

技術進化とコミュニティの文化変容モデル

高 大 八 伸 川 田 水 神 伸 史 行 正 行 道

1. はじめに

画期的な新製品をもたらすイノベーションは、当然、その製品を作るメーカーが生み出したのだろうと考える“常識”に対して、von Hippel (1988) は異を唱えた。イノベーションを商業生産に適用可能な状態にまで最初に推進した特定の個人または企業をイノベーターとして定義し、ガス・クロマトグラフ、核磁気共鳴分光器といった科学機器では、111のイノベーションの77%，半導体製造装置、プリント基板の組み立て装置といったエレクトロニクス製品の製造装置でも49のイノベーションの67%でユーザーがイノベーターであったことを明らかにしたのである（楣山、2000a）。実際、イノベーションは、科学技術の進歩と市場ニーズが相互に刺激し合いながら次第に明確化・具体化していくプロセスであり、純粹なテクノロジー・ッシュや純粹なマーケット・プルの事例を探すことは難しい。

そこで本稿の目的は、こうしたイノベーションのプロセスを技術コミュニティの進化のプロセスとしてとらえ、技術展開の方向性が技術コミュニティの力学によって決まるという従来からの学説を一步進め、その開発スピードも技術コミュニティの内部変化によって大きく変化することを説明するモデルを構築することにある。そのために、技術コミュニティ内のサブコミュニティとして開発者コミュニティとユーザー・

コミュニティを考え、この二つのサブコミュニティの文化を有するルールに則って脱ぎ替えることで、あるときには開発者コミュニティの一員として、あるときにはユーザー・コミュニティの一員として行動するオーディエンスが存在するというモデルを考える。このサブコミュニティ文化の脱ぎ替えルールとして、文化変容の領域で整理されてきた分離志向と同化志向を採用し、実際にそれをエージェントのルールとして組み込んだマルチエージェント型のコンピュータ・シミュレーション・モデルを構築する。さらに、将来的にコンピュータ・シミュレーションの予測精度を向上させるために、分離志向と同化志向の測定尺度も開発し、実際の調査データにより、その妥当性の検証も行う。

2. 技術展開の方向性と開発スピード

(1) 技術展開の方向性を決めるコミュニティの力学：社会構築主義

イノベーションを科学技術の進歩と市場ニーズが相互に刺激し合いながら次第に明確化・具体化していくプロセスとしてとらえるというアイデアは、経営学の分野にとどまらず、1980年代には科学技術論で技術の社会的構成(social construction of technology; SCOT)あるいは社会構築主義(social constructionism)として登場していた(綾部、2006)。たとえば、現在のような自転車一すなわち前輪と

後輪と同じ大きさの空気入りタイヤで、ペダルをこいでチェーンで後輪を駆動する自転車ーは1880年代から90年代にかけて登場するが、それはリニア・モデルが主張するように、研究→開発→製造→販売という一連の不可逆な流れではなかった。

現実には、1870年代、前輪が後輪に比べて極端に大きく、前輪についたペダルを直接こぐオーディナリー型と呼ばれる自転車タイプが主流であった時代があり、それから現在の自転車がドミナント・デザインとして定着するまでの間、亜種を含めて様々なタイプの自転車が登場しては消えていった。それどころか、個々の部品が定着するプロセス一つをとっても、そこには様々な社会的グループが影響を与えていたのである。たとえば空気入りタイヤは、当初は振動問題を解決するために開発されたが、自転車レースで従来のタイヤよりも速く走れたことから、振動を問題視していなかったスポーツ用の自転車ユーザーにも受け入れられて定着した。このように社会構築主義では、技術展開の方向性は「関連社会グループ」(relevant social group)の力学によって決まるとして主張される(Bijker, Hughes & Pinch, 1987)。

これにさらに技術的サイクルについての考察も加味して、Tushman and Rosenkopf (1992)は、特に技術の複雑性が高いときには、技術発展の利害関係者の集団であるコミュニティの論理によって技術的進化が生じると主張した(大神, 2009a)。しかし実際には、Tushman and Rosenkopf (1992)が最も複雑性の低い製品、つまりコミュニティの論理の影響が最も小さい例として挙げていた板ガラスについてさえ、その製造工程である板ガラス成形技術は、コミュニティの論理あるいは技術とコミュニティの相互作用によって技術展開の方向性を説明できる。

板ガラスの大量生産は1900年初頭のラバース式機械円筒法の誕生に始まり、1910年代半ば以降に自動平板引上法によるラバース式の駆逐・代替、そして1950年代のフロート法の誕生とその台頭により、Tushman and Rosen-

kopf (1992)も指摘していたように、ラバース式の板ガラス製造工程で大きく六つに分けられる工程のうち、それまで多くのコストを発生させていた三つの工程が不要となった(大神, 2009c)。そして、フロート法によって片隅に追いやられて生き残っていた自動平板引上法のコミュニティとの棲み分けにより、フロート法は厚さ2mm以上の厚いガラス用という技術展開の方向性が生じた(大神, 2009c)。ところがこのフロート法による成型可能な厚さ2mmの「下方限界」も、フロート法のコミュニティに、それまでの建築業界だけではなく、自動車業界、液晶ディスプレイ業界といった新しい「関連社会グループ」が加わったことで、その力学の結果として、2mmよりも薄いガラスをフロート法で作る方向性が明確になり、やがてはその方向での技術展開が急速に進むことになるのである。

このフロート法の2mmの成型可能下方限界を巡る技術展開の方向性の攻防は、自転車の空気入りタイヤに対する理解が、振動軽減用部品から高速走行用部品へと大きく変わったように、自らがもっている知識や世界理解の方法が唯一無二のものではなく、別の状況ではまた違った知識や世界理解の方法がありえるという社会構築主義の主張(綾部, 2006)を如実に支持している。すなわち、それまで、厚い板ガラス用の成形技術として理解されていたフロート法が、そのコミュニティに新しい関連社会グループ、特に液晶ディスプレイ業界が加わったことで、今度は、薄い板ガラス用の成形技術へと理解を180度転換したのである。

それでは、技術展開の方向性が決まった後の開発スピードの急激な変化は、どのように説明できるのであろうか。実は、TushmanとRosenkopfは、技術の不安定期にコミュニティの影響が大きく、技術が安定定期に入ると支配的な評価軸が確立されて、それに基づいた技術開発が行われると考えていた。これは、おそらくKuhn (1962)のパラダイム論からの類推なのであろう。しかし、それはどのようなメカ

ニズムによるのであろうか。

(2) 開発スピードの急速な変化と情報粘着性

開発スピードの急速な変化という現象で注目されるのが、オープン・ソースのソフトウェア開発である。近年のオープン・ソース・ソフトウェアやオンライン百科事典 Wikipedia のようなある種のオンライン・コンテンツの開発では、アクティブに開発の対象となっている最先端（ヘッド）が即公開されており、開発者とユーザーは同じヘッドを共有している。これまでリリース管理¹⁾の名の下で行われてきた独立した品質保証の工程を放棄し、十分なテストやチェックを受けない不完全かつ不安定な製品が渡る可能性をユーザーに了解してもらう代わりに、常に真の意味での最新バージョンが提供されるのである（八田，2009）。そして、Linux のように、ソフトウェアが急速に成長したり、あるいは急に停滞したりと開発スピードが急速に変化する現象が見られると同時に、同じオープン・ソース同士でありながら、従来から存在した BSD を新興の Linux が一気に抜き去るような現象も見られるようになった（高橋・高松，2002）。

こうした現象の理解の仕方として、特に Linux や BSD のようなとともに UNIX 系のオペレーティング・システム (OS: Operating System) の場合、たとえば普及のプロセスとしてとらえ、情報の移転の難易度によって開発スピードを説明する方法も考えられる。その際に、これまで一番有望とされてきたのが、情報粘着性の概念である。「情報粘着性」(information stickiness) とは、企業の問題解決活動における情報の移転の難しさに関わる概念で、von Hippel (1994) によって提唱された。情報粘着性が高くなるほど情報の移転が難しくなり、情報粘着性が低くなるほど移転は容易になるとされる（楣山，2000b）。von Hippel (1994) の考えでは、①たとえば情報がノウハウのような暗黙知の場合には情報粘着性は高くなるが、設計図やマニュアルなどの移転できる形にあら

かじめ形式知化されている場合には、情報粘着性は低くなる。また同時に、②もともと大きなシステムの一部を構成している情報を移転先で期待通りに機能させるには、チューニングをする必要があり、それには余計な費用がかかることになる。

企業内でのベスト・プラクティスの移転を研究した Szulanski (1996) は、このうち①については、ベスト・プラクティスの存在を仮定することで、形式知化が既に行われているという前提で議論を単純化し、その上で②について、そのベスト・プラクティスの base case に対して、移転前のパフォーマンスを出すために行わなくてはならないチューニングの手間を eventfulness すなわち「移転の間に経験された問題のある状況の程度」(p. 30) と呼んだ。つまり、「移植性」(portability) を eventfulness として測定できる可能性を示唆したのである（若林・大木，2009）。

実際、ソフトウェアの世界では、移植性は移植コストによって測定可能で、現在、比較的よく使われている Basic COCOMO もそのツールの一つである。ここで COCOMO (COnstructive COSt MOdel) とは、Boehm (1981) がソフトウェア開発に要する工数や費用、期間を推定するために開発したモデルである。Boehm 自身がその後数回の改訂を行っているために、1981 年のオリジナル・モデルは “Basic COCOMO” と呼ばれるようになったが、単純で使い勝手もいいので、現在でも一般的に使われている。Basic COCOMO は、プログラム・ソースコード（コメント行などは含まない）の物理行数 SLOC (Source Lines of Code) と、ソフトウェア開発に要した工数や期間、費用との関係を示したモデルで、たとえば、SLOC から開発作業量 E (人月単位) を求める計算式

$$E = a \text{ SLOC}^b$$

などがよく使われる。ここでの SLOC は 1,000 行単位（いわゆる KSLOC）である。Boehm は COCOMO 開発当時、ある米軍需産

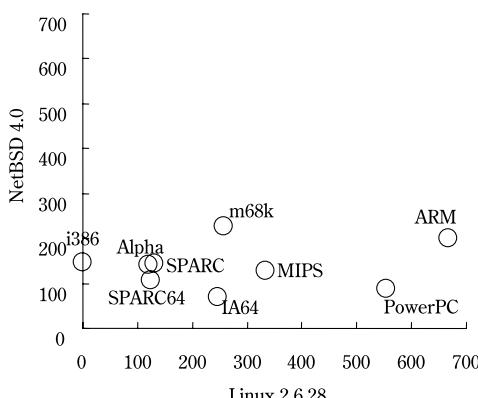
業において空軍関係の多数のプログラム開発プロジェクトを統括する立場にあり、そのデータから得られた回帰係数の値は、 $a=2.4$, $b=1.05$ であった。当然のことながら、プログラムによって開発の難易度は異なるし、プログラマの生産性にも大きな個人差があるのだが、仮に、 $a=2.4$, $b=1.05$ で2,000行のプログラムだったとすると、 E は4.97人月、すなわち、平均的な難度の2,000行のプログラムを書くには、平均的なプログラマ1人の5ヶ月分の作業量を要することになる。この開発作業量 E を求める計算式を使って、ソフトウェアの移植性・移植コストを測定することができる。

実際、ソフトウェアは、元々想定されたものとは異なるアーキテクチャでも動作するように移植(porting)することができる。たとえば、Linuxカーネルは、もともとi386用に開発さ

第1表 LinuxとNetBSDの移植のための開発作業量

アーキテクチャ	Linux 2.6.28		NetBSD 4.0	
	SLOC	工数(人月)	SLOC	工数(人月)
i386	0	0.00	49,948	145.77
Alpha	41,610	120.33	48,743	142.08
SPARC64	43,222	125.23	37,604	108.19
SPARC	45,107	130.97	49,122	143.24
IA64	82,424	246.64	25,074	70.70
m68k	85,928	257.66	76,113	226.85
MIPS	109,715	333.04	43,881	127.23
Power PC	178,212	554.24	30,516	86.89
ARM	212,564	666.92	67,607	200.31

第1図 LinuxとNetBSDの移植開発作業量

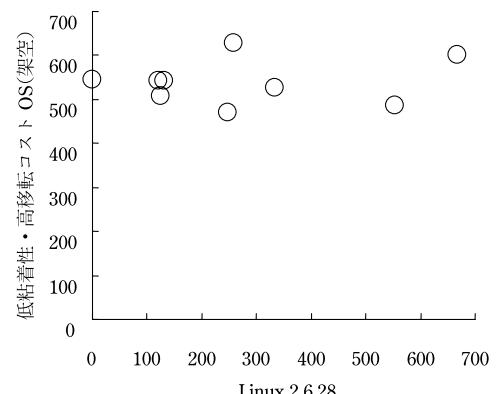


れたが、その後、移植作業が行われ、いまや様々なアーキテクチャ上で動作するようになっている。この移植に要するコストは、同じソフトウェアであっても移植先のアーキテクチャによって異なる。たとえば、Linux 2.6.28とNetBSD 4.0の主要アーキテクチャへの移植のための開発作業量を計算すると第1表のようになる。

第1表をグラフ化した第1図から、もともとi386用に開発されたLinuxの移植のための開発作業量はアーキテクチャによってかなりのばらつきがあることが分かる。それと比較して、NetBSDはそれほどばらつきがない。つまり、NetBSDに比べて、Linuxは最初のアーキテクチャに対する粘着性が強いのである。しかも、NetBSDに比べて、Linuxの開発作業量のレベルは高い水準にある。つまり、Linuxは、最初のアーキテクチャへの粘着性が強く、なおかつ他のアーキテクチャへの移植コストが高いことになる。

このことをより分かりやすくするために、比較対照のために架空の数値例として第2図を作成した。第2図は第1図のグラフを上方に400人月シフトさせた架空OSの例であるが、仮に、第2図のようであれば、Linuxは最初のアーキテクチャへの粘着性は強いが、移植コストはむしろ架空OSの方が高水準にあるということになる。つまり、von Hippel (1994) の

第2図 Linuxと比べて低粘着性・高移転コストの架空OSの例



情報粘着性は

- (i) 情報の暗黙知/形式知の程度（ただしソフツウェアの場合には、形式知化は既に行われている）
- (ii) 最初のアーキテクチャに対する狭義の粘着性
- (iii) 他のアーキテクチャへの移植コスト

の三つの要素に分解できることになる。

しかし、以上の分析は情報粘着性に関する議論を深めたことにはなるが、NetBSDと比べて、(ii)の最初のアーキテクチャに対する粘着性が強く、しかも(iii)の移植のための開発作業量のレベルが高水準のLinuxが、なぜBSDを抜き去ったのかについては、全く説明力がない。情報粘着性の観点からは、むしろ、逆の現象が起きてしかるべきだったのである。そこで本稿が改めて着目するのが、技術展開の方向性を考察する際にも登場した技術コミュニティなのである。

実は、既に述べたように、TushmanとRosenkopfは、技術の不安定期にコミュニティの影響が大きく、技術が定期に入ると支配的な評価軸が確立されて、それに基づいた技術開発が行われると考えていた。ところが、現実の板ガラス成形技術の進化プロセスでは、フロート法がドミナントになった定期に入り、「成形可能な厚み」が評価軸として確立した以降になって、コミュニティに新しい関連社会グループが加わって、技術展開の方向性のみならず、「成形可能な厚み」に関する技術開発の開発スピードも急激に変化したのである。

実際、Linux対BSDという比較をした場合、先行していたBSDが、ライセンスを巡ってAT&T（正確にはその子会社USL）が起こした訴訟に巻き込まれたことで、開発者はBSDの開発を手控え、開発は急速に下火になってしまった。もし裁判でAT&Tが勝つようになると、BSDをベースにして開発したソフトには、いつまでもAT&Tのライセンスが必要になってしまふからである。この訴訟に巻き

込まれた1992年から1994年までの約1年半という時期が、実は、AT&Tのライセンスに抵触しない同じUNIX系のOSであるLinuxが急成長した時期、すなわちLinuxのコミュニティが急成長した時期と一致するのである（高橋・高松、2002）。つまりコミュニティは、技術展開の方向性を決めるだけではなく、開発スピードにも大きな影響を及ぼしている可能性がある。それでは、どのようなメカニズムで、コミュニティは開発スピードに影響を与えるのであろうか。

3. コミュニティの文化変容モデル

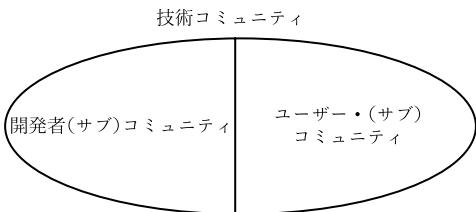
(1) 開発スピードの急激な変化と「オーディエンス」

オープン・ソースのソフトウェア開発に見られるようなユーザーの存在は、たとえばvon Hippel（2005）のような既存研究でも指摘されてきたが、八田（2009）はこの概念を一步進めて、時と場合によって開発者とユーザーのどちらかの顔に使い分ける存在を「オーディエンス」（audience）と呼んだ。オーディエンスはvon Hippel（1986）のいう「リード・ユーザー」²⁾とは異なり、ユーザーのときもあれば開発者のときもある存在である。より現実のソフトウェア開発の姿に近づけて記述すれば、オーディエンスは、あるときは開発者コミュニティの一員として振舞い、あるときはユーザー・コミュニティの一員として振舞う（第3図）。

それでは、技術コミュニティを構成する個々のオーディエンスに、サブコミュニティである開発者コミュニティあるいはユーザー・コミュニティの一員としての顔を自由に使い分けることを可能にさせている条件とは何か？ それこそが実は、ソフトウェア開発においては改変可能なヘッドの存在、つまりヘッドをソースコードで提供するオープン・ソースだったのである。

仮に、オープン・ソースのように、各オーディエンスのレベルでの開発者とユーザーのスイッチング・コストが非常に低くなると、「開

第3図 技術コミュニティ（＝オーディエンスの集合）内のサブコミュニティ



「発スピードの変化」が大きくなると考えられる。実際、リリース管理方式の場合にはあまり見られなかった急速な進歩や突然の停滞といった特徴的な現象がオープン・ソースでしばしば観察される際には、オーディエンスがユーザー・コミュニティから次々と開発者コミュニティに鞍替えして開発が急速に進んだり、逆に、たとえば開発者コミュニティ内でのコンフリクトが深刻になり、大多数のオーディエンスが開発者コミュニティを離脱してしまって開発が突然に停滞したりする裏事情が存在していた。つまり、たとえ、ある特定の製品のオーディエンスの集合が固定されていても、改変可能なヘッドがオーディエンスに公開されていれば、その製品の開発スピードは急激に変化する可能性が高まる。

しかし、考えてみれば、これはソフトウェア開発に特有の現象ではない。そもそも通常の技術でも、たとえば特許のライセンサーは、ライセンサーがたんなる特許のユーザーではなく、力をつけてやがて強力な競争相手となることを想定し、恐れているが故に、厳格なライセンス契約を結ぶのである。実際、板ガラス成形技術のライセンシング・コミュニティ³⁾は、まさにオーディエンスの集合であった。ただしフローント法の場合、ピルキントン社の特許権が切れるまでは、グラントバック条項を掲げるピルキントン社の決めた方向性の下で、2 mmという成型可能な厚さの下方限界を容認あるいは放置して、より厚いガラスを作ることを追求し、フローント法は固定されたライセンシング・コミュニティの中で劇的に技術展開を遂げた。そしてやがて開発は停滞し、ライセンサーはたんなる特許のユーザーへと回帰していく。

しかし、特許権が切れ、ピルキントン社がライセンス契約等でオーディエンスを縛りつけておくことができなくなったとき、かつてのランセンサーは技術展開の方向性の自由度を獲得し、今度は、液晶ディスプレイ業界のような、より薄いガラスを必要とする関連社会グループの力学によって、より薄いガラスを作るというまったく逆の技術展開の方向性が決まり、さらに、それまで特許のユーザー・コミュニティの一員として振舞っていたガラス・メーカーが、ピルキントン社の特許権が切れる 것을契機に、今度は開発者コミュニティの一員として、フローント法による2 mmの壁の打破を目指し始めたことで、開発スピードを上げていくのである。

(2) 文化変容の Berry モデル

このように、オーディエンスが、開発者コミュニティとユーザー・コミュニティの二つのサブコミュニティの文化のあるルールに則って脱ぎ替え、あるときには開発者コミュニティの一員として、あるときにはユーザー・コミュニティの一員として振舞うと考えることで、開発スピードの変化は説明できそうである。このオーディエンスが「開発者コミュニティとユーザー・コミュニティの一員としての顔を自由に使い分ける」という現象は、本稿では、オーディエンスが、開発者コミュニティとユーザー・コミュニティの二つの「コミュニティの文化」をあたかもOSのようなソフトウェアを移植するかのごとく、集合的レベルの心理的プログラムを入れ替える現象だととらえる。⁴⁾ そうとらえることで、二つ以上の異なる文化が接触することによって、双方、もしくは一方の文化すなわちプログラムが変容するというプロセス—これを英語では“acculturation”と呼び、日本語では「文化変容」（あるいは「文化触変」）と訳される—として、技術コミュニティの進化のプロセスをモデル化することができる。

1970年代、文化人類学の分野で、Berryは文化変容の概念を成熟させていった。マイノリ

ティにおける生態と個人の発達に関する研究 (Berry, 1971), 西洋化に直面しているマイノリティの心理ストレス（これを「文化変容ストレス (acculturative stress)」と呼んでいた）に関する研究 (Berry and Annis, 1974) を経て、最初の著作となる Berry (1976) 以後、彼は文化変容 (acculturation) のモデルを提示するようになる (Berry, 1976, 1983, 1984)。Berry (1976) ではマイノリティが西洋文化に接した際の、自文化と西洋文化に対する心理的傾向を表す二つの質問を軸として第2表のように文化変容を四つに類型化して整理した。これが後に様々な分野で「文化変容モデル」として引用されることになる。⁵⁾

第2表の各類型には、次のような説明が加えられている（大川、2009a）。

同化……伝統的文化を捨て、より大きな社会へ移動を示す考え方。

第2表 文化変容の4類型

Are positive relations with the larger society to be sought? より大きな社会との肯定的な関係が求められるべきか？		Is traditional culture of value and to be retained? 伝統的な文化は価値があり保持されるべきか？	
Yes	Integration 統合	No	Assimilation 同化
No	Rejection (Segregation) 拒絶（分離）	(Deculturation) (失文化)	

出所: Berry (1976, p.181) Table 8.4 に日本語訳を付したもの。

第3表 文化変容の4類型とエージェントのルール

異質な人と接触している		自分の文化を変えない	
Yes	分離志向	Yes	No
		Integration 統合	Assimilation 同化
No		Rejection (Segregation) 拒絶（分離）	(Deculturation) (失文化)

統合……伝統的文化の維持とより大きな社会との積極的協力 (pulling together) を示す考え方。

拒絶（分離）……より大きな社会から離れた文化の維持を表す考え方。アメリカンインディアンが拒絶 (rejection) の傾向を示すと、それは伝統的隔離 (conventional segregation) を意味する。拒否は自己隔離である。

失文化……常識的にも、予備調査でも、この結果は誰も選択しなかった。しかし、周辺性 (marginality) の概念の特徴の一部には関連がある。

Berry自身は考えもしなかつただろうが、今日的な意味でのBerryの貢献は、文化変容の現象が生じる際の個々人のレベルでのルール（つまり後述するエージェントのルール）を単純明確に表すことに成功したということである。単純化すると、個人レベルのルールは異質な人と接触している非「分離志向」と自分の文化を変えない非「同化志向」に整理される（第3表）。

ところで、Berryの4類型のうち、Berry自身も明確な解釈を与えてこなかったのが「失文化 (deculturation)」の類型である。分離志向のレベルも同化志向のレベルも高いということを素直に理解すれば、「失文化」の状態の例として、次のような行動・姿を想像することができる。

「私はカメレオンの女。ダサいって言われて、孔雀みたいにおしゃれになって。顔が地味って

言われて、インコみたいに派手な化粧になって。子供っぽいって言われて、ヒョウ柄のブランド服で着飾って。女らしくないって言われて、馬の尻尾みたいに髪の毛を伸ばして。短い方が似合うって言われて、アメリカン・ショートヘアになって。肌焼いた方がイケてるよって言われて、カラスみたいに真っ黒になって。今もう美白ブームだよって言われて、厚化粧で隠したらシマウマみたいに白黒になって。もう少しほっちゃりの方がいいよって言われて、子豚みたいに丸くなっている。やっぱやせてる方が好きだなって言われて、ダイエットしたら身体だけやせて胸がつるっとしちゃったりして。で、私一体、本当は何になりたいんだろう。」⁶⁾

要するに、周囲から影響されやすく、しかも接触するとすぐに離れるという行為を繰り返すので、「周囲の人」が流動的となり、いつも自信がもてずに不安で、自分が何なのか分からぬというストレスを抱えることになる。これが「失文化」の状態だと考えられる。今流行の「自分探し」も、失文化の一形態だといえるかもしれません。

(3) マルチエージェント・シミュレーション

このように、Berryは個々人の類型（個々のエージェントのルール）を記述することには成功したが、だからといって、こうした個人の認知や行動から、単純に社会全体のあり方までを推測することは難しい。たとえば、「同化」に分類される個人が集まつたからといって、社会全体が一つの文化に収斂するはずだと結論づけるのは、明らかに論理の飛躍である。実際、Berryは、個々人の類型によって、どのような社会が出現するのかについては、明確な主張をする手段をもっていなかったし、彼の研究は、あくまでも個人レベルの心理学的な考察であり、社会全体の状況に関する調査結果も記述も欠いていた。そこで、本稿では、マルチエージェント型のコンピュータ・シミュレーションにより、どのような社会状況が出現するのかを実験してみることにしよう。

まず、 10×10 セルの二次元格子状の空間に50人のエージェントをランダムに配置する。各エージェントは白色か黒色の「文化」をもち、第4図のような自分の1近傍（エージェントAの1近傍は網掛けした8つのセルが該当する）の「異質度」を計算する。「異質度」は、1近傍にいるエージェントのうち、異なる「文化」のエージェントが占める割合であり、以下の式によって求められる。

$$\text{異質度} = \frac{\text{1近傍にいる自分と異なる「文化」のエージェント数}}{\text{1近傍のエージェント総数}}$$

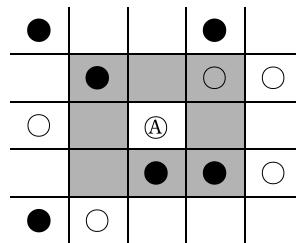
第4図の例で計算すると、白色の文化をもつエージェントAの異質度は $3/4$ となる。各エージェントは「分離基準」と「同化基準」の二つの基準をもっていて、どちらの基準も0から1の値をとり、次のように異質度との大小関係でエージェントの行動を規定する。

- (i) もし 異質度 > 分離基準 ならば、1近傍の空セルをランダムに選び移動する。
- (ii) もし 異質度 > 同化基準 ならば、自分の文化を変える。

これらの分離基準・同化基準と分離志向・同化志向との関係は第4表のようになる。

こうして、各エージェントが順番にルールを実行して、全エージェントが実行し終えたら1ステップとする。1回のシミュレーションは1,000ステップとし、1,000ステップ終了後の

第4図 エージェントAの1近傍



エージェントAの1近傍

状態を次の四つの「社会のタイプ」に分類した。

- (a) I (Integration) ……二つの文化が入り混じって、移動も文化変化もゼロの状態。
- (b) S (Segregation) ……二つの文化圏に分かれて、移動も文化変化もゼロの状態。
- (c) A (Assimilation) ……一つの文化に収斂して、移動も文化変化もゼロの状態。
- (d) D (Deculturation)……移動も文化変化も

続いている状態。

ここでは混乱を避けるために、「統合」「分離」「同化」「失文化」という個人レベルの文化変容の4類型のラベルを使わずに、「I」「S」「A」「D」というラベルで社会全体の状況を分類することにした。

白色の文化を多数派で45個のエージェント、黒色の文化を少数派で5個のエージェントという初期設定にして、第5表のような分離基準と同化基準の各組み合わせについて、50回

第4表 文化変容の4類型とエージェントのルールと分離基準・同化基準

異質な人と接触している		自分の文化を変えない	
		Yes	No
Yes	高	高	低
	低	分離志向	同化志向
Integration 統合	Assimilation 同化	Rejection (Segregation) 拒絶(分離)	(Deculturation) (失文化)

第5表 多数派45・少数派5のときの最終的な「社会のタイプ」

分離基準	同化基準									
	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
1.0	I	I, A	A	A	A	A	A	A	D	D
0.9	I	I	A	A	A	A	A	A	D	D
0.8	A	A	A	A	A	A	A	A	D	D
0.7	A	A	A	A	A	A	A	S	D	D
0.6	A	A	A	A	A	A	A	S	D	D
0.5	A	A	A	A	A	A	A	S	D	D
0.4	A	A	A	A	A	A	S	S	D	D
0.3	A	A	A	A	A	A	S	S	S	S
0.2	A	A	A	A	A	A	S	S	S	S
0.1	A	A	A	A	A	A	S	S	S	S

注) 各組合せで50回の試行のうち、もっとも頻繁に出現した「社会のタイプ」を表示。

のシミュレーションを実行し、その結果、「I」「S」「A」「D」のうちで、もっとも頻繁に出現した「社会のタイプ」を表すと、第5表のようになった。

第4表と第5表を比較してすぐに分かることは、「統合」「I」以外の3類型は、いずれも位置関係が大きくずれているということである。個々のエージェントのルールは単純でも、エージェントが相互に作用しあうことで、個々のルールからは想像もできないような社会現象が出現するというのが、マルチエージェント型のシミュレーションの特色・利点であるが、この第5表は、まさにその典型的な例だといえる。

(4) インプリケーション

第5表で注目されるのは、文化変容の類型「失文化」に対応しているはずで、移動も文化変化も続いている「D」のポジションである。この「D」の状態にあるということは、前節の技術コミュニティの議論で解釈すると、いわゆるドミナント・デザインにいつまでも到達しなかったり、あるいは開発スピードが不安定でいつまでも収束しなかったりといった特徴が見られると考えられる。ところが第5表は、この失文化の状態「D」が、実は分離志向が低く、同化志向が強い状態のとき、すなわち文化変容の4類型で「同化」の人々が集まった社会で「社会のタイプ」として出現することを示している。そして、文化変容の4類型で「失文化」の人々の集まった社会では、「D」ではなく、分離「S」が「社会のタイプ」となるのである。

さらに、この第5表の組合せのうち、太枠で表示した分離基準=0.5、同化基準=0.5に設定した50回のシミュレーションの結果を、ステップごとに追いかけてみよう。その様子は第6表と第5図に示されているが、図表には、実際に行った1,000ステップのうち10ステップまでしか結果を表示していない。実はこのくらい多数派と少数派に差があると、最長でも4ステップまでで形勢が決まってしまうのである。それ以降のステップでは多数派、少数派のエー

ジェントの数は動かなくなる。第5表からも分かるように、この分離基準=0.5、同化基準=0.5の組合せでもっとも頻繁に出現した「社会のタイプ」は「A」であったが、その内訳は第6表のように、50回のシミュレーションのうち39回78%が单一文化で、いずれも多数派白色文化に収斂し、「A」に分類される。しかもそのうち30回は「パターン1」であり、ステップ1だけが少数派5で、ステップ2からは多数派の文化に収斂して少数派は0になってしまう。

この結果はどんな意味をもっているのだろうか。多数派をユーザー・コミュニティ、少数派を開発者コミュニティとすると、要するに開発者が5/50すなわち10%しかいないときは、分離基準=0.5、同化基準=0.5では、80%近い確率で、早い段階で開発者はいなくなってしまい、全員がユーザーになってしまうということを意味している。現実にも、圧倒的多数のユーザーが開発者に対して発言、クレームの集中砲火を浴びせたために、嫌気がさした開発者が離脱してしまうという現象が起きているが、それを連想させるような結果になっている。そのことと表裏一体であるが、圧倒的多数派のユーザーの中で開発者コミュニティが維持されている「パターン3」や「パターン7」では、少数派は1つまり2%しか残っていないが、実は「社会のタイプ」は「S」であり、開発者コミュニティはユーザーから距離を置くことで存続していることも、現実味のある姿である。

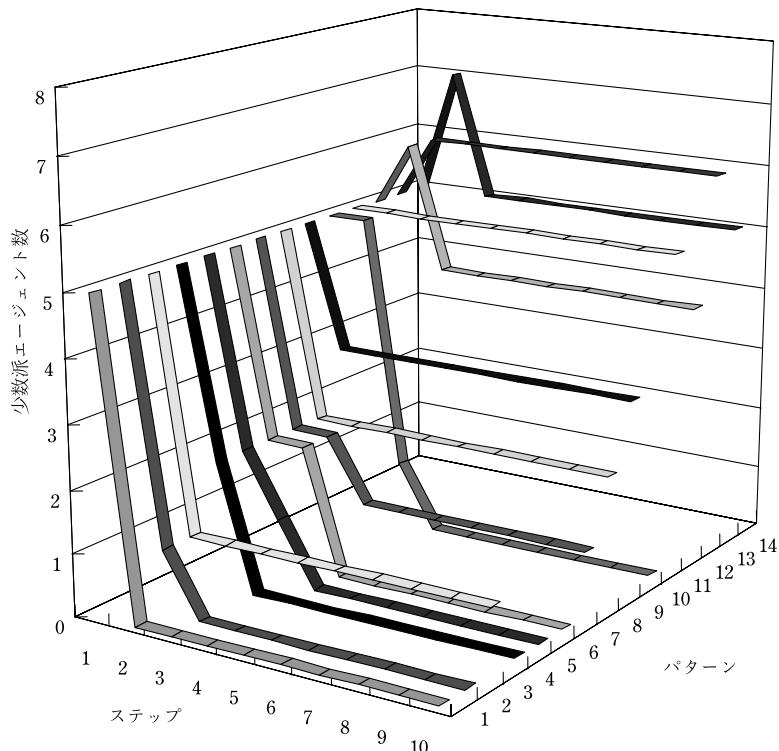
しかし他方で、第5図が分かりやすいが、「パターン13」では、少数派は数を5から6に増やしているし、「パターン14」では瞬間に5から7まで40%増やし、またすぐに5に戻っている。どちらのパターンも「社会のタイプ」は「I」だが、これはもっと分離志向が強ければ周囲に異文化者が1人でもいれば距離を取るのだが、分離基準=0.5程度であれば、ユーザーから逃げずに接触は保ったままで開発者コミュニティを維持していることを意味している。現実の世界では、この状態で開発者が

第6表 少数派エージェント数の増減パターン

ステップ	少数派エージェント数の増減パターン													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	5	5	5	5	5	5	5	5	5	5	5	5	5	5
2	0	1	1	2	2	2	2	3	5	5	6	6	7	
3	0	0	1	0	1	2	2	3	1	5	4	6	5	
4	0	0	1	0	0	0	1	2	3	0	5	4	6	5
5	0	0	1	0	0	0	1	2	3	0	5	4	6	5
6	0	0	1	0	0	0	1	2	3	0	5	4	6	5
7	0	0	1	0	0	0	1	2	3	0	5	4	6	5
8	0	0	1	0	0	0	1	2	3	0	5	4	6	5
9	0	0	1	0	0	0	1	2	3	0	5	4	6	5
10	0	0	1	0	0	0	1	2	3	0	5	4	6	5
該当数	30	5	4	1	1	1	1	1	1	1	1	1	1	1
社会のタイプ	A	A	S	A	A	A	S	I	I	A	I	I	I	I

多数派 45・少数派 5, 分離基準=0.5, 同化基準=0.5

第5図 少数派エージェント数の増減パターン

40%増減することは大きな意味をもつ。⁷⁾

4. 分離志向と同化志向の測定

(1) 測定尺度

現実のイノベーションで、以上のようなメカ

ニズムで現象が起こっていると結論するには、実際のコミュニティの分離志向と同化志向を測定し、その組合せの違いにより、生起する現象が異なることを示す必要がある。さらに、一步進めて、測定した個人の分離志向と同化志向をもとにして個々のエージェントの分離基準、同

化基準を設定できるようになると、特定のコミュニティの進化の様子をより高い精度でシミュレートすることができるはずである。

そこで、第2表にも記載されているもとのBerryの二つの質問

【非「分離志向】

Are positive relations with the larger society to be sought?

より大きな社会との肯定的な関係が求められるべきか？

【非「同化志向】

Is traditional culture of value and to be retained?

伝統的な文化は価値があり保持されるべきか？

を企業内で使用できるように、次のようなそれぞれ五つの質問項目を開発した。

【分離志向】

S1. 職場に新しく人が配置された場合には、自分から積極的に交わっていく方だ。

S2. 他の会社から転職して来た人と話すのは楽しい。

S3. 新卒採用よりも中途採用の方が、良い人材を採用できると思う。

S4. 職場のコミュニケーションでは、世代による壁は存在しない。

S5. 色々な経験を積んできた人からもっとたくさん学びたい。

それぞれの質問項目について、「はい」ならば1; 「いいえ」ならば0とした上で、次のように分離志向 SEGRE を定義した。

【分離志向】

$$\text{SEGRE} = (1 - S1) + (1 - S2) + (1 - S3) + (1 - S4) + (1 - S5)$$

【同化志向】

A1. うちの会社には守るべき「うちの会社らしさ」があるように思う。

A2. うちの会社のやり方よりも少しでも優れ

た方法を他社が採用していると思われる場合には、たとえ試行錯誤的であっても、積極的に取り入れてみるべきだと思う。

A3. 今よりももっといい仕事の仕方があるよう思う。

A4. 他の会社の成功事例は参考にはなっても、自分の会社には使えないと思う。

A5. もしも上司を選べるのであれば、うちの会社の文化を体現した上司の下で働きたい。

それぞれの質問項目について、「はい」ならば1; 「いいえ」ならば0とした上で、次のように同化志向 ASSIM を定義した。

【同化志向】

$$\text{ASSIM} = (1 - A1) + A2 + A3 + (1 - A4) + (1 - A5)$$

この二つの測定尺度 SEGRE と ASSIM が、それぞれ分離志向と同化志向を反映したものになっているかどうかを確認するために、本稿では文化変容の4類型のうちの「失文化」に着目した。既に述べたように、周りから影響されやすく、しかも周囲の人とは接触したらすぐに離れる傾向のある人の場合、「失文化」の状態になりやすいと考えられるが、実は、現実の企業の中にも、こうした「失文化」になりやすい状況に置かれている人々がいるのである。それは、他の会社からの出向者である。完全に転職したわけではなく、いずれは元の会社に戻るかもしれない出向という形で今の会社に勤めている場合、そもそも付き合うべき周囲の人が2グループ存在し、しかも、自分は出向者であるという自覚があればあるほど、どちらにも合せようとするだろう。そこで、次のような仮説を立てた。

仮説1. 他の会社からの出向者は、分離志向のレベルも同化志向のレベルも高くなる傾向がある。すなわち失文化の状態になりやすい。

(2) 調査方法と結果

二つの測定尺度 SEGRE と ASSIM を使って、この仮説を検証できれば、二つの測定尺度がそれぞれ分離志向と同化志向を反映したものになっていると考えていいはずである。

今回、調査対象に選んだ A 社には、親会社からの出向者が、管理職を中心に少数派として存在している。しかし、回答者に対して、出向者であるかどうかを直接質問したのでは、回答にバイアスが入る可能性がある。そこで今回の検証で着目したのが、A 社の特殊事情である。実は A 社では、たまたま調査時点で勤続年数 5 年以上 10 年未満の階層が、ほぼ全員、親会社からの出向組で占められていた。そのため、出向者であるかどうかを問うのではなく、勤続年数を「1. ~3 年未満」「2. 3 年以上~5 年未満」「3. 5 年以上~10 年未満」「4. 10 年以上

~15 年未満」「5. 15 年以上~20 年未満」「6. 20 年以上~25 年未満」「7. 25 年以上」のカテゴリーの中から選ばせる通常の勤続年数に関する質問を使って、出向者を特定することができる。調査は留置法による全数調査で、回収率は 99% であった。

そこで、実際に勤続年数の階層別に分離志向 SEGRE と同化志向 ASSIM を使ってクロス表を作成してみた。文化変容の 4 類型では、分離志向 SEGRE と同化志向 ASSIM がともに高い個人が失文化の状態にあると考えられているので、第 5 表のクロス表で分離志向 SEGRE と同化志向 ASSIM がともに高い太い線で枠をした部分に着目してみると、第 5 表 (B) の「勤続年数 5 年以上 10 年未満」の階層では 13 人中 7 人 (54%) がこの失文化の状態にあった。

第 5 表 A 社の SEGRE と ASSIM

(A) 全体

SEGREGATION	ASSIMILATION						計
	0	1	2	3	4	5	
0	0	9	39	135	79	11	273
1	2	11	49	207	99	23	391
2	2	9	47	135	61	19	273
3	2	3	23	45	48	11	132
4	1	2	3	20	10	4	40
5	0	1	2	6	7	1	17
計	7	35	163	548	304	69	1,126

太枠の内部: 232 人 (21%)

網掛けのセルは相対度数 5% 以上

(B) 「勤続年数 5 年以上 10 年未満」=出向者

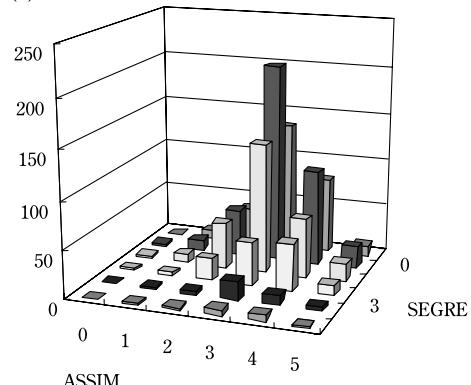
SEGREGATION	ASSIMILATION						計
	0	1	2	3	4	5	
0	0	0	0	0	1	1	2
1	0	1	1	0	0	0	2
2	0	0	1	1	2	0	4
3	0	0	0	1	2	0	3
4	0	0	0	1	0	0	1
5	0	0	0	0	1	0	1
計	0	1	2	3	6	1	13

太枠の内部: 7 人 (54%)

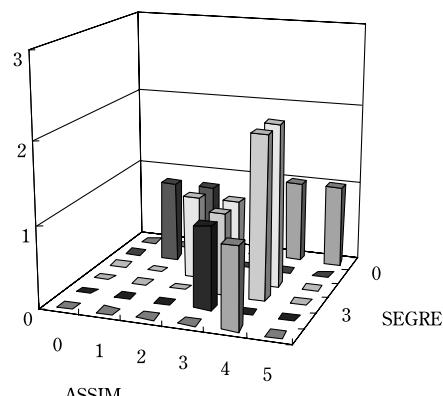
網掛けのセルは相対度数 5% 以上

第 4 図 A 社の SEGREGATION と ASSIMILATION

(A) 全体



(B) 「勤続年数 5 年以上 10 年未満」



ところが、その他の勤続年数階層ではこの太枠内には10%～25%しか入らず、第5表(A)の全体で見ても21%しか失文化の状態の個人はいないのである。つまり、親会社からの出向者は、他の従業員と比べて、2倍以上の割合で失文化の状態にいるのである。

以上のことから、仮説1は検証され、企業の中では、出向者の場合に失文化になりやすい傾向があると考えられる。同時に、まだ開発の緒に就いたばかりで、改良の必要はあるものの、⁸⁾われわれが開発したSEGREとASSIMが、それぞれ分離志向と同化志向の測定尺度として有用であることを示すことができた。

5. む す び

分離志向と同化志向の測定尺度の開発は、まだ緒に就いたばかりであるが、開発スピードの変化を技術コミュニティの内部変化によって説明するモデルを構築するという本稿の目的はある程度達成されたのではないだろうか。技術コミュニティ内のサブコミュニティとして開発者コミュニティとユーザー・コミュニティを考え、文化変容でいわれる分離志向と同化志向をエージェントのルールとして組み込んだマルチエージェント型のコンピュータ・シミュレーション・モデルを構築するという本稿のアプローチは、情報粘着性のようなものと比べて有望なものであると考える。今後は、実際の技術コミュニティにおいて分離志向と同化志向の測定を行い、さらに測定尺度の改良を行うとともに、現実の開発スピードの変化の説明力の検証とコンピュータ・シミュレーションの予測精度の向上に取り組みたい。

- 1) 典型的な製品開発プロセスにおいては、製品の市場リリース時期を目標として開発が行われる。こうした開発方式をここでは「リリース管理方式」と呼んでいるが、少なくとも初期のソフトウェア開発においても踏襲され、従来は、多かれ少なかれ似たような生産管理が行われてきた(Brooks, 1975)。すなわち、社内の開発部門にお

いて試作が繰り返され、ある程度目途が立った(あるいはスケジュール的なデッドラインが迫った)時点でプロトタイプが作成され、このプロトタイプに対して徹底したテストや細かな改善といった品質保証(Quality Assurance)作業が行われてから、最終的な製品として市場にリリースされることが当たり前のように行われてきたのである。これはこれまでの経緯から直感的に、オープン・ソースのソフトウェア開発と対峙させて議論されることが多かった。しかし論理的には「オープン・ソース・ソフトウェアをリリース管理する」ことが可能である。実際、代表的なLinuxディストリビューションとして知られるDebianから派生したUbuntuは、リリース管理を行うことでDebianを上回る人気を得ているオープン・ソース・ソフトウェアである(八田, 2009)。

- 2) von Hippel (1986)は、新しい(novel)もしくは改良された(enhanced)製品・プロセス・サービスの「リード・ユーザー」(lead user)を、次の二つの特徴を示すユーザーだと定義している(p. 796). (i) 市場の大半が遭遇するよりも数ヶ月から数年早く、将来、市場で一般的になるであろうニーズに直面している。(ii) それらのニーズへのソリューションを得ることで、有意な便益を得るポジションにいる。
- 3) 板ガラス成形技術のライセンシング・コミュニティは、ラバース式では一つだったものが、自動平板引上法では複数の亜種が存立する群雄割拠の状態となり、さらにフロート法ではふたたび一つのライセンシング・コミュニティに収束するという経過をたどる(大神, 2009b)。
- 4) 実際、Hofstede (1984, p. 21 邦訳 p. 12)によれば、文化は「一つの人間集団のメンバーを他の集団のメンバーから区別することができる心理の集合的プログラミング」と定義される。
- 5) 二つの質問のYes/Noで 2×2 の四つのセルを作り、そこに同化、統合、拒絶のラベルを入れるという表形式をBerryが提示したのはBerry (1976)が初めてだった。ただし、この表についていたタイトルは「文化変容しているコミュニティに向けられる主要な二つの質問」である。実は、この表は最初、自文化と西洋文化に対する心理的姿勢に関する24の質問項目の分類ラベル、同化、統合、拒絶の三つの整理のために考え出さ

れたものだった。この表が「文化変容の類型」へと変化し、該当する質問がなかったセルにつけられた仮ラベル「(失文化)」の括弧が外されて、「多元社会における文化的関係の可能な形態のモデル」と称した表になるのはBerry (1984)のことである。その後もBerry自身は「失文化」の代わりに「周辺性」を使うことがときどきあり、ややぶれがあるので注意がいる。こうした事情など詳しくは大川(2009b)を参照のこと。

- 6) JFN (TOKYO FM系) 毎週日曜日 14:00～14:55 『山下達郎のJACCS CARD サンデーソングブック』で2009年第一クール(1月～3月)に何度か放送されていたスポンサーであるジャックスのCM。女性の声で、この台詞が流れた後で、男性の「探そう」の声で締めくくられる。
- 7) これはオープン・ソース・ソフトウェアの技術コミュニティのイメージである。リリース管理を行っているような組織的なソフトウェア開発の場合、ドラマ1人で1ヶ月を要するプロジェクトにドラマ2人を投入しても、多くの場合半月では終わらず、むしろコミュニケーション・コストの増大から2ヶ月以上を要する事態に陥ることも多い(ブルックスの法則)ので注意がいる(Brooks, 1975)。
- 8) ただし、もともと互いにあまり相関の高くないうな質問を設定したこともあり、Cronbachの α 係数はASSIMが0.04、SEGREが0.42と低い。またS1～S5の主成分分析の第一主成分の固有値は1.34、固有ベクトルは(0.32, 0.62, 0.42, -0.55, 0.17)。A1～A5の主成分分析の第一主成分の固有値は1.65、固有ベクトルは(0.53, 0.60, 0.21, 0.31, 0.47)。A社1社のデータセットだけを使って結論を出すのはまだ早いが、残念ながらASSIMもSEGREも測定尺度としては、まだまだ改善の必要がある。

参考文献

- 綾部広則(2006)「技術の社会的構成とは何か」『赤門マネジメント・レビュー』Vol. 5, No. 1, pp. 1-18, 2006年1月25日, <http://www.gbrc.jp/>.
- Berry, John W. (1971) "Ecological and cultural factors in spatial perceptual development," *Canadian Journal of Behavioral Science*, 3(4), pp. 324-336.
- Berry, John W. (1976) *Human Ecology and Cognitive Style*, Sage, Beverly Hills (Distributed by Halsted Press, Division of John Wiley & Sons, New York).
- Berry, John W. (1983) "Acculturation: A comparative analysis of alternative forms," In R. J. Samuda, & S. L. Woods (eds.), *Perspectives in immigrant and minority education*, pp. 65-78, New York: University Press of America.
- Berry, John W. (1984) "Cultural relations in plural societies: Alternatives to Segregation and their sociopsychological implications," In N. Miller & M. B. Brewer (eds.), *Groups in contact: The psychology of desegregation*, pp. 11-27, Orlando, Florida: Academic.
- Berry, John W., & Annis, R. C. (1974) "Acculturative stress: The role of ecology, culture and differentiation," *Journal of Cross-Cultural Psychology*, 5(4), pp. 382-406.
- Bijker, Wiebe E., Thomas P. Hughes, & Trevor J. Pinch (eds.) (1987) *The Social Construction of Technological systems*, MIT Press, Cambridge, MA.
- Boehm, Barry W. (1981) *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ.
- Brooks, Frederick P. Jr. (1975; 1995) *The mythical man-month: essays on software engineering*, 1st edition and anniversary edition with four new chapters, Addison-Wesley, Reading, Mass (山内正弥訳『ソフトウェア開発の神話』企画センター, 1977; 原著増訂版(1995)の翻訳: 滝沢徹, 牧野祐子, 富澤昇訳『人月の神話—狼人間を撃つ銀の弾はない—』(原著発行20周年記念増訂版)アジアン・ウェスレイ・パブリッシャーズ・ジャパン1996, 第7刷(2001)以降および新装版(2002)はピアソン・エデュケーション).
- 八田真行(2009)「Linuxディストリビューションの比較—DebianとUbuntu—」『赤門マネジメント・レビュー』Vol. 8, No. 1, pp. 1-18, 2009年1月25日, <http://www.gbrc.jp/>.
- Hirschman, Albert O. (1970) *Exit, voice, and loyalty: responses to decline in firms, organizations, and states*, Harvard University Press, Cam-

- bridge, Mass (三浦隆之訳『組織社会の論理構造: 退出・告発・ロイヤルティ』ミネルヴァ書房, 1975; 第2刷 1980; 矢野修一訳『離脱・発言・忠誠: 企業・組織・国家における衰退への反応』ミネルヴァ書房, 2005).
- Hofstede, Geert H. (1980; 1984) *Culture's Consequences: International Differences in Work-Related Values*, Sage, Newbury Park, Calif, Abridged (ed.), Sage, Beverly Hills (1984年版の訳: 萬成博・安藤文四郎監訳『経営文化の国際比較—多国籍企業の中の国民性—』産業能率大学出版部, 1984).
- Kuhn, Thomas S. (1962) *The Structure of Scientific Revolutions*, University of Chicago Press, Chicago (中山茂訳『科学革命の構造』みすず書房, 1971).
- 大神正道 (2009a) 「技術進化における社会政治的ダイナミクスと技術サイクル、製品の複雑性—経営学輪講 Tushman and Rosenkopf (1992)—」『赤門マネジメント・レビュー』Vol. 8, No. 2, pp. 71–81, 2009年2月25日, <http://www.gbrc.jp/>.
- 大神正道 (2009b) 「板ガラス成形技術の変遷—フロー法の台頭と技術の棲み分け—」『赤門マネジメント・レビュー』 Vol. 8, No. 4, pp. 141–168, 2009年4月25日, <http://www.gbrc.jp/>.
- 大神正道 (2009c) 「支配的技術の変遷におけるコミュニティの影響—板ガラス成形技術の事例研究—」 Discussion Paper Series 2009-MMRC-266, 東京大学ものづくり経営研究センター, 2009年6月, http://merc.e.u-tokyo.ac.jp/mmrc/dp/pdf/MMRC266_2009.pdf.
- 大川洋史 (2009a) 「M&A研究における文化変容アプローチの可能性」『赤門マネジメント・レビュー』 Vol. 8, No. 3, pp. 93–112, 2009年3月25日, <http://www.gbrc.jp/>.
- 大川洋史 (2009b) 「文化変容モデルの誕生—経営学輪講 Berry (1976)—」『赤門マネジメント・レビュー』 Vol. 8, No. 7, pp. 393–408, 2009年7月25日, <http://www.gbrc.jp/>.
- 帽山泰生 (2000a) 「ユーザー・イノベーション」高橋伸夫編著『超企業・組織論』有斐閣, pp. 109–118.
- 帽山泰生 (2000b) 「情報粘着性」高橋伸夫編著『超企業・組織論』有斐閣, pp. 203–212.
- Szulanski, Gabriel (1996) "Exploring internal stickiness: Impediments to the transfer of best practice within the firm," *Strategic Management Journal*, 17 (Special Issue), pp. 27–43.
- 高橋伸夫・高松朋史 (2002) 「オープン・ソース戦略の誤解—Linuxはなぜ成功したのか—」『赤門マネジメント・レビュー』 1(4), pp. 283–308, 2002年7月25日, <http://www.gbrc.jp/>.
- Tushman, Michael L., & Rosenkopf, Lori (1992), "Organizational determinants of technological change: Toward a sociology of technological evolution," In B. M. Staw & L. L. Cummings (eds.), *Research in Organizational Behavior*, Vol. 14, pp. 311–347, JAI Press, Greenwich, CT.
- von Hippel, Eric (1976) "The dominant role of users in the scientific instrument innovation process," *Research Policy*, Vol. 5, pp. 212–239.
- von Hippel, Eric (1986) "Lead users: A source of novel product concepts," *Management Science*, Vol. 32, No. 7, pp. 791–805.
- von Hippel, Eric (1988) *The sources of innovation*, Oxford University Press, New York.
- von Hippel, Eric (1994) " 'Sticky information' and the locus of problem solving: Implications for innovation," *Management Science*, 40 (4), pp. 429–439.
- von Hippel, Eric (2005). *Democratizing innovation*, MIT Press, Cambridge, MA (サイマル・インターナショナル訳『民主化するイノベーションの時代』ファーストプレス, 2005).
- 若林隆久・大木清弘 (2009) 「知識の移転: 粘着性的測定—経営学輪講 Szulanski (1996)—」『赤門マネジメント・レビュー』 Vol. 8, No. 4, pp. 169–180, 2009年4月25日, <http://www.gbrc.jp/>.
- (高橋) 東京大学大学院経済学研究科教授
 (大川) 東京大学大学院経済学研究科博士課程
 (八田) 東京大学大学院経済学研究科博士課程
 (稻水) 東京大学大学院経済学研究科附属
 経営教育研究センター特任助教
 (大神) 東京大学大学院経済学研究科博士課程