column
研究員の視点

単にモジュール化することが重要なのではない

井上明人(いのうえ・あきと)
国際大学GLOCOM研究員

1980年生まれ. 国際大学GLOCOM研究員/助教. 慶應義塾大学SFC研究所上席研究員(訪問). 慶應義塾大学総合政策学部卒, 同大学院政策・メディア研究科修士課程修了. 2006年より国際大学GLOCOM研究員. 専門はコンピュータ・ゲームの理論, ゲーム産業論. 論文「遊びとゲームをめぐる試論―たとえば, にらめっこはコンピュータ・ゲームになるだろうか―」(『Mobile Society Review未来心理』vol. 13)など.

「モジュール化」は, ボールドウィン, クラークらによって提唱された概念で, 日本では経済学者の青木昌彦[★1]や池田信夫[★2]らが中心となって紹介している. 注目度の高い概念であるが, 垂直統合/水平分離という概念との違いが明確に意識されずに,「擦り合わせか, モジュールか」というような理解がされている場合も多い. また, 特に実質的な内容の伴わない理想的な形態であるかのように, この概念が使われることも多い. しかし「モジュール化」は, すべての産業が目指すべき理想的な形態ではないし, 水平分離のことを指しているわけでもない. 本稿では, この「モジュール化」を考えるときにどういったことが重要となるのかについて, 筆者の理解の及ぶ範囲で関連概念を整理し, その上で「モジュール化」を扱う難しさについて考えてみたい.

I. モジュール化とは

モジュール化と言うとき, 通常は供給サイドの製品開発方式として整理される. 最も有名なものはIBMが行ったIBM/360の設計だろう. ボールドウィンとクラーク[★3]によれば, 非モジュール型(インテグラル型)のアーキテクチャ(図1)とは, 簡単に言えば, 部品間の相互依存性が高いアーキテクチャのことである. ある特定の部品Xを改良しようという場合には, 他の部品Yとの関係性を調整しなければならない.

部品間で相互依存性が高いアーキテクチャでは, 部品開発を行う担当者間の擦り合わせが必須になる. 部品開発の間に相互のフィードバックをどれだけ成功させら

れるかが，最終的な仕上がりの鍵となる．もしも開発担当者間の相互フィードバックがうまく機能せず，コミュニケーションが今ひとつうまくいってないチームならば，製品の機能は発揮されず，ちぐはぐなものとなってしまうことになる．「擦り合わせ」という概念が重要になるのは，こういう理屈を前提にしている．「擦り合わせ」というと，とかく「日本人は擦り合わせが得意」という日本特殊論のフレームだけで理解されてしまいがちだが，「擦り合わせ」の概念を言い始めた藤本隆宏は，単に日本特殊論を唱えているわけではない．

自動車の部品を例にあげると，エンジンの開発者はスピードだけを考えればよいわけではない．燃費，ステアリング，サイズなど，考えるべきパラメータが多い．そうなってくるとエンジンの開発者は，ホイールや車体など，他の部品の開発担当

図1：非モジュール型（インテグラル型）のアーキテクチャ★4

機能ごとの相互依存性が高く，分離して設計することが難しい．（●は調整が必要）

	機能A	機能B	機能C	機能D	機能E	機能F	機能G	機能H
機能A			●					
機能B	●		●	●		●		●
機能C				●			●	
機能D	●	●			●		●	
機能E						●		●
機能F		●		●	●		●	
機能G					●			●
機能H		●		●		●		

出典：カーリス・Y・ボールドウィン＋キム・B・クラーク［2004］『デザイン・ルール―モジュール化パワー』（東洋経済新報社）の図を参考に筆者作成

図2：モジュール型のアーキテクチャ

機能ごとの相互依存性が低く，個別のモジュールを独立して設計しやすい．

	機能A	機能B	機能C	機能D	機能E	機能F	機能G	機能H
機能A		●	●	●				
機能B	●		●	●				
機能C	●	●		●				
機能D	●	●	●					
機能E						●	●	●
機能F					●		●	●
機能G					●	●		●
機能H					●	●	●	

出典：カーリス・Y・ボールドウィン＋キム・B・クラーク［2004］『デザイン・ルール―モジュール化パワー』（東洋経済新報社）の図を参考に筆者作成

者との連携を取ることが必須となる．この相互調整プロセスがよく知られている「擦り合わせ」である★5．

しかし，部品ごとの相互依存性を低くして，部品間の相互調整をあまり考えなくて済む――すなわち，担当者間のコミュニケーションがそれほど密でなくてもどうにかなる――アーキテクチャを作ることもできる．担当者間での相互調整コストをどうやって減らすのか．それを実現するために図1のアーキテクチャを，モジュール型に設計しなおしたものが図2である．

一般的なデスクトップパソコンの設計を考えてみよう．ハードディスク，CPU，メモリ，マザーボード，筐体，モニタといった部品は相互に独立している．エイサー社のモニタから，デル社のモニタに切り替えても問題なく使えるし，ソニー製の512MBのDRAMがマザーボードにささっているのを，バッファロー製の2048MBのDRAMに差し替えても問題なく使うことができる★6．ここで，エイサー，デル，バッファロー，ソニー各社の担当者に集まってもらって，わざわざ相互に擦り合わせを行う必要はない．インタフェースが同じであれば動作は保証される．つまり，部品と部品をつなぐインタフェースを固定化することで，部品間の相互調整コストをゼロに近くしている．それがモジュール化を行うということである．すなわち，製品を，単に分解可能な部品にできることがモジュール化なのではなく，部品と部品の相互依存性を低くし，調整コストを下げることがモジュール化なのである．

部品間の相互調整コストを下げることにはさまざまなメリットがある．何よりも重要なのは，個別の部品の開発担当者が他の部品との調整を考えずに開発を進められることだろう．そうすれば外部との調整を考えず，労力を一点に注ぐことができる．こうした労力の集中が積み上げ式の技術革新をブーストさせることにもつながる．一方でデメリットとしては，部品と部品の間で独立して開発を進めるため，全体としては余分な機能やコストがどこかにかかってくる．そのため，全体としての余分な機能やコスト低減のためには，モジュール型ではないほうがよく機能する場合もある．モジュール化は万能なわけではないが，同種のモジュール市場での競争を促進することが重要な鍵となっているような場合には，モジュール型アーキテクチャは大きな機能を発揮するものと考えられている．

2. 垂直統合／水平分業

こうしたモジュール化の概念は，垂直統合／水平分業といった概念と一緒くたにして理解されがちである．だがモジュール化の議論において，「垂直統合的なモジュール化」や「水平分業的なインテグラル化」が成立しうることが，すでに指摘されている★7．垂直／水平という概念の核は，調整のメカニズムの時間的な問題，

あるいはエージェント（提供事業者）を（1企業に）集中化させるか（複数企業に）分散化させるかという程度の意味で，あまり厳密に定義せずに使われることが多いという印象がある．

したがって，垂直／水平概念によってモジュール化にかかわる概念を見ていくと，おそらく混乱をきたすだろう．たとえば，ソフトウェア開発の調整段階を時間的な問題に絞って言えば，時間的な集中化＝ウォーターフォール型開発であり，時間的な分散化＝アジャイル型開発であると整理される．一方，開発にかかわるエージェント（プログラマー）の問題について言えば，エージェントの集中化＝社内での内製化であり，エージェントの分散化＝アウトソーシング・他社製品の利用，と整理できるだろう．では，社内でアジャイル型開発を行うのは垂直統合型なのか，水平分離型なのか？　複数の会社が協働するウォーターフォール型開発は垂直統合型なのか，水平分離型なのか？

垂直／水平という概念の抽象化が有用な場合も少なくはないだろうが，少し事態が複雑さを帯びてくると，この概念はいささか粒が粗すぎるきらいがある★8．事態が複雑になってくれば，論じるべき概念の粒度はもっと細かくする必要がある．一体，何の調整が集中化されていて，何の調整が分散化されているのか――たとえば，時間なのか，エージェントなのか――これをきちんと整理しないと空疎な議論が繰り返されてしまうことになるだろう．

たとえば，モジュール型のアーキテクチャにかかわるエージェント（エンジニア）たちは，それぞれに他のエージェントとの間の調整コストを支払わなくても済み，自分のやりたいことをある程度まで自由にやれることにはなる．だが，同時にモジュール型のアーキテクチャというのは部品と部品をつなぐインタフェースを固定化するということとほぼ等しい．つまり，モジュール型アーキテクチャでは，インタフェースをどのような形で固定するのか，ということはあらかじめ調整が済んでおり，最初の時点でそのアーキテクチャ設計にかかわることのできなかったエージェントにとっては，ときにそれはイノベーションの範囲の制約にもなりうる．

たとえば，任天堂が1983年に発売したファミリーコンピュータは，ゲームソフトとゲーム機の間のモジュール化を成し遂げて成功した例だ．だが，ファミリーコンピュータが技術的に陳腐化していくにつれて，ソフトウェア開発会社にとっては，モジュールの設計が制約となってしまう．これを回避するためには，ファミリーコンピュータの進化版としてスーパーファミコンが発売されたり，インターネットに接続可能なドリームキャストのような別の設計がなされたゲーム機が発売されたりしたように，インタフェースが適度に進化していく仕組みが重要になる．そうしなければ，固定化されたモジュールのインタフェースは便益だけでなく，束縛として機能してしまうことになる．

また，モジュール型アーキテクチャのほうが，非モジュール型アーキテクチャよりも，財の多様性が増加し，自由度が高くなると思われているが，これも一律には言えない．確かに，ファミリーコンピュータは，ソフトウェアのバリエーションを増やすことには成功しただろう．しかし，インタフェースまで含めた上でのコンピュータゲームのバリエーションの多さという点では，ハードからソフトまですべてを設計することのできるゲームセンターのゲームのほうがはるかにバラエティに富んでいる．指先だけでなく手足を活用するようなゲームはゲームセンターではかなり昔から広く流通していたが，家庭用ゲーム市場では近年になってWiiが流行するまで，身体を活用したゲームという部分でのイノベーションは抑えられていた．

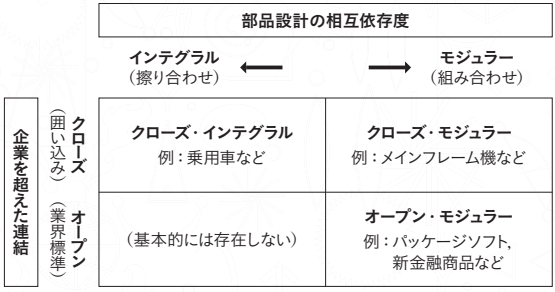
3. クローズ化とオープン化

さて，もう一つ重要な関連概念として，クローズ／オープンという区分についても見ていきたい．

オープン型とは何か．部品と部品をつなぐインタフェースの情報が一企業の中だけで閉じておらず，業界レベルでの標準化がなされている状態が「オープン型」にあたる．一方で，モジュール化がなされてはいても，部品間をつなぐインタフェースの情報が標準化されていなかったり，API（Application Program Interface）などの形でインタフェースが公開されていなかったりする状態のものを「クローズ型」や「囲い込み型」★9と呼ぶ．

図3は，藤本隆宏による，モジュール化とオープン化の区分を表した図である．横軸にインテグラル／モジュールの区別★10をとったとき，ここにクローズ化／オープン化という区別を重ねることができる．図の右下のような，モジュール型でかつオープンな形態であれば，「企業を超えた『寄せ集め設計』が可能であり，異なる企業から素性のよい部品を集めて連結すれば複雑な『擦り合わせ』なしに，た

図3：藤本隆宏による「モジュール化とオープン化の区分」



出典：藤本隆宏［2003］『能力構築競争 日本の自動車産業はなぜ強いのか』中公新書

だちに機能性の高い製品が生み出される」★11ことが可能になる．

一方で，先述したメインフレームの名機であったIBM/360について言えば，高度なモジュール化を達成してはいたが，インタフェースは公開されておらず，基本的にはIBM社内でのクローズ型のアーキテクチャの製品であった．単に「モジュール型」と言ったとき，すぐさまオープン・モジュール型のアーキテクチャと考えられることが多いが，それだけではないということを押さえておきたい．なお，クローズなモジュール型アーキテクチャであっても，もちろん調整コストの削減というメリットは発生する．

ただ，ややこしいことに，クローズ／オープンと言っても，明確に区分けができるものではない．図4を見ればわかるとおり，細かく論じていけばかなりグラデーションを持つことになる．トヨタの生産工程はある程度クローズではあるかもしれないが，Windowsはどうだろうか？　Windowsは確かにクローズな領域が多いが，かなり多くのインタフェースを公開し，ソフトウェア開発者がWindows上で自由に開発できるような環境を提供しているのは事実だ．もちろん，Linuxほどにオープンなプロジェクトではないが，完全にクローズなわけでもない．マイクロソフトは一時期「オープンな企業」と言われたが，同時に「ブラックボックスを設けているクローズな企業」という批判もされる．

つまり，オープンとクローズという概念を細かく見ていくと，どこをクローズにしてどこをオープンにするか，という切り分けこそが，より実質的な議論にとって重要だということがわかるだろう．

図4：クローズとオープンのグラデーション

クローズ

	IBM／360	トヨタ式生産システム	ファミリーコンピュータ	Windows	Linux（オープンソース）	W3C
社内完結	●	●	●	●		
数社協働作業	●	●	●	●		
他社製品互換	▲	▲	●	●	●	–
API公開				●		
業界標準化						●
パワーユーザーが利用可能				●	●	●
ライトユーザーが利用可能				●	●	●

オープン

●該当する　▲ほぼ該当する

出典：筆者作成

4. モジュール化が重要なのか

以上，関連概念について簡単に整理を試みてきたが，モジュール化とこれにかかわる問題は単純な話ではないということが，理解していただけたのではないだろうか．すでに池田や藤本らによって指摘がなされているように，モジュール化は常に，どの産業にでも有用なわけではない．ある時期にはモジュール型アーキテクチャよりも，インテグラル型アーキテクチャのほうが有利に働いたり，ある時期にはモジュール型が有利に働いたりする．また，産業別，国別の競争環境によっても何がよりよい選択肢なのかは異なってくる．

すでに触れたように，モジュール化とは，多種多様なエージェントの自由な活動を保証する，というだけの話ではない．むしろ，ある部分に制約を伴わせることによってはじめて，相互調整コストや初期投資コストの低下といった便益を可能にすることができるのである．相互調整コストを高くしてでも無駄のないシステムを構築しようとする場合など，モジュール化が適さない場合もあり，決して万能のシステムではない．単一のモジュール型アーキテクチャしか選択できないような市場設計は，むしろ市場全体にとってマイナスに働く危険性もありうる．どのようなモジュール型アーキテクチャが個別の市場にとって適切なのか――すなわち，どこからどこまでをひとまとまりにして分断し，どこの調整コストを低下させることが適切なのか――は，個別の市場からのフィードバック抜きに判断することはできない．こうしたフィードバックのないモジュール化は，不利な仕組みを強制してしまう危険も伴う．

先のファミリーコンピュータの例で言えば，家庭用コンピュータゲームの歴史上「任天堂のファミリーコンピュータがモジュール型アーキテクチャだったから成功した」という観点のみから，ゲームの産業史を考えるのは浅薄な理解である．アタリVCSやカセットビジョンといった家庭用ゲーム機市場での先行製品や，ゲーム機と入門用パソコンとしての性能を併せ持ったMSXパソコンなどの製品も存在するなかで，競争の末に勝ち残ったのがファミリーコンピュータである★12．高性能，低価格を実現するとともに，市場の絞り込みといった戦略が，市場による淘汰のなかで選ばれたことを考えなければ，ファミリーコンピュータの普及は説明することができない．当時（1980年代前半）は，いかに高性能であってもゲームしかできないファミリーコンピュータよりも，MSXパソコンのような低価格パソコンの方向性こそより優れた解だという意見は決して少なくなかった．しかし，MSXパソコンは生き残らなかった．

モジュール化が重要なのではない．重要なことは，競争条件に合わせて，その解の一つとして★13，いかなるモジュール化を構想するかということだ．しかし，い

かなるモジュール設計が望ましいのか，という問いは悩ましい．今のところ，その望ましさを最終的に測る指標は，市場による淘汰しかないのかもしれない．

註

★1――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の本質―』東洋経済新報社

★2――池田信夫［2005］『情報技術と組織のアーキテクチャ　モジュール化の経済学』NTT出版

★3――カーリス・Y・ボールドウィン＋キム・B・クラーク［2004］『デザイン・ルール―モジュール化パワー』東洋経済新報社

★4――図1では，複数の機能の関係性に，階層的な順序のある意思決定がかかわる場合を考慮に入れている．たとえば飲料の容器を設計する際には，「ふたを付ける／付けない」（機能A）を決定してから→「ふたの直径」（機能B）を決める．この場合，機能Aは機能Bに影響するが，機能Bは機能Aに強い影響を与えない．詳しくは，ボールドウィン＋クラークの前掲書p.46を参照．

★5――藤本隆宏［2003］『能力構築競争　日本の自動車産業はなぜ強いのか』中公新書

★6――正確には，すべてのマザーボードでDRAMやCPUのインタフェースが共通しているわけではないので，それぞれのマザーボードが指定しているインタフェースを持ったものを選ぶ必要はある．

★7――日本の製造業が1970年代から80年代にかけて自動車・電機・精密機械などの業種で高い競争力を発揮した原因は，この種の製品では連続的に細かい改善を重ねる必要があるため工程の補完性が高くなったことにある．特に製造業においては，二次元の設計図ですべての情報を表現することは不可能であり（青島・延岡・竹田［2000］），設計部門と製造部門との「擦り合わせ」が不可欠である．このように職能集団を横断した協力が必要な場合には，垂直統合型の組織は職能集団ごとのセクショナリズムを助長し，「設計図を塀越しに投げ渡す」ようなものとなりやすい（Detouzos et al. [1989]）．ここでは，モジュールとしての職能集団の結合様式がウォーターフォール型に固定されているため，かえって官僚主義の弊害が大きいのである．これに対して，トヨタ型の水平分業においては，すべてを内製化するのではなく，下請けにまかせて長期的な信頼関係にもとづいて管理することによって需要の変動に対する柔軟な対応を可能にした（池田，前掲2，p.52）．

★8――また，こうした水平／垂直にかかわるものの中で近年新たな装いを持って登場してきた概念の一つが，マルコ・イアンシティ／ロイ・レビーンによる『キーストーン戦略』（翔泳社，2007）だろう．

★9――國領二郎［1999］『オープン・アーキテクチャ戦略』ダイヤモンド社

★10――藤本によればモジュラー（部品）型と表記されているが，同一のものとしてここでは扱う．

★11――藤本，前掲4，p.89.

★12――詳しくは，クラシックビデオゲームステーションオデッセイ掲載の「家庭用テレビゲーム機究極年表」を参照：<http://www.ne.jp/asahi/cvs/odyssey/dataroom/gamelist.html>．国内で発売された家庭用テレビゲーム機はほぼ網羅されている．ファミリーコンピュータがいかに競争的な市場の中での淘汰を経たゲーム機であったかがわかるだろう．

★13――もちろん，モジュール化が解でない，ということもありうる．