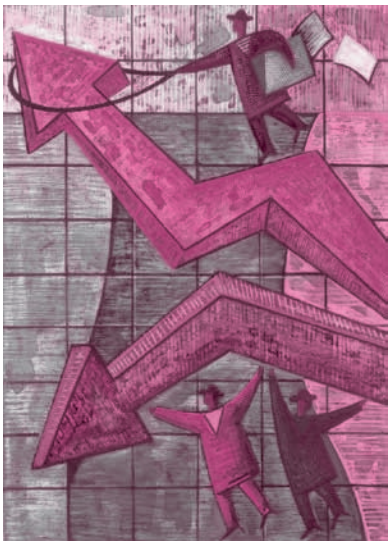

パターンランゲージから ソフトウェアパターンへ

江渡浩一郎 （独）産業技術総合研究所 社会知能技術研究ラボ

[パターン，Wiki，XP]

プログラミングにおける定石集「デザインパターン」が提唱され，久しい．デザインパターンはプログラミングに繰り返し現れる構造や設計を集め，プログラマが再利用しやすい形でまとめたものだが，提唱された当初はデザインパターンが本当にプログラミングを改善するものなのか，しばしば議論されていた．また，開発初期からユーザを参加させ，短期間にリリースを繰り返しながら環境の変化に対して適応的に開発をすすめる手法「アジャイルソフトウェア開発」も，一般に普及した．その実例が，エクストリームプログラミング（XP）やスクラム（Scrum）である．また，利用者が共同で Web サイトを構築するシステム「WikiWikiWeb」（以下 Wiki）は，その代表例であるインターネット上の百科事典「Wikipedia」の普及によって誰もが知っている存在へと成長した．

さて，デザインパターン，アジャイルソフトウェア開発，Wiki，この三者に共通するものは何かと聞かれたら，何と答えるだろうか．私ならば「パターンランゲージの影響から生まれた」と答える．実はこの三者は，一見異なる分野のものに見えるが，パターンランゲージに影響を受けて生まれたという共通点を持っているのだ．

筆者は 2002 年より Wiki に関する研究を続けてきた．その過程で，Wiki が持つ根本的な特性についての疑問につきあたった．Wiki が誕生した経緯を調べてみると，一見関係ないと思える複数の概念が結びついていることが分かった．このようなさまざまな概念が生まれてきた歴史を調べ，著書『パターン，Wiki，XP ―時を超えた創造の原則―』（技術評論社，2009）として公開したところ，大きな反響があった．本稿では，その内容を踏まえ，パターンランゲージがソフトウェア開発の分野に影響を与えてきた歴史の概略を紹介する．

■ 自分なりの振り返り

私がこの問題に興味を持つようになったきっかけは，Wiki だった．2002 年，オフィスの情報共有に Wiki が導入された．はじめは，簡素な見かけに戸惑い，ごちゃごちゃした中身に使いにくいと思ったりもしたが，使いこなすにつれて，その情報整理の威力を知ることになった．

私は当時，東京芸術大学で新入生のコンピュータ教育を担当していたのだが，そこに Wiki を導入することにした．学生はすぐに Wiki を使いこなすようになった．最終的には先生方も参加し，Wiki は第 2 のキャンパスとなった．その後，私はメーリングリストと Wiki を統合したシステム「qwikWeb」を構築した．これらの経験を通じ，私は Wiki の本質について考えるようになった．Wiki はシステムとしてはそれほど複雑なものではない．しかし，Wiki には独特の力がある．それはいったい何なのか．

私は Wiki の歴史を調べることにした．そして，私は，建築家 Alexander が 50 年ほど前に考え出した独自の建築設計手法に突き当たる．

いささか迂遠な道のりではあるが，Alexander による建築設計手法から順に，ソフトウェアパターンへの展開を追ってみよう．

[Alexander による建築設計手法]

■ 形の合成に関するノート

建築家 Christopher Alexander は，1936 年オーストリアのウィーンに生まれた．イギリスのケンブリッジ大学の数学科に入学し，数学の学位取得後，同大の建築学科に入学する．アメリカに渡り，ハーバード大学大学院の建築学科で博士号を得る．ハーバードの博士論文として提出された『形の合成に関するノート』[1] は，1964 年に書籍化され大きな反響を呼んだ．

ここで彼が試みたのは，デザインプロセスの数学的形式化である．彼は，デザインをコンテクストから適切な形を導きだす過程と定義し，従来デザイナが感覚や経験，勘などによって主観的に行ってきたデザインを，数学的形式に還元した．

最初に挙げている例は，ヤカンのデザインである．ヤカンがヤカンとして使えるためには，満たさなければならない要求条件として，次のようなものが考えられる．「水を入れられること」「持ち手がついていること」「平らな面に置けること」「簡単に加熱できること」など．また，製造過程においては「安価に製造できること」「丈夫に製造できること」といった要求条件がある．それぞれの要求条件は細分化することができる．たとえば，「水を入れられること」は，口がついていて，その中が水を入れられる大きさの容器になっていて，水が漏れない構造になっていることなどである．Alexander はこれらの要求条件をなるべく多く満たす形態がすなわち良いデザインであると考えた．

デザインの対象に関して，考え得るかぎりすべての要求条件（コンテクスト）を洗い出す．その際，既存の建築設計では，物理的条件と非物理的条件（社会構成や人間関係など）を分けて考えることが一般的であるが，ここではそのような区別を前提としない．こうして列挙された要求条件の全体集合を「システム」と呼ぶ．次に，要求条件同士の関係をすべて考慮する．その中から，互いに影響を与え合うと思われる要求条件を集め「サブシステム」（部分集合）に分割する．サブシステムは，可能な限りさらにサブシステムへと分割される．こうして，最終的にシステムはシステムとサブシステムの親子関係（ツリー構造）に分解される．最後に個々のサブシステムごとに最適の形態を求め，それらを合成するのだが，その際，ダイアグラムという図解が使われる．ダイアグラムはサブシステムごとの解を図式化して表したものである．ダイアグラムを複数重ね合わせることで，最終的な形態が決定されるというわけである．

ここではヤカンを例にとって説明しているが，実際にデザインしたい対象は都市である．彼はこの手法を用いて，インドにある人口 600 人の農村の計画案をつくった．農村が必要とする条件をフィールドワークによって 141 個見つけ出し，次にそれを 4 つのグループからなる 12 個のサブシステムに分割した．それぞれのサブシステムはさらに 10 個程度のサブシステムに分割されている．このツリー構造に対応するダイアグラムが作られ，最終的なデザインが示された．

この Alexander の方法論は多くの人の耳目を引きつけた．建築家の磯崎新は「ひとつの論文が，方法上の大変革をもたらすのはサイエンスの世界にはしばしばみられることだが，建築では稀なことである．彼の論文は，デザインの科学とでもいうべき領域に決定的な地平をひらいたという点において，はかりしれない影響をあたえている」[☆1] と述べている．

1964 年，Alexander はカリフォルニア大学バークレー校の教授となり，ベイエリア高速鉄道（BART）から，駅の要求条件の分析を依頼される．具体的には，駅舎の切符を売るチケットブースや改札，駅の入口などの関係性の計画で，30 個の駅舎が対象となっていた．彼は 390 個の要求条件を挙げ，これをサブシステムに整理しようとした．しかし，

☆1 磯崎　新：建築の解体，美術出版社，p.176（1975）．

ここで予想外の問題が発生した．たとえば，客はチケットブースに並んでチケットを買うが，列が膨らむと駅の入口から人が溢れ，人の通行が妨げられてしまう．関係が薄いと判断して切断したはずのサブシステム間に連携が生じてしまったのである．彼はここから，自分の方法論の欠点を発見する．要求条件を列挙し，条件間の関係性を導き出し，そこから解を求めるためには，いったんツリー構造へ還元し，階層構造にしなければならない．しかし，現実の都市空間は，必ずしも秩序立った階層構造に落とし込むことはできない，複雑に絡み合った構造をしていることに気がついたのである．

■都市はツリーではない

その反省を踏まえ，Alexander は 1965 年「都市はツリーではない」[2] を発表する．『形の合成に関するノート』の出版の時点で，すでに Alexander は有名だった．日本でも建築関係者周辺で『形の合成に関するノート』の海賊本，コピー本が出回っていたほどだ．しかし，この「都市はツリーではない」によって，彼の名声は決定的になった．

この論文において，彼は都市計画家によって設計された都市を人工都市，長い年月を経て作られた都市を自然都市と名付けた．そして，少数の専門家によって設計された都市の構造はすべてツリー構造に収まることを過去の都市計画の分析から明らかにした．反対に，自然都市は，1 つの要素に複数の意味が重ねあわされる．彼の言葉で言う「セミラティス」の状態になることを示した．その上で，人工都市には自然都市のような心地良さなどの質が備わっていないとし，都市はツリーであってはならないと批判した．

彼は高い質を持った自然都市の例として，カルフォルニアのバークレーや日本の京都を例にあげている．ここで，実際には京都のような都市も都市計画によってできあがったものだという点に注意しよう．碁盤の目のような道路からも分かる通り，京都は，少数の設計者が俯瞰的視点から全体を設計し，基盤の整備を行ってできた都市だ．つまり，人工都市と自然都市は，実際には単純に対立する概念ではない．Alexander が批判したのは，ツリー構造に還元されてしまうような抽象的な設計のされ方をした都市計画だった．

しかし，この時点で，Alexander は，ではどのようにすれば都市をツリーではない方法で設計できるのか，という疑問に対して，答えを持っていなかった．この後，彼は実践を通して，パターンランゲージという考え方に辿りつく．

■パターンランゲージ

自然都市のような質を備えた都市を設計するためにはどうすればよいか．その問題に答えを与えようとしたのがパターンランゲージであり，パターンランゲージを用いた利用者参加型の設計手法である．

パターンランゲージは，都市や建築において繰り返しあらわれる形態をパターンという形式でまとめたものである．Alexander を含めた 3 人は，8 年以上の歳月をかけ，建築において繰り返し現れる形態を 253 個のパターンとして収集し，1977 年に『パタン・ランゲージ』[3] として発表した．ここでいう形態は，一般的な建築の形にとどまらず．地球上での都市の配置から，街中の街路，建物の玄関や居間，屋根や壁の施工から，部屋の中での小物の配置まで，さまざまなスケールでバラエティに富んだ形態が対象となっている．

パターンには，ある状況において発生する問題を解決するために望ましいとされる形態が，自然言語で記述されている．1 つのパターンは，一定の形式（パターン名，写真，上位パターンへのつながり，本文，下位パターンへのつながり）によって記述される．他のパターンへのつながりが明記されていることによって，パターンの全体はネットワーク状の集合になる．パターンランゲージではランゲージという言葉が使われているが，ここには 3 つの意味が込められている．

1. 詩をつくるように都市や建築をつくる

Alexander は，パターンランゲージを用いた設計を詩の創作に喩えている．詩人が有限の単語を組み

合わせて無数の美しい詩を生み出すように，人々は，複数のパターンを組み合わせてすばらしい建築や都市を作り出すことができる．誰もが知っているように，言語は有限個数の文字の組合せから成り立っている．そのような制約があるからこそ，人々の間で共通して理解可能な表現ができる．同じように，パターンは，設計者に適度な制約と一定の自由度を与え，それが設計者の創造性を触発するきっかけとなる．

パターンランゲージに関するよくある誤解は，パターンをレゴブロックのような部品と解釈し，それをつなげるだけで建築を作れるようにしたものと理解してしまうことである．工場で部品をつくり，現場でそれを組み合わせるだけで短期間に施工できるようにすることをプレファブと言うが，それはAlexander が目指していたものではない．パターンは，現実の建築や都市を複数の人間が観察することによって発見・共有される言語化されたイメージであり，設計に参加する人々が変わればその結果も自ずと異なったものになる．

2. コミュニケーションの道具としての言語

パターンランゲージを用いることで，建築家と住民は互いの意図を伝え，議論が可能になる．図面や数式の分からない一般人であっても，パターンを記述することはできるため，パターンというドキュメントを共同編集していくことで，複数の人々の主観的な意見が客観的な事実に変換されていく．そのような伝達と記録，共有の媒体として，パターンランゲージは機能する．このように，複数の人間のコラボレーションをサポートする道具として使うという意図を持ってパターンランゲージはデザインされている．

3. システムを生成するシステムとしての言語

1967 年のアスペン・デザイン会議で，Alexander は「システムを生成するシステム（Systems Generating Systems）」という発表を行った．ここで彼は 2 つのシステムを定義した．ある対象を全体として捉えたときに見える構造「包括システム」と，そのような包括システムを生み出す「生成システム」である．自然言語は，言語が自己言及性を持つがゆえに，自分自身の言語そのものを更新していくことができる．ある単語や文章の意味は，他の単語や文章との関係性によって定義されるが，このように言語が言語に自己言及することで，言語システムの全体はつねに変容し続ける．つまり，自然言語は生成システムである．Alexander は，パターンランゲージもそのような「生成システム」であるべきと考えた．このような問題意識は，同時代のサイバネティクスや Gregory Bateson の思想と似通っている．

■時を超えた建設の道

Alexander は，1979 年に『時を超えた建設の道』[4] というパターンランゲージの背景にあたる理念を語った本を出している．この本はしばしば「分かりにくい」と評される．その理由の 1 つに，彼が明白に東洋思想の影響を受け，そこから都市や建築への考えを巡らせているという点が挙げられる．「道」や「門」といった老子の言葉の借用にそれがよくあらわれている．

しかし，ここで強調しておきたいのは，Alexander が徹底した形式主義者としてスタートしたという点である．建築や都市は，設計者の直感や経験，自己主張などによって作られるべきではなく，もっと普遍的で客観的な，定義可能な美に基づいて設計されるべきであると Alexander は考えていた．パターンランゲージもそのような普遍的美を備えた構築物を実現するための方法論として提起されたことを理解する必要がある．

日本では，Alexander の思想は，建築家の磯崎新によって，ほぼリアルタイムに紹介され，多くの人々に影響を与えた．

パターンランゲージを用いた設計の実作として，1982 年に開始され，1985 年に完成した盈進学園東野高校（埼玉県）がよく知られている．しかし，この仕事は Alexander にとっても満足のいくものではなかった．彼は利用者参加型プロセスの直営方式による漸進的な整備を主張したが，施工期間の長期化と建築費の高騰を招き，最終的にゼネコンが導入された．Alexander は日本の伝統的な木造建築技術を用

いることを主張したが，最終プランではコンクリート造との折衷となってしまった．理論面での多大な影響にもかかわらず，その後 Alexander の実作としての建築物が日の目を見る機会はほとんどなくなってしまった．

[パターンランゲージからソフトウェアパターンへ]

■オブジェクト指向プログラミングへの応用

さてここで，このようなパターンランゲージをソフトウェアの世界に展開した 2 人の研究者を紹介したい．Kent Beck と Ward Cunningham である．1986 年，2 人はテクトロニクス社の研究所に勤務する同僚だった．その頃，ソフトウェア開発の世界では，オブジェクト指向と GUI の大きな波が押し寄せていた．1972 年にゼロックスパロアルト研究所（Xerox PARC）の Alan Kay は「オブジェクト指向」に基づく対話型プログラミング環境 Smalltalk を発表し，大きな反響を呼んでいた．1984 年にはアップル社の「Macintosh」が発表され，GUI が一般化しつつあった．このような流れを受け，ACM は，オブジェクト指向に関する学会 OOPSLA（Object-Oriented Programming, Systems, Languages, and Applications）を立ち上げ，1986 年にオレゴンで第 1 回を開催した．この第 1 回の OOPSLA で，Beck と Cunningham は共に発表を行った．

もともと Beck と Cunningham は，プログラムを分かりやすくすることに関心を持っていた．彼らは議論を進めるうちに，2 人とも Alexander のパターンランゲージに強い興味を持っていることに気づいた．Beck は大学生のとき，生協で『時を超えた建設の道』と出会い，そのまま立ち読みで全部読み終えてしまったほどだった．意気投合した 2 人はパターンランゲージをコンピュータのシステム設計やプログラミングに取り入れようと試みる．

当時，テクトロニクス社内では，あるグループが Smalltalk でつくられたシステムの最終設計について悩みを抱えていたのだが，Beck と Cunningham は彼らの相談に乗り「パターンを用いたシステム設計」を提案する．Alexander が，建築はその利用者自身が設計すべきであると考えたのと同じように，2 人はそのシステムの利用者自身に最終設計をしてもらおうと考えたのである．利用者の代表には，トレーナーと現場のエンジニアが選ばれた．こうしてユーザインタフェースに関する 5 つのパターンからなるパターンランゲージがまとめられた．できあがったユーザインタフェースを見て，2 人は驚いた．簡素だが，非常にエレガントなデザインが実現していたのである．2 人はこの結果を論文にまとめ，第 2 回の OOPSLA 1987 において「オブジェクト指向プログラムのためのパターン言語の使用」[5] として発表した．

ここから OOPSLA を舞台に，ソフトウェア開発におけるパターンランゲージの応用を追究することとなった．その中でも Erich Gamma，Richard Helm，Ralph Johnson，John Vlissides の，GoF（Gang of Four）と呼ばれる 4 人が記した『デザインパターン』は，オブジェクト指向開発における名著として有名になり，以降ソフトウェア開発の分野でパターンと言えば，このデザインパターンのことを指すようになる．

■パターンブラウザとしての Wiki の誕生

実は Beck と Cunningham の論文は，デザインパターンだけでなく，Wiki の誕生と深く結びついている．Cunningham は，同じ時期に，収集したパターンを記録し閲覧するためのツールとして，Macintosh の HyperCard を用いて「パターンブラウザ」をつくっていた（図 -1）．はじめは自分自身のためのツールだったのだが，HyperCard でつくられたコンテンツ（「スタック」と呼ばれる）は，コピーを共有することによって，複数の利用者が共同編集を行うことが可能だった．

1993 年に「Mosaic 2.0」と「NCSA httpd」が発表され，Web にフォームと CGI という動的なページをつくるための仕組みが用意されると，Cunningham は，パターンブラウザを Web 上に展開すること

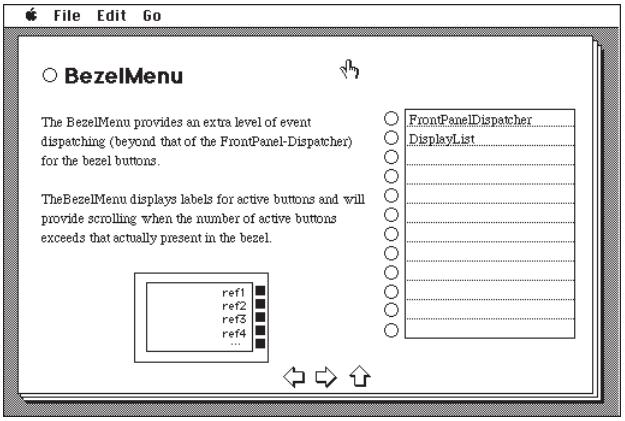

図 -1　HyperCard によるパターンブラウザの画面

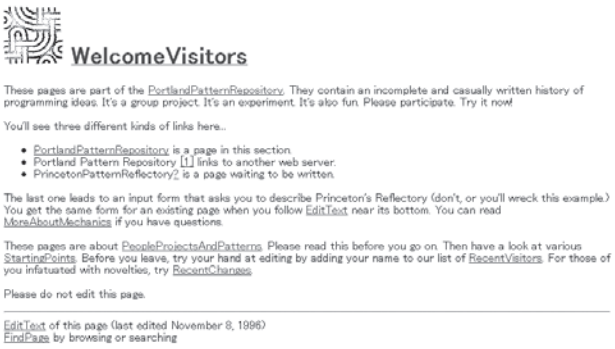

図 -2　最初期の WikiWikiWeb の画面

を思いつく．1994 年にプロトタイプを作り上げ，1995 年 3 月 25 日，みんなで書き換えられるデータベースという位置付けの Web サイトが公開される（図 -2）．

最初の WikiWikiWeb のコードである WikiBase は，説明文章とともに誰でも編集できる形で公開されていたため，同じ仕組みを利用者が自身のサイトで動かすことは難しくなかった．こうして，WikiWikiWeb と同じ機能を持つ Web サイトが世界中に現れ，それらはやがて「Wiki サイト」と呼ばれるようになった．

今日もっとも知られている Wiki サイトは，Wikipedia だろう．Wikipedia の前身は，Jimmy Wales と Larry Sanger が 2000 年にはじめた Nupedia という新しいフリーの百科事典のプロジェクトだった．Nupedia は従来の百科事典の編集方針を踏襲しており，執筆者は博士号を持っていなければならず，記事は 7 段階の厳格な査読を経て公開されることになっていた．実際には Nupedia の執筆者は増えず，なかなか記事は充実しなかった．Sanger は，友人から Wiki についての話を聞き，それを Nupedia の一部として導入するアイディアを思いつく．Wales を説得するが，Nupedia の執筆者と査読者から強硬な反対にあってしまう．結局，2 人は独自に「wikipedia.com」という新しいドメイン名を取得し，2001 年 1 月 15 日に Wiki による百科事典 Wikipedia がスタートした．図 -3 が最初期の

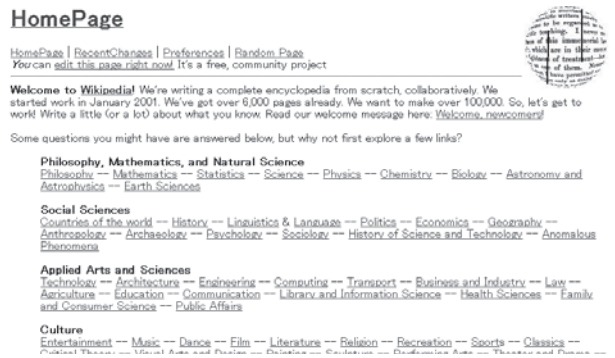

図 -3　最初期の Wikipedia の画面

Wikipedia の画面である．図 -2 と比較すると，よく似た画面であることが分かる．この段階ではまだ実際に WikiWikiWeb のコードを引き継いで使っていたので，画面が似ているのは当然なのである．

■ソフトウェア開発におけるプロセスパターン

エクストリームプログラミング（XP）は，パターンランゲージをソフトウェア開発を行う組織やプロセスの全体に適用するという Beck のアイディアによって生まれた．XP は，ソフトウェア開発現場において良いとされる一連の行動指針を，「プラクティス」と呼ばれる形でまとめたものである．

1987 年にパターンランゲージのプログラミングへの応用を提唱した Beck は『デザインパターン』の大成功に，かならずしも 100% 賛同していたとは言えなかった．自身が提唱したプログラミングへの

パターンランゲージの応用が世に普及したとはいえ，元々の趣旨だった「利用者自身によるシステム設計」という理想が実現したわけではなかったからだ．

1999 年，Beck は『XP エクストリーム・プログラミング入門』[6] を出版した．これは，パターンランゲージのプログラミングへの応用のリベンジだと言える．元々の趣旨だった利用者自身によるシステム設計という理想を目指し，システム開発のプロセス全般にパターンランゲージを応用することを試みた．Beck のツイート☆2 にあるように，「漸進的開発」や「チームにユーザを入れる」といったプラクティスは，Alexander に触発されて考えたプラクティスである．本書の執筆時に彼はこの本で「パターン」という言葉を使わないという決断をしている．それは，プログラミングの世界でパターンという言葉がデザインパターンの意味で定着してしまっていたからだ．しかし，第二版では記述を見直し，第 23 章「時を超えたプログラミングの道」として Alexander のパターンランゲージとの関係を語っている．彼はここで，自分の祖父が大工であり，引越をするたびに自分たち家族が住む家を自分たち自身で設計していた思い出を語り，そのような利用者自身による設計が目標であると語っている．Alexander の理論を称賛しつつ，建築という最古からある職業を変革することの限界に触れ，いま出現しつつあるソフトウェア開発業界であれば「新しい社会構造を作りあげることができる」と主張した．つまり，XP の最終的な目標は，新しい社会構造を作り上げることにあるのだ．

[新しい社会構造を作り上げる]

ここまでで，パターンランゲージの誕生とその発展を振り返り，パターンランゲージがソフトウェアの世界に応用されていった歴史を概観した．より詳細を知りたい方は，前述の『パターン，Wiki，XP』を参照していただければ幸いである．パターンという言葉は日常的に使われる言葉であり，ともすれば漠然としたイメージのまま理解したような気になってしまう．しかし，Alexander が考えるパターンランゲージは単なる定石集ではなく，利用者参加型設計プロセスや，利用者の主観を漸進的に客観に変えていく仕組みが含まれている．それこそが，パターンランゲージが Wiki や XP に与えた影響の本質だろう．Beck が語ったように，究極的には我々自身の手で新しい社会構造を作り上げることこそが目標なのだ．近年，建築やソフトウェアなどといった我々を取り巻く環境をアーキテクチャという語彙で考察する試みが活発になってきているが，パターンランゲージの影響は，そのような社会変革への意思という観点から参照されるべきだと考えている．本稿がそのような議論を推進する手助けになれば幸いである．

☆2 https://twitter.com/#!/KentBeck/status/3405661648

参考文献

1) Alexander, C. : *Notes on the Synthesis of Form*, Harvard University Press (1964)．（翻訳：形の合成に関するノート，鹿島出版会，1978）．
2) Alexander, C. : A City is Not a Tree., *Architectural Forum*, Vol.122, No.1, pp.58–62 (1965)．（翻訳：都市はツリーではない，別冊国文学 22 号，1984）．
3) Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. : *A Pattern Language : Towns, Buildings, Construction*, Oxford University Press (1977)．（翻訳：パタン・ランゲージ―環境設計の手引，鹿島出版会，1984）．
4) Alexander, C. : *The Timeless Way of Building,* Harvard University Press (1979)．（翻訳：時を超えた建設の道，鹿島出版会，1993）．
5) Beck, K. and Cunningham, W. : Using Pattern Languages for Object-Oriented Programs, *Technical Report No. CR-87-43*, Vol.67 (1987)．（翻訳：オブジェクト指向プログラムのためのパターン言語の使用，http://capsctrl.que.jp/kdmsnr/wiki/transl/?UsingPatternLanguagesForOOP）．
6) Beck, K. : *Extreme Programming Explained : Embrace Change,* Addison Wesley (1999)．（翻訳：XP エクストリーム・プログラミング入門，ピアソン・エデュケーション，2005）．

（2011 年 6 月 14 日受付）

江渡浩一郎（正会員） k-eto@aist.go.jp

産業技術総合研究所社会知能技術研究ラボ研究員．1997 年慶應義塾大学大学院政策・メディア研究科修了．国際メディア研究財団を経て，2002 年より産業技術総合研究所に勤務．2010 年東京大学大学院情報理工学系研究科博士課程修了．博士（情報理工学）．集団的創作活動に興味を持つ．著書に『パターン，Wiki，XP』（技術評論社）．日本ソフトウェア科学会会員．