

## 製品アーキテクチャーをめぐる研究の系譜

馮, 全

<https://doi.org/10.15017/3000346>

---

出版情報：経済論究. 119, pp.55-68, 2004-07. 九州大学大学院経済学会  
バージョン：  
権利関係：

# 製品アーキテクチャーをめぐる研究の系譜

馮 全

## 目次

1. はじめに
2. 製品アーキテクチャーの概念
  - 2.1 アーキテクチャーと製品アーキテクチャーの概念
  - 2.2 製品アーキテクチャーの分類
  - 2.3 製品アーキテクチャーの表現形式
3. 製品アーキテクチャーをめぐる諸研究
  - 3.1 製品アーキテクチャーの特性分析
  - 3.2 アーキテクチャーの戦略マネジメント
4. 終わり

## 1. はじめに

初期のコンピュータにはソフトや関連機器の互換性はなかった。1964年、IBM/360<sup>1)</sup>の開発を契機にモジュール部品が作られ、それを共有することで機種内での互換性をもたせることが可能となった。これにより、コンピュータ産業は大きく発展した。このような驚くべき進歩をもたらしたのは部品のモジュール化である。Simon (1969) は、すべての人工システムが階層構造を持つと指摘し、その複雑なシステムの分析枠組みに関して「準分解可能性」という概念を提示した。準分解可能性を持つシステムとは、サブシステム間の相互作用は弱いですが、サブシステムの内部には強い相互作用が働いているシステムである。そのようなシステムは、サブシステム間の相互作用と、サブシステム内部の相互作用とを区別することが出来るため、全体システムを階層構造として記述できる。もともと建築学で使われていたアーキテクチャーの概念を、システムの設計思想へ転用することによって、複雑なシステムをサブシステムに分割するという新しい分析手段を提供することになった。アーキテクチャーの概念を製品設計からサービス、工程、組織、物流プロセス、情報ネットワーク事業にまで拡張してみると、人間が構想し設計する人工システムは、複数の要素を連結する構造だという考えも導き出せる。それを分割し、分析することによって、今までとの違った側面から企業活動を把握することができる。製品アーキテクチャーは、ますます企業分析にとって有効な道具となっている。一方で、自動車は典型的なインテグラル型製品として知られている。「自動車を分解すると、まずエンジ、ボディ、シャシー、内装など約10の領域に分かれる。さらにばらせば、数10のサブシステム、1千を超える機能部品、そしてボルトナットまでばらせば約3万の単体部品に分解できる」(藤本, 2003, pp.100-101)。このような複雑なシステムでは、部品の製品アーキテクチャー特性により、部品メーカーにおけるイノベーション

ン生成プロセスが違ふと考えられる。また、企業間関係もそのイノベーション生成プロセスに影響を与えるか否かどうかも研究の対象である。以上の問題意識を踏まえ、本稿はアーキテクチャーの概念及び類型を明確化する上で、コンピュータ産業及び他の産業の製品アーキテクチャーの先行研究を整理する。その中でも、自動車産業のモジュール化をめぐる研究を焦点に当てる。最後に今後の研究課題を提示する。

## 2. 製品アーキテクチャーの概念

### 2.1 アーキテクチャーと製品アーキテクチャーの概念

アーキテクチャーは、古代ギリシア語の *architekturaw* を語源とし、「偉大な技術」を意味する。そして、具体的には建築術建造物のことを指す（小原・所・渡辺・国吉，1995，p.9）。

1964年、IBM/360の設計者であるアムダール（G.Amdahl）らによって、初めてコンピュータ・アーキテクチャーという言葉に正確な定義が与えられた。すなわち、「計算機アーキテクチャーとは、データの流れ、制御法、論理設計、物理的実現法などの機構を踏まない、プログラマーからみた概念的構成、機能動作などの基本構造である」。その定義を簡単に一言でいえば、「プログラマーから見えるコンピュータの論理的な仕様」であり、言い換えれば、命令セットの定義である（Amdahl, 1964, p.87）。現在のコンピュータ・アーキテクチャーとは、狭い意味ではソフトウェアとハードウェアのインターフェースの定義を意味するが、通常はもっと広く、「コンピュータ・システムを階層化して設計する際、階層間のインターフェース定義、および階層内の構成方式」として定義され、コンピュータの構成技術全般を指す（馬場，1995，p.12）。

Ulrich (1995a) は、一般的なアーキテクチャー概念を提示した。彼によれば、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る」。そして製品アーキテクチャーの概念は、「製品における機能要素の配置」、「機能要素から物質的な構成コンポーネントに写像すること」、「相互作用している構成コンポーネントの間でのインターフェースの規定仕様」から成り立っている（Ulrich, 1995a, p.419）。また製品アーキテクチャーは、「a. 製品システムの概略的な構成図を描くこと、b. 相互関係がある要素（機能あるいは物理）を固まりにすること、c. 概略的な幾何様式レイアウトを作ること、d. 要素間の主要で基本的なかつ付属的な相互依存関係を識別しあるいは確認すること」という四つの手順によって立てられる（Ulrich, 1995b）。

藤本（2003）によれば、「製品を構成するコア部品をどのように連結するか」に関する基本コンセプトを、製品アーキテクチャーと呼ぶ。製品の要求機能をどのように展開し、製品をどのような部品に切り分け、機能をどのように配分し、部品間の接合部分（インターフェース）をどのように設計するかなどに関する基本ルールが、アーキテクチャーの選択により決まっている。

一方で、本郷・国立（2001）によれば、製品アーキテクチャーとは、「ある製品の機能をどのように物質的な構成コンポーネントに分配するかという計画あるいは体系概念である」。アーキテクチャーとは、「人工物システムの特質をとらえる為の概念であり」（本郷・国立，2001，p.6），すなわち人工物システムをどう解釈するかに関する基本的な設計構想のことである。言い換えれば、そのシステムの「わ

け方」と、分けた構成要素間の「つなぎ方」に関する基本的なものの考え方ということになる。そのため、アーキテクチャーという概念は、あらゆる人工的なシステムに対して適用することができる。企業活動や経営現象に応用する場合は、a. 企業が開発し販売する製品のシステム、b. その製品を生産する工程のシステム、c. その製品を最終顧客に販売するための流通システム、d. 開発・生産・販売を統括するビジネス・プロセスのシステム、e. 活動の主体となる企業の内部組織のシステム、f. 連携して開発・生産・販売を行う複数の会社の企業間ネットワークのシステム、g. 企業間でやり取りされる取引情報や物流のシステムなどが考えられる。これがいわゆる「ビジネス・アーキテクチャー」である。

上記のように、アーキテクチャーの概念は、建築構造の意味からコンピュータ・アーキテクチャー、製品アーキテクチャー、ビジネス・アーキテクチャーなどを経て、その意味をますます拡張して進化している。これは下記のアーキテクチャーの理論をめぐる研究の展開と一致する。以上の議論をふまえて、本稿では製品アーキテクチャーを、「ある製品の機能をどのように物質的な構成コンポーネントに分配するか、という計画あるいは体系概念である」と定義する。

## 2.2 製品アーキテクチャーの分類

製品アーキテクチャーは、一般にモジュール型とインテグラル型、オープン型とクローズ型に分けられる (Ulrich, 1995; Fine, 1998; 藤本, 1998; Baldwin and Clark, 2000)。ただし、藤本 (2001) が指摘したように、「厳密的に言えば、ある製品が全体として「インテグラル型」であるか「モジュール型」であるかという問い自体が正確でない。その製品アーキテクチャーがモジュール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思う方が合理的である。つまり、ある「モジュール型」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上位の1階層で強いモジュール性が強く現れる製品」を指している (藤本, 2001, p.7)。例えば、典型的モジュール型製品であるパソコンは、各部機能要素と構造要素との写像関係が極めて単純で、かつ構造要素の間のインターフェースが階層化・標準化されている。しかし、内部のコア・コンポーネントであるCPUは、他のコンポーネントに対する外部のインターフェースがオープンであるに対して、その中身は実に比較的にインテグラル・クローズド製品に偏っている。

### ①モジュール・アーキテクチャー

モジュール・アーキテクチャーの製品とは、機能と部品 (モジュール) との関係が1対1に近い。各部品を見ると、それぞれ自己完結的な機能があり、一つ一つの部品に非常に独立性の高い機能が与えられている。そのため、部品相互間の信号やエネルギーのやり取りもそれほど必要ではない。したがって、インターフェースが比較的にシンプルで済む。各モジュール (部品) の設計者は、インターフェースの設計ルールについて事前の知識があれば、ほかの部品の設計の影響を受けずに独自の設計ができる (藤本, 2001)。

上記のモジュールとは、その内部では構造的要素が強く結びつき、他のユニットの要素とは比較的弱く結びついている。その結びつきには明らかに程度の差があり、従って、モジュール化には濃淡がある。言い換えれば、モジュールとは、構造的に互いに独立しているが、一緒になって働く大きなシ

システム中の単位である。モジュール化は、モジュール内で相互依存し、モジュール間で独立している。そのため、モジュール化によってある複雑なシステムは、より小さな部分に分割され、それぞれを別々にみることで管理できる。ある要素の複雑さが一定の限界を超えるときには、単純なインターフェースを持つ別個の抽出を定義することで、その複雑性を隔離できる。抽出によって、その要素の複雑性が隠される。すなわち、インターフェースは、より大きなシステムにおいて要素がどのように作用するかを示す (Baldwin and Clark, 2000)。モジュール化は、構成要素間のインターフェースの数を減らし、相互作用を弱めることによる複雑性の削減を意図している。したがって、構成要素間の調整や擦りあわせにかかるコストを大幅に削減できると期待される。また、各モジュールの独立性が確保されるため、システムに対する変化をモジュール・レベルに局部化することが可能になる。それだけでなく、各モジュールはほかのモジュールと無関係に進化できる。モジュール化はシステムの複雑性を削減する戦略となっている。

②インテグラル・アーキテクチャー

インテグラル・アーキテクチャーの製品とは、機能群と部品群との関係が錯綜しているものを指す。部品群が相互に調整しあって、トータル・システムとしての機能をもつ。また、逆に一つのモジュールが多く機能を担っている。そのため機能と部品の関係が多対多の関係にある。したがって、各モジュール (部品) の設計者は、互いに設計の微調整を行い、相互に緊密な連携を採る必要がある (藤本, 2001)。

以上の分類に、複数企業間の連携関係を加えると、表 1 に示すように、「オープン・アーキテクチャー型」と「クローズ・アーキテクチャー型」という、もう一つのアーキテクチャー分類となる。そうすると、写像関係がより複雑かつインターフェースが複雑でルール化されていないタイプと、写像関係がより単純かつインターフェースが単純化されているタイプの 2 種類に大別することができる。注意しなければならない点は、インテグラル・アーキテクチャーにはクローズとオープンが存在し、モジュール・アーキテクチャーにもクローズとオープンが存在する。だが、現実的には、写像関係が複雑でインターフェースがルール化されていないものが公開されても意味がないため、殆どありえないはずである。したがって、オープンなものはモジュール・アーキテクチャーであると考えてもさしつ

表 1 製品アーキテクチャーの類型化

	インテグラル・アーキテクチャー	モジュール・アーキテクチャー
クローズ	<ul style="list-style-type: none"> <li>● 機能と部品との関係が複雑</li> <li>● 部品間インターフェースは複雑でルール化されていない。</li> <li>● インターフェースは一般に公開されていない。</li> </ul>	<ul style="list-style-type: none"> <li>● 機能と部品との関係が単純で 1 対 1 に近い</li> <li>● 部品間インターフェースは単純でルール化されている。</li> <li>● インターフェースは一般に公開されていない。</li> </ul>
オープン	<ul style="list-style-type: none"> <li>● 機能と部品との関係が複雑</li> <li>● 部品間インターフェースは複雑でルール化されていない。</li> <li>● インターフェースは外部に公開されている。</li> </ul>	<ul style="list-style-type: none"> <li>● 機能と部品との関係が単純で 1 対 1 に近い</li> <li>● 部品間インターフェースは単純でルール化されている。</li> <li>● インターフェースは一般に公開されており、業界内で共有されている。</li> </ul>

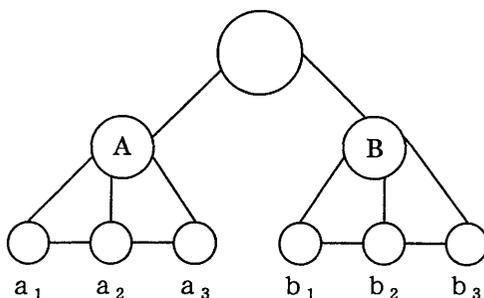
かえない (柴田・玄場・児玉・2002)。

インターフェースをオープン化するメリットとしては、製品システムのユーザー数が増加することで、自社の製品システムの魅力を増大する点である。補完財メーカーに対して製品システムをオープン化することによって、補完財市場における競争を促進し、補完財メーカーにおける規模の経済性を高めることを通じて、間接的に自社システムの価格性能比を高めることができる。また、オープン化が多様な人々の知識の集結をもたらすことで、自社システムの向上も果たすことができる。

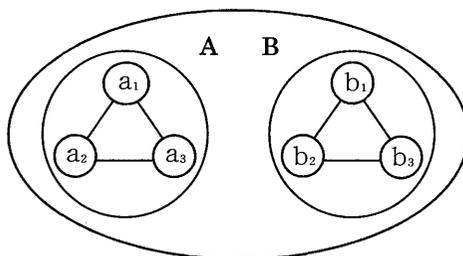
### 2.3 製品アーキテクチャーの表現形式

一般にシステムはヒエラルキー、ネットワーク、マトリックスといった形式で表現することができる。たとえば、6つの構成要素( $a_1, a_2, a_3, b_1, b_2, b_3$ )からなるシステムで、 $a_1, a_2, a_3$ はAというサブシステム、 $b_1, b_2, b_3$ はBというサブシステムを構成し、多層的なヒエラルキーとなっている。そして、サブシステムA、Bはそれぞれインテグラル的である。サブシステム内の構成要素には相互作用が存在するが、サブシステム間は相互依存性がないものと想定できる。3つの形式で以下のように表現できる。

①ヒエラルキー表現 (Goepfert and Steinbrecher, 1999; 藤本, 2002)



②ネットワーク表現 (Simon, 1969; 青島・武石, 2001)



③マトリックス表現 (Baldwin and Clark, 2000 ; ボールドウィン, 2002)

		A			B		
		a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	b <sub>1</sub>	b <sub>2</sub>	b <sub>3</sub>
A	a <sub>1</sub>		×	×			
	a <sub>2</sub>	×		×			
	a <sub>3</sub>	×	×				
B	b <sub>1</sub>					×	×
	b <sub>2</sub>				×		×
	b <sub>3</sub>				×	×	

マトリックス表現の方式は、明確に各要素の相互関係を明示できるが、3つ要素が以上になると、表現しにくくなる。それに対して、ヒエラルキー表現は要素が多くても明確に表現できるが、要素間相互依存の関係が明示できない。ネットワーク型は要素が多いほど、表現しにくくなる。

### 3. 製品アーキテクチャーをめぐる諸研究

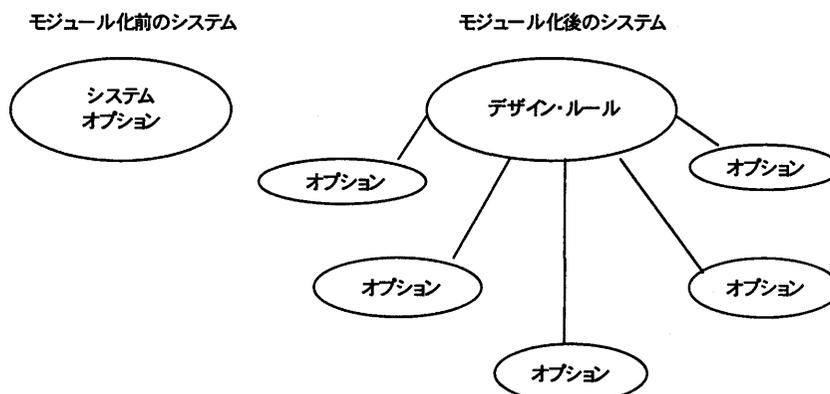
#### 3.1 製品アーキテクチャーの特性分析

Mudge (1978) は、コンピュータの設計の進化理論を提示した。市場で売れた最初のシステムを最小限とすれば、技術進歩はどのように進化するかについて研究した。そして技術進歩は基本的に異なる2つの設計スタイルに翻訳できると指摘した。第1に、(時間とともに減少する)最小価格で基本的に一定の機能性を提供すること、第2に、コストを一定に保って機能性を増すことである (Mudge, 1978, pp.315-316)。彼の研究により、設計の方向性をつけられた。

また、Langlois and Robertson (1992) は、はじめてモジュール化と柔軟性との関係を明らかにした。最初の段階で不確実性を許容する構造は、後の段階でも不確実性を受容できるので、モジュール型の設計・タスク構造は柔軟なものとなる。仮に、後の段階で、新たな知識によって隠されたモジュールの設計の一つでよりよい解決策が生まれ出されたなら、システムの他の部分を大きく変更せず比較的簡単に新たな解決策が導入できる (Langlois and Robertson, 1992, pp.312-313)。

コンピュータ産業における製品アーキテクチャーの研究の権威者は Baldwin and Clark である。彼らはモジュール化の三つのメリットを明確した。①「管理可能な」複雑性の範囲を増大する。それは、要素またはタスク間の相互作用の範囲を限定し、設計または製造プロセスで生じる循環の量と範囲を減らす。②大規模設計において異なる部分が同時に作業することを可能にする。③モジュール化は不確実性に適応する。それは、設計パラメータを可視のものと隠されたものに分離することである。隠されたパラメータ<sup>2)</sup>は、他の設計の部分から隔離され、変更可能である。その隠されたパラメータの値は不確かなものとなる (Baldwin and Clark, 2000, p.107)。また、モジュール化になるために、6つのモジュール化オペレータ<sup>3)</sup>が挙げられる。①ある設計(とそのタスク)をモジュールに「分離」(split-

図1 モジュール化が選択枝を生む



出所) Baldwin, Carliss Y. and Kim B. Clark (2000), *Design Rules: The Power of Modularity*, MIT press  
 (安藤晴彦訳 (2004) 『デザイン・ルール：モジュール化パワー』, 東洋経済新報社, 邦訳p.277を参考に加筆作成。

ting)する。②1つのモジュール設計を他のものに「交換」(substituting)する。③「追加」(augmenting)する。つまり、ある新しいモジュールをそのシステムに加える。④あるモジュールをシステムから「削除」(excluding)する。⑤新たなデザイン・ルールを創造するために「抽出」(inverting)する。⑥あるモジュールを他のシステムに「転用」(porting)する。また、分離と交換は非モジュール型にも適応でき、追加、削除、抽出、転用はモジュール型しか適応しないと指摘した。(Baldwin and Clark, 2000, pp.145-146)。彼らのもう一つ大きな貢献は金融理論において発達したオプション理論<sup>4)</sup>を使って、モジュール化オペレータの価値を創造する役割を説明した。モジュール型設計プロセスには3つの基本的段階がある。すなわち、①デザイン・ルールの定式化、つまりモジュールの数、境界線及びインターフェースを決める段階、②隠されたルールの平行作業、すなわちそのタスク構造が実行される段階、③統合・検証の段階である(同上書, pp.288-291)。このプロセスを経て、モジュール型設計では、多数のオプションを創造する。設計プロセスをモジュール化することで、設計オプションの「サイズ」と「位置」の両方が変化する。システム全体ではなくオプションの一部に影響するので、このオプションは「より小さく」なる(図1)。モジュール化は、設計の選択枝を増加させ、同時に分散させる。このため、このような選択枝は中枢のアーキテクトの許可をえなくても、多くの人が把握できる。多数の非集権化した行為主体が有益な選択枝を追求することは、システム全体の変化のスピードを加速させる(同上書, pp.259-262)。開発の不確実性が非常に高いときには、個別モジュールの革新のための競争が複数の単位によって同時に行われることが、資源の重複投入にもかかわらず、社会的に効率的であることを論理的に示したことの意義は大きい。

### 3.2 アーキテクチャーの戦略マネジメント

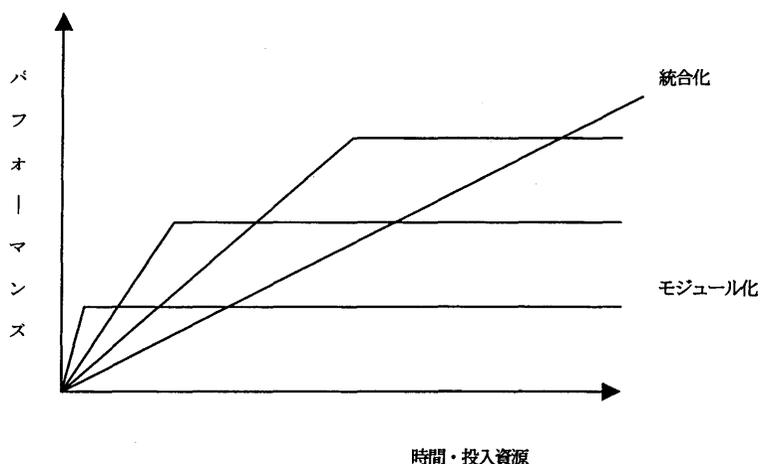
以上のアーキテクチャーの特性およびモジュール化の必要性についての研究は主にコンピュータ産業界を研究対象にして行われたものである。それを踏まえて1990年代半ばから、日本の研究者は上記の

コンピュータ産業における研究をベースにして、他の産業をめぐる研究を盛んに行った。代表的な研究者に国領、藤本、青島、武石などを挙げることができる。そのなかでも、国領は情報産業について独自の研究を行い、藤本は自動車産業について世界をリードする研究を行った。

国領 (1999) は、モジュールのインターフェースに目を向けてオープン・アーキテクチャー戦略を行うべきだと主張した。彼によれば、オープン・アーキテクチャー戦略とは、自社の持つ情報をより積極的に公開、発信し、他社の多様な情報と結合させることによって、情報価値の自己増殖現象を発生させ、その価値を自社の利益として取り込んでいく戦略である。すでに情報産業で大きな力を得ている企業は、付加価値の全てを自社で提供するのではなく、ユーザーに他企業の製品と自社製品を組み合わせて使用してもらう戦略をとるべきである。より多く企業に自社の製品を補完する製品を開発してもらうことで、自社製品の魅力はますます高まっていく (国領, 1999, pp.1-2)。彼は、オープン・アーキテクチャー戦略はものの価値を最大化するのではなく、情報の価値を最大化させる戦略なのだと強調した。また、情報産業において垂直固い込み型戦略を水平展開型戦略へ加速的に転換すべきだと主張した。つまり、自社の得意領域に経営資源を集中投入しつつ、自社に足りない機能についてはそれをもっとも得意としている企業に補完してもらう。それは、単に効率を高めるだけではなく、多様な主体が持つ情報を結合させ、価値の増大を図るメカニズムでもある (同上書, pp.100-115)。国領の研究は、規模の経済性よりネットワーク経済の重要性を示した。インテグラル型技術を分解して、自分の得意なモジュールの設計及び製造に集中して、それ以外の技術及び製造について他の企業と連携して資源を活用する有効な企業戦略である。ただし、国領は自らも指摘したように、この戦略の適応範囲はモジュール化可能の製品であることが必須条件である。

楠木とヘンリ (2001) は、ハードディスク・ドライブの技術発展の分析を通して、製品アーキテクチャーのシフトについて研究した。彼らの研究によれば、製品システムや生産システムのアーキテクチャーは、時代とともにダイナミックに変化している。つまり、インテグラル化とモジュール化の相対的優位性が変化する。産業のごく初期の段階では、製品アーキテクチャーはインテグラル的であるのが普通である。製品システムを構成する要素技術がどのように組み合わせられて動くのかが明確に定義できず、要素間の相互依存や相互作用も十分に理解されていない。インテグラル・アーキテクチャーのもとでは、要素技術やコンポーネントの相互依存や相互作用についての知識、すなわち統合に関わる知識を学習し蓄積することが、競争力のある製品を開発するために決定的に重要である。そのような製品開発はインテグラル・イノベーション<sup>6)</sup>をもたらす。インテグラル・イノベーションは、相互依存の関係にある要素技術やコンポーネントの組み合わせについての新しい知識の創造に基づいて、製品の機能や品質、コストを改善する。インテグラル・イノベーションは、相対的に長い時間をかけた実験や試行錯誤や継続的な学習を必要とする。このようなノウハウの蓄積を通じて、企業は徐々に要素技術やコンポーネントがどのように相互作用するのかについての理解を深めていく。相互作用についての理解に基づいて、企業は開発に必要なツールや専用装置、テストや実験の手法、シミュレーション・モデルなどを開発する。その結果、要素間の技術的な相互依存性は段々と小さくなり、インターフェースが明確になっていき、製品アーキテクチャーは次第にモジュール化する。モジュール化によって各モジュールの独立性が確保され、ほかのモジュールとの調整なしで進歩できる。その代わりモ

図2 モジュール化と統合化の単純化された表現



出所) 青島矢一・武石彰 (2001)「アーキテクチャーという考え方」藤本隆宏・武石彰・青島矢一編 (2001)『ビジネス・アーキテクチャー——製品・組織・プロセスの戦略的設計』, p.46より抜粋。

ジュール化に付随するインターフェースの集約化・ルール化は、システムが達成できる最大パフォーマンス・レベルに一定の制約を与える。標準が確立するため、製品システムの構成要素の相互作用や相互依存がコード化される。そのとき、製品アーキテクチャーは徐々にインテグラル化にシフトする。したがって、製品アーキテクチャーは、進化の過程でモジュールとインテグラルを繰り返すプロセスである。この意味で、製品アーキテクチャーの変化は傾向ではなくサイクルである (楠木・ヘンリ, 2001, pp.270-273)。図2に示すように、モジュール化することによって性能向上のスピードを高めることができるため、同じ時間とコストをかければ、モジュール化の戦略の方が相対的に高いパフォーマンス水準を実現できる。彼らの研究は主に内生的要素技術イノベーションによって、ハードディスク産業のアーキテクチャーのシフトをもたらしたと表明した。

また、楠木とヘンリは上記の研究を踏まえて製品アーキテクチャーと組織のダイナミックな相互作用についても理論的な研究を行った。楠木とヘンリ(2001)によれば、インテグラルな状態からモジュラーな状態へシフトし、またインテグラルなアーキテクチャーへと逆転するという製品アーキテクチャーのダイナミクスは製品技術と組織の不適合をもたらしかねない。要素技術のインターフェースが明確になるにつれてアーキテクチャーがモジュール化し、統合型組織戦略をとる企業が初期の段階で構築していた価値を消失させてしまう。そして、市場での要素間の調整が効率的にできるようになったにもかかわらず、イノベーション活動の内部調整を追及する。このような時に統合組織の限界が顕在化する。インテグラルな知識のより深い理解はしばしばモジュラー・イノベーションとして結実しうるし、要素技術レベルの変化がインテグラル・イノベーションを引き起こすこともある。製品アーキテクチャーがシフトするときに、イノベーションの源泉・機会についての事前期待と事後的な実現形態とのギャップが生まれる。これがいわゆる「組織の罅」である。製品アーキテクチャーがインテグラルな段階でリーダーシップを持つ企業は、インテグラルな知識に優れており、その強みに基づい

表 2 製品アーキテクチャーと組織の適合・不適合

バーチャル組織	適合 <ul style="list-style-type: none"> <li>● 構成要素内部での価値の創造</li> <li>● 非効率な調整の排除</li> </ul>	不適合 <ul style="list-style-type: none"> <li>● 相互作用や相互依存を管理できない</li> <li>● 不十分な開発活動ノインフラクチャ</li> </ul>
統合組織	不適合 <ul style="list-style-type: none"> <li>● 不必要で非効率な内部調整</li> <li>● 規模の経済の阻害</li> </ul>	適合 <ul style="list-style-type: none"> <li>● システム全体での価値の創造</li> <li>● 不明確な相互作用や相互依存を効果的に調整</li> </ul>
	モジュラー・アーキテクチャー	インテグラル・アーキテクチャー

出所) 楠木・ヘンリ (2001) 「製品アーキテクチャーのダイナミック・シフトーバーチャル組織の落とし穴」 藤本隆宏・武石彰・青島矢一編 (2001) 『ビジネス・アーキテクチャー製品・組織・プロセスの戦略的設計』, 有斐閣, p.266より抜粋。

てさらに製品をよりよいものにしていくことができる。よりよい製品を開発するためには、インテグラル・イノベーションを追求しつづけた方が、はるかに効率で効果的に見えるのである。それゆえに、このような「合理的」なアプローチは、企業の活動をモジュラー・イノベーションと振り向けるのを困難にする。その結果として組織的な慣性が生まれ、モジュール化していく製品アーキテクチャーと統合組織でありつづけることとの不適合が生まれる。これはいわゆる統合組織の罠である。上述の組織の罠の論理は逆の方向でも成り立つ(同上書, pp.265-273)。表 2 に示すように、モジュール・アーキテクチャーはバーチャル組織と適合するに対して、インテグラル・アーキテクチャーは統合組織と適合するという結論が導き出される。

柴田・玄場・児玉 (2002) は「分断による学習」という概念を提示した。「分断による学習」とは、製品システムを適切なサブシステムに分断しようという試行錯誤的設計行為を通じ、製品システム全体に関する新しい知識を獲得し、システム全体に関する理解を深め、より良い分断方法を発見し、サブシステム間のよりよい関係性を発見してゆく、という一連のプロセスとして認識できる。また、ファナックの NC システム技術の発展事例に基づいて、システムの複雑性を持つ製品システムについて「分断による学習」をベースにした製品アーキテクチャーの進化モデルを提唱している。このモデルは 3 つの論理から構成されている。①製品アーキテクチャーが基本的にインテグラルからモジュールへ、さらにオープンな方向へ向かって進化する。②画期的な要素技術が採用される場合に、製品アーキテクチャーが再びインテグラルの方向へ逆転してくる。このような製品アーキテクチャーの転換をもたらす要素技術の革新は産業外部から外生的にもたらされる場合と内生的にもたらされる場合がある。③製品アーキテクチャーの変化にしたがって、ビジネス・システムの仕組みも再考する必要がある。つまり、製品アーキテクチャーとビジネス・システムとの間に適応関係が存在していると考えられる(柴田・玄場・児玉, 2002, p.157)。この研究のユニークな所は、サブシステムへの分断方法という視点から、モジュール型アーキテクチャーとインテグラル型アーキテクチャーは対立的な関係ではなく、連続的な進化過程で実現される一つの分断方法としてとらえていることである。「分断による学

習」によって、企業が知識を蓄積してゆき、製品アーキテクチャーの進化を実現する。柴田・玄場・児玉のアーキテクチャーの進化論は、上記の楠木とヘンリの研究と類似しているが、前者はアーキテクチャーの転換をもたらす原因は内生だけでなく、産業外部から外生的要因もあると強調した。

自動車産業のモジュール化に関する研究も盛んに行われている。藤本（2003）によれば、自動車においては、企業を超えて共用できる汎用部品の比率は10%に満たず、インテグラル型のアーキテクチャーを持つ製品である（藤本，2003，pp.101-103）。典型的なクローズ・インテグラル型製品として、自動車産業でのモジュール化の進行についての研究が進んでいる。武石・藤本・具（2001）は自動車産業においては少なくとも3つのモジュール化があると指摘した。すなわち、①「製品アーキテクチャーのモジュール化」（製品開発におけるモジュール化）、②「生産のモジュール化」、③「企業間システムのモジュール化」（調達部品の集成化）である。自動車産業では製品の本質によって、製品アーキテクチャーのモジュール化より、むしろ、生産のモジュール化及び企業間システムのモジュール化により、製品アーキテクチャーのモジュール化を促進していると言えるだろう。彼らの研究によって、欧米の自動車産業では「企業間システムのモジュール化」が「アーキテクチャーのモジュール化」を促していく。日本の自動車産業では「生産のモジュール化」→「製品アーキテクチャーのモジュール化」→「企業間システムのモジュール化」が進んでいると表明した（同上書，p.100）。

また、部品自動車の部品メーカーに焦点を当て、製品のアーキテクチャー的特性が、承認図方式<sup>7)</sup>、貸与図方式<sup>8)</sup>、委託図方式<sup>9)</sup>、市販品方式<sup>10)</sup>など企業間の取引方式に与える影響について藤本、葛（2001）は実証分析を行った。結論として、①部品の機能的モジュラー性（相互依存性）は承認図と貸与図方式の選択に影響を与える。すなわち、機能の相互依存性が低い部品の設計は部品メーカーが設計保証を行う承認図方式が選択されやすい。機能的相互依存性の高い部品の設計の場合は、貸与図方式が選択されやすい。②部品の構造的モジュラー性（相互依存性）は取引方式の選択にあまり影響しない（藤本・葛，2001，pp.211-228）。そこからみて、機能的モジュラー性は部品取引方式に影響するといえるだろう。

さらに、韓（2000）は製品のアーキテクチャー特性と効果的な製品開発パターン間の適合関係と両者間の相互適応プロセスを明らかにする研究を行った。Thompson（1967）のいう相互依存性の概念を援用し、部品ないし子部品同士の関わりの程度を「相互依存性」と規定した。そして、自動車のエアコンとラジエータを例にして実証研究をした。研究の前提として、エアコンはインテグラル・アーキテクチャー型に近い製品であるに対して、ラジエータはモジュール・アーキテクチャー型に近い製品であると設定している。研究の結果からみて、エアコンは、高い外的及び内的相互依存性と強い企業間及び企業内企業メカニズムという適応関係を達成しているのに対して、ラジエータは、低い外的及び内的相互依存性と弱い企業間及び企業内調整メカニズムという適応関係を形成している（韓，2002，pp.144-160）。効果的な製品開発パターンは、製品アーキテクチャー特性に対応して決められるだけでなく、状況によっては製品アーキテクチャー特性を変更することにより、すでに存在する製品開発パターンにそれを適応させることも重要であることを明らかにした。

武石（2003）は、アーキテクチャーの視点から自動車産業のアウトソーシングジレンマについて実証研究を行った。アウトソーシングのジレンマとは、「企業には競争上重要な業務を外部の組織に任せ

ざるを得ない場合がある。しかし、アウトソーシングを行った企業は、一方で競争優位を構築しにくくなってしまふ。分業のパートナーが自分の競争相手とも取引があり、結果的に競争相手に差をつけることが難しくなってしまう。アウトソーシングによってとりあえず自らの負担やリスクを軽くすることができても、競争相手に差をつけられなければ、アウトソーシングは競争上は結局のところ積極的な価値を持たないことになる」(武石, 2003, pp.225-226)。彼の研究によって、①アウトソーシングの問題は多層的にとらえるべきであって、特に組織内部に目を向けることが重要である。②業務と知識を区別して考えることが重要である。既存の技術体系を前提とした効率的な知識分業ではなく、相互の知識領域が重複する冗長な分業が必要になることが明らかになった(同上書, pp.229-236)。

また、藤本(2001)は、モジュールの特性により、「20世紀後半の日本企業は、インテグラル型・クローズ型アーキテクチャーの製品で国際競争力を持つ傾向があった」(藤本, 2001, p.10)という仮説を提出した。米国企業が得意なのは「組み合わせ」重視の製品、日本企業の得意なのは「擦り合わせ」重視の製品である。そしてさらに一歩進んで、「アーキテクチャーの両面戦略」という戦略を提示した。21世紀の日本製造企業が目指すところは、オペレーション(業務)のレベルでの強みをしっかり活用しつつ、その上で従来弱かった「戦略構想能力」を強化し、オペレーション力と戦略力のバランスのとれた企業体質を作り上げていく点であると主張した(同上書, pp.25-27)。

典型的クローズ・インテグラル型製品としての自動車の製品アーキテクチャーの変化について、藤本(2003)は車両制御ソフトの統合化が、自動車のハードウェア部分のオープン・モジュール化を促進する可能性があるとし唆した。つまり、車両制御ソフトの発達によって、厳しい相互調整の必要性から解放された機能部品系のハードウェア設計が標準化へと向かい、全体として自動車のハードがモジュール化を高める(藤本, 2003, pp.350-356)。これはインテグラル・アーキテクチャー製品のインターフェースのオープン化と部品差別化の矛盾を解決する新しい技術マネジメントの方法であろう。

#### 4. 終わり

以上、製品アーキテクチャーをめぐる先行研究についての整理を行った。主に製品アーキテクチャーの進化、製品アーキテクチャーの転換と組織のダイナミックス、オープン・アーキテクチャー戦略及び自動車産業における研究を考察した。それぞれの研究成果にはそれぞれの研究対象に対応する。その中で、製品アーキテクチャーの進化は、分析の次元によって自動車産業に当てはまるかどうかは研究する必要がある。自動車は多種類の材料、部品、技術からなる総合的な組立型製品である。また、開発、生産、販売、サービス、リサイクルという一連のプロセスにわたる、巨大で複雑な人工システムである。自動車に対して、上記の研究に明記したように、製品アーキテクチャーをモジュール化するというより、むしろ生産システム及び企業間関係のモジュール化することにより、製品アーキテクチャーのモジュール化を促進できるといえよう。「はじめ」で書いてある自動車のイノベーションの生成プロセスが、製品アーキテクチャーの特性によりどのように変わっていくのかという研究は、私の知る限り、まだ少ない。一方で Robertson and Ulrich (1998) は指摘したように、モジュール・アーキテクチャーやオープン・アーキテクチャーの場合、製品差別化と部品共通化は両立しうる。すなわち、

部品の共通化を進めたとしても、部品相互の関係性を変えることにより、製品の差別化を実現することが可能である。インテグラル・アーキテクチャーの製品では、部品を共通化しようとするほど、製品の差別化は難しくなる。そして、製品を差別化しようとするほど、異なった部品を使用せざるをえない。すなわち、部品共通化度と製品差別化度は負の相関関係にある。自動車産業では典型的なクローズ・インテグラル型のシステムとしてそのモジュール化が進むとともに、部品間のインターフェースがどのように変化していくか、それに企業内部の差別化を実現するためにどのようにイノベーションを行うかを今後の課題とする。

### 参 考 文 献

- 馬場敬信 (1995) 『コンピュータ・アーキテクチャー』 オーム社。
- 韓美京 (2002) 『製品アーキテクチャーと製品開発—自動車部品開発のケース』 信山社。
- 小原二郎・所荘吉・渡辺優・国吉恵梨子 (1995) 『インテリア学辞典』 壁装材料協会。
- 本郷・国立 (2001) 「はじめに」 藤本隆宏・武石彰・青島矢一編 (2001) 『ビジネス・アーキテクチャー—製品・組織・プロセスの戦略的設計』 有斐閣。
- 藤本隆宏 (1998) 「アーキテクチャー：競争力確保の重要要素に」 『日本経済新聞』 [経済教室] 1998年3月23日。
- 藤本隆宏 (2001) 「アーキテクチャーの産業論」 藤本隆宏，武石彰，青島矢一編 (2001) 『ビジネス・アーキテクチャー—製品・組織・プロセスの戦略的設計』 有斐閣。
- 藤本隆宏 (2002) 「日本型サプライヤ・システムとモジュール化—自動車産業を事例として」 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。
- ボールドウィン・クラーク (2002) 「モジュール化時代の経営」 青木昌彦編著 (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。
- 藤本隆宏 (2003) 『能力構築競争』 中央公論新社。
- 青島矢一・武石彰 (2001) 「アーキテクチャーという考え方」 藤本隆宏，武石彰，青島矢一編 (2001) 『ビジネス・アーキテクチャー—製品・組織・プロセスの戦略的設計』 有斐閣。
- 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論—システム複雑性と分断による学習』 白桃書房。
- 吉田省三・見塚啓明・巖山昌一・神田秀樹編著 (2000) 『金融実務大辞典』 (社) 金融財政事情研究会。
- 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』 ダイアモンド社。
- 武石彰 (2003) 『分業と競争 競争優位のアウトソーシング・マネジメント』 有斐閣。
- 武石彰・藤本隆宏・具承桓 (2001) 「自動車産業におけるモジュール化—製品・生産・調整システムの複合ヒエラルキー」 藤本隆宏・武石彰・青島矢一編 (2001) 『ビジネス・アーキテクチャー—製品・組織・プロセスの戦略的設計』 有斐閣。
- 楠木建・ヘンリ・W・チェスプロウ (2001) 「製品アーキテクチャーのダイナミクス—バーチャル組織の落とし穴」 藤本隆宏・武石彰・青島矢一編 (2001) 『ビジネス・アーキテクチャー—製品・組織・プロセスの戦略的設計』 有斐閣。
- カーリス・Y・ボールドウィン (2002) 「モジュール化のコストと価値」 青木昌彦編著 (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。
- 藤本隆宏・葛東昇 (2001) 「アーキテクチャーの特性と取引方式の選択—自動車部品のケース」 藤本隆宏・武石彰・青島矢一編 (2001) 『ビジネス・アーキテクチャー—製品・組織・プロセスの戦略的設計』 有斐閣。
- Simon, Harbert A (1969), "The Architecture of Complexity: Hierarchic Systems," *The Science of the Artificial*, Cambridge, MA: MIT Press (稲葉元吉・吉原英樹訳 (1977), 『システムの科学』 第1版, パーソナル・メディア), pp.172-184.
- Gopfert, Fan and Michael Steinbrechier (1999), *Modular Product Development: Managing Technical and Organizational Independencies*, mimeo, p.55.
- Ulrich, Karl T. (1995a) "The role of Product Architecture in the Manufacturing Firm," *Research Policy Policy*, Vol 24, pp.419-440.

- Ulrich, Karl T. and Steven D. Eppinger (1995b), *Product Design and Development*, New York: McGraw-Hill.
- Fine, Charles H. (1998) *Clockspeed: Winning Industry Control in the Age of Temporary Advantage*, Reading, MA: Peruseus Books. pp.166-175.
- Baldwin, Carliss Y. and Kim.B.Clark (2000), *Design Rules: The Power of Modularity*, Cambridge, MA: MIT Press.  
(安藤晴彦訳 (2004) 『デザイン・ルール：モジュール化パワー』, 東洋経済新報社).
- Mudge, J.C (1978). "Design Decisions for the PDP-11/60 Mid-Range Minicomputer." in *Computer Engineering: A DEC View of Hardware System*.ed.C.Bell, J.C.Mudge, and J.E.McNamara, Maynard, Mass: Digital, pp315-316.
- Amdahl.G.M., G.A.Blauw, and F.P.Brooks, Jr.(1964). "The Architecture of the IBM System/360." *IBM Journal of Research and Development* 8. April: pp.87-101.
- Thompson, J. D. (1967) *Organizations in Action*, New York: McGraw-Hill (鎌田伸一他訳 (1987) 『オーガニゼーション・イン・アクション：管理理論の社会科学的基礎』, 同文館出版).
- Langlois, R.N., and P.L Robertson (1992) "Network and Innovation in a Modular System: Lessons from the Microcomputer and stereo Component Industries." *Research Policy* 21: pp.297-313.

## 注

- 1) IBM/360：1964年にIBMが開発したコンピュータである。360という名前は、角度の360°にちなんで、あらゆる用途に性能を発揮する汎用機という設計思想を表している。アーキテクチャーという用語の意味を明確に定義して使用したのも、システム/360がはじめて、プログラマーから見たハードウェアの論理仕様として定義され、同一アーキテクチャーのもとで、規模の異なる複数のコンピュータモデルが開発された。また、制御部の設計に、商用機として初めてマイクロプログラミング方式を採用した（馬場，1995，pp.5-6）。
- 2) 隠されたパラメータ（「隠された情報」とも言われる）：そのモジュールを超えて他の設計に影響を及ぼさない意思決定である。隠されたモジュールは、後で選ぶこともできるし、しばしば取り替えることもできるし、そのモジュール設計チーム以外の誰とも相談しなくともよいのである（ポールドウィン，2002，p.41）。
- 3) オペレータ：モジュールシステムに適用可能な一般の行為である（ポールドウィン，2002，p.79）。
- 4) オプション理論：オプションとは、ある資産等をあらかじめ定められた価格で、ある特定の期日あるいは期間内に、「買う」もしくは「売る」という権利である。オプション理論とはオプション価格決定モデルである。これらを使用すると、対象証券の価格、ボラティリティオプションの行使価格と行使期間、金利といった入手可能な変数の関数として、オプションの価格を求めることができる（吉田・見塚・蠟山・神田，2000）。
- 5) 水平展開型のビジネスモデルとは、自社が関わる商品のサプライチェーンにおいて、市場に提供する分野を絞る一方で、自社がコミットする分野についてはより多くの地域で高いシェアを取ろうとする経営方式のことである（国領，1999，p.99）。
- 6) インテグラル・イノベーションとは、引き起こす変化が要素間の結びつき方の変化によるイノベーションのことをさす。それと対照にモジュール・イノベーションとは、引き起こす変化が特定の要素技術やコンポーネントで完結しているイノベーションのことを指す（楠木・ヘンリ，2001，p.268）。
- 7) 承認図方式：部品の開発作業は自動車メーカーと部品サプライヤーの間で分担される。自動車メーカーは目標コスト、目標性能、外形、取り付け部詳細などの基本設計情報を車全体の車両計画やレイアウトに基づいて作成する。一方、部品サプライヤーは、詳細設計図・組立図作成、部品試作、部品単体実験などを行う。これに対して、自動車メーカー側は部品図を検討し、試作部品を実車に装着してテストし、要求性能を満たしているかを確認したうえでその設計図を承認する（藤本・葛，2001，p.213）。
- 8) 貸与図方式：自動車メーカー自らが部品詳細設計図を含む殆どの開発作業を社内で行う方式である。つまり、基本設計能力のみならず詳細設計能力も自動車メーカーに集中しているのである。この場合は、サプライヤーは単なる製造能力の提供者とみなされる（同上書，p.214）。
- 9) 委託図方式：設計開発作業の分担関係は、上述の「承認図方式」と基本的に同じである。しかし、承認図とは異なり、最終図面は自動車メーカーが所有する（同上書，p.213）。
- 10) 市販品方式：サプライヤーが特定の部品をコンセプトから生産まで一貫して行い、市販品・汎用品として売る。自動車メーカーは単にサプライヤーのカタログの中から商品番号を選んで発注する（同上書，p.213）。