

分解—統合プロセスとしての製品イノベーション ン：製品開発と技術開発の同期化

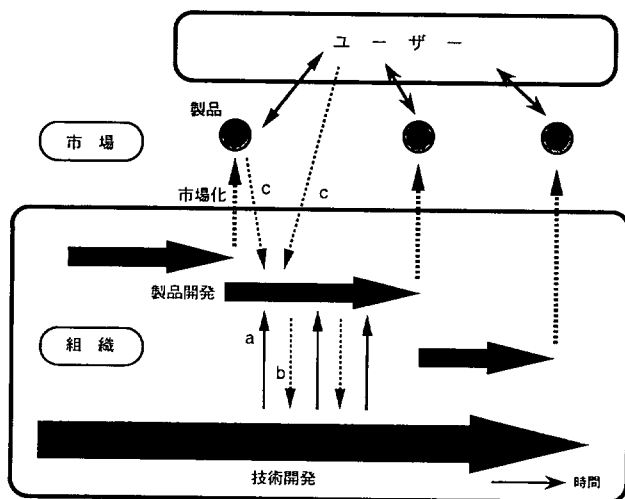
楠 木 建

1 はじめに

組織論・戦略論の領域において、製品イノベーション (product innovation) は近年もっとも注目を集めている研究対象のひとつである。しかし、興味深いことに、従来の研究のほとんどは製品開発 (product development) と技術開発 (technology development) というイノベーションにおける2つの活動を明確に区別せず、暗黙のうちに同一もしくは同質のものとして扱ってきた。従来のイノベーションの研究の中には、製品開発のレベルでの議論もあれば、個別の技術開発に焦点をあわせているものもある。そして多くの研究はこの両者を意識的・無意識的に混同し、いずれをも部分的に含むような形で製品イノベーションを議論しているのである。

現実の製品イノベーションのプロセスにおいては、確かに製品開発と技術開発とは密接に関連しており、現象を観察する上ではこの両者は混然一体となっている場合が多いだろう。しかし、このことは両者が同一ないし同質の活動であるということの意味するわけではない。ここで強調しておきたいことは、製品開発と技術開発は相互的に異なる活動であるということである。従来の研究は、製品開発と技術開発の差異を無視ないしは軽視していたために、製品イノベーションのひとつの重要な側面に十分に光りを当てることができないという限界を含んでいたのではないか、というのがこの論文の基本的な問題意識である。この論文の目的は、製品開発と技術開発の本質的な差異に注目した上で、製品イノベーションのプロセスを記述する概念的なモデ

図1 製品開発と技術開発



ルを提示することにある。「製品イノベーションのプロセスは、製品開発と技術開発がダイナミックに同期化するプロセスとして記述できる」というのがこの論文の基本的な主張である。

2 製品開発と技術開発の平行な関係

技術開発のアウトプット（技術）は製品開発に必要なインプットとして重要なもののひとつである。しかし、このことは単純に技術開発と製品開発との間にリニアな関係があることを意味するものではない。現実のイノベーション・プロセスを記述する上では、製品開発と技術開発との関係を一種の平行なものとしてとらえた方がむしろ説得的である¹⁾。製品開発と技術開発との平行な関係を概念的に図示したものが図1である。

図1に示すように、製品開発のアウトプットは製品であり、特定の製品が開発され市場化された時点でその製品開発プロジェクトは終了する。ここで製品イノベーションとは、開発された製品が従来にない機能を含んでいるか、

機能水準を向上させた場合を指している。いずれにせよ、製品イノベーションを直接的にもたらすものは個別の技術開発ではなく、あくまでも製品開発である。

一方で、製品開発にとって必要となる技術は技術開発のアウトプットである(aの矢印)。逆にいえば、製品開発は技術開発の成果である個々の技術を利用したり組み合わせたりしながら製品に体现させ、特定の形をもった製品へとまとめあげていく活動である。また、製品開発のプロセスで特定の技術が必要になり、製品開発からの要請に基づいて技術開発の一部が方向づけられるという逆の関係もあるだろう(bの矢印)。さらに、製品開発のプロセスではユーザーのニーズといった市場情報も重要な役割を果たす(cの矢印)²⁾。たとえば、市場化された製品に対するユーザーの反応は重要な市場情報として次期の製品開発にフィードバックされるであろう。

製品開発と技術開発は、いずれも一連の問題解決(problem solving)のプロセスとして記述できる³⁾。すなわち、製品開発と技術開発は、いずれも特定の問題を発見・設定し、それを解決することによって、最終的に「製品」ないしは「技術」という解を導出するプロセスとして理解できる。

製品開発と技術開発の間にはいくつかの共通点がある。たとえば、製品開発と技術開発はいずれも一種の「実験」であり、組織にとって未知の情報・知識を創造していくプロセスである。このような意味で、生産のようなルーチン化が容易な活動と比較して、製品開発と技術開発はいずれも高い不確実性を取り込む活動である⁴⁾。しかし、その一方で製品開発と技術開発とでは、問題解決プロセスとしての特性がさまざまな側面で大きく異なる。上述したように製品開発と技術開発は相互に密接に関係しているけれども、この両者はパラレルに進行するものであり、概念的には区別されるべきものであるというのがここでの基本的な発想である。

以下の2つの節では、製品開発と技術開発それぞれにおける問題解決プロセスを概念的に記述するという作業を行う。この作業の目的は、それぞれの問題解決プロセスに組み込まれた特性を考察した上で、両者の差異を分析す

ることにある。

3 技術開発における問題解決

技術開発における問題解決は、まず解決すべき大きな製品技術上の問題を設定することから始まる。ここで技術開発の目的は、設定された技術的問題に対して、「技術」という特定の解の集合を与えることにある。しかし、初期に設定された上位問題から一足とびに特定の解（技術）に到達することは一般的にいて不可能である。なぜならば、初期に設定された上位問題と最終的に導出されるべき解との間には大きな距離があり、両者の対応関係は一般に不明確であるからである。

このことの背景には「技術」という解がもっている次のような特性が関連している。そもそも技術とは、特定少数のパラメータによってそのパフォーマンスを把握できる程度にまで定式化された因果関係についての知識であり、典型的には特許に申請されるような内容規定の明確な知識である。すなわち、問題解決のアウトプットとしての技術的な解は、結局のところ因果関係が明確になる程度までに特定化された個別技術を単位として現出するのである。

ところが、その一方で技術にはさまざまな個別技術によって構成される複雑なシステムとしての側面がある⁵⁾。上の議論との関連でいえば、初期に設定された技術的問題は、このレベルではあくまでも「システム」が提示する問題であり、多数のパラメータを潜在的に含んでいるといえる。上位の技術的問題は単純に既知のパラメータを集計したものではなく、それが暗黙のうちに組み合わせられた「プロトタイプ」(prototype)として把握されるのである⁶⁾。それゆえ、プロトタイプとして設定された上位問題を単一の技術的な解によって解決することは一般的にいて困難であり、ひとつの問題に対応して最終的には複数単位の解（技術）が必要となる。

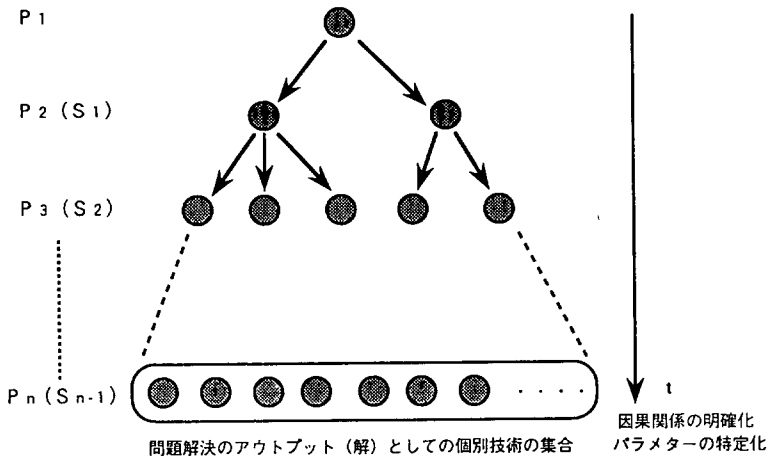
自動車の具体例で考えてみよう。いま仮に「走行安全性の向上」という製品技術に関わる上位問題を設定したとしよう。この問題に対する技術的な解としては、単純な例をあげれば「(ボディ剛性を向上させるような)スタイリ

ング技術」が考えられる。しかし、自動車の製品技術が複数の個別技術から構成されるシステムである以上、「スタイリング技術」という解だけではなく、これ以外にもボディに用いる素材に関連する技術やより機敏な操舵を実現する技術、より強力なブレーキングを実現する技術、といった可能な解が他にも数多く存在する。「走行安全性の向上」という問題解決をより効率的・効果的に進めるためには、複数の解が導出されなければならない。

それゆえ、現実の技術開発における問題解決は、上位問題を一度に完全に解決するのではなく、部分的に解決することによって新たに複数の問題を発見・導出し、再度これを部分的に解決するというように、上位問題の部分的な解決と新たな下位問題の設定（ないしは上位問題の再設定）という作業を繰り返しながら階層的（hierarchical）に進行する。ここで問題が部分的に解決されるということは、下位問題に含まれる因果関係の知識が上位問題に比較して一定程度まで明確になるということである。いいかえれば、このプロセスは相対的に相互依存性が低く、因果関係に含まれる変数がより制御（control）された下位問題へと上位問題が分解されるプロセスである⁷⁾。問題解決が階層的に進行するにしたがって、問題は特定化され、パフォーマンスを測るパラメータが絞り込まれ、因果関係についての知識が明確になり、導出される技術的な解の性能が高まっていく。ようするに、技術開発における問題解決は、上位問題を階層的に解決しながら、より定式化された操作可能性が高い解へと徐々に接近していくというプロセスをたどるのである。

技術開発における問題解決のプロセスを概念的に図示したものが図2である。技術開発における問題解決はまず最上位の問題（P1）の設定からはじまり、問題が相対的に相互依存性が低い下位問題（P2；P3）へと分解され、特定化されていく。この下位問題への分解は、同時に上位問題に対する部分的な解（S1；S2）を導出していくプロセスでもある。すなわち、 $P1 \rightarrow P2 \rightarrow P3$ という階層的な問題解決は「目的-手段」の連鎖として理解できる。このプロセスを繰り返しながら、最終的には問題解決の最下位に解の集合（ S_{n-1} ）が導出される。図2の問題解決プロセスにおいて、因果関係の知識はこの

図2 技術開発における問題解決



レベルでもっとも明確になるのであり、技術開発のアウトプットとしてもたらされる技術とは、結局のところ階層的な問題解決の最下位に位置する解集合にほかならない。

自動車の例に戻ろう。「走行安全性の向上」として設定された上位問題 (P1) は、たとえば「ボディ剛性の向上」や「ブレーキ性能の向上」といった部分的な解 (S1) を導出することによって下位問題 (P2) へと分解される。さらに「ボディ剛性の向上」という問題は、「スタイリングの向上」や「ボディ素材の改良」という部分的な解 (S2)、すなわち下位問題 (P3) のレベルへと分解される。このように問題解決の階層を次々に下降することによって、たとえば「素材 A と素材 B を $x:y$ の比率で合成した鋼板をボディ材として採用した場合にもっとも剛性が高くなる」といった、 S_{n-1} のレベルに位置するような操作性の高い解 (技術的知識) に到達するのである。

技術開発をこのような問題解決プロセスとしてみた場合、次のような技術開発に固有の特性が指摘できる。第1に、技術開発は本質的にインクリメンタルに進行していくプロセスであり、高い連続性 (continuity) と累積性

(accumulativity) で特徴づけられるということである。上で議論したように、技術開発における問題解決は階層的であり、階層をひとつひとつ下降しながら部分的な問題解決を繰り返すという形で、技術的な知識が徐々に獲得されるからである。設定される技術的な問題が複雑なものであるほど、特定の解集合に到達するまでには多くの階層を経る必要がある。したがって、設定される上位問題が複雑なものであるほど、このような連続的・累積的な特性は顕著になるだろう。いずれにせよ、技術開発における問題解決は、ある時点で突然起こるという性格のものではなく、インクリメンタルにはあるけれども「常に」進行しているのである。

ただし、このことは問題解決のアウトプットとしての技術的知識に「飛躍」がないということの意味するものではない。しばしば観察されるように、「技術の飛躍」という現象は起こりうる。しかし、この場合でも一連の問題解決のプロセスそれ自体に飛躍があったと考えるべきではない。「飛躍」を含むような技術的知識がもたらされるのは、階層的な問題解決そのものにある種の非連続性があるからではなく、むしろ既存の上位のレベルで新しい問題が追加されたり、上位問題の設定が大幅に変更されたり、下位問題への分解の仕方が変更されることがあるからである。ある種の「飛躍」をとまなうような技術開発であっても、その解に至る問題解決のプロセスそれ自体は同様にインクリメンタルに進行しているのである。このような意味で、技術開発における問題解決は、それがもたらす解の性質に関わらず、本来的に連続的で累積的なプロセスをとる。

技術的知識はある種の「ストック」として組織に連続的に累積されていくのであり、あらゆる技術的知識はそれまでの連続的な問題解決の上になりたっている。したがって、下位レベルの特定化された技術的知識だけを直接的に取り出して変更を加えたり、技術的知識の組み替えを行うということは一般に困難である。いいかえれば、技術開発の問題解決においては部分的な学習棄却 (unlearning) はきわめて困難なのである。「技術の飛躍」を現実するためにはより上位レベルでの問題の組み替えが必要であるけれども、このこ

とはそれまでに累積された既存の技術的知識を包括的に「棄却」ということを意味している。

第2に、技術開発における問題解決はクローズド・システムとしての側面をもっている。繰り返し述べるように、技術開発における問題解決は、準分解可能な単位への上位問題のブレイクダウンを繰り返すことによって「因果関係の知識」という操作性の高い解を導出することを目的としている。これは論理の一貫性と完結性を第一義的に追求するプロセスであり⁸⁾、介在する変数を徐々に排除しながら技術の論理を深耕 (narrow focusing) していくプロセスである。逆にいえば、問題解決プロセスを深耕していくということは、導出される下位問題ないし解を不特定多数の影響源から隔離していくことであり、問題解決のプロセスのクローズド・システムとしての側面を強化していくということである。ということは、技術開発における問題解決はある意味で不可逆的なプロセスであり、マネジメントの意思や市場情報などの技術の論理の外部にある影響源から遮断されて「自走する」という側面をもっているということの意味している⁹⁾。それゆえ、技術開発の問題解決は安定した方向性をもったトラジェクトリー (technological trajectory) に沿って進行するのである¹⁰⁾。

このように考えると、技術開発をインテンシブに押し進めている組織であるほど技術開発がクローズド・システムをなす傾向が強まるのであり、そこでの問題解決プロセスはより不可逆的になると仮定できる。技術的な知識を蓄積し、技術のフロンティアを追求するということは、問題解決のプロセスを深耕していき、結局のところ上述したクローズド・システムとしての側面を強化することになるからである。このような場合、より上位レベルの問題の組み換えを必要とするトラジェクトリーのシフトや「飛躍」を含むような解の導出は相当に困難になるだろう。

第3に、技術開発における問題解決にはきわめて高い継続性 (sustainability) が組み込まれており、技術開発はきわめて長い時間志向性をもっている。技術開発における階層的な問題解決は、「部分的な解を導出する

ことによって上位問題を下位問題へと分割する」という作業を繰り返すプロセスである。つまり、ある問題に対する解を導出することは、同時に新しい下位問題を提出することを意味している。したがって、仮にいま階層的な問題解決を繰り返すことによって図2の S_{n-1} のレベルの技術的知識を獲得していたとしても、別の見方をすれば、依然として P_n というレベルの問題が未解決のまま残されているのである。すなわち、 P_n という問題に対応した S_n のレベルの解をさらに探索し、 P_n をさらに下位問題 (P_{n+1}) に分割し、よりいっそう明確な因果関係の知識を求めて問題解決プロセスを引き続き深耕していくことが論理的には可能である。より上位のレベルで問題が変更されたり、新しい問題が追加されない限り、さらに明確な因果関係をもつ解を求めて技術開発は末広がりに行進し続ける。ようするに、技術開発における問題解決それ自体には特定の終着点がないのであって、技術開発のダイナミクスにはきわめて高い継続性と拡散性 (divergency) が組み込まれているのである。

4 製品開発における問題解決

次に、製品開発における問題解決のプロセスを概念的に記述してみよう。ここで強調したいことは、製品開発における問題解決は、技術開発の場合と異なって、問題の「分解」(disintegration) と分解された問題の「統合」(integration) という2階段のプロセスを必要とするということである。

製品開発における問題解決も、まず開発されるべき製品についての上位問題を設定することからはじまる。この上位問題は、開発される製品が含むべき市場や技術についての属性を圧縮したものであり、しばしば「製品コンセプト」(product concept) と呼ばれる。製品開発における問題解決の目的は、設定された上位問題を満足させるような「製品」という解を導出することにある。ここで技術開発の場合と異なるのは、製品開発においては初期に設定された上位問題と「製品」という解の明確な対応関係が維持されるという点である。

技術開発の場合、問題解決の解としての技術は特定の因果関係についての局部分化された知識であるから、明確なパラメータないしは「次元」(dimension)によって論理的に把握することができる反面、それ自体ではあくまでも「目に見えない」ものである。これに対して、製品開発における問題解決の解は多種多様な次元を暗黙のうちに含んでいる。つまり、解は個別の次元によってではなく、不特定多数の次元を包括したプロトタイプとして把握される。製品開発においては、問題解決の解を論理的かつ明確に把握することが困難になる反面、解全体が「かたち」(form)をもつため、そこに一定の可能性 (visibility) が確保できるのである。一方で、初期に設定される上位問題もさまざまな属性を圧縮した製品コンセプトであるから、数多くの次元を潜在的に含むプロトタイプの側面をもっている。すなわち、製品開発においては初期に設定される上位問題と問題解決の解とはいずれもプロトタイプとして把握されるものであり、この意味で両者の対応関係は相対的に明確なるのである。

ふたたび自動車の具体例で考えてみよう。たとえば、日産のプリメーラの開発における問題解決では、「アウトバーンで、しっかり、速く、快適に」(Sure, Fast, and Comfortable on the Autobahn) という製品コンセプトが上位問題として設定されている¹¹⁾。この表現をみても明らかなように、この上位問題にはさまざまな次元ないしは属性が包括的に含まれている。一方で、最終的に問題解決の解としてもたらされる製品 (プリメーラ) も、さまざまな次元を包括的に含むプロトタイプとして把握される。このように、上位問題と解との間にプロトタイプとしての共通性が確保されるので、製品開発においては両者の対応関係が1対1の関係として明確になるのである。製品開発における問題解決は、製品コンセプトという抽象度の高いプロトタイプを、製品という実体のあるプロトタイプへと変換していく形態化のプロセスであるといえる。

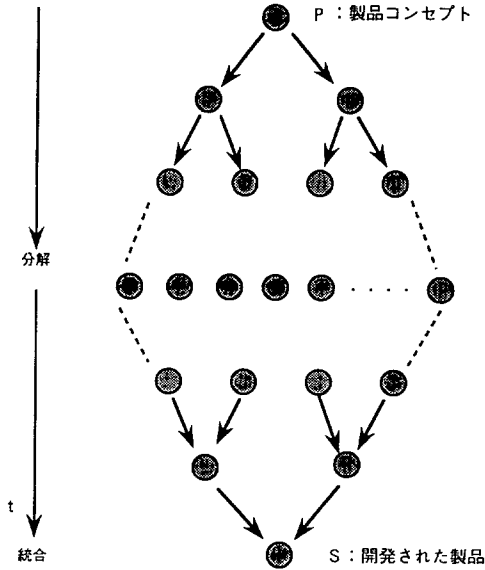
しかしながら、上位問題と解との対応関係が明確であるからといって、問題解決が上位問題から解へと一足とびに進行するわけではない。製品開発に

においても、問題解決はやはり階層的に進行する。なぜならば、製品開発における問題解決は技術的知識のインプットを必要とするからである。技術開発の場合と同様に、製品開発においても上位問題それ自体はきわめて操作性を欠いた形で設定されている。すでに議論したように、技術的知識はそもそも特定の因果関係についての知識であるから、このように問題の操作性が低く、問題が特定化されていない状態のままでは、個別の技術的知識(技術開発における問題解決の解)を製品開発の問題解決において利用することは一般に不可能である。したがって、技術開発の場合と同様に、製品開発における上位問題も問題の分解を繰り返しながら一定の程度にまで特定化されなければならない。製品開発においても、上位問題を部分的に解決しながら複数の下位問題へと分解するという、「目的-手段」の連鎖からなる階層的な問題解決のプロセスがまず現れるのである。

ただし、技術開発の場合とは異なって、製品開発における問題解決はどこまでも継続的に拡散していくわけではない。問題が一定のレベルにまで分解され、技術的知識がインプットされたならば、その後の問題解決は、製品というさまざまな属性を包括的に含む解に向かって、逆に個別の解を階層的に統合していくプロセスをたどる。この問題解決のプロセスは、いわば連立方程式を解くように、小さな単位に分解された部分解を組み合わせ、調整し、製品という解へとまとめあげていくプロセスである。以上の議論を総合すると、製品開発における問題解決は、はじめに上位問題の分解によって問題を操作化していき、後に操作化された各部分の統合によって製品という解を導出するという2段階のプロセス(分解-統合プロセス)として記述できる。これを概念的に図示したものが図3である。

プリメーラの開発の場合、上述した上位問題(製品コンセプト)は、「ハイスピード・パフォーマンス」、「快適設計(コンフォート10)」、「スタイル」という下位問題に分割された。さらに、たとえばこのうちの「ハイスピード・パフォーマンス」は、「軽量化」、「エンジン/ミッション/シャーシ」、「車体&エクステリア」という下位問題に分割され、このような問題解決を通じ

図3 製品開発における問題解決



で「フラッシュサーフェスにより CD 値 0.29 以下」（「車体&エクステリア」の下位問題）という、相対的に操作化された下位問題が導出された。このように、問題解決を階層的に進めることによって問題は徐々に操作化され、その結果、特定の因果関係についての技術的知識との対応関係が明確化していく。ようするに、この問題解決における「分解」の段階は、製品開発における技術的知識の利用可能性を高めていくプロセスである。そして、技術的知識のインプットを受けた後は、プリメーラという製品として完全に形態化された解に到達するまで、「統合」による問題解決が階層的に繰り返されるのである。

製品開発をこのような2段階の問題解決プロセスとしてみた場合、次のような製品開発の特性が指摘できる。第1に、技術開発が高い連続性と累積性で特徴づけられるのに対して、製品開発には一定の非連続性（discreteness）が定義的に組み込まれている。上述したように、製品開発においても問題解

決のプロセスそれ自体はインクリメンタルに進行する。しかし、製品コンセプトという上位問題の設定にはじまる問題解決のプロセスは、製品という解を導出し、市場化した時点で終了するのであり、これが技術的知識を「ストック」として累積していく技術開発と決定的に異なる点である。仮に次の製品開発が引き続き進められるとしても、それはあくまでも新たに製品コンセプトを設定するところからはじまるのであるから、そこには必然的に一定の非連続性があることになる。

ただし、製品開発が非連続的であるという上の議論には一定の制約がある。なぜならば、製品開発における問題解決は、製品というアウトプットをもたらすと同時に、その副産物としてある種の知識をもたらすからである。それは個別の技術そのものについての知識ではなく、個別の技術的知識のあり方についてのアーキテクチャルな知識 (architectural knowledge) である¹²⁾。すでに述べたように、技術開発における問題解決は次々にパラメータを絞り込み、因果関係を特定化していくプロセスであり、このプロセスをどこまで深耕したとしてもアーキテクチャルな知識は生まれない。これに対して、製品開発における問題解決は、問題を分解し特定化するだけでなく、部分的な解の組み合わせや調整を通じて問題を統合していくプロセスを含んでいる。「分解—統合」の2段階のプロセスであるがゆえに、製品開発における問題解決はアーキテクチャルな知識をもたらすのである。個別の技術的な知識と同様に、問題解決の副産物であるアーキテクチャルな知識は製品開発においてきわめて重要な役割を果たす。アーキテクチャルな知識は、製品という直接のアウトプットとは別に組織に蓄積されていくものであり、この意味で製品開発には累積的な側面も一部含まれるということに注意が必要である。

第2に、技術開発の場合と比較して、製品開発における問題解決は相対的にオープンなシステムである。すでに述べたように、最終的に導出される製品という解はさまざまな次元を包括的に含むプロトタイプとして把握されるものであり、そのパフォーマンスを評価するパラメータも多岐に渡る。これ

表1 製品開発と技術開発における問題解決の特性

項目	技術開発	製品開発
問題解決の解 解の特性	<ul style="list-style-type: none"> —技術 —因果関係についての明確な知識 —特定化された次元によって把握可能 —可視性が低い 	<ul style="list-style-type: none"> —製品 —形態化されたプロトタイプ（アーキテクチャ的な知識） —不特定多数の次元を暗黙のうちに含む —可視性が高い
問題解決プロセスの形態	—分解プロセス	—分解-統合プロセス
システム特性	—クローズド・システム	—オープン・システム
ダイナミクス	—連続的・累積的・拡散的	—非連続的・収束的
時間志向性	—長期的	—短期的

は論理の一貫性と完結性を追求する技術開発と決定的に異なる点である。製品が最終的には市場化されるものである以上、製品開発はさまざまな市場情報や競合他社の製品と無関係におこなうことはできない。製品開発における問題解決のプロセスは、市場情報やマネジメントの意思などのさまざまな影響源に対して相対的にオープンである必要がある。

第3に、技術開発の場合とは異なって、製品開発における問題解決には「特定のまとまりをもった製品の開発（市場化）」という明示的な終着点がある。製品開発における問題解決は最終的には製品という解に向かって収束していく。製品開発のダイナミクスには本来的に収束性（convergency）が組み込まれているのである。この意味で、問題解決の時間志向性は相対的に短くなるといえる。したがって、図1に示したように、技術開発が長期に渡って連続的・継続的に展開される間に、複数の製品開発が断続的に繰り返されるのが普通である。

以上で考察してきたように、技術開発と製品開発の問題解決には相互に異なる特性が一貫したパターンをもって組み込まれている。技術開発と製品開発それぞれの特性と両者の差異を要約したものが表1である。

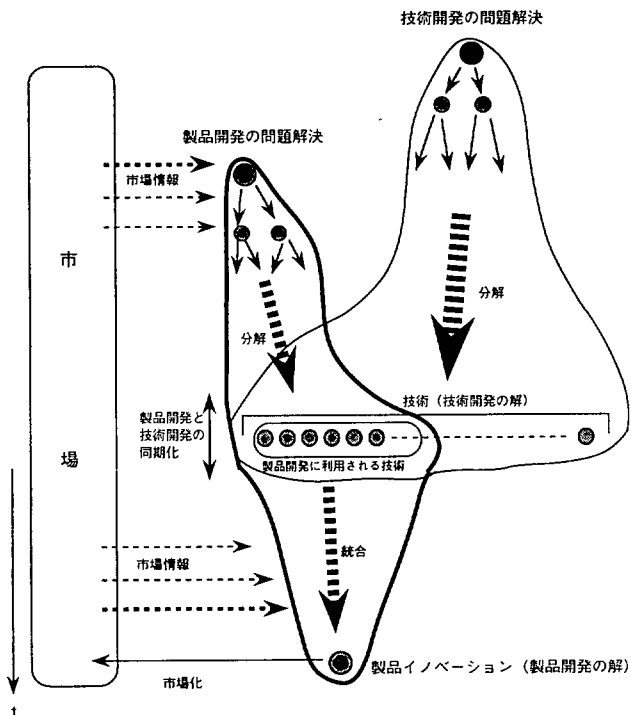
5 製品開発と技術開発の同期化

製品イノベーションは直接的には製品開発のアウトプットであるから、製品イノベーションを生み出すプロセスも一般に「分解—統合」の問題解決プロセスとして記述できる。すでに述べたように、上位問題と解との対応が明確であるにも関わらず、ここでの問題解決プロセスが「分解—統合」という2段階の形態をとる理由は、そうしなければ上位問題を操作化できず、したがって問題解決において必要となる個別の技術的知識との関係が把握できなくなるからである。これは製品開発と技術開発の「インターフェース・ギャップ」の問題である。

製品イノベーションないし製品開発のプロセスとは、見方を変えれば、製品開発と技術開発のインターフェース・ギャップを解消することによって、上位問題を解決する上で必要となる技術的知識（技術開発における問題解決の解）を一連の問題解決に取り込もうとするプロセスとして理解できる。ようするに、製品イノベーションを実現するためには、製品開発と技術開発という異なる2種類の問題解決があるレベルで同期化(synchronize)する必要があるのである。製品イノベーションのプロセスとは、一面では製品開発と技術開発の同期化(synchronization)のプロセスである。

製品開発と技術開発の同期化に注目して、製品イノベーションの問題解決プロセスを概念的に図示したものが図4である。製品イノベーションという解を導出するためには、個別の技術的知識を問題解決のプロセスに取り込まなくてはならない。そのためには、特定化された個別の技術的知識とのインターフェース・ギャップが解消されるまで、製品開発上の問題が分解されなくてはならない。製品開発と技術開発それぞれの下位問題の対応関係が明確になり、インターフェース・ギャップが解消されたときに、はじめて技術的知識を製品開発に利用することが可能になるのである。これが製品開発と技術開発との同期化が実現された状態であり、異なる2つの問題解決が実質的に重なり合った状態である。製品開発が技術開発に同期化することによって、

図4 製品開発と技術開発の同期化



個別の技術的知識の一部が製品開発に選択的に取り込まれ、局部化された問題に対する部分解が与えられる。この部分解を階層的に統合していくことによって、最終的に製品イノベーションという形態化された解が導出される。

このように考えると、製品開発の問題解決における「分解」の段階は、技術開発の問題解決との同期化が実現されるまで継続するといえる。ということは、製品開発の問題解決は技術開発から独立に進行するのではなく、技術開発の問題解決が提示している解（個別の技術的知識）をにらみながら、それとの同期化を実現しやすいような方向に進んでいく、ということの意味している。

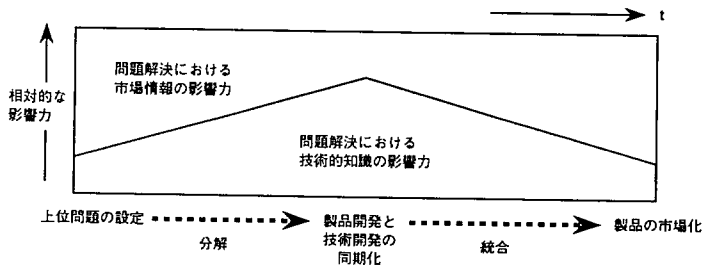
もちろん、製品開発と技術開発の同期化は相互作用的(interactive)に実現されるのであって、技術開発の問題解決を所与として、製品開発の問題解決が一方的にそれに従属するわけではない。同期化が実現されるプロセスには、製品開発の問題解決が技術開発の問題解決を方向づけるという側面もある。しかし、同期化は完全な意味での「相互作用的」なプロセスではない。一般的にいて、同期化が実現されるプロセスでは、「技術開発の問題解決が製品開発の問題解決を方向づける」という方向の影響関係が相対的に強くなる傾向がある。

その理由には、主として次の2つがある。第1に、技術開発に時間的先行性があるからである。一般的にいて、まったく技術的な知識が蓄積されていない状態で製品開発が行われるということはまれであり、特定の製品開発がスタートする時点で、技術開発の問題解決はすでに一定の程度まで進行しているのが普通である。たとえば、モデル・チェンジを繰り返す自動車の場合、このことが典型的に当てはまる。この場合、個別の技術的知識は製品開発に先行してかなりの程度まで蓄積されているのであって、それゆえ、製品開発と技術開発の同期化は、先行的に蓄積されている個別の技術的知識をかなりの部分所与として、製品開発における「分解」のプロセスがそれに歩み寄るという形で実現される。

第2の理由は、技術開発の問題解決に本来的に組み込まれている特性に関連している。すでに議論したように、技術開発における問題解決は連続的・蓄積的・継続的に進行する。そして、問題解決は進行するにしたがってクローズド・システムをなすため、技術の論理を自己完結的かつ不可逆的に追求するプロセスとなる。それゆえ、技術開発が製品開発の問題解決からくる要請に柔軟に対応するということは一般に困難なのである。

いずれにせよ、組織にすでに一定の技術的知識が蓄積されており、それを利用しようとする以上、同期化を実現するプロセスにおいては、「技術開発が製品開発に同期化する」というよりも「製品開発が技術開発に同期化する」という色彩が相対的に強くなるというバイアスが発生するのである。技術開

図5 製品イノベーションの問題解決における影響要因のミックス



発の問題解決をより深耕している組織であるほど、このバイアスは強く作用するであろう。

さて、冒頭で述べたように、製品イノベーションは技術と市場情報という2つの要因によって方向づけられる¹³⁾。しかし、製品イノベーションにおける問題解決のプロセスを製品開発と技術開発との同期化を中間にはさんだ「分解—統合プロセス」として考えると、この2つの要因のミックスは問題解決のプロセス全般に渡って均等ではなく、問題解決の段階によって図5のように変化すると仮定できる。

製品開発と技術開発の同期化が実現される段階やその前後では、問題解決に影響を与える要因はまさに技術的知識の側にある。この段階では問題が特定化された部分に分解されているため、製品についての市場情報との対応関係はきわめて不明確であり、市場情報は直接的にはほとんど影響力をもち得なくなる。これに対して、上位問題を設定するというような分解プロセスの初期の段階や最終的に製品という解へと部分解を統合していく段階では、問題はプロトタイプとして把握されるのであり、製品についてのさまざまな市場情報に対してオープンである。この段階の問題解決に影響を与える要因は、特定化された技術的知識よりもむしろ市場情報の側にあるといえるだろう。したがってこの場合、問題解決を方向づける要因のミックスにおいては市場情報が支配的になる。

以上、この論文では、製品イノベーションのプロセスを概念的に記述するモデルを提示してきた。製品開発と技術開発とでは問題解決の特性が本質的に異なるけれども、製品イノベーションのプロセスにおいてはこの異なる2種類の問題解決がともに必要とされる。それゆえ、製品イノベーションのプロセスでは、製品開発と技術開発それぞれの問題解決の同期化が決定的に重要な意味をもっている。この論文の考察は、企業における製品イノベーションないしは製品開発と技術開発との関係に関わるユニークな洞察を提供するはずである。その点の考察は別稿で改めて展開することにしたい¹⁴⁾。

- 1) Cf. Kline, Stephen J. and Nathan Rosenberg, "An Overview of Innovation." in Ralph Randa and Nathan Rosenberg (ed), *The Positive Sum Strategy: Harnessing Technology for Economic Growth*, Washington DC: National Academy Press, 1986.
- 2) Cf. Utterback, James M., "Innovation in Industry and the Diffusion of Technology." *Science*, No. 183, 1974, pp. 620-626.
- 3) Cf. von Hippel, Eric, "Task Partitioning: An Innovation Variable." *Research Policy*, No. 19, 1990, pp. 407-418.
- 4) Cf. Kline, Stephen J., and Nathan Rosenberg, op. cit.
- 5) 今井賢一「イノベーションと企業の戦略・組織」今井賢一（編著）『イノベーションと組織』東洋経済新報社, 1986年。
- 6) Cf. Jaikumar, R. and R. E. Bohn, "The Development of Intelligent Systems for Industrial Use: A Conceptual Framework." *Research on Technological Innovation, Management and Policy*. Vol. 3, JAI Press, 1986.
- 7) Cf. Simon, Herbert A., *Administrative Behavior* (3rd edn.). New York: Free Press, 1976.
- 8) 榊原清則「メタテクノロジー：技術のダイナミクス」『一橋論叢』第87巻第3号, 1982年。
- 9) Cf. Rosenberg, Nathan, *Perspectives on Technology*. Cambridge: Cambridge University Press, 1976.
- 10) Cf. Doci, Giovanni, "Technological Paradigms and Technological Trajectories: A Suggested Interpretation of the Determinants and Directions of Technical Change." *Research Policy*, Vol. 11, 1982, pp. 147-162.
- 11) 引用したプリメーラの事例は次による。

野中郁次郎「グローバル組織経営と知識創造」『組織科学』第25巻第4号，1992年。

- 12) Cf. Henderson, Rebecca M. and Kim B. Clark, "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms." *Administrative Science Quarterly*, Vol. 35, No. 1, 1990, pp. 9-30.
- 13) Cf. Kline, Stephen J., and Nathan Rosenberg, op. cit.
- 14) Cf. Kusunoki, Takeru, "The Dilemma of Technological Leadership: A Conceptual Framework." *Hifotsubashi Journal of Commerce & Management*, Vol. 27, No. 1, 1992. (forthcoming)

(1992年8月)

(一橋大学専任講師)