

連載「オブジェクト指向と哲学」

第13回 知識とは何か（7） - 暗黙知

河合 昭男

<http://www1.u-netsurf.ne.jp/~Kawai>

前回に引き続き、今回もプラトン「テアイテトス - あるいは知識について」[1]を考えてみたいと思います。

テアイテトスは「知識とは何か」という問いかけに順番に3つの説明を試みます。

- ①感覚 perception
- ②真なる思いなし true belief
- ③真なる思いなしに言論が加わったもの true belief with the addition of a rational account

これらは結局すべて棄却されて本書は終了してしまいます。

ソ：・・・従って、知識であるのは、テアイテトス、君のいう感覚でもなければ、また真なる思いなしでもなく、そうかといってまた真なる思いなしに言論の加わってできるものでもないということになるだろう。

テ：ええ、そういうことになるようです。[210A]

②と③はメノン[2]の「徳とは何か？」にも同様の議論がありました。

●知識は教えられる

前に参照したメノンの「徳とは何か？」という対話の中で、徳とは知識であるというひとつの説明がありました。しかし知識なら教えられるはずであるが、徳があると思われている人達の子は必ずしもそうでない。徳が教えられるものならば自分の子に教えないはずはない。つまり徳とは教えられるものではない。だからそれは知識ではない・・・という議論がありました（連載第9回参照）。

その次に、徳ある人は正しい思わく(correct opinion)を持っており、それが人に伝えることができないのだとしました。メノンでは、知識とは静的なもので言葉で表現できるもの、思わくは動的なもの、人の振る舞いに関するもので言葉に表現できないもののようなのです。

この正しい思わくはテアイテトスの②真なる思いなし(true belief)と同じもののようなのです。

言葉で表現できない真なる思いなしに言葉で表現できるものを合わせたものが③となります。

●UMLで整理する

ここまですをUMLで整理します。図1左は連載第9回のもので、右はテアイテトスの③です。左は徳そのものではなく徳ある人なので、右の知識と同等には比較できないのですが、並べて見ました。また左の属性区画に入れた知識は右の知識ではなく、右の知識の属性の区画に入れた言葉で表現できるものです。操作の区画に入れた正しい思わくと真なる思いなしはどうやら同じもののようです。

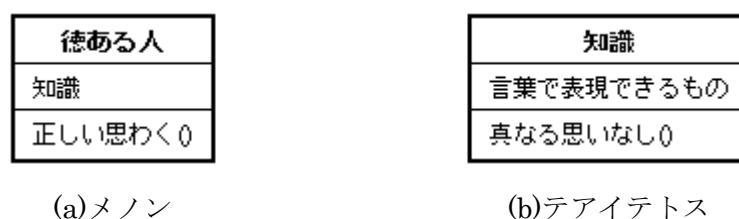


図1 徳ある人と知識の対比

つまり言葉で表現できるものとそうでないものをセットとして持っているとか何か強力なパワーを発揮する。むしろ言葉で表現できないものの方が重要な役割を果たす。

●具体例の本質とは

徳のある人なら具体的に例を挙げられますが、ソクラテスの問いかけは徳そのものの説明です。知識も例えば具体的な医者や大工の知識、靴職人の知識などの説明ならばできますが、知識そのものの説明はテアイテトスにはできませんでしたしソクラテスも説明しません。

ソクラテスは、人に対しては自分は知らないから、もしも知っているなら教えて欲しいと識者に頼み、次々と質問を繰り返して相手の説明では充分でないことを対話の相手にも気付かせます。

しかし、ソクラテスは人に対して言っているように、本当に無知であったとはとても信じられません。そんな筈はあり得ません。自分にはじつは徳も知識ももちろん分かっている。残念ながらそれが言葉で説明できないものだという事も分かっている。しかし何とかしてその大切なものを人に伝えたい。

自分こそは徳ある人である、あるいは知識を備えた人であると思っている一部の人がじつはそうでないのだと言いたい。ストレートにそういっても理解できないし、受け入れられることはない。なぜなら大抵その人達は本当に徳や知識を備えた人でないからです。

●暗黙知

結局、正しい思わくや真なる思いなしとは暗黙知なのです。暗黙知とはなにか？それは言葉で表現できないもので、知識の奥にあるものです。

「我々は語ることができるより多くのことを知ることができる」[3]

例えばある人の顔を知っている。その特徴をことばのみで他の人に伝えるのは難しい。説明できないから知らないということにはならない。

頭で覚えたものを形式知、体で覚えたものが暗黙知であるということが出来ます。自転車の乗り方は言葉では説明しきれません。練習して体で覚えます。

ソクラテスの対話編に出てくる知識は一見形式知のようです。正しい思わくや真なる思いなしは暗黙知です。しかしソクラテスの意図している知識は形式知と暗黙知を含んだ知識です(図2)。

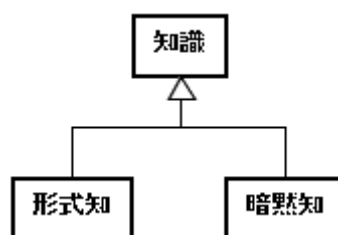


図2 形式知と暗黙知

●産婆術は暗黙知形成の術

暗黙知は教えられるものではなく自分で習得するものです。ソクラテスの産婆術とは暗黙知を習得させるための術です。適切な質問を繰り返し、対話者は苦しみながら自分で考え抜いて暗黙知を形成してゆくのです。

メノンの冒頭を思い出します。

メノン：こういう問題に、あなたは答えられますか、ソクラテス。人間の徳性というものは、はたしてひとに教えることのできるものであるか。それとも、それは教えられることはできずに、訓練によって身につけられるものであるか。それともまた、訓練しても学んでも得られるものではなくて人間に徳が備わるのは、生まれつきの素質、ないしはほかの何らかの仕方によるものなのか…。[70A]

テアイテトスの最後の場面です。

ソ：それでは、知識について、どうだね、君、僕たちは産むものを何かまだお腹のうちにもっているかしら。産もうとする苦しみはまだ僕たちにのこっているかしらん。それとも、産むだけのものはもうすっかり産んでしまったのかしらん。

テ：はい、それは神明に誓って申し上げますが、それでもうすっかりなのです。少なくとも私は、私が私の中にもっておりましただけのものというよりは、それ以上のものまであなたのおかげで口にできてしまいました。[210B]

口にしたものはすべて棄却されてきましたが、テアイテトスは口にはできないものをしっかり暗黙知として体得した筈です。

ソ：それならば、この後、テアイテトス、もし君が他のものをお腹にもつようにしようと試みるものがあって、もしそれをもつようになるとしたならば、君は今のこの吟味のおかげで、もっとよいものをもって充たされることになるだろうし、またもしお腹が空のまま産まれるものできない場合には、君は君の知らないものを知っていると思ったりしないだけの思慮深さを持つことによって、一緒にいる人たちを悩ますような重荷となることが一段と少なくなって、人々と一層よく折り合っていけることになるだろう。つまり、僕の技術でできるのは、ただそれだけのことであって、それ以上はなんにもできないのだ。そしてまた僕も、他のおよそ過去現在にわたる偉大な驚異すべき人物の知っているようなものは何ひとつ知ってはいないのだ。

この暗黙知形成の術、何も生み出していないようでじつは大切なものをしっかり体得させる術がソクラテスの産婆術です。

【参考書籍】

- [1] プラトン著、田中美知太郎訳、テアイテトス、岩波文庫、1966
- [2] プラトン著、藤沢令夫訳、メノン、岩波文庫、1994
- [3] マイケル・ボラニー著、佐藤敬三訳、暗黙知の次元-言語から非言語へ、紀伊国屋書店、1980

連載「オブジェクト指向と哲学」 第14回 学習パターンと SECI モデル

河合 昭男

<http://www1.u-netsurf.ne.jp/~Kawai>

前回は、知識には表面的な形式知とその奥にある暗黙知があり、ソクラテスが追求している知識とはどうやら暗黙知らしいというお話でした。

形式知は言葉で表現できるが暗黙知は言葉で表現できない。だから知らないとはならない。

「我々は語ることができるより多くのことを知ることができる」 [3]

ではどうやって暗黙知を確認できるのか？それは実践できるかどうかです。実際に自転車に乗れるかどうかということです。

大工は大工の知識を、医者や医師は医者や医師の知識を、靴職人は靴職人の知識を持っているからそれぞれの職業が務まる。それぞれの知識とは「形式知+暗黙知」です。形式知は言葉で伝達できるが暗黙知は自分で体を使って時間をかけて習得するものです。

●パターン言語

当連載第1回プロローグで述べましたが、オブジェクト指向のみで世界を捉えるには何か欠けていることに気が付きます。パターン言語はその欠けている部分に当たりそうです。

世界を場と其中で活動するオブジェクトと捉えます。その場の力を扱う考え方にクリストファ・アレグザンダーが提唱するパターン言語が関係しそうです。パターン言語は氏のライフワークの初期の成果です。それは一里塚であり、氏の追求しているものは、人に働く場の力という直接目に見えないものなのではないかと筆者は感じています。

パターン言語の議論に踏み込む前にそのひとつの分かりやすい事例である「学習パターン[1]」から入ってゆきたいとおもいます。学習パターンは形式知と暗黙知との議論にもつながってゆきます。

●SFC 学習パターン

SFC 学習パターンは、慶應大学湘南藤沢キャンパス (SFC) に入学した新1年生がこれからどのように勉強してゆけばよいかを示すヒント集のようなもので、小冊子として全新入生に配布されました。学校として作成したものではなく井庭先生の指導の下で学生が主体となって作成したことが特徴です。内容はパターン言語の形式で 39 のパターンから構成され、すべてのパターンに以下図 2~図 4 に示されているようなわかりやすいイラストが付いています。工夫されたパター

ン名とこのイラストにより、一目でパターンの内容を推測することができます。

学習パターン詳細は SFC オフィシャルサイト[1]から公開されています。その他筆者の寄稿記事[2]をご参照ください。

この学習パターンは、公開された当初からパターン言語のよくまとめられた適用事例として注目しており、一般的にどこでも使えるのではないかと頭の隅に焼き付いていました。筆者は新人研修の講師と講師サポートをそれぞれ担当しましたが、その現場でこの学習パターンがひらめき、早速実践して「成程こんな時に使える」とその効果を実感しました。

●新人研修

個人的な話になってしまいますが、筆者はプログラマ向け新人研修の講師もしています。研修は座学とプログラミング演習の繰り返しが基本です。座学は形式知を言葉で伝達することで、プログラミング演習は受講生が体を使って形式知を暗黙知に変換する作業です。

頭で覚えた形式知は忘れやすいですが、体で覚えた暗黙知は忘れません。暗黙知は応用が利き、まったく同じ場面でなくても実践することができます。実践を繰り返すことにより暗黙知は更に深まります。研修後の現場での OJT がこの暗黙知を深める経験です。

研修の最後にグループで課題に取り組み発表を行います。数名のメンバーが各自の得意分野を分担し、共同でひとつの仕事を成し遂げるという体験をします。

グループ演習は各自が様々な経験を通して学びとった暗黙知をグループのメンバーで共有する場です。発表はあらたな形式知の創出です。

●SECI モデル

SECI モデルとは「知識創造企業」[4]で紹介されているナレッジ・マネジメントのプロセス・モデルです。創造性のある製品開発をしている先進的企業では、形式知と暗黙知の循環サイクル「連結化、内面化、共同化、表出化」により創発が行われているとするものです。

新人研修のプロセスは SECI モデルにうまく当てはまります (図 1)。

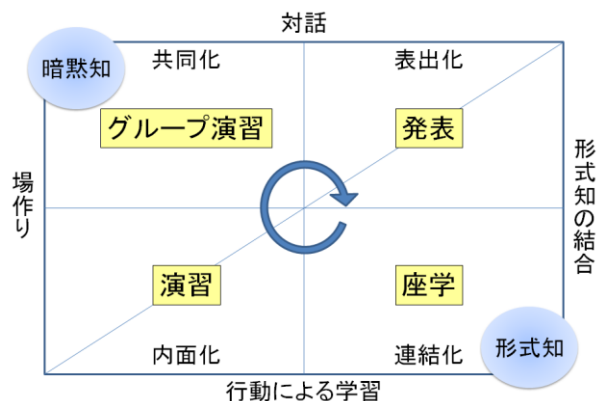


図1 新人研修のプロセスと SECI モデル

①座学

講師はテキストを使って形式知を受講生に伝達する。受講生は新たな知識を各自がすでに持っている形式知や暗黙知とつなげ**連結化**してゆく。

学習パターンには例えばふたつのパターンがあります。座学は「まねぶ」ことから(No.8) 始まり、**教わり上手になる(No.9)** ことで一層各自理解を深めることができる。



「まねぶ」ことから(No.8) 教わり上手になる(No.9)

図2 座学プロセスに使える学習パターン[1]

②演習

各自が PC でプログラミング演習を行いながらあらたな形式知を暗黙知に**内面化**してゆく。このプロセスにも学習パターンには例えばふたつのパターンがあります。

演習を通して**自分で考える(No.25)** 訓練を行い、手を動かし**身体で覚える(No.10)** で学んだ知識を自分のものにする。

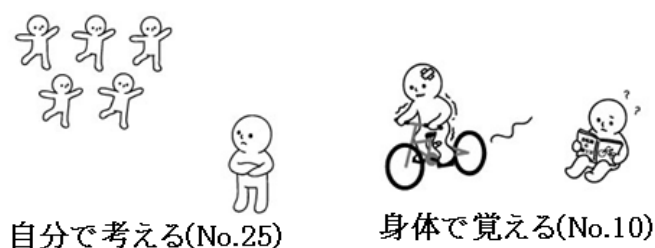


図3 演習プロセスに使える学習パターン[1]

③グループ演習

共同作業でドキュメントとプログラムを作成する。その過程でメンバーの暗黙知を**共同化**してゆく。

グループ演習は未知のメンバーとの**偶有的な出会い(No.18)** の場であり、演習課題を通してフィールドに**飛び込む(No.17)** 仮想体験を行い、学びを共有する。

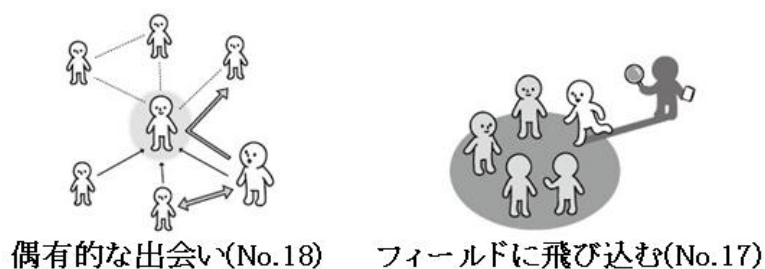


図4 グループ演習プロセスに使える学習パターン[1]

④発表

異なる暗黙知を持つメンバーが共同作業で作上げた成果には新たな知識が創発され、発表により形式知として**表出化**される。

成果物をまとめるという作業と発表を通して**アウトプットから始まる学び(No.13)**、**教えることによる学び(No.31)** を体験する。さらにどうしたらよりうまく伝えられるかという**魅せる力(No.34)** を工夫する場にもなる。

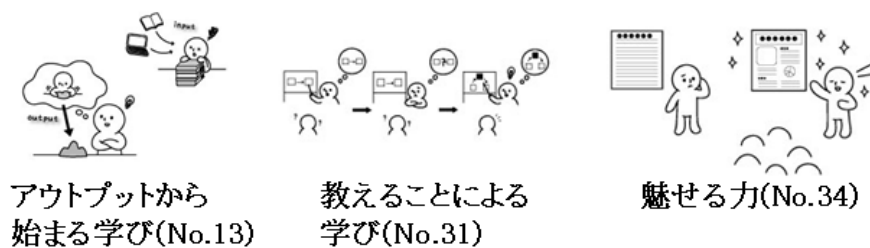


図5 発表プロセスに使える学習パターン[1]

今回は新人研修のプロセスが形式知と暗黙知の視点で SECI モデルに対応し、学習パターンが各プロセスの実践をサポートするということを、ほんの少しですが、示しました。

次回からクリストファ・アレグザンダーの世界に入ってゆきたいと考えています。

【参考書籍とサイト】

- [1]慶應義塾大学 湘南藤沢キャンパス 学習パターンプロジェクト、
<http://learningpatterns.sfc.keio.ac.jp/> (本稿のパターンのイラスト含む)
- [2] パターン言語事例 - 慶應 SFC の『学習パターン』、
<http://www.atmarkit.co.jp/im/carc/serial/world/28/01.html>
- [3]マイケル・ポラニー著、佐藤敬三訳、“暗黙知の次元-言語から非言語へ”、紀伊国屋書店、1980
- [4]野中郁次郎、竹内弘高、“知識創造企業”、東洋経済新報社、1996

連載「オブジェクト指向と哲学」 第15回 パターン言語 - 無名の質

河合 昭男

<http://www1.u-netsurf.ne.jp/~Kawai>

前回は、パターン言語の事例として「SFC 学習パターン」を取り上げました。特にわざわざパターン言語を意識することなく、学習法のノウハウ集として利用しても良い内容です。筆者はたまたまいくつかのパターンが新人研修のプロセスをカバーしていることに気付きました。さらに注意深く読んでみると、それらは SECI モデルのサイクルにも重なることを発見しました。

SFC 学習パターンを作成した人は特に SECI モデルは意識していなかったそうです。にもかかわらず結果として SECI モデルの 4 つのサイクルをカバーしてしまっている。それがパターン言語の潜在的パワーだと筆者は思います。一つひとつのパターンの力は小さくても、関連するパターンをまとめてネットワークにすれば大きな力となります。

今回は建築家クリストファ・アレグザンダー（以下 C.A.）が提唱したパターン言語のキーワードである無名の質を考えてみたいと思います。

● パターン言語の動機

C.A.の問題意識は、鉄とガラスとコンクリートでできた近代建築には何か大切なものが欠けているという疑問がひとつの開始点です。人間疎外している。そこからは住民の活気ある生活が生まれてこない。そこで実際には生活しないアーキテクトが設計し、そもそも住民は設計に参加できない。住民の生活よりハードウェアの方が重視される。

建築に対する価値感にはアーキテクトと住民で異なるであろう。しかし近代建築の価値観はアーキテクト側が決める、住民はそこに文句を言わずに黙って住むしかない。

では住民側に立った町や建築の価値とは何であろう？

C.A.はパターン言語を 3 部作として世に問いました。彼の理想とする町や建築のイメージや価値は、その一冊「時を超えた建設の道 (The Timeless Way Of Building)」[1]にちょっと変わったスタイルで表現されています。

● 無名の質

人に感動を与える町や建築って何だろう？世界遺産となるようなすばらしい町や建築は一体どうしたらできるのだろうか？

これはかなり重い問題です。もっと身近な例では、買い物や食事をするとき、はやっている店とそうでない店の違いは何だろう？やはり人がいっぱい入っている店で買い物をし、行列のでき

るレストランに並びます。その差は販売している商品やレストランのメニューだけではない何かがある。

人はただそれが欲しいからだけでものを買うのではなく、その店だから買いたくなることがあります。人はただコーヒーを飲みたいだけで喫茶店にはいるのではなく、例えばスターバックスで一時を過ごしたくなることがあります。何か活気がある。店の人も客も生き生きしており (alive)、店舗の内装も統一感があって店員も客もその場で一体感が感じられ (whole)、居心地の良さを感じます (comfortable)。

建築物という単なるハードウェアだけでなく、そこに参加する人も含めた「場」として、そこに備わる品質特性というものを考えます。これはなかなかひとことで表現できません。C.A.はこの品質特性そのものの定義はできないのでとりあえず「無名の質-QWAN (Quality Without A Name)」と名付けました。そこには次のような7つの品質特性が含まれている[1]。

- 1.生き生きとした alive
- 2.全一的 whole
- 3.居心地のよい comfortable
- 4.捕われのない free
- 5.正確な exact
- 6.無我の egoless
- 7.永遠の eternal

無名の質とはどうやらこれらの7つの特性の共通部分の中にあるものらしい (図1)。

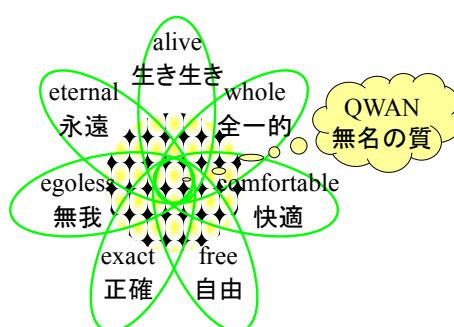


図1 無名の質は7つの特性の共通部分に含まれる

UML では汎化関係で表現できます。無名の質は7つの特性を継承します (図2)。

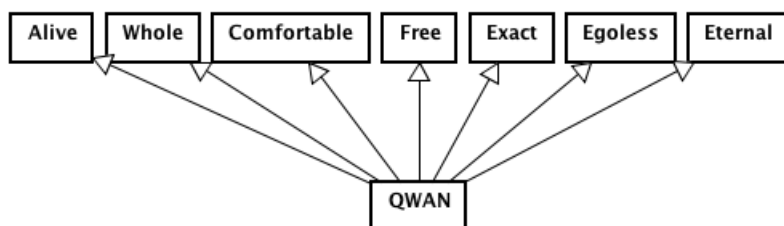


図2 無名の質は7つの特性を継承する

●無名の質と徳

無名の質の発想はソクラテスの「徳とは何か？」の議論とそっくりです。C.A.の探し求めている質にはまだ名前が付けられていませんが、徳はまず名前があります。その違いはありますが、両者共その本質が何であるかという説明はうまくできません。徳のある人なら例を挙げることができますが、徳そのものを説明することができず、正義・節制・敬虔・知恵・勇気などをすべて満たすものの中にあるという説明がなされます。当連載第5回の図を再掲します（図3）。無名の質の図とそっくりです。

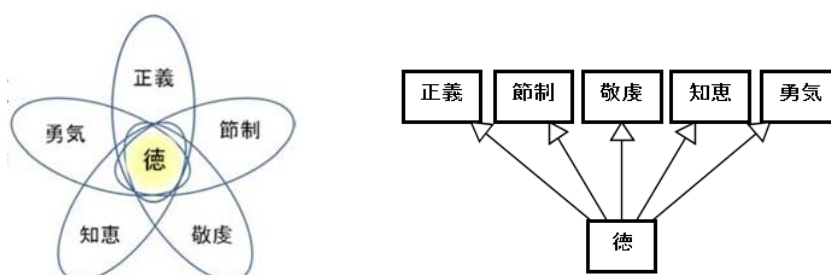


図3 徳は5つの徳目を継承する

●生き生き - alive

QWANの中でこれが一番重要な品質特性です。はやっている店は活気があり、店員も客も生き生きしています。行列のできるレストランもそうです。店舗やレストランという場として活気があります。店員の顔が輝いています。そういった場にはさらに人が吸い寄せられ、集まってきます。

●働きたい会社ベスト100

Fortune という雑誌で毎年 100 Best Companies To Work For のランキングが発表されます。日本での就職希望者の多い会社のランキングとは価値観が異なるようです。日本は安定志向・大企業志向ですが、米国では小規模の会社もランキングされます。日本ではあまり知られていない会社がほとんどですが、ちなみに本年は Google が 1 位に返り咲きました。おなじみの Starbucks もランキングされています。

□

筆者はここにランキングされている小売店舗の見学ツアーに参加したことがあります。その中

でも印象に残っているのはニューヨーク郊外の食品スーパーStew Leonard's です。入り口には100 Best Companies To Work For に選ばれたことが誇らしげに掲示されています。その後10年連続で選ばれています。当スーパーはNY周辺にわずか4店舗あるだけで、会社規模は大きくありません。にもかかわらず米国の就職人気企業です。特定の契約農家から仕入れるのがポリシーなので生産規模は限られており、店舗数は増やせないそうです。

●Stew Leonard's

筆者が訪れたのはマンハッタン北部のヨンカーズ店で、もともと牧場のあったところで周辺にはなにもない、都心からも離れていてちょっと買い物にゆく距離でもありません。中に入るとちょっとしたジャングルの露天市場というテーマパーク風の感じで、そのあたりの木から取り立ての果実を店に並べた風のレイアウトで、通路はくねくねしていてこの先何があるのかも見えません。わくわくします。活気 (alive) があり、店員と客が、そこはニューヨーク郊外ではなく南国の島か東南アジアに居るのではないかといった店舗と人との一体感 (whole) があります。居心地の良さ (comfortable) を感じます。ここには確かに QWAN があります。

--

今回はパターン言語のキーワードである無名の質-QWAN について筆者の感じていることを述べました。

【参考書籍とサイト】

クリストファ・アレグザンダー 3 部作

[1] The Timeless Way of Building,1979

□邦訳「時を超えた建設の道」鹿島出版会

[2] A Pattern Language,1977

□邦訳「パタン・ランゲージ - 環境設計の手引き」鹿島出版会

[3] The Oregon Experiment,1975

□邦訳「オレゴン大学の実験」鹿島出版会

□

連載「オブジェクト指向と哲学」

第16回 パターン言語 - ソフトウェアへの浸透

河合 昭男

<http://www1.u-netsurf.ne.jp/~Kawai>

今回は、パターン言語のキーワードである無名の質について考えました。「パターン言語とは何か」の説明は後にして、外堀から埋めてゆきます。

パターン言語は本来の建築の世界よりもむしろソフトウェアの世界に取り込まれてきています。その流れは大きくふたつあります。第1はデザイン・パターンから始まった一連のソフトウェア・パターンの一大潮流です。第2は一連のオブジェクト指向開発プロセスへの影響です。統一プロセス RUP や、その対極として生まれた XP から始まったアジャイル開発ブームにパターン言語の考え方がとり込まれてきました。

●デザイン・パターン

’94年に出版された GoF のデザイン・パターンはソフトウェア開発の世界に一大センセーションを引き起こしました。

--

“Design Patterns – Elements of Reusable Object-Oriented Software”[1]

邦訳「オブジェクト指向における再利用のためのデザインパターン」[2]

--

この書籍はクリストファ・アレグザンダー（C.A.）のパターン言語に触発されてまとめられたものです。同著の始めに C.A.からの引用があります。

「それぞれのパターンは我々の身のまわりで何回も起きる問題、および、それぞれの問題に対する解法のポイントを記述している。そこで我々は、これらの解法を何万回でも使うことができる。同じ問題に対する同じ解法を何度も何度も最初から考え直さずに済むというわけだ」[2]

オブジェクト指向が「ソフトウェア部品の再利用技術」なら、パターンは「知識の再利用技術」と言える。

単純化すれば「問題と解法」のペアを C.A.はパターンと名付け、そのアイデアがデザイン・パターンになったわけです。パターンという普通に使われている名詞に C.A.は別の意味付けを行い、それが支持されたわけです。

GoF のデザイン・パターンは、過去様々な開発で何度も使われた設計のテクニック調査してパターンという視点で 23 のパターンとして集大成したものです。

ちなみにこの書籍の初版では、各パターンのモデルは UML ではなく OMT 法で記述されています。UML0.9 が OMG に提出されたのが'96 年なので本書が出版された'94 年は正に UML の前夜です。オブジェクト指向開発方法論の統一が議論されていた時代です。

アーキテクチャ・パターン、アナリシス・パターンなど一連のソフトウェア・パターンが次々出版されてきました。従来の開発プロセスの問題に焦点を当てたアンチ・パターンという風変わりな書籍もやはり C.A. のスタイルを踏襲しています。

●パターン集とパターン言語

一連のソフトウェア・パターンはパターン・カタログとしてとても有益なものです。役に立つのだからそれが単なるパターン集なのかパターン言語なのかという議論は利用者にとってはささいな問題です。

UML で表すと、パターン集はパターンで構成されています。個々のパターンは単独で存在するのでなく他のパターンと関連します。(図 1)

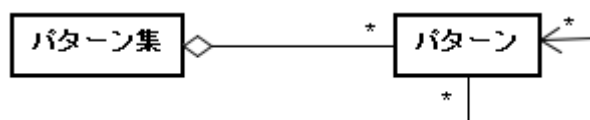


図 1 パターン集

パターン集とパターン言語の違いを UML で表現するのは困難です。パターン言語にはパターン集にはない何らかの特性と振る舞いがある筈です。(図 2)

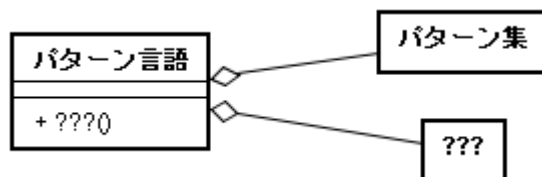


図 2 パターン言語

前回「SFC 学習パターン」をパターン言語の事例として紹介しましたが、これは単なるパターン集、ノウハウ集以上に強力なパワーが内包されています。一つひとつのパターンはそれなりに有益なものですが、相互に関連するパターンのネットワークが強力なパワーを発揮します。前回述べたようにパターン言語と SECI モデルは全く別々の概念ですが、アレグザンダー方式でパターンを整理してパターン言語化してゆくと、いつのまにか知らず SECI モデルの 4 つの象限をカバーしてしまっています。学習パターン (言語) はいつのまにか知識創造パターン (言語) に進化しているのです。

第 1 のムーブメントである一連のソフトウェア・パターンは必ずしもパターン言語化を目指し

たものではないようです。しかし第2のムーブメントであるソフトウェア開発プロセスへの流れはパターン言語を目指したものです。

●XP – eXtreme Programming

’99年に出版されたケント・ベックのXP[3]は、GoFのデザイン・パターンと並ぶ一大センセーションを引き起こしました。正にアレグザンダーのパターン言語なのですが、それを前面に出すと開発技術者は引けてしまうので、敢えてパターン言語という言葉を表に出さないでまとめられています。

有名な「ペア・プロ」、「オンサイト・カスタマ」、「テストイング」、「週40時間」など12のプラクティスがパターンです。これらは単独でも役立ちますが、相互に関連があり、すべてを最大限に、極端に(extreme)実践するのがXPという極端な開発プロセスです。

この12のプラクティスは、ソフトウェア工学などの理論から導き出されたものではなく、著者達の開発プロジェクトの経験に基づき集大成されたものなので、プログラマには説得力があります。

●場のクオリティ

C.A.の目指しているものは建物や町のハードウェアだけのクオリティではなく、そこで生活する住民も含めた場のクオリティです。場のプレイヤーである住民が生き生きと活動でき(alive)、住民と建物や町が全体として一体感があり(whole)、住民は居心地のよさを感じられる(comfortable)など人間系を重視しています。

●品質とは？

開発プロジェクトが目指すものはQCDです。顧客の要求を満足する高品質の製品を決められた予算で決められた納期までに完成させることです。

品質という言葉、当然ながら通常は製品の品質を意味します。しかし品質が必要なのは製品だけではありません。要求の品質というものもあります。要求の品質は製品の品質に大きく影響します。さらに開発プロジェクトの品質も製品の品質に大きく影響します。

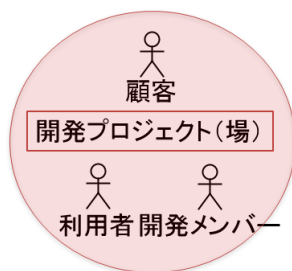
XPから始まったアジャイル開発がプログラマに支持されたのは、人間系も含めた開発の場のクオリティに焦点を当てたからです。クオリティのある開発プロジェクトからクオリティのある製品ができる。

●XPが目指す「場のクオリティ」

XPが目指すものは、開発チームのメンバーをいかに活性化させるか、モチベーションを上げるか、生産性を上げるかにあります。12のプラクティスは、開発プロジェクトという場のクオリティを作り出す源となる実践原理となるものです。プロジェクトのクオリティが結果として製品の

クオリティを創出します。

場としての開発プロジェクトの主なプレイヤーは、依頼主である顧客、完成したシステムの利用者、開発メンバーです。XPにはオンサイト・カスタマというプラクティスがありますが、このカスタマは依頼主と利用者の代表です。XPの12のプラクティスはこの開発プロジェクトのクオリティを作ります。XPがなぜプログラマに支持されているかという、人間重視の開発の場のプレイヤーみんなが生き生きと仕事ができ(alive)、メンバー間および開発する製品の一体感のある組織であり(whole)、開発メンバーは快適で居心地のよい環境で仕事ができる(comfortable)からです。



●ふたつの知識

パターン言語とは知識の形式化の技術です。組織および個人の経験や試行錯誤から蓄積された暗黙知から新たな形式知を創出する技術です。

第1のムーブメントはパターンを設計知識の再利用技術として取り込んだ。第2のムーブメントはパターン言語を開発プロセスの知識として取り込んだ。

【参考書籍】

[1] Erick Gamma 他, “Design Patterns – Elements of Reusable Object-Oriented Software”, ADDISON-WESLEY, 1994

[2]本位田真一、吉田和樹【監訳】、「オブジェクト指向における再利用のためのデザインパターン」、ソフトバンク、1995

[3]Kent Beck, ”eXtreme Programming eXplained”, ADDISON-WESLEY, 1999

連載「オブジェクト指向と哲学」

第17回 パターン言語 - 育てるとということ

河合 昭男

<http://www1.u-netsurf.ne.jp/~Kawai>

前回は、パターン言語のソフトウェアへの浸透をテーマとし、2大潮流があるとしました。

第1はデザイン・パターンから始まった一連のソフトウェア・パターンのムーブメント、第2は一連のオブジェクト指向開発プロセスへの影響です。統一プロセス RUP や、その対極として生まれた XP から始まったアジャイル開発ブームにパターン言語の考え方がとり込まれてきました。

知識の観点から捉えるなら、パターン言語とは知識の形式化の技術です。組織および個人の経験や試行錯誤を通して蓄積されてきた暗黙知を新たな形式知として創出する技術です。

第1のムーブメントは設計知識の再利用技術として取り込み、第2のムーブメントは開発プロセスの知識表現技術として取り込みました。

●アレグザンダーの3部作

クリストファ・アレグザンダー (C.A.) の3部作は C.A.の考える建築の価値、価値を創造するための基本戦略および具体的な基本戦術から構成されています。

--

[1] The Timeless Way of Building,1979

邦訳「時を超えた建設の道」鹿島出版会

[2] A Pattern Language,1977

邦訳「パタン・ランゲージ・環境設計の手引き」鹿島出版会

[3] The Oregon Experiment,1975

邦訳「オレゴン大学の実験」鹿島出版会

--

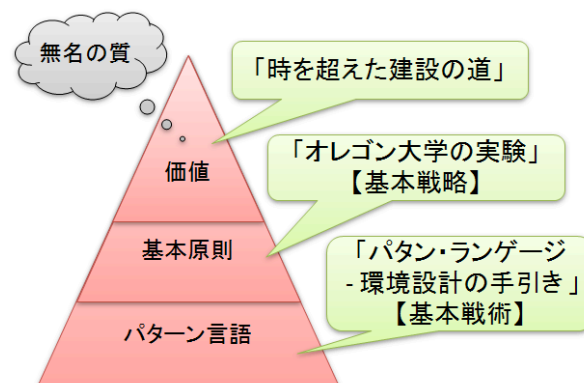


図1 アレグザンダーの3部作の位置付け

[1]は当連載第15回で紹介した**無名の質**がキーワードです。建築物というハードウェアだけの品質ではなく、そこで実際に生活する住民も含めた場のクオリティというものこそが求められるべきだとします。

[2]はずばり「パターン言語」をタイトルとし、具体的な253パターンが記述されています。このパターン記述形式は、ソフトウェア・パターンでも踏襲された問題とその解決策として記述するC.A.独特のスタイルです。

原題は“A Pattern Language”で“The Pattern Language”ではありません。パターン言語は対象とする問題領域ごとに自由に作れるものなので“Another Pattern Language”はいくつでも作れます。例えば、同じ建築がテーマでも具体的プロジェクト毎に利害関係者が集まって独自のパターン言語を作成することができます。本書はどこでも使える一般的な建築のためのものです。

●オレゴン大学の実験

[3]はプロセスがテーマです。パターン言語を用いて具体的にどのようにして建築を行おうとしているのかを説明する最初の書です。オレゴン大学プロジェクトの建設と計画に関するマスタープランとして示されていますが、他のプロジェクトでも活用できるものです。

「従来のマスタープランや予算処置に替わるべきものとして、これら6箇条の原理を採用するように私たちは勧めることとする。」[3]

プロセスの基本原則を6つの原理として挙げています[3]。

--

- | | |
|-----------|--|
| ①有機的秩序の原理 | The principle of organic order |
| ②参加の原理 | The principle of participation |
| ③漸進的成長の原理 | The principle of piecemeal growth |
| ④パターンの原理 | The principle of patterns |
| ⑤診断の原理 | The principle of diagnosis |
| ⑥調整の原理 | The principle of coordination |

--

さて、ここでは②と③に注目します。従来の建築はハードウェアの品質に重点があるが、C.A.はそこで実際に生活する住民が満足できる品質を重視する。その品質は人間系を含むものであり、具体的には説明困難であり、C.A.はとりあえず**無名の質**と名付けた。そのようなあいまいなものをどうしてアーキテクトが始めに顧客の要望を聞いて設計できるのあろうか？顧客も言葉ではうまく表現できないであろう。実際に完成してそこで生活してみないことにはわからない。

つまり住民参加でゆっくりと時間を掛けて作り上げてゆくしかない。それが②参加の原理+③漸進的成長の原理です。一気に作ろうとせず、育ててゆくのです。

●反復型開発

オブジェクト指向開発プロセスの特徴は反復型開発ができることです。これは統一プロセスRUPでもXPなどのアジャイル開発でも同じです。なぜ反復型開発にするのかは建築と同じです。利用者にとって、システムは実際に使用してみないとそれが本当に使いやすく有用なものであるかどうかはわからないからです。反復型開発のメリットは、全体完成前に途中で何度も利用者が実際に使ってみて評価しフィードバックできることです。反復型開発とユーザ参加はセットで効果を発揮します。実際XPにはオンサイト・カスタマというプラクティスが含まれています。

●場のクオリティ

C.A.の考え方は「建築物+住民」からなる場の人間重視のクオリティを、住民参加で時間を掛けて作り上げてゆくことです。システム開発も同様に捉えることができます。「システム+利用者」からなる場の人間重視のクオリティを、利用者参加で時間を掛けて作り上げてゆくことです。

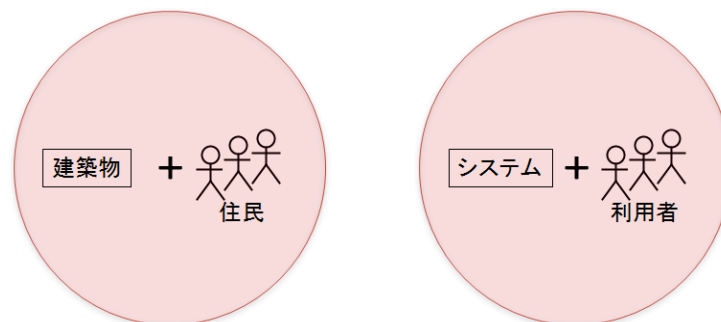


図2 人間系を含む場のクオリティを上げる

●花と種

「建物や町の無名の質は、人々の平凡な日常活動から間接的に生成されるだけで、人為的にそれをつくることはできない。ちょうど、花が種から生成されるだけで、人為的にそれをつくれないのと同じである。」[1]

建築もシステム開発も実際は予算と納期があり、ゆっくりと作ってゆく訳にはゆきません。直接的効果が数値化できないクオリティと、経済の制約のトレードオフの調整が問題となります。

●有機的秩序の原理

これは、人工物を作るのも自然のように始めから完成品を作るのではなく育ててゆくという考え方です。規模が大きくなるほど、利害関係者が多くなるほど時間を掛けて全体調和を保ちつつ成長させてゆかねばなりません。

「現在認められているようなマスタープランが、ある全体を創造し得ないこと、つまり、

ある総体性 (totality) を創造できても、全体性 (whole) は創造できず、また全体主義的秩序を生みだし得ても、有機的秩序は生み出し得ないことを論じて見たい。」[3]

“whole” は無名の質の2つ目に挙げられていた特性で、C.A.のキーワードです。

有機的秩序とは「部分の要求と全体の要求との間に完璧なる均衡が存在する場合に達成されるような秩序」[3]と定義している。

この考え方も C.A.の書籍の様々な場面で登場します。また稿を改めて考えて見たいと思います。

【C.アレグザンダー著書】

[1] The Timeless Way of Building,1979

平田翰那訳「時を超えた建設の道」鹿島出版会、1993

[2] A Pattern Language,1977

平田翰那訳「パタン・ランゲージ - 環境設計の手引き」鹿島出版会、1984

[3] The Oregon Experiment,1975

宮本雅明訳「オレゴン大学の実験」鹿島出版会、1977

連載「オブジェクト指向と哲学」

第18回 パターン言語 - 成長する全体と癒し

河合 昭男

<http://www1.u-netsurf.ne.jp/~Kawai>

前回は「パターン言語 - 育てるということ」をタイトルに、アレグザンダーが目指す価値を作り出すプロセスについて、パターン言語3部作の「オレゴン大学の実験」をテキストに、オブジェクト指向開発プロセスと重ね合わせて考えました。

参加の原理と漸進的成長の原理は、オブジェクト指向開発プロセスの雛型として完成された統一プロセス(RUP)やアジャイル開発にも入っています。そこで実際に生活する住民やシステムの利用者を含んだ場のクオリティを上げるための基本原理は同じです。

今回のテキストは「まちづくりの新しい理論」です。監修者難波和彦氏による「アンビバレントなオマージュ」と題された中身の濃いまえがきにはアレグザンダーの思想が凝縮されています。

●成長する全体という概念

シリーズ第6巻「まちづくりの新しい理論」は、サンフランシスコのウォーターフロントの一角を数年に渡って開発・管理を行うプロセスを想定したシミュレーションがベースです。

C.A.が目指す「成長する全体」には4つの特徴があります。[1]

--

1. 全体は少しずつ成長していきます。
2. 全体は予測できません。
3. 全体ははっきりしたまとまりをもっています。
4. 全体は情感に満ちています。

--

成長する全体のイメージを図にするなら調和がひとつの軸になります。全体とそれを構成する部分との間の調和、部分はそれを全体として構成するさらなる部分との間の調和、というように全体と部分の階層構造全般に細部までゆきわたる調和を保ちながらより大きな全体に少しずつ成長してゆきます。

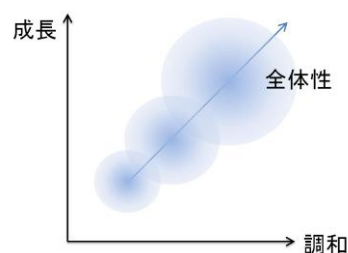


図1 成長する全体

前回挙げたプロセスの6つの基本原理（以下6原理）と比べて見ます。

1. 全体は少しずつ成長していきます。

6原理の②漸進的成長です。①参加の原理と合わせて全体調和を保ちつつ成長します。

2. 全体は予測できません。

従来型のマスタープランを否定します。しかし否定するだけで終わらないで、そこで「オレゴン大学の実験」や「まちづくりの新しい理論」の提唱です。

3. 全体ははっきりしたまとまりをもっています。

6原理の①有機的秩序つまり「部分の要求と全体の要求との間に完璧なる均衡が存在する場合に達成されるような秩序」[2]が内包されている状態です。全体が単なる部分の集まりではないというのはどういうことかを説明しようとしています。

この全体をどこまでの範囲ととらえるかという問題があります。そこで生活する住民は第1ですが、C.A.は建築の場所や環境を重視します。机上で設計したものを無条件にどこにでも建設することはありません。形は似ていてもその場所に最適な設計は世界にただひとつです。①有機的秩序は他の建築物（人工物）との関連+その場所や環境（自然）との適合性を含みます。自然との調和がとれていないと大きな成長はできないという考え方です。

自然との一体感について京都の東福寺を訪れたときの感想があります。

--

たとえば京都の東福寺はきわめて洗練されていて、自然環境からの連続体のように感じられる。・・・まるで誰かが埋もれていた自然の核心に手を触れ、そのまま何かを地上に持ちあげるようにして建物をつくってくれたかのようなのだ。[3]

--

4. 全体は情感 (feeling) に満ちています。

--

というのも全体性それ自体が、私たちに働きかけ、心の底に届き、私たちの気持ちを揺り動かし、涙を誘ったり、幸せにしたりする力を持っているからです。[1]

--

C.A.の重視する価値は人に軸足があります。人の感覚は論理的に捉えられない要素が大で、ここでは feeling ということばで表現しています。無名の質では alive と comfortable の部分です。

●成長と癒し

--

様々な場所で次々に行われる建設は、都市を癒すような方法でなされるべきである。(*1)

この「癒す」とは、「全体を取り戻す」という本来の意味で使われています。それはすでにある全体を修復するだけでなく、新しい全体を作りだすことも含んでいます。[1]

--

この「癒し(heal)」や「修復(repair)」も C.A.のキーワードのひとつで、時間をかけた漸進的成長プロセスの一部です。

(*1)は次のように言い換えることができます。

--

新しい建設行為は、すべてひとつの基本的義務を担っている。その周囲に連続した全体性のある構造を生み出さなければならないということである。[1]

--

癒しとは全体性を補強するものです。成長を担保するものです。

●癒しとリファクタリング

ちなみにオブジェクト指向開発にはリファクタリングという技法があります。一度作成したプログラムの振る舞いは変えずに構造を見直します。せっかく完成して動くプログラムができたのになぜそのようなことをするのか、それはソフトウェアは完成して納品すれば終わりではなく、不具合修正や機能拡張などライフサイクルを通して保守しやすい構造にするのが目的です。やはりソフトウェアにも漸進的成長があります。構造を修復し癒すのです。新たな全体を作りだすのです。

建築の癒しや修復は住民の生活を改善しますが、プログラムのリファクタリングは利用者には直接見えないという違いがあります。ライフサイクルで効いてきます。

このリファクタリングの強力なツールがデザイン・パターンです。デザイン・パターンは最初から設計に取り込むこともできますが、後から組み込むのがリファクタリングです。パターンがクオリティを生み出します。

●アーキテクチャ設計と癒し

オブジェクト指向開発を成功させるひとつの要はアーキテクチャ設計です。全体をモジュール(部品)に分割します。各モジュールはさらに小さなモジュールに分割します。特にインタフェースと実装が明確に分離されたモジュールをコンポーネントと呼びます。これがオブジェクト指向技術の特徴です。

コンポーネントの特徴は、外部からはインタフェースしか見えず実装は見えないことです。一旦実装が完成してから、インタフェースを変えずに実装のみ差し替えても利用側は修正不要なことです。図 2 の Component_A をリファクタリングして Component_A1 として差し替えても、

Interface_A を変えなければ利用側のクラス X は修正不要です。

コンポーネントは癒しが容易になる仕組みです。これは全体性に副作用的悪影響を与えることのない小さな癒しといえます。

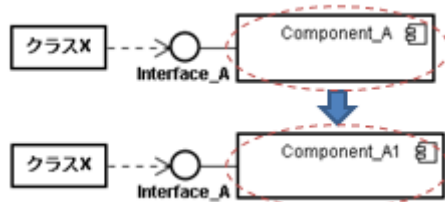


図2 インタフェースと実装を分離する

さらに新たなコンポーネントを追加してシステムを膨らませることができます。全体性に副作用的悪影響を与えないより大きな癒しといえます。

【参考書籍】

[1]C.Alexander, A New Theory of Urban Design,1987

□難波和彦監訳「まちづくりの新しい理論」鹿島出版会、1989

[2] C.Alexander, The Oregon Experiment,1975

□宮本雅明訳「オレゴン大学の実験」鹿島出版会、1977

[3]S.グラボー「クリストファー・アレグザンダー」工作舎、1989