

ナレッジマネジメントとしてのパターン活動

チュートリアル「形から入らないパターン活動綴り方教室」に向けて

細谷 竜一† 金澤 典子†† 沖田 直幸††† 佃 軍治††††

組織の中におけるノウハウを知識化し、共有することすなわちなレッジマネジメント(knowledge management)の重要性が叫ばれてから久しい。ソフトウェア開発の分野では、「パターン」の考え方がソフトウェア開発にまつわるナレッジマネジメントの手段の一つとして捉えられ始めている。組織の中でパターンを効果的に運用することでナレッジマネジメントを行うための活動を我々は「パターンライフサイクルマネジメント(pattern lifecycle management / PLCM)」と呼ぶ。本稿では、パターンをナレッジマネジメントの概念と対比させながら捉え、そして PLCM の視点から組織におけるパターンの実践について議論する。

Pattern Activities As Knowledge Management

For the Tutorial “A Workshop for Pattern Activities without Formalism”

Hosoya Ryuichi† Noriko Kanazawa†† Naoyuki Okita††† Gunji Tsukuda††††

It has been long since the importance of turning know-how into knowledge and sharing it within the organization, i.e., knowledge management was recognized. In the field of software development, the concept of “patterns” is beginning to be highlighted as a means of managing software development knowledge. We refer to the activities of knowledge management within an organization by effectively using patterns as “patter lifecycle management (PLCM).” In this paper, we contrast patterns with the concept of knowledge management and discuss from the PLCM point of view practice of patterns within an organization.

1. はじめに

組織の中においてノウハウを知識化し、共有することすなわちなレッジマネジメント(knowledge management) [1]の重要性が叫ばれてから久しい。ソフトウェア開発の分野では、パターン[2]の考え方がソフトウェア開発にまつわるナレッジマネジメントの手段の一つとして捉えられ始めている[3][4]。ここでは、知識化されたノウハウはパターンの形態をとり、組織内で共有され、そして成長していく。

出版物などを通じて一般によく知られたパターンは初めから出版された形で生まれ出たわけではない。そこへ至るまでのパターンの成長がある。パターンは、まず特定の集団あるいは組織の中で発見され、書かれ、共有され、洗練されていき(フェーズ I)、その中から遂に公にさ

れるものが出てくる(フェーズ II)。

組織の中に潜在する知恵を組織内で共有し、効果的に活用するためには、まずフェーズ Iにあるパターンの発見と育成に焦点を当てるべきである。一方、組織の外からフェーズ II のパターンを導入するためには、それを組織の文化やプロジェクトのスコープに合わせてあつらえ直した上で、組織の中で根付かせていくプロセスが必要である。

パターンの発見、導入、共有そして洗練を支援する活動を我々は「パターンライフサイクルマネジメント(pattern lifecycle management / PLCM)」と呼び、[3]の中でその定義や性質を議論した。本稿では、パターンをナレッジマネジメントの概念と対比させながら捉え、そしてパターンライフサイクルマネジメント(以下 PLCM)の視点から、ソフトウェア開発組織におけるパターンの実践について論じる。

2. パターンのスキル

パターンを運用するために必要なスキルの一つは、パターンを記述するスキルである。パターンにはいろいろな記述形式があるが、そこから得られる情報の性質には共

†株式会社東芝(Toshiba Corporation) e-ソリューション社 SI 技術開発センター

††独立コンサルタント(Independent Consultant)

†††横河電機株式会社(Yokogawa Electric Corporation) システム事業部

††††株式会社日立製作所(Hitachi, Ltd.) システム開発研究所

通性がある。すなわち、そのパターンを識別するための名前、いつどういときに使うかの選択基準、適用前の状況(文脈 / context)、適用結果のイメージ(結果文脈 / resulting context)、適用前の状況から適用後の状況へと導く手順(解 / solution)の詳しい説明、説得(事例や理論などの根拠 / rationale)、そして適用対象となる活動の中で集めたパターンの一覧表(関連するパターン / related patterns)あるいはそれらを有機的に結合したパターンランゲージ(pattern language)である。パターンはこうした情報のセットによって知識を形式化するので、パターンを運用する活動はナレッジマネジメントの一形態と捉えられる。ナレッジマネジメントの用語では、知識の形式化は表出化(externalization)と呼ばれる[1]。パターンを記述するスキルはパターンによるナレッジマネジメントの基本的なスキルの一つである。

パターンを記述することが組織に潜在するノウハウを知識化するのに対し、外部から提供されたパターンの記述を読み解き、自らが直面している問題や状況に当てはめて理解することは、組織の中に外部から提供された知識を獲得してノウハウ化する。ナレッジマネジメントの用語では、形式化された知識の獲得は内面化(internalization)と呼ばれる[1]。これもまた基本的なスキルである。

パターンは、もとをただとパターンランゲージ(pattern languages)[5]という考え方から始まった。個々のパターンは組織の活動の中の一つの局面に関する知識なのに対し、パターンランゲージは関連するパターンが有機的に結びついて、パターンを適用する局面を次々と導いていく。言い換えれば、パターンランゲージは、あるパターンを適用すると次にどんな局面が訪れ、どんなパターンが適用可能なかを教える。従って、パターンは一つ一つばらばらに記述されるよりも、パターン適用の局面のシナリオを思い描きながら、互いに関連付けていくとより効果的だと考えられる。ナレッジマネジメントの用語では、形式化された知識同士を結びつけて新しい知識を合成することは連結化(combination)と呼ばれる[1]。これを行うためには、パターンランゲージを編成するスキルが用いられる。このスキルの獲得にはその組織活動の対象分野(ドメイン)におけるより高い専門性と連結化のための適性が求められる。

3. PLCM におけるパターンの役割

前節では、パターンのスキルをナレッジマネジメントに見られる考え方と対比して挙げた。この中で、ナレッジマネジメントにおける「SECI(セキ)モデル」の四つのプロセスの内、三つに言及した;表出化、連結化、内面化である。もう一つのプロセスである「共同化(socialization)」[1]は、暗黙知から暗黙知のプロセスとして理解され、徒弟制度を通じて伝授される[4]ものとされてきた。このため、パターンが直接作用することはないと捉える人もいるだろう。しかし、PLCM では、フェーズ I のパターンの運用においては、共同化のためにもやはりパターンを活用する。

PLCM では、パターンを書くことは手段に過ぎないので、フェーズ I のパターンに関してはその記述形式の洗練に重きを置くよりも、組織の活動の中で次々とパターンを発見して掴み取り、そのライフサイクルに乗せていくことを優先する。その過程では、パターンが発見されると、それを完全には記述するよりも、まずそのパターンが存在するという事実を組織内で周知させる。例えば、活動の中で、メンバの成果の中にパターンを見出したとすると、次を行う:

1. パターンに名前を付ける。
2. そのパターンの適用例として成果物を見やすい場所に置く。
3. 組織で用いているパターン一覧表の中にこのパターンの名前と適用例を記録し、閲覧できるようにする。

これにより、ノウハウがパターン化(形式化)へ備えて芽を出した状態を作り出す。同じ組織やプロジェクトの中で、密なコミュニケーションがとれるメンバ同士で、パターンの記述についてこれ以上する必要は必ずしもない。適用例に基づいてコミュニケーションをとりながらノウハウをノウハウとして伝授できるからだ。つまりこれが共同化である。重要なのは、常に充実し、洗練されたパターン記述を前倒して用意することではなく、PLCM としてノウハウの共同化を行うと同時にその表出化への備えを行っておくことである。これによって、組織は必要なときに最適なタイミングで表出化から先のパターンのライフサイクルを進められる。すなわち、PLCM はパターンを用いて組織における知識の共同化を含めた SECI モデルのプロセスのサイクルを支援する活動だといえる。

4. PLCM の体制

PLCM を実施し効果を上げるためには、それを支援するための体制があるとよい。これは、その組織の中のパターン文化の成熟とともに整えられていくべきだろう。そうした支援体制の中には組織内でのフェーズ I パターンのライフサイクルを管理する枠組みが含まれるだろう。また、PLCM のスキル獲得を支援するためのワークショップの開催や、ナレッジマネジメントツールの活用などの可能性もある。また、パターン文化が一つの組織の中だけで発達し続けると、逆に自前主義に陥る恐れがある。そのため、外界から遮断せずに、フェーズ I とフェーズ II の間を行ったりきたりしながらパターンの交配を図れる組織活動の体制が望ましいだろう。

5. 今後の展開

付録として PLCM のパターンランゲージとしての定義を示した。ここに示される各パターンについては、今後パターンワーキンググループ[7]内の実践タスクの活動の中で徐々に定義していきながら、Web 上で事例収集を行う予定である。PLCM それ自身も、形式の完全性を重視

ナレッジマネジメントとしてのパターン活動
～チュートリアル「形から入らないパターン活動綴り方教室」に向けて～

した方法論としてよりも、組織のノウハウとして定着しうる実践技術 (practice) として発展することを狙っている。そのために、今後 PLCM をソフトウェア開発プロジェクトの中で実験的に実施したり、PLCM を支援するためのパターンワークショップを開くなどして、PLCM の実践性を練り上げていきたい。

付録: パターンランゲージとしての PLCM

PLCM をソフトウェア開発組織を例題たパターンランゲージとして整理し、そこに含まれる活動をパターンとして定義する。このパターンランゲージは[3]に基づいている。パターンの記述形式は Coplien 形式[6]を模している。但し、**根拠 (Rationale)** は省いた。これについては[3]を参照されたい。ここに示すパターンランゲージは PLCM の「形から入った」定義である。このパターンランゲージは、今後パターンワーキンググループ[7]の活動の中で発展していくものである。

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
(Pattern Lifecycle Management / PLCM)

問題 (Problem): 組織の開発力を高めるため、組織の中に存在したり、外から導入したパターンを運用し、知識を共有したい。

文脈 (Context):

× 例えばソフトウェア開発組織において、開発をしながら次のような状況に遭遇する:

- 要求仕様があいまいなままなのに、なぜか設計が進んでいる(ように見える)。
- アーキテクトは口先ばかり。
- 個々を組み合わせで動く確信がない。
- 開発の段取りを知らない。
- 設計の悪いところを直すとき、目に付いたところだけ直す。
- プログラムは動いてはいるけれど、見た目や設計に一貫性がない。

結果文脈 (Resulting Context):

例えばソフトウェア開発組織において、開発において知識や情報がよくコミュニケーションされ、共有される文化がある:

- 要求仕様を共有している。
- アーキテクトが自分の言ったことを手本(パターン)で示してくれる。
- 作ったものを仲間の目にさらしている。
- 他人の設計を自分がやっても同じ質を作り出せる。
- 設計の一つを正せば、同じ間違いを全て正せる。

影響力 (Forces): パターンの運用に当たっては、注意し、制御せねばならない様々な影響力がひしめく。次にそのいくつかを挙げる:

- パターンによるコミュニケーション加速 vs. パターンの共有しない者の疎外感
- 効率化のための気の利いた仕掛け vs. コスト削減のための高コストな仕掛け
- 自前主義の否定による品質向上 vs. 無批判、先例主義、モラル低下
- 同じ発明を二度しない vs. 一つの発明を二度と見直さない

解 (Solution): 次のように定義されるパターンライフサイクルマネジメントを実践する:

[パターンライフサイクルマネジメントの定義]

- パターンが発見または外部から導入され、組織内で共有され、成長する過程を支援すること(フェーズ I)
- オプションとして、フェーズ I で成熟したパターンを、公の場に公開して、さらに成長する過程を支援すること(フェーズ II)

同時に、パターンを生み出し活用できるソフトウェア開発を実践する。

これらの実践のため、次に挙げるパターンからなるパターンランゲージを用いる:

[パターンマネジメント実践のパターン

(Patterns of Patterns Management Practice)]

- 即席パターン (Instant Patterns) … プロジェクトの期間内、顔の知れた仲間内ではパターンには名前を与えるだけで共有できる。こうして軽く、すばやくパターンを捉え、そのライフサイクルに乗せる。
- 外部パターンを導入する (Retrofit External Patterns) … 世の中に知られたパターンや他の組織から持ってきたパターンは抽象的だったり違った文化を前提にしたりしているので、自分の組織やプロジェクトのスコープに合わせてあつらえ直して導入する。これはフェーズ II のパターンをフェーズ I へと転移させることである。
- パターンを公開する (Publish Patterns) … フェーズ I のパターンの中には、プロジェクト、組織あるいはドメインを越えて適用可能な普遍性を備えたものもあるかもしれない。組織は、そうしたパターンを洗練し、フェーズ II のパターンへと昇華させて一般に公開することで、その活動をより広い文脈の中で捉え、改

善しながら競争力を高める機会を得られる。

- パターンを死なせない (Keep Patterns Alive) …パターンは書かれればなしでは死んでしまう。適用すべき場面でパターンが再発見し、共有を進め、洗練していくことでパターンを生き生きと育てる。
- SECI (Socialize, Externalize, Combine and Internalize) …ナレッジマネジメント (knowledge management) の枠組みの中でパターンを運用し、SECI モデル[1]にのっとったライフサイクルに乗せる (図 5-1)。

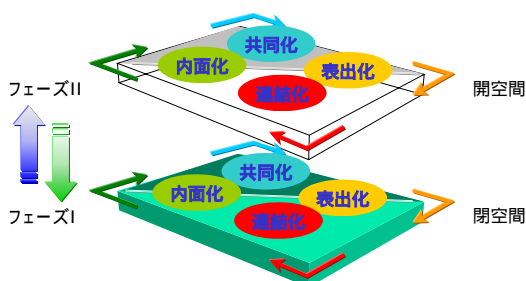


図 5-1 パターンライフサイクルの SECI モデル

- パターンランゲージを編成する (Weave A Pattern Language) …ばらばらのパターンの威力は小さい。パターン同士が有機的に結びついて (パターンランゲージになって) 連鎖的に適用できてこそ最大の効果をもたらす。
- パターン適用を追跡する (Trace Pattern Applications) …パターンを使って、知識が既存の成果物の中で適用されている様子を追跡し、その知識の成長を成果物に再度反映させることを可能にする。
- パターン文化を形成する (Create A Pattern Culture) …パターンを運用するためには、組織そのものもパターンを受け入れ、育てる文化に適応しなくてはならない。パターンが組織文化に適応することと、組織がパターン文化に適応することを同時に支援する。

[ソフトウェア開発実践のパターン
(Patterns of Software Development Practice)]

- 要求を共有せよ (Share Requirements) …要求者 (取得者) でないと決められないことと、開発者 (供給者) のみで決められることを区別し、ソフトウェア開発を通じた知識創造活動のための正しい前提条件を共有する。
- 実践派アーキテクト (Hands-On Architect) …アーキテクトが「このように作ろう」と言い出したら、アーキテクトは二十四時間以内に手本かサンプルを提示しなくてはならない。また、この手本やサンプルはパターン化される。
- 成果をさらせ (Expose Your Work) …自分の成果は常に仲間の目にさらすべき。これによって、パターン発見、組織内での波及、適用の機会を早期に与える。
- 他人の看板を掲げられる (You Can Wear Someone Else's Hat) …パターンを共有し、パターンに基づいた設計を持っていれば、ソフトウェアのあらゆる箇所は元の設計者でなくとも理解、再構築できる。
- 一直せば十直る (Fix One, Get Ten For Free) …「パターン適用を追跡する」ことの結果として、設計のどこかに不適合が見つければ、そこと同じパターンを適用した全ての箇所を追跡し、同時に手直しができる。このようにして、パターンは設計の一貫性を継続的に高めて行く。また、これはそのパターン自身の洗練化をも意味する。

参考文献

- [1] 野中郁次郎 and 竹中弘高. *The Knowledge-Creating Company*. Oxford University Press, 1995.
- [2] Johnson, Ralph E., 中村弘明 et al. *パターンとフレームワーク*. ソフトウェアテクノロジーシリーズ, 共立出版, 1999.
- [3] 細谷竜一. 形から入らないパターン活動. *Proceedings, 第 141 回ソフトウェア工学研究会*, 2003. URL: <http://patterns-wg.fuka.info.waseda.ac.jp/opening.html>.
- [4] May, Daniel and Taylor, Paul. *Knowledge Management With Patterns*. *Communications of the ACM*, 46(7), July 2003. pp94-99.
- [5] Alexander, Christopher. *The Timeless Way of Building*. Oxford University Press, 1979.
- [6] 細谷竜一 and 中山裕子 (監訳). *プログラムデザインのためのパターン言語*. ソフトバンクパブリッシング, 2001.
- [7] パターンワーキンググループ. URL: <http://patterns-wg.fuka.info.waseda.ac.jp/>.