

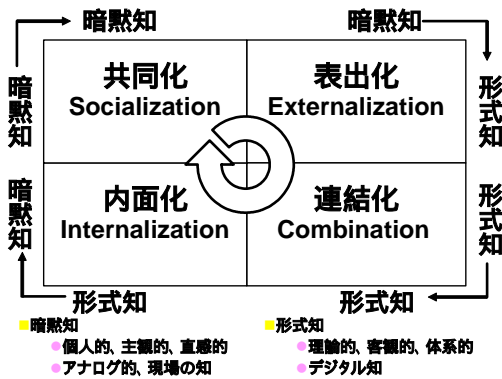
パターン活動とナレッジマネジメント

佃 軍治†

1. はじめに

パターン活動とナレッジマネジメント (KM) の関係について既にいくつか報告されている [1][2]。本稿ではパターンとKMに関する筆者の考えを述べる。

KMでは、図1に示すSECIスパイラルにより知識を増幅させるモデル[3]が有名である。本稿でもこのモデルに従いパターン活動を説明する。



ただし、通常はパターンを記述する場面よりパターンを利用する場面が先に訪れるため、本稿では共同化でなく内面化を基点にして説明する。

2. 内面化 (Internalization)

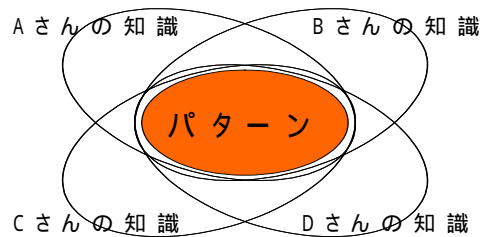
知識を内面化する場合、知識そのものの正しい理解が前提になる。知識の理解において、パターンの記述形式 (文脈、課題、解決策、フォース、結果) は有用である。しかし、記述形式が整備されたことにより安易に表面的に理解するという危険があることを忘れてはならない。表出化されたパターンは暗黙知のエッセンス部分であり、記述からこぼれ落ちた暗黙知を類推し、パターンの心を掴み取ることが重要である。そのためには、そこに記載されている課題と解決策にだけ目を向けるのではなく、類似の課題にパターンを適用しようとしたらどのようなことになるのか、別の解決策はないのか、自己の暗黙知を融合させることができな

いか、など様々なことに思いを馳せる必要がある。また、このときに考えたことや気づいたことは、実適用、共同化、表出化などの各種場面で活用できるので、これらの内容をメモしておくことは重要である。

理解した知識を実適用し、知識を自己の暗黙知に変換することにより内面化のステップが終了する。しかし、すぐに知識を適用する場面が現れるとは限らない。知識の理解と知識の適用に時間的なずれが生ずる場合、上述の気づきメモが存在すれば、理解時に考えたことを脳裏にすばやく呼び起こすことができるので、実適用を効率よく実施できる。

3. 共同化 (Socialization)

チームで仕事を行う場合、チーム内で暗黙知の伝播が行われる。しかし、パターンの勉強会などの場でも暗黙知の伝播は行われる。パターン勉強会において暗黙知の伝播を効果的に行うためには、パターンそのものについての議論だけでなく、パターンを基点とした周辺知識を議論することが有効である。これにより勉強会参加者の幅広い知識 (図2) を参加者間に伝播させることが可能になる。



このような周辺知識の伝播も目的とした勉強会をソフトウェア工学パターンWGの勉強会で実践した[4]。この勉強会の特徴は、パターンに関して理解したこと、疑問点、気づいたこと、パターンから発想したことなどに関して各参加者に事前にメモを作成してもらい、集計したメモに基づき勉強会を進行させることである。これにより知識の共有化を効率よく実践できる。

PLoPなどのパターンコミュニティでは、ライターワークショップというパターンを推敲するためのワークショップを開催している。レビューア

† (株) 日立製作所 システム開発研究所

には題材となるパターンに対する具体的な改善策を提案する義務があり、各レビューアの提案内容を知識と捉えれば、ライターズワークショップもワークショップ参加者間の知識の共有化の場と考えることもできる。

4．表出化（Externalization）

表出化には、ふと思いついたコンセプトに基づき、暗黙知から表出した解決策をラフに記載するステップと、解決策を洗練させ、グループ・組織で再利用可能なものに洗練させるステップがある。

パターンコミュニティでは後者のステップを支援する活動として、シェファードイングとライターズワークショップがある。シェファードイングは、パターン界のベテランが電子メールベースでパターンライターに指導する活動であり、ベテランはライターが記載したパターンの良い点への賞賛や改善提案を積極的に行っている。ここで重要なことはパターンライターのやる気と暗黙知を刺激している点である。これによりパターンライターの持っている未統合の暗黙知がパターンに吸収され、パターンが洗練されていく。ライターズワークショップでも同様の効果がある。

洗練化させるステップでは、再利用範囲を明確にしてパターンを記述することがポイントとなる。パターンコミュニティで提案されているパターン記述形式やシェファードイング、ライターズワークショップは、形式知だけで知識を伝播させることを前提としているため、記述が難しく、記述量も多くなる。しかし、知識の再利用範囲が特定のプロジェクト内などに限定されるのであれば、対面コミュニケーションによりパターン記述を補うことが可能になり、形式知のみを前提とした場合ほど詳細に記述する必要はない [5]。要はコミュニケーション量に応じた最適なパターン記述量であればよい。これは、アジャイル開発方法論による設計ドキュメント量と同じことである。ただし、プロジェクト解散と共に他で再利用すべき知識が失われてしまう事態を避けるために、それらの知識に関しては、プロジェクト終了時にプロジェクト終了後に想定されるパターンの再利用範囲でのコミュニケーション量に応じてパターンの記述量を変更する必要がある。

パターン形式に基づいてパターンを記述することにより、自己の暗黙知の思い違いや矛盾などが判明する場合がある。これは自己を成長させるためのよいトリガになる。自分自身を成長させるためにも暗黙知の形式知化を積極的に行うべきである。

表出化の前者のステップ（解決策のラフスケッチステップ）では、知識の理解と適用による内面化ステップや、プロジェクト内の会議、勉強会、ライターズワークショップなどでメモしたちょっとした気づきが役に立つ。気づきは解決策を思う描くときのトリガになるからである。しかし、気づきは、意識下に潜在しやすく、再び同じことを気づくまでには時間がかかる。メモした気づきを頻りに読み直し、気づきを意識させることが表出化の前者のステップでは重要になる。

5．連結化（Combination）

連結化には、コンセプトに基づいてパターンを組み立てる場合と、類似のパターンや同時に利用することが多いパターンへのリンクをそのパターン記述に含ませる場合がある。前者はパターンランゲージと呼ばれ、単なるパターンの集合よりも強力に作用する。GoFのデザインパターンは後者の例になる。

6．パターン適用に向けて

パターンは知識の流通において、強力な道具になりえる。しかし、プロジェクトで課題が生じたときにパターンリポジトリから必要なパターンを検索し、適用すればよい、という考え方には賛同できない。表出化されたパターンの背後には暗黙知が潜んでいる。これを含めてパターンを理解しないと、パターンの誤適用や硬直したパターンの適用しかできなくなるが、課題発生時にはパターンを十分に理解する時間的余裕がない。やはり、常日頃からパターンに触れ、暗黙知を刺激しつつ知識を増やす努力が必要である。この際、広く浅くパターンを理解するよりも、各パターンをじっくり味わい、とっさに使うことのできる知識を増やすことの方がより重要である。

参考文献

- [1] 山田正樹、知識創造とソフトウェア開発、Software People Vol.2 pp38-48、2003
- [2] Daniel May & Paul Taylor, Knowledge Management with Patterns, Communications of ACM Vol.46 No.7 pp94-99, 2003
- [3] 野中郁次郎他、知識創造企業、東洋経済新報社、1996
- [4] 細谷竜一、金澤典子、沖田直行、佃軍治、形から入らないパターン活動綴り方教室、002003 シンポジウム チュートリアル
- [5] 細谷竜一、形から入らないパターン活動、第141回ソフトウェア工学研究会