

アーキテクチャー視点による製品技術体系の戦略的分析

魏, 聡哲

<https://doi.org/10.15017/3000344>

出版情報：経済論究. 119, pp.17-33, 2004-07. 九州大学大学院経済学会
バージョン：
権利関係：

アーキテクチャー視点による製品技術体系の戦略的分析

魏 聡 哲

目次

1. はじめに
2. アーキテクチャーの視点による製品技術体系の分析
3. アーキテクチャーから見た製品技術体系の戦略的なロジック
4. おわり

1. はじめに

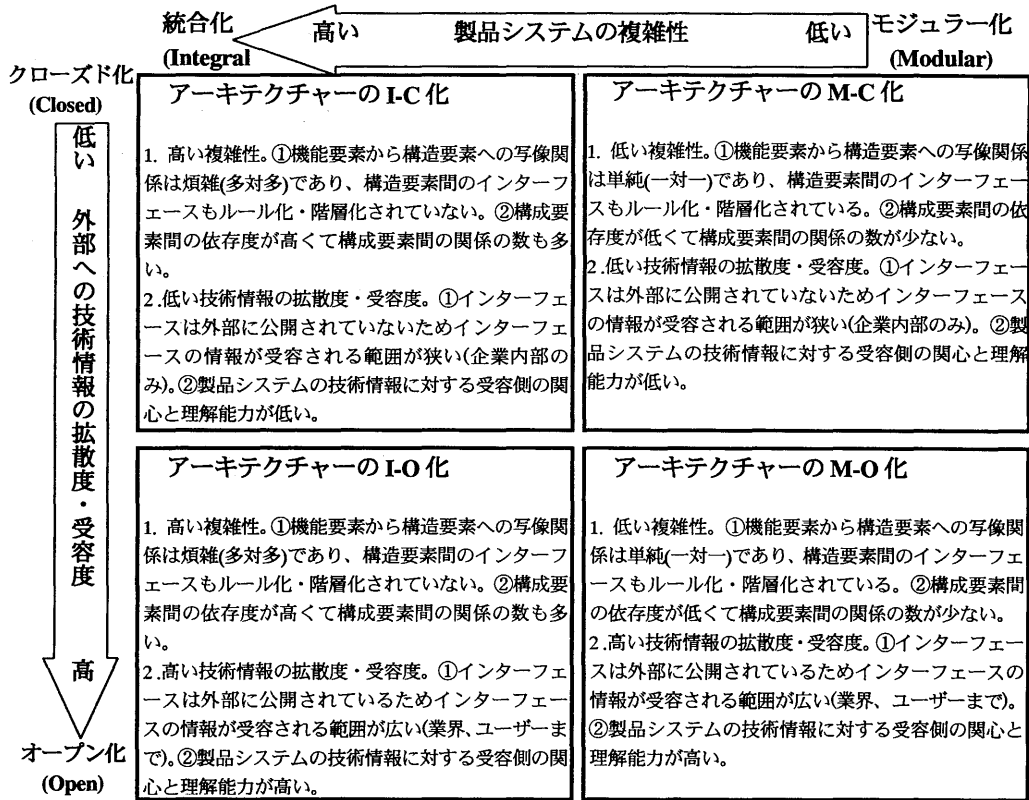
1990年代半ば以降、ハイテク電子産業において、市場に要求される機能に対する部品同士間の自己完結化による機能モジュール市場、EMSやODMという電子製造アウトソーシング市場規模は急成長しているうちに、競争が激化する傾向が明らかに見える。例えば、デスクトップ・パソコンでは、組立に必要な部品ユニット（機能モジュール）企業間の取引以外にも、電気量販店でも自由に買えるようになったため、デスクトップ・パソコンの組立や部品ユニットの製造の分野では、多くの企業が簡単に参入し激しい競争が繰り広げられている。そのような市場競争の特徴はM-O化（Modular/Open）される製品業界において特に明らかに見える。

ある製品の製造分野への技術的な参入障壁が低くなる一つ重要な理由は、M-O化の進展に基づく製品製造のノウハウが理解されやすいことにより技術知識にかかわる障壁も徐々に低下するところにある。同様に製品のライフサイクルに沿って企業が取り組む製品技術のノウハウの体系（System of technologic know-how）は

未成熟な混沌の状態から成熟的な標準化の状態へ変わっていくうちに、業界における製品技術ノウハウの拡散もいっそう進行していく。そうすると、製品技術体系（Product Technology System）¹の成熟化はM-O化という概念で述べられるかもしれない。また、そのような技術ノウハウの体系のM-O化に深く影響しているのは、川上における製品のコンセプトの創出と製品システムの設計の行方にあると考えられる。そして、製品のコンセプトの創出と製品システムの設計との行方を戦略的に理解することができる分析ツールの一つは、アーキテクチャー（Architecture）という概念である。まとめると、近年以来、ハイテク電子産業においては、製品市場の競争が激化する一つ重要な理由は、製品システムのレベルのコンセプトと設計に関するアーキテクチャーのM-O化による製品の製造分野への技術的な参入障壁が低くなることに起因しているかもしれない。したがって、そのような製品のアーキテクチャーの転換は製品

1 システム（System）とは、「それぞれの要素の動きが、その要素同士が集まるグループ内部で特定の関係に依存しているため、各部分に散在している要素の性質だけで説明できない構成的特性をもった複合体であり」、つまり、相互に関係している要素間の性質を説明できる複合体でもある（L.von Bertalanffy [1968], 高橋 [2000], 作者加筆）。それによって、本稿が焦点を当てたい製品技術体系（Product Technology System）とは、「多数構成要素技術間の性質を説明できる製品化された複合体」と定義できる。また、すべての製品は、さまざまな部品から構成されなければならない理由で、業界の製品技術全体の変化を述べる際にして、部品要素技術を単に説明するよりも、むしろ製品技術体系という視点から分析した方が有意義である。

図3 アーキテクチャー視点による製品技術体系の類型化



出所: 藤本[2001], 柴田・児玉[2001]を参考し作成。

図4 製品のアーキテクチャー視点による戦略的なロジック

<p>I-C型アーキテクチャーの戦略ロジック</p> <ol style="list-style-type: none"> 1. 製品技術の差別化優位への追求。 2. 少数企業間での頻繁なコミュニケーションによる系統的な学習の効果が期待できる。 3. 製品の設計 (モデルチェンジ)・生産にかかわる組織間の技術的な制約が存在することによって、ビジネス・プロセスにおける製造の事業活動間関係の内部分断は困難であり、外部資源の活用効果も限られている。 	<p>M-C型アーキテクチャーの戦略ロジック</p> <ol style="list-style-type: none"> 1. 製品全体技術の微妙な差別化と要素技術の差別化優位への追求。 2. 少数企業間でのコミュニケーション・ルールが明確化され、各モジュールにおけるカプセル的学習の効果が期待できる。 3. 製品の設計 (モデルチェンジ)・生産にかかわる組織間の技術的な制約が解放されることによって、ビジネス・プロセスにおける製造の事業活動の内部分断は可能であるが、外部資源の活用効果も限られている。
<p>I-O型アーキテクチャーの戦略ロジック</p> <ol style="list-style-type: none"> 1. 製品技術の最適化・事実上の標準化への追求。 2. 製品システムの特性によって、ビジネス・プロセスにおける製造の事業活動間関係の外部分断は困難なので、多数企業間での頻繁なコミュニケーションはかえってコストがかかる。 3. IT技術などをうまく使った上で多数企業間の頻繁なコミュニケーションとが生み出す膨大な調整コストを抑えるかぎり、外部資源活用とイノベーションの効果は期待できる。 	<p>M-O型アーキテクチャーの戦略ロジック</p> <ol style="list-style-type: none"> 1. 製造上のコスト優位性とコア要素技術の差別化優位への追求。 2. 多数企業間でのコミュニケーション・ルールが明確化され、各モジュール・メーカーによるイノベーションの効果が期待できる。 3. 製品の設計 (モデルチェンジ)・生産にかかわる組織間の技術的な制約が解放されることによって、ビジネス・プロセスにおける製造の事業活動間関係の外部分断は可能であり、外部資源の活用効果も期待できる。

注: 事業活動間関係の内部分断とは、企業グループ内部における製品ビジネスの事業活動に関する組織的関係の明確化である。それに対して、事業活動間関係の外部分断とは、企業グループの枠を越える製品ビジネスの事業活動に関する組織的関係の明確化である。

業界を構成する企業の数、企業のビジネスの競争戦略の策定、さらにその製品ビジネスにかかわる企業組織間関係の発展にも影響を及ぼす可能性があると考えられる。

上述の論理的な背景を踏まえた上で、アーキテクチャーの理論的視点を通して、製品技術体系のパターンを分析する必要がある。そのため、この分析に基づいて製品のアーキテクチャーの違いは、製品ビジネスの競争戦略策定に影響を与える可能性があるのかを解明するを、本稿の目的とする。本稿では、まず製品のアーキテクチャーの解析に関する先行理論の視点を考察したうえで、製品技術体系のパターンを導いてみる。次に、アーキテクチャー視点から述べられる製品技術体系のパターンが示している戦略的なロジックを議論してみる。最後にそのような技術戦略的なロジックに沿って製品ビジネスに潜んでいるもう一つの戦略的な意味を明らかにしたい。

2. アーキテクチャーの視点による製品技術体系の分析

2.1 製品アーキテクチャーの定義

アーキテクチャー (Architecture) という言葉の意味は「建築、建築学、建築様式、構造」あるいは「コンピューターを機能面から見たときの構成方式であり、記憶装置のアドレス方式、入出力装置の構成方式などをさす」と定義されており (三省堂『大辞林 第二版』[1995])、もともと建築業界あるいはコンピューター業界で使用されている用語である。近年以来、アーキテクチャーという言葉も製品技術管理、ビジネス・企業のマネジメント、および産業政策分析などの分野にも応用されつつある。

アーキテクチャーとは「構成要素の相互依存

関係のパターンで記述されるシステムの性質」であり (Karl Ulrich [1995], 青島 [1998])、いわゆるシステムの内的視点で、空間を持つ物質を分析する概念でもある。それはビジネスにかかわる様々なシステム (例えば、製品システム、取引システム、生産システム、組織システム、物流システムなど) のデザイン・マネジメントを考える際に役に立つ概念である (武石 [2000])。そして、それぞれの研究分野で提唱されている「インターフェースの標準化」、「タスクの分割」、「生産のモジュール化」、「プラットフォーム」、「カプセル化」および「相互依存性」など、様々な周知の概念を「アーキテクチャー」という言葉と密接に関連する構成的概念を用いて分析を進めていけば、「いままで見えなかったことが見えたり、曖昧だったことが明瞭になったりする」という効果が期待されている (藤本・武石・青島など [2001])。

そのようなアーキテクチャーの概念を製品技術体系に転写して考えれば、「製品アーキテクチャー (Product Architecture)」という考えが浮かび上がる。製品全体のアーキテクチャーは「製品を構成するコンポーネント²をいかに一緒にうまく働かせるか」という問題を解決するレイアウトである (Rebecca M.Henderson and KimB.Clark [1990])。例えば、ファン製品のアーキテクチャーとそれを構成する数多いコンポーネントが一つルームのエアを流れ動かすシステムを作り上げている。そして、Karl Ulrich [1995] によれば、製品のアーキテクチャーとは「ある製品の機能をどのように物理的なコン

2 コンポーネント (Component) —製品における相互に区別できる物理的な構成部分であり、核心的なデザインの概念と完全な機能を備えており (Clark [1985], Rebecca M.Henderson and KimB.Clark [1990], Karl Ulrich [1995])、基礎部品の集合体あるいは機能モジュールとも言える。

ポーネントに分配するかという計画 (Scheme)」と定義されている。そうすると、この概念を「製品構成要素間の相互依存性 (Dependence) という視点から見た製品システムの性質である」と考えることもできる。また、製品の構成要素間の相互依存性は、製品が市場ニーズを満たすべき諸機能が各物理的構成要素に配分されるパターンと構成要素間のインターフェースのルール化の度合いによって決定されている (Karl Ulrich [1995], 青島 [1998], 武石 [2000])。一般的に、製品・工程の「アーキテクチャー」とは、「製品の要求機能をどのように展開し、製品をどのようにして製品を構成部品や工程に分割し、それに製品機能を配分し、それによって必要となる部品・工程のインターフェースをいかに設計・調整するか」という問題に関する設計構想である (Fine [1998], 藤本 [1998], Baldwin and Clark [2000])。

より具体的に考えれば、製品のアーキテクチャーは「製品における機能要素の配置 (The arrangement of functional elements)」、 「機能要素から物質的な構成コンポーネントへの写像 (Mapping)」、 「物理的なコンポーネント同士が相互に作用している (Interacting) 中でインターフェース (Interface) の規定仕様 (Specification)」から成り立っている (Karl Ulrich [1995])。すなわち、製品システムの技術特性を分類する概念の一つであり、製品全体の機能を各コンポーネントにどのように配分するか、コンポーネント間のインターフェースをどのようにデザインするかという問題にかかわる設計思想でもある (近能 [2000])。

また、Karl Ulrich [2000] は、製品のアーキテクチャーが①製品システムの概略的な構成図式を描くこと、②相互関係のある構成要素を固まり (Chunk) にすること、③概略的な幾何様式のレイアウトをつくること、④構成要素間の基本的なかつ付属的な相互作用を識別することなど、四つの意志決定のプロセスによって確立されると主張している。そして、それぞれの意志決定が企業内外の関連情報に基づいて分析的に行われた上で製品のアーキテクチャーを確立しようとするれば、戦略的要素は不可欠となっている。

一方、製品アーキテクチャーの意味を議論する際に、インターフェースという概念をも明確化しなければならない。一つの製品システムは、相互作用あるいは影響し合っているコンポーネント同士を物質的なインターフェースで連結することによって成り立つだろう。そのインターフェースは、シャフトのギアのように二つのコンポーネント (部品) を繋ぐ幾何学様式の接続部品であれば、リモコンとテレビ本体との間の赤外線のような非接触の連結光線でもある。しかも、インターフェースはスタンダード・プロトコル (Standard Protocol) とも指しており、例えばパソコン部品の SCSI 接続規格、自動車の tyre/rim 標準規格など業界あるいは企業間の共通規定である (Karl Ulrich [1995])。また、インターフェースとは、コンポーネント同士が物質的に接続している部分や、コンポーネント間でエネルギーや情報を交換する「継ぎ手」部分を意味している (近能 [2000], 藤本 [2001])。例えば、錠と鍵を繋ぐ鍵の穴、ADSL 通信回線における接続のプロトコル、パソコン内部のマザーボードにおけるモニター・カードや DRAM カードを差し込むときのスロットなど、各構成要素が持っている機能情報を転換・

3 本稿では、製品の構成要素 (Constitutional elements) が機能要素 (Functional elements) と構成要素 (Physical components) を含む整合的な意味であると考える。

転送する役割を担っている。したがって、インターフェースは人体に対する神経あるいは血管のように製品システムをうまく作動させるために不可欠な存在となっている。

以上の先行研究に基づいてアーキテクチャーの定義をまとめると、「製品技術体系の転換を説明するアーキテクチャーは、製品システムの要求された意図的機能をうまく実現するために必要とされるシステムチックな構図である」と定義できる。すなわち、「製品システムの全体機能のアレンジメント」「機能要素から構造要素(コンポーネント)への写像関係」、「構造要素(コンポーネント)を結びつけるインターフェースの形態」など製品システムの構成要素の関係を記述する戦略的な設計思想でもある。そして、その概念は主に製品のバリューチェーンにおける製品設計(特に製品のコンセプト、製品システム全体の設計、詳細設計など)の段階に应用されている。しかも、それは、川上に位置づけられている商品企画や新製品技術の開発、川中における工程設計・試作、生産・物流、および川下に置かれる販売サービスなど製品のビジネス活動とは相互作用している。

2.2 アーキテクチャー視点による製品技術体系の分類

システムの性質を持っているアーキテクチャーはモジュラー化(Modular)／統合化(Integral)という動き、しかも、オープン化(Open)／クローズド化(Closed)という視点から把握することができる(青島、武石[2001])。そして、製品・工程「アーキテクチャー」の代表的な分け方は、「モジュラー型」と「インテグラル型」の区別、また「オープン(開)型」と「クローズ(閉)型」の区別があると考えられる(Karl Ulrich [1995], Fine [1998], 藤本

[1998], Baldwin and Clark [2000])。それによって、製品アーキテクチャーのタイポロジーを論述するに際しては、「モジュラー化」「インテグラル化」「オープン化」「クローズ化」など理論的視点から始めなければならないだろう。そのようなタイポロジーを論じるのは、製品のライフサイクルに沿って製品技術体系が混沌の状態から整列の状態にかけて揺れ動いている実態の解明について役に立つかもしれない。

(1) 製品アーキテクチャーのモジュラー化 (Modular)／統合化 (Integral)

複雑なシステムを処理する方法として、腕時計の有名な寓話を取り上げられている(Simon [1969])。その寓話の中では、腕時計の完成が大きなサブアセンブリとしてのモジュールを最終に組み合わせることによって成り立っているところにモジュラー化の概念が潜んでいると考えられる。

「モジュール(Module)」とは、半自律的なサブシステムであり、他のサブシステム同士と一定のルールに基づいてお互いに繋がることによってより複雑なシステム、またはプロセスを構成するものである。電子製品の場合には、パッケージ化されたエレクトロニクス部品の機能的な集合体で、他の同様な集合体とともに用いられている。そして、一つの複雑なシステムまたプロセスを一定のルールに基づいて独立でかつ半自律的なサブシステムに分解することを「モジュール化」と呼び、あるルールのもとで独立的に設計されるサブシステムを連結して複雑なシステムまたプロセスを構成することを

4 William J. Abernathy, and James M. Utterback [1978], William J. Abernathy, Kim B. Clark and Alan Kantrow [1983], James M. Utterback [1994], 伊丹敬之・加護野忠男 [1993] を参照。

「モジュラリティ (Modularity)」と呼んでいる (青木[2002])。そして、本稿では、「モジュール (Module)」を「特定な機能あるいはシステムの部分的な機能・構造のパフォーマンスに対して自己完結で行うことができるが、システム全体機能のパフォーマンスに自己完結で対応できない副次的なシステム」と定義し、しかも「モジュラー化」は「モジュール化」と「モジュラリティ」を含んでいる総合的な概念として用いる。

一方、モジュラー化は、複雑性も持つ製品システムを効率的に調整する戦略である。デザイナー達は、製品システムの情報を形式的なデザイン・ルール⁵ (visible design rules) と暗黙的なデザイン・パラメーター⁶ (hidden design parameters) など二つの種類に分割するという手段を通じて製品の「モジュラリティ」を実現している。しかも、もしこの情報の分け方は明確的で、曖昧さがなく完全であれば、「モジュラリティ」が利益的な結果をもたらすようになっていく (Baldwin and Clark [1997])。要するに、製品システムの根幹部分に関する情報と枝葉の部分にかかわる情報がはっきり区別できれば、製品を設計するときの複雑性を処理できる程度も上昇しつつあることによって、時間、

カネおよびヒトの節約などコストの削減という便益も享受できる。

なお、システム複雑性の源泉は①構成要素の数＝要素間の関係の多さ、②各構成要素間の相互依存性＝構成要素間関係の強さなど二つの種類に分けることができる。そして、この二つの源泉を持つシステムの複雑性をどのように削減するかという問題はあらゆるシステム設計にとっての重要な課題となる。それに対して、システムにおける階層化・インターフェースの集約化は、要素間の関係の数を削減することを通してシステム複雑性を低下させようとする戦略であり、インターフェースのルール化・固定化は、構造要素間の相互依存性を弱めることを通じてシステム複雑性を低下させようとする戦略である。しかも、これらの二つの戦略を併用するものは「モジュラー化」戦略である (青島・武石 [2001])。つまり、システムにおける構造要素間の関係を繋いでいる数多い不規則のインターフェースを整理し簡略化するのはシステムの「モジュラー化」へ進行する要件となっている。

したがって、本稿では、焦点を絞りたい製品システムの「モジュラー化」は、製品システムにおける機能要素と構造要素との間の情報の分割、製品を構成している構造要素 (コンポーネント、部品の集合体、機能モジュール) およびそのコンポーネント同士を連結するインターフェースとかかわる。それゆえに、製品システムのモジュラー化は、製品のアーキテクチャーの一つパターンであると考えられる。そして、製品システムを自律的な機能モジュールに分け、しかも、そのモジュール間の相互依存関係を構築するインターフェースをルール化・階層化していく過程は、「製品アーキテクチャーのモジュラー化」と呼ばれることができる。

5 形式的なデザイン・ルール (visible design rules or visible information) とは「製品における各モジュール間の連結に関するデザイン・ルールに強く影響している意志決定」であり、そしてこの形式的なデザイン・ルールは製品システム全体のデザイン・ルールにかかわっているプロセスとコミュニケーションが進んでいく初期に立てられるべきであると考えられる (Baldwin and Clark [1997])。

6 暗黙的なデザイン・パラメーター (hidden design parameters or hidden information) とは「製品における各モジュール間の連結に関するデザイン・ルールに影響を及んでいない意志決定」であり、このような暗黙的なデザイン・パラメーターは製品における各モジュールの独自のデザイン・ルールにかかわっているプロセスとコミュニケーションが進んだあとで立てられかつ随時変更することができる性質を持っている (Baldwin and Clark [1997])。

一般的に、モジュラー型製品のアーキテクチャーは主に二つの特徴をもっており、しかも、そのような特徴は製品システムにおける機能・構造のヒエラルキーの一次レイヤーで強く現れている。第一に、製品システムにおいて機能要素から構造要素への写像関係は単純で、一対一 (one to one mapping) であり、それぞれの構造要素は機能モジュールのように非常に独立性の高い機能が与えられている。第二に、二つの構造要素を連結するインターフェースの数と形が唯一でかつ単純の (De Coupled) であり、つまり構造要素間のインターフェースが事前にルール化・階層化されている。このタイプの製品設計では、独立性が高い構造要素を寄せ集めて組み合わせるだけで複雑な製品システムを構成することが可能なので、製品システムの技術体系は著しくシンプルになる⁷。しかも事前に規定されたインターフェースの固定的なルールさえ守れば、構造要素自体は自由に変更できる状況で高い設計自由度が確保されるという「組み合わせ」の概念が強く示されている (Ulrich [1995], 青島 [1998], 藤本 [1998] [2001], 近能 [2000])。すなわち、製品システムの要求される全体機能パフォーマンスの達成の効率化は、製品のアーキテクチャーがモジュラー化にシフトしようとする目的になる。

モジュラー型製品システムのアーキテクチャーに対しては、統合型 (インテグラル) の製品アーキテクチャーの特徴は対照的である。第一は、製品システムにおいて機能要素から構造要素への写像関係は煩雑 (多対多, 一対多, 多対一) であり、しかもそれぞれの構造要素は不完全機能のモジュールのように相互依存性の

高い現象が明らかに見える。第二は、二つの構造要素を連結するインターフェースの数と形が複数でかつ煩雑 (coupled) であり、つまりコンポーネント間のインターフェースを固定的なルールで規定することができない。

このタイプの製品設計では、一つのコンポーネントの構造を変更すると他のコンポーネントに大きな影響を及ぼすので、製品の技術体系は著しく煩雑になる。したがって、各コンポーネントの設計者が互いに設計に関するコミュニケーションと調整を頻繁に行っている複数のコンポーネントに跨る「擦りあわせ」という設計観は大切となっており (Ulrich [1995], 青島 [1998], 藤本 [1998] [2001], 近能 [2000]), すなわち、製品システム・パフォーマンスの最適化は、製品アーキテクチャーが統合化にシフトしようとする目的になる。製品システムの複雑性によるモジュラー型と統合型アーキテクチャーとの違いは図 1 に示したとおりである。

(2) 製品アーキテクチャーのオープン化 (Open) / クローズド化 (Closed)

オープン化 (Open) / クローズド化 (Closed) とは「システムの構築, 改善および維持のために必要と認識されている情報が社会的に共有・受容される範囲」にかかわっている概念である。この範囲が拡大していく動きは「オープン化 (Open)」と見られており、それに対してこの範囲が縮んでいく動きは「クローズド化 (Closed)」であると称することができる (青島, 武石 [2001])。そして、オープンなシステムは「外部環境からの資源と情報との処理に基づいて自己の生存を維持できるシステム (Self-Maintenance)」と定義されている (Richard Scott [1998])。つまり、オープンなシステムが外部環境との資源のやりとりに積極的に取り組

7 製品システムがモジュラー化されると、製品システム自身と技術体系の複雑性が比較的にならなくなると考えることができる。

んでいるばかりでなく、このやりとりはオープンなシステムの生存能力にとってとても重要なファクターである (Walter Buckley [1967])。

したがって、製品システムの構築、改善および維持のための必要な情報のやりとりを公開的に行って、それを業界（製品を製造・生産する企業にかかわっている売り手、買い手および競争相手から成り立つ）あるいは社会全体（ユーザーまで）に拡散し共有させるのは、「製品システムのオープン化」であり、それに対して、「製品システムのクローズド化」とは、製品システムの構築、改善および維持のための必要な情報のやりとりを閉鎖的に行って該当製品を製造・生産する企業の内部だけで共有・受容される過程であると言えるだろう。

ところで、それをアーキテクチャーの概念に転用すれば、「製品アーキテクチャーのオープン化 (Open) / クローズド化 (Closed)」という言葉は導き出される。オープン・アーキテクチャー型の製品システムとは、本来に複雑な機能を持つ製品システムを、ある設計思想に基づいて独立性の高い単位（機能モジュール）に分解しそのモジュール間を社会的に共有されたオープンなインターフェースで連結することによって汎用性を持たせ、多様な主体が発信する情報を組み合わせて付加価値の向上を図る製品システムを指す (国領 [1999])。その定義によれば、オープン・アーキテクチャー型の製品は必ずモジュラー・アーキテクチャー型の製品でなければならないと示唆している。ただし、製品システムのモジュラー化が製品システムの構築、改善および維持に参加する時に必要な情報を大幅に削減することができるという意味で、製品のモジュラー化が製品のオープン化を促しているにもかかわらず、必ずしも製品がモジュラー化されないとオープン化ができないというわけで

はない。つまり、業界あるいは社会における人々がある製品システムに関する情報に強い関心を持ち、その製品構造を理解する能力が受容側が高ければ（かつメーカー側も製品システムの情報公開する意欲があることを前提として）、製品システムのモジュラー化戦略に頼らなくても製品のオープン化が進んでいく可能性がある（青島、武石 [2001]）。それによって、製品アーキテクチャーのオープン化 / クローズド化は製品システムにおける機能要素と構造要素との写像対応の関係、およびインターフェースのルール化・階層化と絶対的にかかわっているわけではないと考えられる。

したがって、製品アーキテクチャーのオープン化とは、「製品システムの構造要素間のインターフェースに関する情報（設計のソース・データ）が業界あるいは社会全体に公開され受容されていく過程」であり、それに対して、製品アーキテクチャーのクローズド化とは、「製品システム構造要素間のインターフェースに関する情報が該当製品を製造・生産する企業内部、あるいは連携関係のある少数企業間だけで共有・受容される過程」であると定義できる。つまり、製品アーキテクチャーのオープン化 / クローズド化は製品システムにおけるインターフェースの公開 / 閉鎖という要因にかかわっていると考えることができる。例えば、デルのデスクトップ型のパソコン本体におけるアダプター・スロット、アダプター・カードとデスク・ドライブとのデータ伝達回線などいろいろなインターフェースに関する設計上の規格情報が業界（パソコンの組立メーカーとそのライバル同士、部品メーカー、電器販売店）、さらにユーザー（White Box）に共有・受容されていることによって、ディスク・タッグ・パソコンの製品アーキテクチャーはオープン化に極端的に偏ってい

る。それに対して、いまの時点でキャノンのインクジェット型のプリンターにおけるカートリッジをはめるスロットというインターフェースに関する製品設計の規格情報が業界に共有・受容されていないため⁸、インクジェット型のプリンターの製品アーキテクチャーはクローズド化にやや偏っていると考えられる。

2.3 アーキテクチャーの視点における四つの製品技術体系のパターン

上述の文献のオーバービューによると、製品のアーキテクチャーの定義における「機能要素から構造要素（コンポーネントあるいは部品）への写像あるいは構図の関係」，「物質的な構造要素（コンポーネントあるいは部品）の間で規定されるインターフェースの形態」という製品設計段階で決められた製品システムの構成要素の関係性に基づいて製品システムの複雑性および技術情報の拡散と理解など理論的な視点をまとめて考えれば、複雑な製品技術体系のアーキテクチャーを複合的に類型化することができる。

図3のように、2（インテグラル化／モジュラー化）×2（クローズド化／オープン化）という二つの次元で構築されたマトリックスから、四つの製品技術体系のパターンを導き出すことができる。「インテグラル／クローズド (I-C 型)」には、自動車、オートバイおよび鑑賞向けの芸術消費財などが比較的にわけられることができる。「モジュラー／クローズド化 (M-C 型)」には、レゴ、特殊作業機能向けの工作機械、メインフレーム（「IBM System/360」IBM 互換機が現れた前の形態）および一部の「マルチメディア

ア／軽薄短小」型のハイエンド電子消費財など、そして「モジュラー／オープン化 (M-O 型)」には、自転車やローエンド電子消費財（デスクトップ PC, GSM 携帯電話本体など）、インターネット向けのパッケージソフトなど、業界標準機能向けの製品が相対的に当てはまる（藤本[2001]，作者が加筆修正）。

また、Linux（リナックス）については、インターフェースだけでなくソフトウェアの内部構造であるソース・コードが公開され、世界中でその技術情報を理解・受容できるプログラマーがインターネットを通じて開発に参加するようになっている。しかも、製品システムの技術構造には、当時に流行っていたモジュラー化されたマイクロ・カーネルでなく、従来の単層カーネルというインテグラルな構造を採用しており、無駄の無い最適な設計の目標を求めている（小山、竹田[2001]）。したがって、「インテグラル／オープン化 (I-O 型)」には、Linux（リナックス）という製品が比較的に当てはまるかもしれない。

ただし、厳密的に言えば、ある製品技術体系が全体として「インテグラル型」であるか「モジュラー型」であるかという問い自体が正確でないことに注意していただきたい。その製品アーキテクチャーがモジュラー的であるかインテグラル的であるかという認識は、部品のレベルによって異なるという考え方は合理的である。つまり、ある「モジュラー型」の製品とは、正確に言えば、少なくとも製品機能・製品構造のヒエラルキーの比較的な上位の一次階層（ある集成度レベル）で強いモジュラー化の特徴が強く現れる製品システムを指す（藤本 [2001]）。例えば、周知的にデスクトップのパソコンは、一次機能要素と一次構造要素との写像関係が極めて単純でかつ一次構造要素の間のインター

8 キャノンのカートリッジは HP あるいは ESPON のインクジェット型のプリンター製品で使えないという現状で見れば分かる。

フェースが階層化・標準化されており「モジュラー／オープン化 (M-O 型)」の製品システムとも言えるが、製品システムにおけるコア・モジュールであるインテル製の CPU は、ほかの機能モジュールにつながる外部のインターフェースがオープン的であるにもかかわらず、その中身は実に比較的「インテグラル／クローズド (I-C 型)」製品に偏っていると考えられる。

事実上、多くのケース・スタディーから観察されれば、企業にとって最適な製品アーキテクチャーの選択という課題は、完全なモジュラー化あるいはインテグラル化の製品アーキテクチャーにこだわるよりも、むしろ製品システムにおいてどのような機能要素がモジュラー化あるいはインテグラル化へ進むべきかという範疇で議論されるべきであると考えられる (Ulrich [1995])。すなわち、製品アーキテクチャーという概念は絶対的な固定の考え方ではなく、異なる製品システムが比較された上で議論すればはじめて有意義となる。そして、企業は単なるこの製品がモジュラー型製品かあるいはインテグラル型製品に分類する作業を行うだけでなく、市場のニーズと業界技術の変化など経営環境の要因を常に製品システムの開発・設計の考慮に入れて製品アーキテクチャーの方向を戦略的に決めていくという視点も、製品アーキテクチャーという概念がもたらす重要なインプリケーションである。

3. アーキテクチャーから見た製品技術体系の戦略的なロジック

上述のように構成要素間の関係性および技術情報の拡散・受容のありさまを戦略的に記述するアーキテクチャー視点によって、製品技術体系を四つのパターンに類型化することができ

る。それぞれのパターンが示している製品技術の内的な特徴は製品ビジネスの推進にどのような戦略的なロジックをもたらすのかという課題は、「アーキテクチャー視点」という理論自体に問われなければならない。次は、製品開発のパターン、製品のパフォーマンス及び製品の多様性と部品の共通性など製品技術管理の側面にそって、製品アーキテクチャーの四つのパターンは、製品技術の競争優位性、企業組織間のコミュニケーション、ビジネス・プロセスの分断および経営資源の外部活用などビジネス戦略に関連する視点に、どのようなヒントを及ぼすのかを議論してみる。

3.1 アーキテクチャー視点による製品開発のパターンの再考

製品のアーキテクチャーがインテグラル化かモジュラー化かは、企業内部あるいは企業間の調整メカニズムという側面で、製品開発活動のパターンにも影響を及ぼしている (Ulrich [1995], 韓・近能 [2001], Chesbrough, Henry W. and Ken Kusunoki [2001], 柴田・玄場・児玉 [2002])。それに加えて、製品アーキテクチャーはオープン化あるいはクローズド化されると、製品開発パターンもある程度に転換してくるかもしれない。

まず、I-C 型アーキテクチャーを有する製品を考えると、製品システムにおける構成要素間の関係性が複雑で⁹、インターフェースのルールが事後の調整に頼っており、しかも構造要素間関係の変化インターフェースに関する情報が閉鎖されるため、製品技術の開発は必ず企業内部のデザイン・スタッフ、生産部門、マーケティング部品および系列下請けのサプライヤー間の

9 本稿での図1と図2の説明を参照。

緊密なコミュニケーションを通じて行われなければならない。また、実際に量産・試作の段階に入り、一つの工程・作業で部品設計上の品質的な欠陥、もしくは顧客の特注に応える若干箇所での部品設計変更など部分的な製品設計上の問題が発生すると、製品システム全体の機能・構造的な仕組みに影響を及ぼす可能性がある。それゆえに、問題を解決する要件として、企業内および少数企業間の設計・生産に関する事後的な調整メカニズムが不可欠とされる。そのような組織的取引内部において、緊密な情報のやりとりや学習によるアーキテクチャな知識 (Architectural Knowledge)¹⁰の創造によって製品全体の急激なイノベーション (Radical Innovation) あるいはアーキテクチャなイノベーション (Architectural Innovation)¹¹も起こしやすくなる。ただし、そのようなテクニカル・シードにかかわる情報の交換は、少数企業間に絞られているので、イノベーションの効果も制約されている。

また、製品システムにおけるインターフェースの情報が公開・受容されつつあり、アーキテクチャーがI-O化されることが可能である。その場合、IT技術を活用し情報のやりとりのコストを最大限に抑えたうえで、製品システムの開発は企業外部にサプライヤー、ライバルおよびユーザーなど業界の多数構成分子と緊密なコミュニケーションを行うことができる。なお、この複雑なインターフェースのルールを理解するために、多数の業界構成分子が時間・金をかけなければならない。それゆえに、自分が払ったコストを還元するために、この製品のデファクト・スタンダード (De facto standard) の進

展に関連する技術的なイノベーションの実現にも助力しなければならないだろう。

一方、アーキテクチャーがM-C化された製品システムの場合には、製品システムにおける構成要素の関係性が明確化され、インターフェースも事前に簡略・規格化され、しかも閉鎖されるため、製品の開発活動を、いくつかのデザイン・タスクに分けて、企業内部、系列下請けのサプライヤーおよび少数の技術連携関係のある企業間で同時・並行的に進行することができる。多品種少量の市場ニーズを満たすための開発期間の短縮化によって、企業内部あるいはサプライヤー間での製品開発活動の分業化は必然的なトレンドとなる。製品全体の急激なイノベーションは難しくなるが、コンポーネント知識 (Component Knowledge)¹²の創造が必要とされる各機能モジュールの内部のイノベーションが効率的に進めることが可能である。また、インターフェースに関連する技術情報が公開・受容されていないため、経営資源の獲得は企業自身あるいは少数関連系列企業グループに制限されている。そのように有限な経営資源によって各機能モジュールの内部における急激なイノベーション、いわゆるモジュラー・イノベーション (Modular Innovation)¹³の効率化は、一定の程度に留まっている。

また、製品システムにおけるインターフェースの情報が公開・受容されつつあり、アーキテクチャーがM-O化されると、製品システムにおける構成要素の関係性が明確化されただけでなく、インターフェースも簡略・規格化され、さらに外部標準化されることになる。それに基づいて、製品の開発活動はいくつかのデザイン・タスクに分けられ、さらに世界に散在して

10 Henderson and Clark [1990] p.11, を参照。

11 Henderson and Clark [1990], Ulrich [1995] p.437 を参照。

12 Henderson and Clark [1990] p.11, を参照。

13 同上。

いる多数の短期的取引志向の企業同士間でも、同時・並行的に進行することが可能である。また、実際に量産・試作の段階に入り、一つの部品に関する機能・構造的な設計変更が発生するとしても、事前に規定されたインターフェースのルールに沿って、特定の担当設計部門あるいは外部の専門モジュール・メーカーに任せれば、その問題を迅速に解決できる。したがって、その製品の開発活動の外部分業化も不可避となる。一方、インターフェースのルールが外部に公開され、多数の業界構成分子が多く時間をかけなくても、この単純なインターフェースのルールを理解できることによって、この製品の製造分野に快適に参入し、この製品のデファクト・スタンダードをいち早く促進することができる。そして、業界における数多い企業同士が持つ膨大な外部経営資源の補完によって、各モジュラー・イノベーションと漸進的なイノベーション (Incremental Innovation) を最大限に実現することが可能である。

3.2 製品のパフォーマンスの達成

製品のパフォーマンスとは、「製品システムを構成する構造要素が機能要素をうまく実行する程度である」と定義できる。そして、製品のグローバル・パフォーマンス特徴¹⁴はインテグラルされたアーキテクチャーによって最適に実現

される一方、製品のローカル・パフォーマンスの特徴はモジュラー化されたアーキテクチャーによって効率的に達成される (Ulrich [1995], 作者が加筆)。そして、モジュラー・アーキテクチャーを持つ製品の構成要素に比べて、インテグラル化された製品システム全体の性能向上に対する構成要素それぞれの独立した貢献度は低くなる (青島 [1998])。言い換えれば、製品のグローバル・パフォーマンス (以下は製品システム・パフォーマンス) を向上させるために、それぞれの構造要素が機能要素をうまく実行する程度の間での調整が重要である。そして、製品のローカル・パフォーマンス (以下はコンポーネント・パフォーマンス) の強化は、それぞれの構造要素が、それぞれ実像関係のある機能要素をうまく実行する程度をどのように上げるのかにかかわっている。また、インテグラ型の製品システムの場合、機能構造の実像関係が煩雑なので、コンポーネント・パフォーマンスの特徴は現れにくい。それに対して、製品システム・パフォーマンスの特徴はモジュラー型製品において見えにくいと考えられる。

アーキテクチャーがI-C化された製品システムの場合には、構成要素間の関係性が組織的取引内部において頻繁に調整されているため、「機能シェアリング¹⁵ (Function sharing)」などのデザイン・アイデアを活用し製品を軽薄短小化するだけでなく、製品システム・パフォーマンスの最適化を実現することもできる。ただし、インターフェースのルールに関する情報が外部に公開・受容されていないため、企業自身ある

14 製品のグローバル・パフォーマンスの特徴 (global performance characteristics) は、製品システムを構成する大部の分のコンポーネント同士から成り立つ物理的特性にかかわっており、とくに製品のサイズ、形、質量および材料の特性に大きく関与している。それに対して、製品のローカル・パフォーマンス特徴 (local performance characteristics) は製品システムにおける特定のコンポーネントが現れた局所的な物理的特性に関わっている。(Ulrich [1995] p.432)。言い換えれば、製品のグローバル・パフォーマンスは製品システム全体のパフォーマンスに偏り、製品のローカル・パフォーマンスはコンポーネントのパフォーマンスにかかわっていると考える。

15 機能のシェアリング (Function Sharing) とは「一つのコンポーネントが同時に複数の機能を実行するという一対多の写像関係を通じて、コンポーネント同士の冗長的な物理特性を削除する設計戦略」と定義されており、それを利用して製品の質量とサイズを最小限に抑えることができる (Ulrich [1995])。

いは少数関連系列企業グループが持つ有限的な経営資源によってコンポーネント・パフォーマンスの向上は制約されるにもかかわらず、ライバル企業の類似製品に比べて製品システム・パフォーマンスの差別化を一定の程度に保つことができる。したがって、アーキテクチャーのI-C化というアプローチを通じて個別顧客適応の手の込んだ製品システムの設計（機能・構造の仕組み）と性能が相対的に実現されることになる。

また、製品システムにおけるインターフェースの情報が公開・受容されつつあることになれば、アーキテクチャーがI-O化される可能性がある。その場合、製品システムにおける機能・構造間のグローバル・パフォーマンスが最適的に実現されるだけでなく、インターネットなどIT技術を利用して技術情報のやりとりに関するコストを抑えられれば、業界全体の構成分子（買い手、既存企業、売り手およびライバル）の協力によって、製品システム・パフォーマンスあるいはコンポーネント・パフォーマンスは共に大幅に上昇していくことになる。しかも、グローバルな市場取引において膨大な資源を集めて顧客適応の手の込んだ製品設計（システムとコンポーネントとの両方）と性能の最適化を最大限に実現することが可能である。なお、その製品システムのインターフェースの技術情報あるいは知識・ノウハウが業界の多数構成分子に共有されればされるほど、製品技術のデファクト・スタンダードも徐々に形成され、業界における製品システム・パフォーマンスの格差あるいは差別化は無意味的になるだろう。

一方、アーキテクチャーがM-C化された製品システムの場合には、構成要素間の関係性が明確され、各要素技術の開発は組織的取引内部においてそれぞれ平行・独立的に進行されるため、コンポーネント・パフォーマンスが比較的に促

進されている。そして、機能要素から構造要素への写像関係は単純（一対一）であれば、製品の軽薄短小を実現するための機能シェアリングを阻害する可能性があり(Ulrich[1995])。そして、インターフェースが集約化・ルール化されることになると、そのような汎用的なインターフェースに対応するために、製品システムにおける各構造要素の追加的あるいは付加的な硬直的スペース（冗長性・リダダンシー）がかえって必要となり(Simon[1969]pp.220-221, Ulrich [1995], 国領 [1999], 青島・武石 [2001] p.43, 安室 [2003] p.67), 各構造要素が示すコンポーネント・パフォーマンス間の不均衡性も発生しやすい。つまり、製品システム・パフォーマンスの最適化を実現することが難しくなる。また、インターフェースのルールが外部に公開されていないために、業界における製品システム・パフォーマンスあるいはコンポーネント・パフォーマンスの差別化を一定の程度に保つことができる。

M-O型アーキテクチャーの製品システムは、M-C型製品システムに比べて、コンポーネント・パフォーマンスが専門モジュール・メーカーによって強く促進されている。そして、インターフェースの業界標準化によって、冗長性・リダダンシー問題の軽減にも多大な企業間調整のコストをかけなければならない。なお、機能要素から構造要素への写像関係は単純（一対一）なので、機能シェアリングを阻害する可能性がある。また、インターフェースのルールが外部に公開・受容されているため、製品システム・パフォーマンスは、外部のモジュール・メーカーから調達するコア・コンポーネントのパフォーマンスに依存する可能性がある。したがって、標準化のモジュールを組み合わせることを通じて製品システムの多様化と部品の共通化、外部

資源の活用による製造コストの削減とコア・コンポーネント技術への「集中と選択」など戦略的なロジックは重要となる。

3.3 製品の多様性と部品の共通性との達成

1990年代以来、様々な産業において市場激動性 (Turbulence) と製品の多様性 (Variety) への需要がともに増大している。そのような経営環境の変化に対応するために、製品とサービスのマス・カスタム化へのシフトは重要な課題となる (Pine [1993])。

したがって、製品の多様化を実現するマス・カスタマイゼーションは多くの企業にとっていまや取りまなければならない戦略的な指針となる。

製品の多様性 (Product Variety) は、「一つの生産システムが市場に提供する様々な製品間の差異性」によって決められる (Ulrich [1995])。そして、そのような製品の多様性を経済的に達成するために、製品間のコンポーネントあるいは部品ユニットの共通性 (Commonality) を高める必要がある。しかし、その共通性を高めると、製品自体の差別化 (Distinctiveness) を多かれ少なかれ犠牲することになり、製品シリーズの多様化の度合いを下げていく。したがって、製品の多様性と部品の共通性との間に、トレードオフが存在している。また、Robertson and Ulrich (1998) は、製品アーキテクチャーが製品の差別化と部品の共通性との関係性の変化に影響を与えていると示唆している。次は、Ulrich (1995) と Robertson and Ulrich (1998) の論文概念に沿って、アーキテクチャーの理論的視点から導く四つの製品技術体系のパターンが製品多様性の実現についてどのような技術的戦略ロジックをもっているのかを議論してみる。

アーキテクチャーが I-C 化された製品システムでは、製品の多様性を実現するために、部品の加工、コンポーネントおよび最終組立など一連の製造工程の柔軟性¹⁶を高めるだけでなく、しかも、製品の製造は必ずこの製品の製造に携わる企業内部あるいは関連少数部品メーカーとの間の緊密なコミュニケーションを通じて行われなければならない。しかし、製品システムの複雑性によって、製品の差別化と部品の共通性とのトレードオフを解消するのは困難である。それゆえに、制限的な製品の多様性と製品技術の一点豪華的な差別化を追求した方が有利であると考えられる。

I-O 型アーキテクチャーの製品システムの場合、製品の多様化の実現は、部品の加工、コンポーネントおよび最終組立など一連の製造工程の柔軟性を高めるほかに、IT 技術を利用することによって企業内部の従業員と外部のパートナーあるいはユーザー間での緊密なコミュニケーションを通じて実現するのは可能である。しかし、製品システムの高い複雑性によって、製品の差別化と部品の共通性との関係性をバランスするのは難しい。それに加えて、インターフェースに関する技術情報が業界で共有・受容されるので、製品システム・レベルの技術の差別化が無くなる可能性がある。それに基づいて、企業にとって製品仕様の単一性、と部品の共通性による製品の製造コストの削減を追求した方が戦略的に考えられる。

アーキテクチャーが M-C 化された製品システムの場合には、製品の多様化の実現は柔軟性が低い量産型の生産システムを通じても可能で

16 たとえば、段取りや機械設備のセッティング時間の短縮、仕掛かり品の削減、少量生産 (Small lots)、ツーリング・コストの削減およびレーザーや NC 制御による快速的な連続加工などが取り上げられている (Ulrich [1995], 作者が加筆)。

ある。そして、コンポーネント間の関係性が明確化されたので、様々な規格化されたコンポーネントあるいは機能モジュールをうまく組み合わせれば、製品の多様化を効率に達成することができる。Pine (1993) によれば、モジュラー・コンポーネントを作って、きわめて多様性に富んだ最終製品とサービスへと組み立てることは、マス・カスタム化を実現する最良な方法である。一方、製品システムの低い複雑性によって、製品の差別化と部品の共通性を両立することが可能である。それゆえに、経済的な製品の多様性の目標（低コストによる製品の多様性）と製品技術の微妙的な差別化を追求した方が有利的であると考えられる。

M-O 型アーキテクチャーの製品システムにおいては、製品の多様化は業界を跨る外部の標準量産型の生産システムを通じても実現されることができる。そして、コンポーネント間の関係性が明確化され、インターフェースに関する技術情報が業界で共有・受容されるので、業界に標準化される様々な規格化されたコンポーネントあるいは機能モジュールをうまく組み合わせれば、製品の多様化を迅速に達成することができる。ただし、製品システム上の技術の差別化がなくなることによって、製品の多様性の範囲は、業界における既存の標準的なコンポーネントあるいは機能モジュールの組み合わせに留まっている。それゆえに、企業にとって、コスト優位性による製品の多様性を追求した方が戦略的に考えられる。

4. おわり

上述の議論をまとめてみると、製品技術体系のありさまを設計するアーキテクチャー・アプローチは、製品技術に関するさまざまなビジネ

ス戦略の策定に影響を及ぼしている。つまり、四つの製品技術体系のアーキテクチャーが示している戦略的なロジックは、製品ビジネスの競争戦略の策定をある程度で支配しているとも言える。また、製品のアーキテクチャーが I-C 化から M-O 化へと変わるにつれて、企業が取り組む製品技術の競争優位の追求も、差別化からコスト・リーダシップへシフトするかもしれない。一方、製品開発のプロセスにおける頻繁に発生する設計の変更、企業の競争ドメインにかかわっている製品シリーズの充実性あるいは多様性、生産段階に関するコンポーネントあるいは部品規格の標準化方針、顧客のニーズにあてはまる製品のパフォーマンス、製品技術のイノベーションなど、製品技術にかかわるビジネスの競争戦略の意志決定の課題に対しては、アーキテクチャー視点による製品技術体系の分析は一定の程度の意味を持っていると思われる。すなわち、製品技術体系が異なることによって企業が取り組むべきビジネスの競争戦略も変わっていくという当たり前のロジックを、アーキテクチャー視点で吟味すれば、新しいインプリケーションを導くことができるかもしれない（図 4）。

参 考 文 献

- Abernathy, William, J., and James M. Utterback [1978], "Patterns of Industrial Innovation." *Technology Review*, Vol.80, No.7, June-July, pp40-47.
- Abernathy, William, J., Kim Clark and Alan Kantrow [1983], *Industrial Renaissance, Basic books* (望月嘉幸監訳 [1984] 『インダストリアル・ルネサンス』 TBS プリタニカ.)
- 安室憲一 [2003] 『中国企業の競争力—徹底検証—』日本経済新聞社。
- 青木昌彦 [2002] 「産業アーキテクチャーのモジュール化—理論的イントロダクション」青木昌彦・安藤晴彦 編著 [2002] 『モジュール化：新しい産業アーキテク

- チャーの本質』東洋経済新報社。
- 青島矢一[1998]「製品アーキテクチャーと製品開発知識の伝承」『ビジネスレビュー』Vol.46 No.1, pp46-60。
- 青島矢一・武石彰 [2001]「アーキテクチャーの考え方」藤本隆宏・武石彰・青島矢一編『ビジネス・アーキテクチャー論』有斐閣, pp.27-70。
- Baldwin, Carliss Y. and Kim B. Clark [1997], “Managing in an Age of Modularity,” *Harvard Business Review*, Sept./Oct, pp.84-93.
- 伊丹敬之・加護野忠男 [1993]『ゼミナール経営学入門』日本経済新聞社。
- Henderson Rebecca and Kim B. Clark [1990] “Architecture Innovation: The Reconfiguration of existing product technologies and the failure of established firms,” *Administrative Science Quarterly*, Vol.35, pp.9-30.
- James M. Utterback [1994], *Mastering The Dynamics Of Innovation, Harvard College*. (大津正和, 小川進 監訳 [1998]『イノベーション・ダイナミクス—事例から学ぶ技術戦略』)
- 小山裕司・竹田陽子 [2001]「ソフトウェアの開発技法と構造—コンピューターソフトウェア開発の特徴と課題—」藤本隆宏・武石彰・青島矢一編『ビジネス・アーキテクチャー論』有斐閣, pp161-171。
- 国領二郎 [1999]『オープン・アーキテクチャー戦略—ネットワーク時代の協働モデル—』ダイヤモンド社。
- 近能善範 [2000]「製品アーキテクチャー」高橋伸夫編著『超企業・組織論』有斐閣, pp.99-108。
- 韓美京・近能善範 [2001]「アーキテクチャー特性と製品開発パターン—自動車部品のケース」藤本隆宏・武石彰・青島矢一編『ビジネス・アーキテクチャー論』有斐閣, pp229-245。
- Michael E. Porter [1985], *Competitive Advantage: Creating and Sustaining Superior Performance*, N. Y.: The Free Press. (土岐坤・中辻万治・小野寺武夫など訳 [1985]『競争優位の戦略—いかに高業績を持続させるか—』ダイヤモンド社)。
- Pine II, B. J. [1993], *Mass Customization: The Frontier In Business Competition*, MA: Harvard Business School Press (江夏健一・坂野智昭 [1994]『マス・カスタマイゼーション革命』日本能率マネジメントセンター)。
- Simon, Harbert A [1969] “The Architecture of Complexity: Hierarchic Systems,” *The Science of Artifi-*
- cial*, Cambridge, MA: MIT Press, pp192-229.
- 下田博次 [1994]『王者 IBM・リストラへの挑戦』PHP 研究所。
- 新宅純二郎・許斐義信・柴田高 編著 [2000]『デファクト・スタンダードの本質』有斐閣。
- 柴田友厚・児玉文雄 [2001]「製品アーキテクチャーの進化—アーキテクチャーから見た技術進化」『一橋ビジネスレビュー』2001年 WIN, pp.180-196。
- 柴田友厚・玄場公規・児玉文雄 [2002]『製品アーキテクチャーの進化—システムの複雑性と分断による学習—』白桃書房。
- 高橋伸夫 [2000]「システムとしての組織・境界としての企業」高橋伸夫編著『超企業・組織論』有斐閣, pp. 223-249。
- 武石彰 [2000]「モジュラー化, オープン化」『一橋ビジネスレビュー』2000年 WIN, pp.182-183。
- Ulrich, Karl T. [1995] “The role of Product Architecture in the Manufacturing Firm,” *Research Policy*, Vol 24, pp.419-440.
- Ulrich, Karl T. and Steven D. Eppinger [2000], *Product Design and Development*, New York: McGraw-Hill, pp.181-206.