

ファームウェア・アーキテクチャの揺れ動きとその要因

デジタル複合機の事例

福澤 光啓

東京大学大学院経済学研究科

[E-mail: ee67016@mail.ecc.u-tokyo.ac.jp](mailto:ee67016@mail.ecc.u-tokyo.ac.jp)

立本 博文

東京大学ものづくり経営研究センター

[E-mail: tatsu@gbrc.jp](mailto:tatsu@gbrc.jp)

新宅 純二郎

東京大学大学院経済学研究科

[E-mail:shintaku@e.u-tokyo.ac.jp](mailto:shintaku@e.u-tokyo.ac.jp)

要約：近年、情報家電や携帯機器などをはじめとした多くの製品分野において、半導体チップとして組み込まれた MPU とソフトウェア（ファームウェア）の重要性が高まっている。機械製品の代表であった自動車ですら、MPU とソフトからなる電子コントロールユニットがいまや 100 程度も搭載されている。そのようなファームウェアが、製品の高機能化、耐久性向上、多機能化、低コスト化に果たす役割は大きく、日本の電子製品、機械製品の競争力を支える重要な要素となっている。しかし、その役割が増大するにともなって、ファームウェアの大規模化や複雑化が進んでおり、開発コスト増大、開発リードタイム長期化、製品不具合の頻発といった深刻な問題も起こしている。そのような状況の中で、多様な機能をつかさどる一群のファームウェアをどのようなアーキテクチャで設計するかが、製品開発の重要なテーマとなっている。機能の増加に対して追加的・場当たりの対応では、全体としての製品機能を達成できず、様々な問題が生じる。本報告では、デジタル複合機におけるファームウェア・アーキテクチャの変遷についての事例を取り上げて考察する。

キーワード：ファームウェア、MPU、ファームウェア・アーキテクチャ

¹ 本稿は 2006 年 4 月 21 日開催のコンピュータ産業研究会での報告をもとに、報告者の一人である福澤光啓（東京大学大学院）が本誌掲載のために作成したものである。

1. はじめに

近年、情報家電や携帯機器などをはじめとした多くの製品分野において、半導体チップとして組み込まれた MPU とソフトウェア（ファームウェアとか組み込みソフトと呼ばれている）の重要性が高まっている。機械製品の代表であった自動車ですら、MPU とソフトウェアからなる電子コントロールユニット（ECU）がいまや 100 程度も搭載されている。そのようなファームウェアが、製品の高機能化、耐久性向上、多機能化、低コスト化に果たす役割は大きく、日本の電子製品、機械製品の競争力を支える重要な要素となっている。しかし、その役割が増大するにともなって、ファームウェアの大規模化や複雑化が進んでおり、開発コスト増大、開発リードタイム長期化、製品不具合の頻発といった深刻な問題も起こしている。

このような傾向が始まったのは、1970 年代である。1969 年に日本の電卓メーカーのひとつであったビジコン社が電卓用半導体チップの開発をインテルに依頼した。この時に、現在さまざまな製品で使われているマイクロプロセッサ（MPU）の概念ができた。それまでは複数の半導体チップが連携して演算処理をおこなっていたのだが、その機能が単一の半導体チップに集約されるようになったのである。これが、パソコンだけでなく、先に述べたようなさまざまな製品の中に組み込まれるようになっていった。MPU には、それを動かすソフトウェアが必要である。そのため、さまざまなハード製品の中に、MPU やマイコンチップとソフトが組み込まれており、いわば簡単なパソコンとソフトが内蔵されている状態になっている。

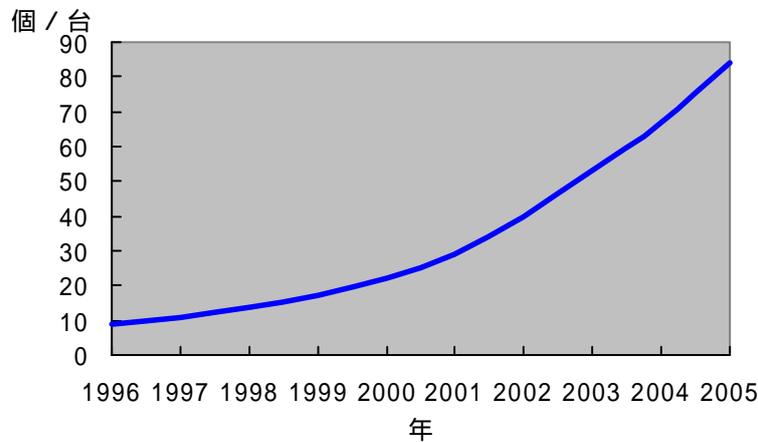
2. 製品に用いられているファームウェアの役割

(1) 自動車：燃費と耐久性の向上

次に、このファームウェアが製品の中で具体的にどのように使われているのか、二つの例を挙げて説明する。最初の例は自動車である。図 1 は、車の中に使われているマイコンチップの使用数量の推移を示している。1990 年代に数がどんどん増えていき、2005 年には約 100 個のマイコンチップが車の中に搭載されるようになった。マイコンチップが使われているのは車だけではない。いわゆる「白物家電」といわれる炊飯器やエアコン、あるいは電話、ファックス、さらには最近のデジタル家電製品であるデジタルカメラ、液晶テレビ、プラズマテレビのような薄型テレビ、携帯電話などにも、マイコンチップが数多く使われるようになっている。これらの家電製品などでも、自動車と同じくだいたい 100 個ぐらいのマイコンチップが使われている。

上で述べたように、自動車の中には約 100 個のマイコンチップが使われている。その場所

図1 自動車に用いられているマイコンチップの使用数量推移

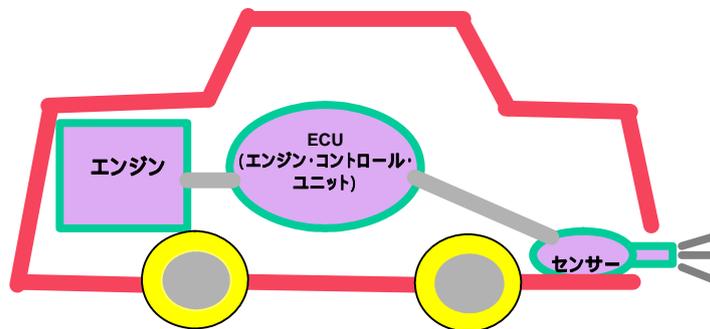


は、わかりやすいところではオーディオやカーナビなどの情報系の部分、そのほか、ブレーキのシステム、あるいはワイパーの動きを制御する部品などに、マイコンチップとソフトウェアがセットで組み込まれている。中でも一番重要なのが、図2に示したエンジン・コントロール・ユニットである。これはエンジンを制御する小型のコンピュータで、ドライバーの運転席の前方、ハンドルの下あたりに組み込まれている。

エンジン・コントロール・ユニットはリアルタイムで情報を処理し、最適な燃焼状態をコントロールしている。そのために重要な役割を果たしている部品がセンサーである。エンジンでガソリンと空気を燃やしてその排気ガスが後ろから吐き出されるわけだが、この排気ガスの状態をチェックしているのがセンサーである。排気ガスの状態を見ることによって、エンジンの中で適切に燃焼が起きているかどうか分かる。1970年代に排気ガス規制が実施されたとき、規制をクリアするためにエンジンの燃焼を最適状態に保つ必要が生じた。つまり、ガソリン量を常にコントロールして、ガソリンと空気の混合比を最適な状態に維持することが必要になったのである。エンジンの爆発によって生じる排気ガスの状態を測定し、その結果をエンジン・コントロール・ユニットというコンピュータに戻す。コンピュータはあらかじめ定められたプログラムにしたがって、燃焼効率を上げるための調整をしてガソリン噴出量などの命令をエンジン関連部品に送る。このようなサイクルをリアルタイムで繰り返していくわけである。それによって、排気ガス規制をクリアできるエンジンができるようになった。

このシステムによって可能になったのは、排ガス規制クリアだけではない。まず第一に、燃費を良くすることができる。近年、ハイブリッド・エンジンなどで燃費の良いものが登場

図2 自動車のエンジン・コントロール・ユニット



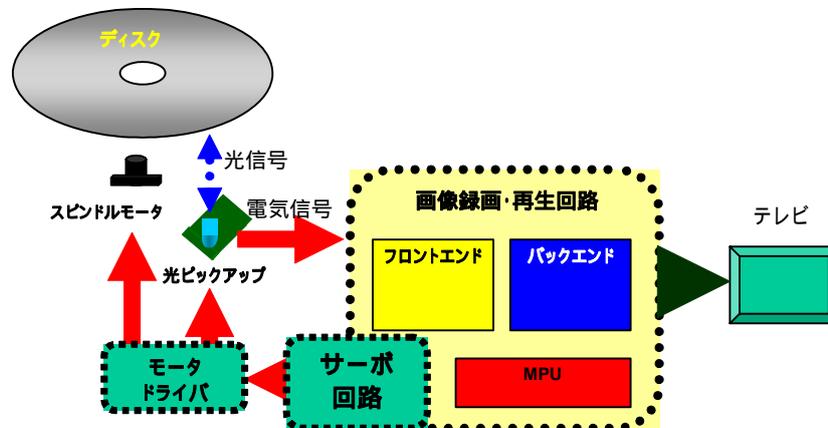
しているが、ガソリン・エンジンも画期的に燃費を向上させている。長い歴史のあるガソリン・エンジンの燃費が最近になって向上している理由の半分ぐらいは、ここに組み込まれているファームウェアの貢献で、あとの半分ぐらいがエンジンそのものの改良によるらしい。ハードの改良とソフトの改良が相まって、全体の燃費が向上しているのである。

もうひとつの効果は、耐久性の向上である。日本の自動車は耐久性が高いといわれる。これも、ハードの部品や機械加工の精度が高いということだけではなく、ファームウェアが貢献している部分がある。ガソリンはバルブを通じて燃料噴射されているわけだが、5年、10年乗っていると、このバルブの中が詰まって噴射されるべきガソリン量が減り、燃焼状態が悪くなってくる。しかし先の排気ガス監視センサーが機能していれば、燃焼状態の悪化を感知して、ガソリンの押し出し量を少し多くするように噴射圧力を強くし、最適な燃焼状態を保つことができる。その結果、5年、10年経っても馬力が落ちない、燃費が落ちない状態が実現されることになる。

(2) 光ディスク装置：モジュラー化の促進

ファームウェアが製品に影響をおよぼしたもうひとつの例は、CD や DVD に使われている光ディスクである。光ディスクは表面に細かな凹凸がたくさんあり、そこに光ピックアップという部品がレーザーを当て、その反射で凹凸を読む。それを (0, 1) のデジタル信号に変換し、電子回路に送り、それを合成して画像情報に戻していくという仕組みになっている。表面の凸凹は非常に細かいので、それを読むためには精密な制御が必要になるが、どうしてもさまざまな読み取りエラーが出てくる。光ピックアップ自体にも、部品による個体差がある。これらの誤差や個体差をできるだけ少なくするよう制御するのがサーボ回路である(図3)。

図3 CDレコーダやDVDレコーダにおけるサーボ回路



情報技術は、このサーボ回路に二つの大きな変化をもたらした。第一は、1990年代半ばに、サーボ回路が完全にデジタル化されてソフトウェアで管理されるようになったことである（デジタル・サーボ回路）。これにより、従来はピックアップと装置の最適セッティングをハード的に追い込んでいかななくてはならなかったのが、ソフトウェアで処理できるようになったのである。ハードウェアを調整して最適な状態を実現しようとするれば、事前に決めた最適値のところに部品の状態を持っていくしかなかったのだが、ソフトウェアによる制御であれば実際にデータを読んでみて、何回か試行を繰り返すことで最適値を得ることができる。部品の個体差やばらつきがあっても、最終的に組み上げた状態で調整することができるようになったのである。

第二点は、ディスクと装置の間のすりあわせを容易にする技術（ライト・ストラテジー； write strategy）である。光ディスクには先に述べたように表面に小さな凸凹信号がある。CD-RやDVD-Rといった記録用の書き込みディスクの場合、ディスクの表面に特殊な色素を塗布し、その色素をレーザーで焼いて凹凸を作っている。しかし、どのような色素を使うか、どのような製造方法を採用しているかは、メーカーによって異なる。その結果、レーザーで焼くときの最適条件がメーカーによって大きく異なるのである。ライト・ストラテジーは、さまざまな特性をもったディスクにメーカーのIDをつけ、まずそれを読んで、ディスクのメーカーを判断し、最適な書き込み方を選んで実行するのである。これがなければ、ディスクメーカーに求められる製造基準は非常に厳しくなり、製造できる企業は限られてしまう。ファームウェアによって、ディスクの製造がはるかに容易になっているのである。

(3) ファームウェアによる製品の高性能化

ファームウェアの導入によって、実際の製品がどのように高性能化しているのだろうか。たとえば、デジタルカメラにもファームウェアが使われている。デジタルカメラの画質を良くするために、最近では600万画素や800万画素という製品も出回るようになった。これはCCDという半導体チップ上に、画素数に応じた電子の眼を作り込んでいることを意味している。その製造工程は画素数が多くなればなるほど、小さな画素を半導体チップの中に作り込まなくてはならないから難しくなる。実際、はじめて300万画素の製品を作った時に、300万個すべてが完璧になっているかということそうではなくて、どこかの画素に欠陥がある。しかし、どこに欠陥があるかは製品によりバラバラである。そこで、欠陥を補正するのにソフトウェアを使う。欠陥のある画素が見つかったら、その画素の周囲の情報を読み取って、欠陥画素の情報を補正復元するのである。周囲が全部肌色であればその画素の情報も肌色だろうし、周辺の色が変化していればその変化の度合いを読み取って欠陥画素の部分の色を推測するのである。このような補正ソフトをファームウェアに組み込むことで、ハード面では完璧ではなくても、数百万画素という高機能の製品を作り出すことができるのである。

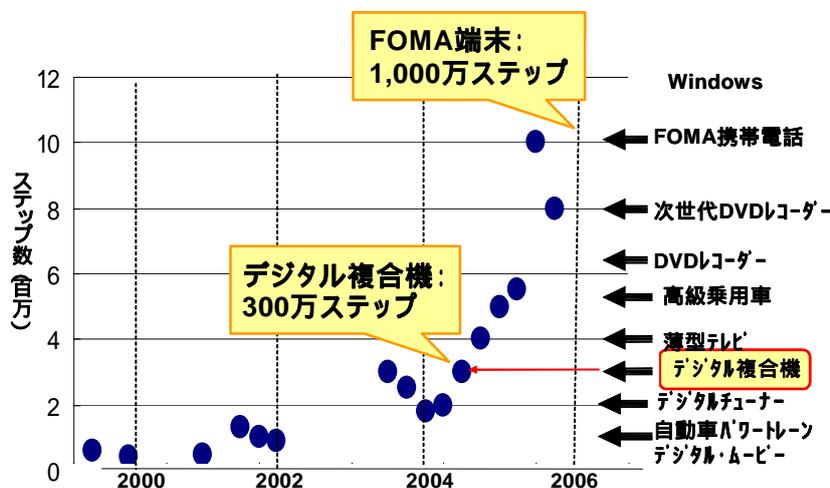
(4) ファームウェアの種類とステップ数の爆発

ファームウェアの種類としては、マイコンチップ・ファームウェア（マイコンチップ（SoCやシステムLSI含）のROM/EPROM内のソフトウェアのこと）、OSレイヤー下のファームウェア（プロトコルスタックやデバイスドライバー等）、OSレイヤー上のファームウェア I（ミドルウェアやライブラリのことであり、OSを含むこともある）、OSレイヤー上のファームウェア II（アプリケーションやユーザーインタフェース）、ネットワークを介したファームウェア（分散処理・統合処理を念頭に置いたファームウェア）の五つが挙げられる。

図4では、製品に用いられているファームウェアのステップ数とその推移が示されている。2000年以降、ファームウェアのステップ数が百万行を超え、急速に巨大化している。このステップ数の急増は、上記の～の種類ファームウェアのうち、後者三つのレベルで起きている現象である。本報告で取り上げる近年のデジタル複合機に用いられているファームウェアのステップ数は300万である。携帯電話のFOMA端末にいたっては、1000万ステップを越えるほどにまで増大している。

このステップ数の爆発的な増大が、製品開発の上では、きわめて深刻な問題をもたらしている。前述のように、ファームウェアは製品の競争力の面で極めて重要な役割を果たしている。したがって、いかにして、効率的かつ効果的にファームウェアを開発していくかが製品

図4 ファームウェアのステップ数の推移と製品の関係



出所) 小川紘一氏(東京大学ものづくり経営研究センター研究員)作成資料

開発の上で重要な課題である。しかし、ステップ数の増大に見られるファームウェアの複雑化は、製品開発に深刻な問題をなげかけている。すなわち、製品開発期間の長期化や、複雑な構造や分業に起因するバグの発生が見られるようになり、ファームウェアのバグによる製品のリコールさえ頻発する分野もある。

本報告では、ファームウェアが製品機能決定の重要な要素であると捉え、比較的早い時期にファームウェアの大規模化が観察されるデジタル複合機の事例を取り上げて、ファームウェアの構造と開発組織について論じていきたい。

3. ファームウェア・アーキテクチャの揺れ動き デジタル複合機の事例

(1) ファームウェア・アーキテクチャ

ここまで見てきたように、ファームウェアを用いた製品において複数の機能を提供しようという場合には、それぞれの機能に対応したアプリケーションソフトやミドルウェア、OSなどが必要になるが、これら複数のファームウェア・コンポーネント間の調整をどのように行えば、製品の多機能化をよりうまく行うことができるのだろうか。

従来の経営学では、どのようにして製品を構成部品(コンポーネント)に分割し、そこに製品機能を配分し、それによって必要となる部品間のインタフェースをいかに設計・調整するかに関する基本的な設計構想を製品アーキテクチャと呼び(藤本, 2001)、それに関する研究が行われてきたが、そこで注目されているのは、主としてハードウェア・コンポーネント

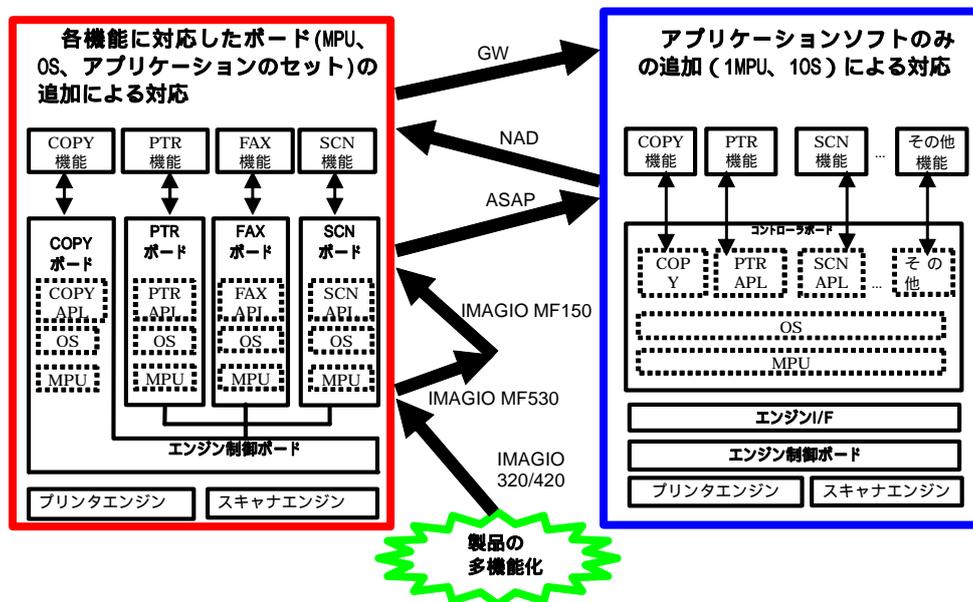
である（青島，武石，2001；Baldwin & Clark，2000；藤本，2001；Henderson & Clark，1990；Ulrich，1995）。そのため、従来の議論では、ファームウェアを用いた製品分野において競争している企業が直面している問題に対して十分な処方箋を提示できていない。

本稿では、製品機能とファームウェア・コンポーネントとの対応関係や、ファームウェア・コンポーネント間の対応関係、ファームウェア・コンポーネントとMPUとの対応関係から記述される製品システムの特徴を「ファームウェア・アーキテクチャ」と呼び、ファームウェアを用いた多機能化を長年にわたって実現してきた製品の好例としてデジタル複合機を取り上げて考察する。²

(2) デジタル複合機におけるファームウェア・アーキテクチャの揺れ動き

複写機業界全体では、多機能化を行なう必要があるということは複写機のデジタル化が行なわれた1980年代中頃には既に明らかとなっていた。1990年代半ばまでは、アナログ複写機が主流であったが、その後デジタル複合機が急速に普及した。³ デジタル複合機においては、複写機能についてはコピー機（プリンタとスキャナの連動）、FAX機能についてはFAX、読み取り機能についてはスキャナ、印刷機能についてはプリンタが対応しているが、それだ

図5 デジタル複合機におけるファームウェア・アーキテクチャの揺れ動き



² 株式会社リコーにおけるファームウェア・アーキテクチャの開発事例に基づいている。

³ 既に1990年代初頭において、デジタル複合機に用いられているファームウェアのステップ数は百万行を越えていた。

けではコピー機やプリンタやスキャナとして機能しない。実際にこれらの機能を実現するためには、これらのハードウェア・コンポーネントを制御するためのファームウェア・コンポーネントが必要である。

図5に示されているように、リコーのデジタル複合機のファームウェア・アーキテクチャにおいては、アプリケーションやOS、MPUをひとつのセットとして、そのボードの追加によって対応するか、ひとつのOS上で複数のアプリケーションを動作させることによって、機能の追加や削除の要求にアプリケーション・レイヤにおいて対応するという、二種類のアーキテクチャ間での揺れ動きが起きていたのである。⁴

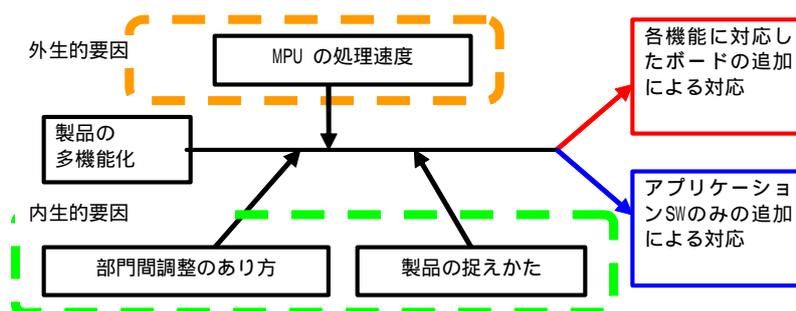
(3) ファームウェア・アーキテクチャの揺れ動きの要因

このようなファームウェア・アーキテクチャの揺れ動きはなぜ生じていたのだろうか。その要因として、図6に示されるように、MPUの技術的境界、製品の捉え方の変化、および、開発組織における相互調整のあり方の変化という三点が考えられる。

まず、MPUの処理速度が不十分な場合には、必要とされる性能を維持しつつ複数のアプリケーションを並列に処理できなかったため分散型の処理が行われていたのだが、その後のMPUの性能向上により並列処理が可能となったため統合型の処理が可能になったのである。

次に、製品の捉え方の変化としては、従来はコピー部門がファームウェア・アーキテクチャを開発していたので、デジタル「複合機」と言いながらも、実際には「もできる複写

図6 ファームウェア・アーキテクチャの揺れ動きの原因



⁴ リコー初の普及型デジタル複合機である「IMAGIO 320/420」が発売された1987年から、最新のアーキテクチャ（「GW アーキテクチャ」）に基づいた製品である「IMAGIO Neo 320/420」が発売された2001年までの間に、「IMAGIO MF530」「IMAGIO MF150」「ASAP アーキテクチャ」「NAD アーキテクチャ」といったファームウェア・アーキテクチャが開発された。

機」として製品が捉えられていた。その後、ASAP アーキテクチャや GW アーキテクチャにおいては、製品の捉え方が「オフィスで生じる様々な情報を処理するための機能を搭載した事務機器」というものへと洗練されてきたのだが、それが最終的に実現したのは GW アーキテクチャのときであった。

最後に、部門間調整のあり方の変化について、最後の三つのファームウェア・アーキテクチャの開発に即して考察する。ASAP アーキテクチャを開発する際には、MPU の処理速度が足りないながらも、ひとつの OS 上で複数のアプリケーションを動作させるアーキテクチャが採用された。そのようなアーキテクチャを可能にするためには、既存のコア部門がファームウェア・アーキテクチャの開発を行うのではなくて、他の機能までを含めた「全体最適」の観点から設計を行うために部門間の緊密な相互調整が必要であったにもかかわらず、それが不十分であったため失敗した。

このように、MPU の処理速度が遅いことと、部門間の調整を行うのが困難であることを受けて、部門間の緊密な相互調整を行うことによってファームウェア・コンポーネントの抽出を行うよりもむしろ、部門間の相互調整をあまり必要としないアーキテクチャに回帰したのが NAD アーキテクチャであった。しかし、これでは当時要求されていた水準の機能を実現できなかったのである。

これらの二度の失敗を経て、GW アーキテクチャにおいては、複数のアプリケーションを並列処理するに足るだけの MPU の処理速度が実現されたことと、部門間の緊密な相互調整を行うための専門のプロジェクトチームが設置されたことによって、多機能化の要求にアプリケーション・レイヤで対応できるようなファームウェア・アーキテクチャの開発に成功したのである。このように、デジタル複合機のファームウェア・アーキテクチャの開発において重要であったのは、各ファームウェア・コンポーネント間の共通部分を洗い出して、新たなファームウェア・コンポーネントとして「抽出」(Baldwin & Clark, 2000) することであったと考えられる。これを最もうまく行うことができたのは、ファームウェア・アーキテクチャを開発するための組織を設けて緊密な部門間調整を行うことが試みられた GW アーキテクチャのときであった。

(4) 事例研究から得られた教訓

デジタル複合機の事例研究では、組織が主体的にファームウェア・アーキテクチャを変えていくということが観察されたのだが、このことは、従来のコアとなる機能に縛られて製品を捉えていたり、それに適した部門間調整のあり方の下でファームウェア・アーキテクチャの開発を行うと、狙った機能を実現することが困難であるということを示唆している。この

ような問題を克服するためには、本事例から示されたように、 トップマネジメントが積極的に関与して、ファームウェア・アーキテクチャの開発を行うためのリソースを大量に投入し、 従来のコア機能を開発してきた部門の意見に縛られることなく、当該製品に対する捉え方の見直しを図り、 部門間での相互調整を緊密に行うことが必要であると考えられる。

4. まとめ

ものづくりにおけるファームウェアの役割として、 製品の機能をソフトウェアで実現するという側面（たとえば、携帯電話のメール・ソフトやウェブ閲覧などの機能は、ソフトで実現された機能である） 光ディスクの例に述べたような、部品のばらつきを許容するという側面（光ピックアップやディスクなどの部品の間の互換性が、ファームウェアによってきわめて高くなり、その結果、装置や製品の低コスト化、さらには製品の高性能化が進む）ということが挙げられる。このようにファームウェアは、製品を大きく変えてきた。

ファームウェアは、ハードウェアとソフトウェアの融合と見ることができる。製品の低コスト化、高性能化の背景にはもちろんハードウェアの進歩そのものがあるわけだが、ハードだけだと限界があるときに、ソフトがそれを補い、両方が合わさって次の段階までいけるといのがファームウェアの真骨頂であり、それによって製品の高性能化が実現している。

参考文献

- 青島矢一, 武石 彰 (2001) 「アーキテクチャという考え方」 藤本隆宏, 武石彰, 青島矢一 編著 『ビジネス・アーキテクチャ』 2章. 有斐閣.
- Baldwin, C. Y., & Clark, K. B. (2000). *Design rules: Vol.1. The power of modularity*. Cambridge, MA: MIT Press.
- 藤本隆宏 (2001) 「アーキテクチャの産業論」 藤本隆宏, 武石彰, 青島矢一 編著 『ビジネス・アーキテクチャ』 1章. 有斐閣.
- Henderson, R. M., & Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly*, 35, 9-30.
- Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24, 419-440.

赤門マネジメント・レビュー編集委員会

編集長 新宅 純二郎

編集委員 阿部 誠 粕谷 誠 片平 秀貴 高橋 伸夫 藤本 隆宏

編集担当 西田 麻希

赤門マネジメント・レビュー 5巻7号 2006年7月25日発行

編集 東京大学大学院経済学研究科 ABAS/AMR 編集委員会

発行 特定非営利活動法人グローバルビジネスリサーチセンター

理事長 高橋 伸夫

東京都文京区本郷

<http://www.gbrc.jp>