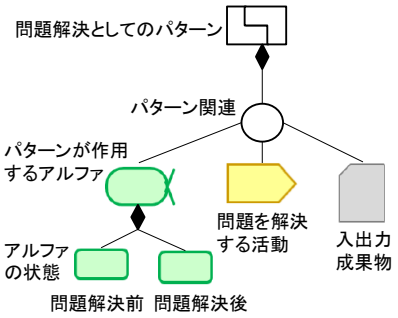


図 2: 問題解決としてのパターンの表現案

4. おわりに

本稿では多様なパターンの種々の利用局面における統一的な表現と扱いを目的として、SEMAT のエッセンスおよびメソッドアーキテクチャにおける問題解決としてのパターンの表現を試みた。今後は、種々の具体的なパターンを対象として、問題解決におけるアルファの状態遷移といった振る舞いを含めた具体的表現、および、手法やプラクティスといったソフトウェア開発・運用上の他の要素との接続表現を通じて、提案する表現の妥当性や有効性を検証したい。

参考文献

1) 鷲崎弘宜ほか, “デザインパターンへのソフトウェア工学的取り組み”, コンピュータソフトウェア, Vol.29, No.1, pp.1_130-1_146, 2012.

2) A. Nhlabatsi, A. Bandara, S. Hayashi, C.B. Haley, J. Jurjens, H. Kaiya, A. Kubo, R. Laney, H. Mouratidis, B. Nuseibeh, Y. Tahara, T.T. Tun, H. Washizaki, N. Yoshioka and Y. Yu, “Security Patterns: Comparing Modeling Approaches”, in “Software Engineering for Secure Systems”, pp.75-111, IGI Global, 2010.

3) Jeffrey K. H. Mak, Clifford S. T. Choy and Daniel P. K. Lun , “Precise Modeling of Design Patterns in UML,” Proceedings of the 26th International Conference on Software Engineering (ICSE’04), pp.252-261, 2004.

4) OMG, “Essence – Kernel and Language for Software Engineering Methods,” ad/2013-02-01, 2013. http://www.omg.org/spec/Essence/1.0

5) Ivar Jacobson, Pan-Wei Ng, Paul E. McMahon, Ian Spence, Svante Lidman, 日本語訳： 鷲崎弘宜, 平鍋健児, 川口恭伸, 小林展英, 吉村健太郎, “ソフトウェアエンジニアリングのエッセンス: SEMAT カーネル”, 日経 ITPro, 2013.

6) Atsuto Kubo, Hironori Washizaki, Atsuhiro Takasu, and Yoshiaki Fukazawa, “Extracting Relations among Embedded Software Design Patterns,” Journal of Integrated Design and Process Science, Vol.9, No.3, pp.39-52, 2005.