

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

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. (クリストファ・アレグザンダー) 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。

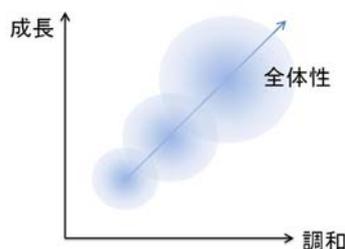


図1 成長する全体

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。

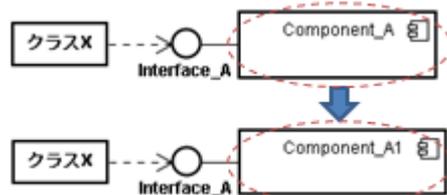


図 2 インタフェースと実装を分離する

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