

まちづくりから学ぶソフトウェア開発へのパターン活用法

Naoyuki Okita
Yokogawa Electric Corporation
Naoyuki.Okita@jp.yokogawa.com

Yuichiro Kato
Hakusan Corporation
ykato@hakusan.co.jp

1. はじめに

Alexander 読書会（以下、読書会）は、情報処理学会ソフトウェア工学研究会パターンワーキンググループの学際タスクの位置づけで 2003 年に活動を開始した。ソフトウェアパターンの源流であるまちづくりのパターンを学ぶため、建築分野の専門家を交えて、C. Alexander 著書の輪読、建築の見学等のフィールドワークを定期的実施している。読書会の活動を通じて、私たちは、まちづくりにおけるパターンの活用方法は、ソフトウェア開発におけるものとは大きく異なり、ソフトウェアパターンでは未開拓の部分で活用されていることを学んだ。建築の世界で得られた知見を、どのようにしたらソフトウェア開発に生かすことができるか考えてみたい。

本稿では、まず、まちづくりにおいて、パターンがどのような位置づけであるかを述べる。次に、パターンの活用方法を、「コミュニケーションツール」「アセット」「ライフサイクル」の3つのパターンとして紹介する。まちづくりにおけるパターンの活用方法をソフトウェア開発にどのように取り入れることができるか、可能性と課題を交えて考察する。

2. まちづくりにおけるパターンについて

本章では、まちづくりにおけるパターン(以下、まちづくりのパターン)の概要を紹介する。ここで説明する「まちづくり」とは、住民や利用者が積極的に参加する、ユーザ参加によるまちづくりである。ユーザ参加のまちづくりは、建築の世界では主流ではないが、実際に適用した例では、その成果はユーザから高い評価を得ていることが分かった。例えば、盈進学園東野高校やオレゴン大学では、ユーザ参加のキャンパス作りが支持され、現在まで継続している。この中で、パターンは重要な役割を担っている。(以降、本稿ではキャンパスなども含めて「まち」と表現することにする)

本稿では、まちづくりに関わるステークホルダーを次のように定義する。

表1 まちづくりに関わるステークホルダー

用語	説明
ユーザ	住民や施設の利用者など、日常的にまちに関わる人々の総称
建築家	ユーザと共にまちを設計し、施工者と共に建設する人
施工者	大工、土木工事者、庭師など、建築物(建物や敷地)を建築する人々の総称

2.1 まちづくりにおけるパターンの位置づけ

まちづくりのパターンは、まちづくりにユーザが参加するためのツールである。ユーザ参加のまちづくりは、ユーザと建築家と施工者の立場の異なる3者による共同開発である。立場の異なる人々による共同開発では、相互理解や合意形成が不可欠である。まちづくりのパターンは3者の相互理解や合意形成を助け、ユーザが望む街の建設や維持に貢献する。

一方で、ソフトウェアパターンはまちづくりのパターンを源流に持つが、独自の進化を遂げてきた。パターンが持つ記述力や知識共有の力にフォーカスすることにより、ソフトウェア開発者間の知識共有ツールとしての地位を確立した。たとえば、デザインパターンはプログラム設計者の基礎知識として広く知られている。ソフトウェア開発者の中で普及し、設計や実装の領域で成功を収めていると言える。

このように、まちづくりのパターンとソフトウェアパターンでは位置づけが異なる。表1にまちづくりとソフトウェア開発でのパターンの違いを5つのWの形式でまとめてみた。

表2 まちづくりとソフトウェア開発におけるパターンの違い

視点	まちづくり	ソフトウェア開発
なぜパターンを使うか(Why)	どのような街をつくるかの合意形成のため	どのようにソフトウェアを設計するかのため
誰がパターンを使うか(Who)	ユーザと建築家と施工者	ソフトウェア開発者
何にパターンを使うか(What)	「まち」そのもの(開発対象物の要求)	ソフトウェアの設計(開発対象物の設計)
いつパターンを使うか(When)	ライフサイクル全般	設計～実装
どこでパターンを使うか(Where)	ユーザとのワークショップ(打ち合わせ)、建設現場	開発部署内

2.2 まちづくりのプロセスとパターンの関係

ユーザ参加のまちづくりは、ユーザと建築家と施工者による共同開発のプロセスである。ここでは、ユーザ参加のまちづくりのプロセスの概要を示し、パターンがどのように関連しているかを示す。また、ソフトウェアの開発プロセスとの対応を示す。

街づくりのプロセス

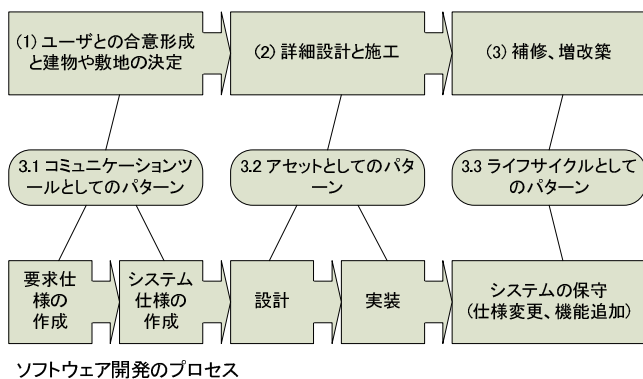


図1 まちづくりプロセスとパターンとの関係、およびソフトウェア開発プロセスと対応

(1) ユーザとの合意形成と建物や敷地の決定

ユーザと建築家は共同でワークショップを行い、ユーザの要求を表すパターンの有機的な集合、パターンランゲージを作り出す。個々のパターンは、例えば Alexander の書著「パタン・ランゲージ」に掲載されているパターンから選び出すか、新たに作り出す。

次に、パターンランゲージを活用してまちの構造を設計する。パターンは設計の方針を示すが、設計の詳細までは定めない。模型や、建築予定地での原寸大プロトタイプを作成し、ユーザの要求がパターンランゲージに十分反映されているかを確認する。この結果をもとに、具体的な建物や敷地の構造を決定していく。

ここで作り出されたパターンの集合は、その「まち」のパターンランゲージとなる。これはユーザ要求のリストであり、設計方針でもある。

(2) 施工

建築家と施工者は共同で、ユーザと合意したパターンに従って建物の内部構造や細部を設計し、施工する。必要に応じて、ユーザと建築家と施工者の3者で、細部を決定する場合もある。

このとき、パターンランゲージは3者をつなぐ、コミュニケーションツールおよび、「まち」の設計方針として機能する。

(3) 補修・増改築

まちづくりに終わりはない。ユーザと合意したパターンは、「まち」を補修・改修する際の指針となり、一貫性の

ある街並みを維持するために利用される。ユーザ要求の変化に対応して、パターンランゲージも定期的に見直される。パターンの入れ替えも行われる。

3. まちづくりのパターンと、ソフトウェア開発への適用可能性と課題

本章では、まちづくりのプロセスにおけるパターンの利用法を、パターンの形式で紹介する。各パターンについて、ソフトウェア開発への適用可能性と、それに際して生じる課題について考察する。

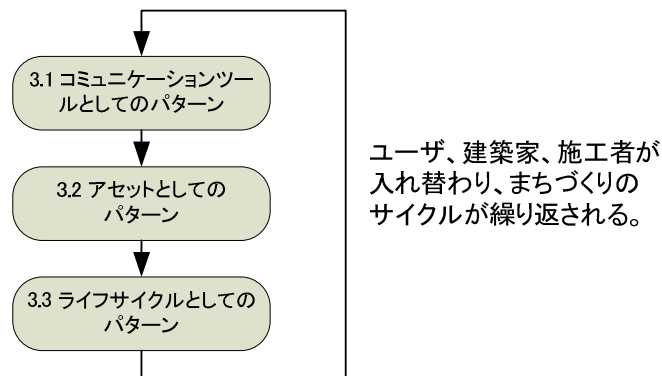


図2 まちづくりのパターンの関係

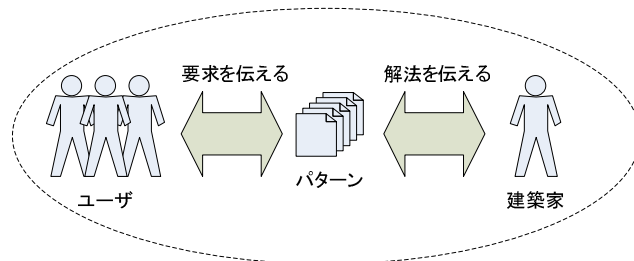
3.1 コミュニケーションツールとしてのパターン

ユーザ参加のまちづくりは、ユーザと建築家や施工者との共同作業である。この共同作業を実現するには、ユーザと建築家らのコミュニケーションが必須であり、パターンはその実現に役立つ。実際に、盈進学園のキャンパスづくりにおいては、教職員と建築家の間でのコミュニケーションに活用され、ユーザの要求掘り出しやキャンパスの設計に活用された。

ソフトウェア開発では、仕様リーダーがユーザから要求を引き出す際のコミュニケーションに相当する。

コミュニケーションツールとしてのパターン

Patterns as a communication tool



Context :

ユーザ参加のまちづくりを開始しようとしている。ユーザは自らの要求を建築家に伝えたい。建築家はユーザからまちに対する要求を引き出し、どのようなまちを建築するかをユーザと合意したい。

Problem :

要求を伝えることは意外に難しい。ユーザは要求(What)を伝える代わりに、解 (How)を伝えてしまいがちである。ユーザは限られた建築知識を元にして、建物や敷地のアイデアを出ししまい、真の要求を建築家に伝えることができない。このようにして獲得された要求を元にとると、まちの設計が不安定になる。ユーザの本当の要求を捉えていないためぶれやすく、設計変更が相次ぐためである。

Solution :

ユーザの要求とその解決ポイントをパターンとして記述し、ユーザとのコミュニケーションに使う。

パターンを記述するに当たり、パターンの形式でいう「Problem」の部分にユーザが要求する建築物と直面する課題を記述し、「Solution」の部分にユーザが理解可能な言葉で、建築物が満たすべき要件を書こう。パターンの「Problem」は、ユーザが自らの要求を引き出すための助けとなり、その要求を建築家へ伝え、共有することを助ける。パターンの「Solution」はユーザと建築家に対して、コミュニケーションをとりながら、まちを設計する方法を提供する。このようにして設計されたまちは、ユーザの要求が満たされていることが検証されており、まちづくりの基盤となりうる。

例として、「パタン・ランゲージ」の「手近な緑」のパターンを紹介する。「Problem」の部分は、ユーザは気軽に出かけられる緑地を要求している。さらに、緑地は近くにあれば利用されるが、遠すぎると利用されないとの課題を示している。「Solution」の部分は、すべての家や仕事場から徒歩3分以内の距離に公開緑地を作り、都市全域に分散させるとの要件を記述している。このようにして、ユーザの要求と課題に対して、建築家は具体的な提案をし、ユーザの要求を満たしているかを検証できる。

ソフトウェア開発への適用可能性と課題 :

ソフトウェア開発においても、ユーザ要求の獲得は重要な課題である。ユーザの「要求を記述する手法」はユースケースや USDM を初めとして開拓されているが、ユーザの「要求を引き出す手法」は開拓の余地があると思われる。そこで、ユーザと開発者間の双方向のコミュニケーションを実現するためのツールとして、パターンを活用できないだろうか。パターンを利用することに関しては、ソフトウェア業界は建築業界よりも普及しており、ユーザ参加のシステム仕様作成に貢献できる可能性を持っている。

このときの課題として、ソフトウェア開発ではパターンを作成する習慣が少ないことが考えられる。パターンは普遍的な知識であるべきとの思い込みが開発者にあり、パターン作成への敷居を高くしていると思われる。

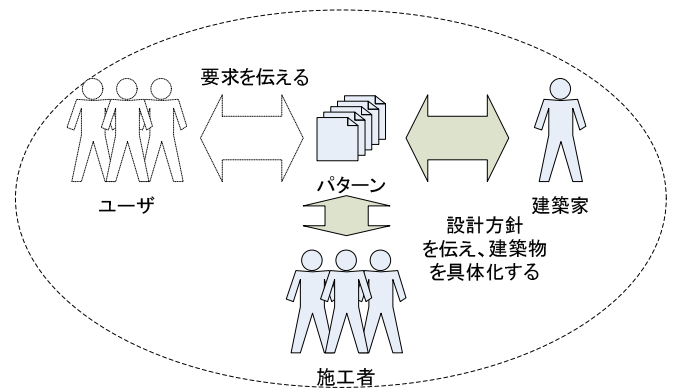
したがって、パターン作成への敷居を低くすることに取り組もう。第一に、ユーザとの双方向のコミュニケーションを主とし、ユーザの要求に対して、その理由を問いかけ、得られた回答を含めてパターンに記述しよう。第二に、パターンは開発中のシステムに適用できれば十分としよう。普遍的なパターンに抽象化する必要はない。

3.2 アセットとしてのパターン

ユーザと開発者が作り出したパターンは、システムの資産である。まちづくりにおいては、一貫性のある街並みや建築物の実現に貢献する。ソフトウェア開発では、一貫性のあるシステムの設計や実現に貢献する。

街の資産としてのパターンランゲージ

A pattern language as an asset of a town



Context :

建築家とユーザはパターンを通じて、まちの概観に合意した。建築家は施工者と共に、建築物を施工している。ユーザと建築家と施工者の3者で、建築物の装飾や手すりなどの細部を調整し、最終的な形を決定しながら、施工を進めている。

Problem :

ユーザ要求を満たしつつ、一貫性のある建築物を実現するには、2点の課題がある。

1点目は、ユーザ要求が施工者に理解されることである。建築物は数多くの施工者によって施工されるため、ユーザ要求を設計や施工の具体的な方針として、すべての施工者に浸透させる必要がある。

2点目は、細部に対するユーザ要求のゆれを制御することである。最初から完全な建築物の設計図を作成して施工する方法は、ユーザ要求を十分には満たせない。建築物を施工する過程で、ユーザは細部への具体的な要求に気づくが、その時点で要求を反映することができなくなるためである。一方で、アドホックに細部を決定していくと、建築物の一貫性がなくなり、まち全体の価値を損なう恐れがある。

Solution :

ユーザと合意したパターンを、その街のパターンランゲージとして資産化しよう。パターンランゲージは、ユーザ要求の集合であるため、建築物の設計・施工方針となる。これは、建築家と施工者が共通の認識を持つことを助ける。

また、パターンランゲージは、ユーザと建築家と施工者が建築物の細部を決定する上でのガイドになるため、細部への要求のぶれをユーザ自身が自律的に制御できる。

ソフトウェア開発への適用可能性と課題 :

ソフトウェア開発においても、要求仕様書やシステムの機能仕様書は完全ではない。特に、異常や内部エラーなどの非正常系のケースでは、設計や実装の段階でシステムの振る舞いを決めることが多く発生する。このとき、開発者がアドホックに振る舞いを決定すると、システムの一貫性を損なう。似たような例外事項に対して、ある機能ではエラー終了し、別の機能では継続処理するような食い違いは避けたい。

多くのシステム開発では、設計ルールやコーディングルールの利用によって対応している。これらのルールは普遍的がゆえに、開発者のスキルを一定水準に底上げするには有効だが、きめ細かなユーザ要求を実現するには向いていない。

そこで、パターンランゲージを、システムの詳細仕様や設計実装を進める上での基本方針として活用できないだろうか。要件定義時にユーザと開発者で合意した事項についても、新たなパターンとして、パターンランゲージに加えることにより、ルールよりも柔軟な運用ができると思われる。

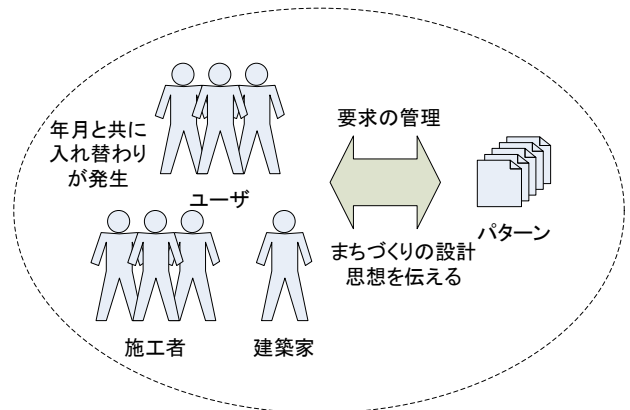
このとき課題として、システムレベルから機能実装レベルまで様々なパターンが作成され、粒度がまちまちであることから運用が難しくなることが考えられる。これに対しては、「パターン・ランゲージ」が町・建物・施工のようにレイヤを分割したのと同様に、粒度ごとにレイヤを分けよう。たとえば、システムの観点、コンポーネントの観点、内部設計や実装の観点に整理してみよう。実際に、内部設計や実装の観点では、パターンの利用が普及しており、受け入れる素地は十分にあると思われる。

3.3 ライフサイクルとしてのパターン

まちづくりやシステム構築に終わりはない。保守活動を現状維持でなく発展につなげるために、パターンランゲージは役に立つ。まちづくりにおいては、建築物の補修や改築を通じて、まちを発展することである。ソフトウェア開発では、システムの仕様変更や機能追加を通じて、システムを拡張することである。

まちのライフサイクルへのパターン

A pattern language for town's lifecycle



Context:

ユーザ参加のプロセスにしたがって、まちは建設された。まちづくりに参加した建築家や施工者たちは現場から離れ、ユーザの何人かも入れ替わった。これから先も、まちを維持し、発展させていきたい。

Problem:

まちづくりの当初の思いを維持しながら、まちを継続的に発展させたい。

まちづくりに終わりはない。まちのいたるところで、建設が行われる。したがって、当初の建設が終わっても、継続的なユーザ参加が求められる。しかし、まちづくりに参加するユーザは年月と共に入れ替わり、まちの一貫性を維持することは難しくなる。

このとき、属人性をなくすため、長期にわたる建設計画を固定したくなる。しかし、固定化された計画は早々に陳腐化する。まちの発展や世の中の流れと共に、ユーザ要求が変化するためである。計画当初にはない新たな要求が生まれる一方で、もはや必要とされない要求も出てくる。よって、まちを継続的に発展させるには、ユーザと要求の両方の変化に対処しなければならない。

Solution :

パターンランゲージを、まちを発展・維持するための建設方針としても活用しよう。定期的にパターンランゲージを見直し、無用になった要求を示すパターンは捨て、新たな要求を示すパターンを追加しよう。

パターンの「Solution」は建物や敷地が満たすべき要件を示しており、補修や改築時のチェックリストになる。パターンでは建物や敷地の詳細までは規定せず、適用する際にユーザと建築家らによって詳細を決定しよう。

1985年に建てられた盈進学園東野高校では、教員と生徒だけでなく保護者も参加したキャンパスづくりが現在も継続している。筆者らは何度か訪問したことがあるが、生徒や保護者らによりPTA会館の改修や遊歩道の整備などが継続的に実施され、キャンパスが改善される姿に驚いた。高

校は、3年で生徒や保護者が入れ替わるにも関わらず、ユーザ参加のキャンパス作りが継続されている源泉にパターンランゲージも寄与しているのではないだろうか。

オレゴン大学ユージンキャンパスでも、40年間に渡って、パターンランゲージが大学キャンパスを補修・改築するための方針として使われている。パターンランゲージは大学内のコミュニティによって定期的に見直されている。

ソフトウェア開発への適用可能性と課題：

ソフトウェア開発においても、開発対象のシステムの継続的な発展は共通した課題である。ユーザ要求に従って開発したシステムも、年月と共に陳腐化し機能の追加や変更を必要とする。システム開発当初のユーザや開発者が現場から離れたとき、最初の危機は訪れる。新たなユーザ要求に対応できないシステムは利用されなくなり、陳腐化が加速する。一方で、安易な追加・変更はシステムの一貫性を損ないやすい。システムの追加・変更にも、まちづくりと同様にパターンランゲージを活用できないだろうか。

このとき、(1)パターンランゲージを見直す間隔と、(2)ユーザや開発者の入れ替わりについての課題が考えられる。

(1)に関しては、適切な見直しの間隔は、システムが置かれた環境に依存するだろう。ユーザ要求の変化に対応できなかったパターンランゲージは使われなくなる。一方で、頻繁なパターンランゲージの変更は、ユーザや開発者の負担を大きくする。

(2)に関しては、パターンランゲージはユーザと開発者とのコミュニケーションを置き換えるものではなく、コミュニケーションと共にあることを意識しよう。一度に、ユーザと開発者全員を入れ替えることは避け、システムの語り部となる人を残そう。

また、別の可能性として、仕様変更そのものをパターン化することが考えられる。ソフトウェアの設計や実装を改善する手法としてリファクタリングが存在し、アジャイル開発を中心に普及している。仕様変更についても、リファクタリングのような手法は使えないだろうか。このとき、パターン「Solution」部分は仕様変更時のチェック項目として活用できると思われる。

4. おわりに

本稿では、ソフトウェア開発者に対して、まちづくりのパターンを紹介すると共に、ソフトウェア開発への適用可能性を考察した。

2章では、最初にまちづくりにおけるパターンの位置づけを紹介し、まちづくりパターンはユーザ参加のまちづくりのためにありことを示した。次に、まちづくりのプロセスとパターンの関係を紹介し、まちづくりのライフサイクル全般にパターンが関わっていることを示した。

3章では、まちづくりにおけるパターンの活用方法を「コミュニケーションツール」「アセット」「ライフサイクル」の3つのパターンとして紹介し、ソフトウェア開発への適用可能性と課題を交えて考察した。本稿では、幅広いソフトウェア開発者に向けてパターンの可能性を示すた

め、プロセス全体を俯瞰して、工程ごとに1つのパターンだけを紹介した。実際には、工程毎にパターンランゲージが存在すると思われる。

ソフトウェアパターンは、まちづくりのパターンを源流に持つが、目的や利用法の違いや、業界の違いによって、それぞれが異なる発展を遂げてきた。どちらが優れているかではなく、お互いの良い点を見習い、ものづくりの発展のために、パターンの可能性を求めていきたい。

Alexander 読書会では、ソフトウェア開発者と建築家が集い、パターンやアジャイルプロセスについての創発的な議論の場になることを目指している。本稿に興味を持ち、もっと突っ込んで議論したい方は、ぜひ読書会にご参加いただきたい。

謝辞 本稿のきっかけとなり、読書会を支え続けているメンバ各位に感謝します。

5. 参考文献

- [1] 中埜博: パタン・ランゲージによる住まいづくり, 井上書院(1988)
- [2] Christopher Alexander, 難波和彦(訳): まちづくりの新しい理論, 鹿島出版会(1989)
- [3] Christopher Alexander, 平田 翰那(訳): パタン・ランゲージ環境設計の手引, 鹿島出版会(1989)
- [4] Christopher Alexander, 平田 翰那(訳): 時を超えた建設の道, 鹿島出版会 (1993)
- [5] Christopher Alexander, 宮本 雅明(訳): オレゴン大学の実験, 鹿島出版会 (2000)
- [6] Hajo Neis: Urban Systems: Generative Urban Process: Systems of Rules - Projects, 明治大学特別講義(2010)
- [7] Martin Fowler, 児玉 公信(訳), 平澤 章(訳), 友野 晶夫(訳), 梅澤 真史(訳): リファクタリングプログラムの体質改善テクニック, ピアソンエデュケーション(2000)
- [8] Erich Gamma, Ralph Johnson, Richard Helm, John Vlissides, 本位田 真一 (訳), 吉田 和樹(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, ソフトバンククリエイティブ(1999)
- [9] 清水 吉男: 要求を仕様化する技術・表現する技術, 技術評論社(2005)