ANNALS OF BUSINESS ADMINISTRATIVE SCIENCE 7 (2008) 1–18
Available at www.gbrc.jp

Online ISSN 1347-4456 Print ISSN 1347-4464

A Generation-Selection Process of Product Architecture: A Case of Development of the Firmware in Digital MFP

Mitsuhiro FUKUZAWA

Graduate School of Economics, the University of Tokyo

E-mail: mifukuzawa5@gmail.com

Abstract: In this article, we focus on the firmware development process that has recently become larger and more complex in the context of digitalization. From a case study involving the development of the firmware for a digital multifunction peripheral (MFP), the firmware architecture have changed alongside the change in organizational structure. As a result, some suggestions are made for the generation and selection process of a product architecture suitable for multi-functionalization, as well as for the ideal form of interdepartmental coordination in an organization. Moreover, the limits of the existing arguments about the development of product architecture are revealed and a new viewpoint is proposed to investigate the dynamics of product architecture.

Keywords: product architecture, firmware, digitalization, multi-functionalization, interdepartmental coordination, digital MFP

1. Introduction

Recently, in many product areas such as information household appliances, portable devices, automobiles, and industrial instruments, there has been an increase in the functions realized by the firmware[1] or by the interaction between the hardware and the firmware. Because of this, the scale and complexity of the firmware has increased. Concurrently, the development cost of the firmware has come to occupy a large portion in the whole product development cost. This has placed an urgent need for an efficient firmware development. For example, a high-end model automobile, which can be defined as a typical machine product, is now equipped with around a hundred electronic control units (ECU, each

[1] It can also be called embedded software, but in this article, we use the term firmware.

consisting of a microcomputer and its firmware), which are used to realize numerous functions. In order to ensure that these functions have a high quality, it is important that multiple ECUs are controlled comprehensively, which is an urgent task for automobile manufacturers and parts suppliers.

In existing arguments about product development and product architecture management (Aoshima & Takeishi, 2001; Baldwin & Clark, 2000; Fujimoto, 2001; Henderson & Clark, 1990; Ulrich, 1995), hardware components have attracted most of the attention. Therefore, an insufficient number of product architecture studies have considered both the hardware and firmware. In addition, in the field of software engineering, software development models have been studied (Tatsumoto, 2002), but there has been insufficient study on the influence that embedded software has on the product architecture as a whole.

However, the influence of the firmware on the product architecture increases as the firmware used for a product increases in size and importance (Kato, 2002; Shintaku, Ogawa, & Yoshimoto, 2006). To efficiently provide a product with good multi-functionalization, it is important to develop methods for selecting the functions to incorporate into the product and for coordinating/integrating these functions. In a digitalized product, application software, middleware, and an operating system (OS) corresponding to each function are necessary. How should these firmware components be coordinated/integrated, in order to provide good multi-functionalization? What kind of organization is suitable then, in order to accomplish coordination and integration of these firmware components?

To answer these questions, in this article, we first define the characteristics of a product system described by the correspondence between a) the product functions and firmware components, b) the firmware components themselves, and c) the firmware components and CPU as the "firmware architecture." We then consider a case involving the development of the firmware for a digital multifunction peripheral (MFP), released by Ricoh Ltd., which has been used to realize multiple functions for many years. Our point of view is different from the existing argument, which considers a change in the product architecture as a kind of environment that requires manufacturing firms to adapt. Our view is, however, that a change in the product architecture can be seen as the result of an organization *selecting* a different product architecture. From this point of view, we pay attention to the changes in the firmware architecture and the changes in the ideal method of interdepartmental coordination in the organization developing these architectures. Thereby, we can offer a way to perform the product development activity more effectively and efficiently.

2. Existing argument about product architecture and its limits

Arguments about product architecture have involved the relationship between the function and structure of a product (Aoshima & Takeishi, 2001; Baldwin & Clark, 2000; Fujimoto, 2001; Ulrich, 1995). It has been shown that there are some constant conformity relations between the type of product architecture (integral type or modular type) and the ideal organizational structure for its development (interdepartmental coordination or a division of labor between companies) (Aoshima & Takeishi, 2001; Fine, 1998; Fujimoto, 2001; Kusunoki & Chesbrough, 2001; Sanchez & Mahoney, 1996; von Hippel, 1990).

Furthermore, it is pointed that the product architecture changes dynamically with progress over time, so that it generally shifts from an integral type to a modular type (Baldwin & Clark, 2000), or, in contrast, that it shifts from a modular type to an integral type (Fine, 1998; Kusunoki & Chesbrough, 2001). Existing studies focused on mainly two occasions that the product architecture changes, that is, when the function that is going to be realized by the product changes (Henderson & Clark, 1990), and when the technology used for a product changes (Kusunoki & Chesbrough, 2001). However, neither approach has provided sufficient investigation into the process used by a company when changing the product architecture.

As above, when the existing argument about the product architecture are analyzed, it is thought that "a change in the product architecture" is recognized as a kind of "environmental change" to which the organization should respond. For example, as for the claim "that the organization strategy of the modular type is suitable for the product architecture of the modular type" in Kusunoki and Chesbrough (2001), the focus of the argument is fixed on the kind of organization suitable for a modular product architecture. This reflects the point of view that the product architecture is recognized as a certain "environment" to which an organization must adapt. As a result, it is not sufficiently clear what kind of mutual adjustment has to be made in an organization to make the product architecture modular. Going to extremes, it has only been discussed whether an organization of a certain type can adapt to a certain product architecture. The existing arguments do not recognize an observed change in the product architecture as the result of an organization making a decision to adopt a different product architecture. Accordingly, the process by which an organization changes its product architecture has been disregarded. Thus, under this point of view, it is difficult to make useful suggestions on how a company should act when changing simultaneously the product architecture and the organization.

Therefore, we adopted the point of view that "an organization proactively selects a product architecture." We observed cases of product architecture shift to consider "what kind of intention the organization held when moving to a different

Figure 1. Investigation area covered in this article

		focused architecture	
		hardware	firmware
view point	product architecture →organization	most of existing study	
	organization →product architecture		this article

Source: Author.

product architecture” and “what kind of organization was suitable for realizing such a change.” When we take this point of view and reinterpret the product architecture framework, the following three points must be considered. First, behind the observed product architecture, a series of decisions must take place about how to combine the necessary functions and components. Secondly, only the change in the product architecture is observable ex post in a market, and behind such a change, certain decision making is generated. Then, a series of processes wherein a particular architecture is selected take place over time. Finally, after the selection and manufacturing of a specific architectural product, yet another product architecture can be generated and selected. In particular, here we consider how these processes are influenced by interdepartmental coordination, product concepts, and available resources.

Figure 1 shows the position of this study based on the above-mentioned argument. In many of the existing studies about product architecture, attention has been given to the architecture of the hardware and the response of an organization to the product’s architectural change as given. In contrast, in this study, we give attention to the architecture of the firmware and the process by which an organization selects the product architecture.

3. A case study: Development of the firmware for the digital MFP by Ricoh

3.1. Object and method

Nowadays, multiple functions, such as copying, faxing, printing, scanning, and network function are offered in a single digital MFP. However, this has not always been the case. Through trial and error, the number of feasible functions gradually increased (i.e., multi-functionalization) as a means of market differentiation. In order to integrate multiple functions in a single product, it was important to determine a method to allocate the firmware components that controlled each function and to let them work in harmony. However, existing arguments that the firmware facilitates the modular product architecture (Kato, 2002; Shibata, Genba, & Kodama, 2002; Shintaku, et al., 2006), analyzed the individual firmware components but overlooked the design concept behind firmware as a group. Thus, in this article, we consider the change in the firmware architecture and the desirable style of interdepartmental coordination.

This case study is based on multiple interviews with the software engineers developing the digital MFP at Ricoh in 2005,[2] along with Ricoh internal information; the "Ricoh Technical Report," and Ricoh Corporate History Editing Committee (1996). We chose digital MFP as the subject of our analysis for the following reasons: a) The role of the firmware is more significant in digital MFP though both hardware and firmware are incorporated. b) It is a representative product that shifted from an analog method to a digital method at the same time multi-functionalization was pushed forward. c) It is a product that had already seen a remarkable increase in the scale of the firmware by the early 1990s.

We start our analysis from 1987, when IMAGIO 320/420, which was Ricoh's first integrated type digital MFP was released, and up to 2001, when IMAGIO Neo350/450 was released. During this period, six firmware architectures were developed, the IMAGIO 320/420 (1987), IMAGIO MF530 (1991), IMAGIO MF150 (1993), ASAP architecture (about 1994), NAD architecture (about 1998), and GW architecture (2001). These architectures can be roughly classified into two types. One involves application software, an OS, and a CPU on a circuit board where multi-functionalization can be added by the addition of a board (ASAP architecture and GW architecture). The other is a method that copes with the addition or deletion of functions in the application layer by operating multiple application software on one OS (IMAGIO 320/420, IMAGIO MF530, IMAGIO MF150, and NAD architecture).

3.2. Product architecture of the digital MFP

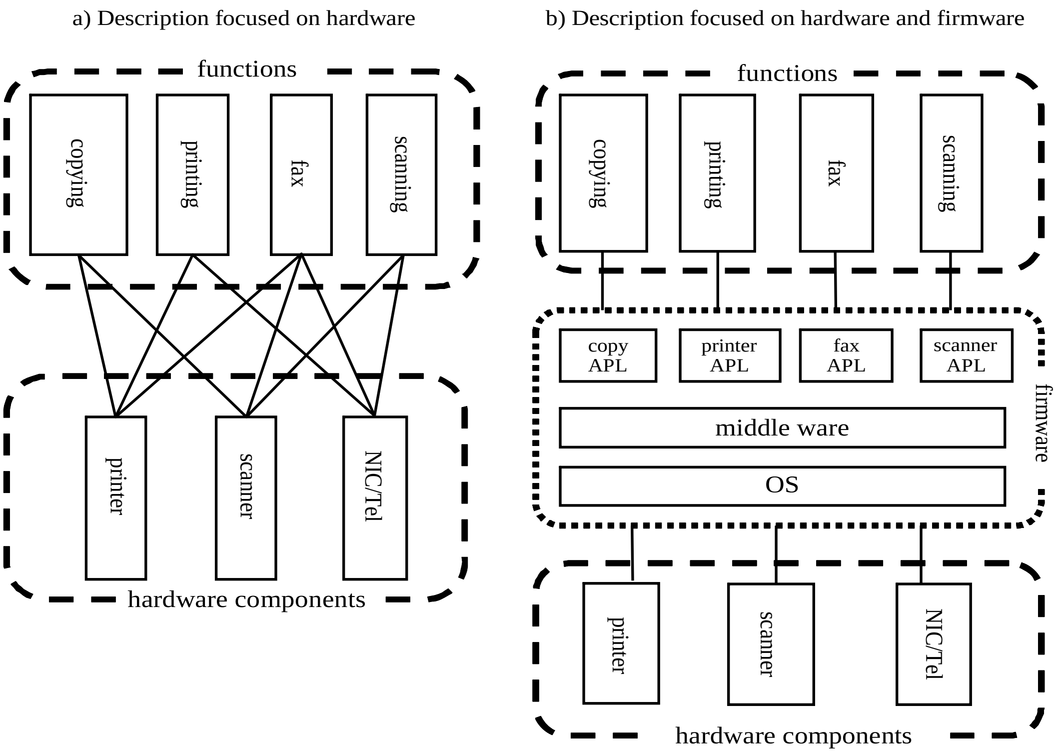
The digital MFP is comprised of both hardware (i.e., mechanical components and electronic circuits) and firmware, working together to provide various functions. The main differences of a digital MFP from an analog copier are, a) the firmware is heavily involved in the processing function,[3] b) multiple functions as a fax machine, printer, and scanner, for example, are offered in one machine,[4] c) there is

[2] Interviews were performed four times in 2005. On September 20 (three hours), October 11 (three hours), October 24 (four hours), and November 29 (questions and answers in e-mails).

[3] In the beginning of the 1970s, an analog circuit was used to control the amount of light that comes in from the light source and a relay was used for sequence control. Although these systems were highly precise and complex, they were replaced by microcomputers and firmware control. The characteristics of microcomputer control are as follows: a) capable of carrying out complicated operation, b) easy to program on-demand functions, c) hardware defects discovered in the latter half of the systems design stage can be modified by firmware. Microcomputer and firmware control were adopted widely because of these advantages, and the development of the firmware became increasingly important.

[4] The biggest problems facing digitization were

Figure 2. Product architecture of digital MFP



Source: Author.

roughly a division between the engine[5] unit and the controller unit.

When we examine the hardware and describe the product architecture of a digital MFP, it can be seen that each product function and hardware component have a one-on-many correspondence. For example, the interlocking movements of the printer and scanner constitute the copying function; the interlocking movements of the printer, scanner, and phone line constitute the fax function; and the printer[6] is also used for the print function (Figure 2a). However, each of these functions is not realized exclusively with these hardware components, but with the interlocking actions of the hardware components and firmware components used to control them.

In the digital MFP, the firmware components and product functions have a one-on-one correspondence. For example, the copying application software supports the copying function, the fax application supports the fax function, the scanner application supports the scanning function,

colorization and multi-functionalization. Because in the case of an analog copier, the copying function is performed by a large number of functional parts with complicated mechanical linkages, and it is thought that the degree of integrity in the product architecture is high.

5 Generally, the "engine" in digital MFP refer to the printer. The scanner is called scanner engine.

6 To use digital MFP as a network scanner and a network printer, an NIC (Network Interface Card) is necessary.

and the printer application supports the print function (Figure 2b). The middleware compiles the functions that are common to these applications, and the OS is used as a firmware component to make adjustments between tasks. When we describe the product architecture by the firmware, we see clear modularization in the total product (Figure 2b). In contrast, when we describe the product architecture by the hardware, we can observe characteristics of an integral type. In such a product that has a complicated function-component relation, development becomes extremely redundant if one wish to leave each component's development independent. Otherwise, an organization could perform total optimization through intricate coordination effort in an extremely integrated framework. Or if a manufacturing firm wished to reduce such bizarre complexity, one should change the assignment function given to each hardware component, which is a hardware-centric solution. Yet, a firmware-centric solution is mostly adopted in practice where the arrangement and coordination of firmware components would be manipulated. In this approach the firmware architecture becomes a crucial point of argument, and thus, it is essential that we also consider the steps taken by the organization to accomplish this task.

3.3. Firmware development activity at Ricoh

Figure 3 shows a schematic representation of the transition in the firmware architecture of Ricoh's digital MFP. In this section, we divide the firmware development at Ricoh into four phases.

3.3.1. Before ASAP architecture development

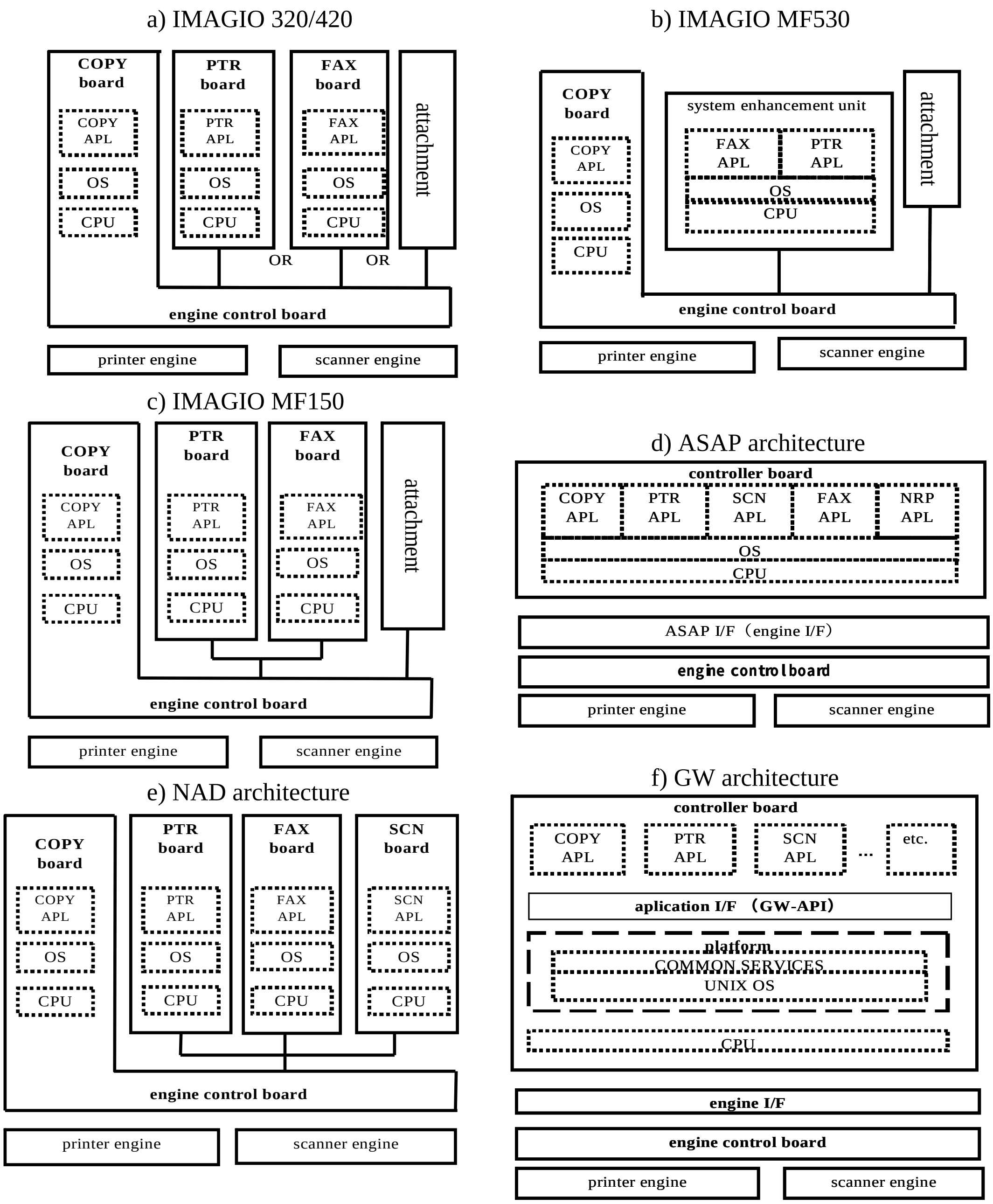
Before the ASAP architecture (i.e., IMAGIO 320/420, IMAGIO MF530, and IMAGIO MF150), product development was carried out individually in different functional units such as the copier (or a digital MFP) division, fax division, and printer division. Accordingly, firmware for each function was developed separately.[7] Mutual adjustment was little considered between different groups, because idiosyncratic divisional cultures had existed and the technical language differed among each group. In addition, the leader in the development of the firmware architecture of these models were engineers from the copier division. As a result, the firmware architecture became centered on the copying function.

3.3.2. Development and failure of the ASAP architecture

Because higher performance and multi-functionalization were promoted, the firmware in digital MFP grew in scale and complexity. This made firmware development even more difficult and pushed forward the development of the ASAP architecture, which had started from 1992. Since the project leader was a printer engineer, the ASAP

[7] According to the organizational chart that is listed in each year's version of the "Securities Report" for Ricoh Ltd., and the positions and posts given for the writers of articles in each year's version of the "Ricoh Technical Report," different divisions were involved in the development of a copier (a MFP), fax, and printer (including a scanner) from 1986 to 2000.

Figure 3. Transition of firmware architecture of digital MFP in Ricoh



Source: Interview by author and "Ricoh Technical Report."

architecture was developed based on an orientation that was different from previous models. The ASAP architecture placed each function in parallel, breaking away from the conventional copy-function centered design. Furthermore, though firmware components in each model had been standardized by then, standardization between multiple models had not been aimed at. Therefore, by standardizing firmware components between models, it was expected that development cost, quality, and lead-time will improve.

However, because until then, the firmware to realize each function had been developed with little mutual coordination between groups, it was difficult to integrate them to function adequately on one OS, and coordinating between groups were far from easy. The most particular problem was that Ricoh's original OS for the ASAP architecture had been developed independently from all divisional groups, and the disclosure of information about the OS interface was seriously delayed. This resulted in each group developing an OS and unique application software on their own. The original plan failed to let multiple functions perform in parallel on one OS. Consequently, the first ASAP architecture machines were basic copiers neither realizing OS standardization nor application software compatibility between the models.

We would like to point the following causes for the failure of the ASAP architecture. First, it was not sufficiently understood at Ricoh that the firmware plays the heavy role in achieving multi-functionalization, so that scarcely enough development resources, such as engineers and budget, were allocated. Second, although close mutual coordination between groups was necessary to develop the ASAP architecture, this was lacking. Finally, the processing speed of the CPU was insufficient.

3.3.3. Going back to the past architecture

Learning from the lessons of the ASAP architecture, the NAD architecture was developed under a project leader from the copier division. In this case, the architecture was a throwback to what had already existed, that is, it was developed independently in each group from the OS to an application rather than trying to achieve standardization between the firmware components. Reverting from the ASAP architecture to the NAD architecture aimed at reducing mutual coordination costs between various divisions necessary if an entirely new architecture was to be developed. However, the following problems existed with this architecture. First, it was weak in networking performance. Second, the development cost increased because the amount of hardware increased. Finally, a firmware had to be developed each time a new model is released because standardization between past architectures or firmware components was not realized.

3.3.4. Development of the GW architecture

The development of the GW architecture began with the launch of a project team called the GW-PT in 1998. This was the first time that a project team was organized to develop the firmware architecture at Ricoh. In developing the GW architecture, a large number of engineers were consulted, learning from the failure of the ASAP architecture, and more development resources were spent. Because, in the

NAD architecture, the firmware to realize each function was developed separately in different organizational divisions, each group developed an application and an OS, and there were many functional overlaps. In contrast, the GW architecture was developed under a leadership of the GW-PT project team that facilitated close coordination among each division. This approach succeeded in extracting the common functions realized by the firmware as "a platform," and enabled this platform used commonly between multiple models. The structure of the GW architecture resembles the ASAP architecture, but the approach taken in their development differed, and this difference affected the success of the development (Table 1).

3.4. Wavering of the firmware architecture and the factors involved

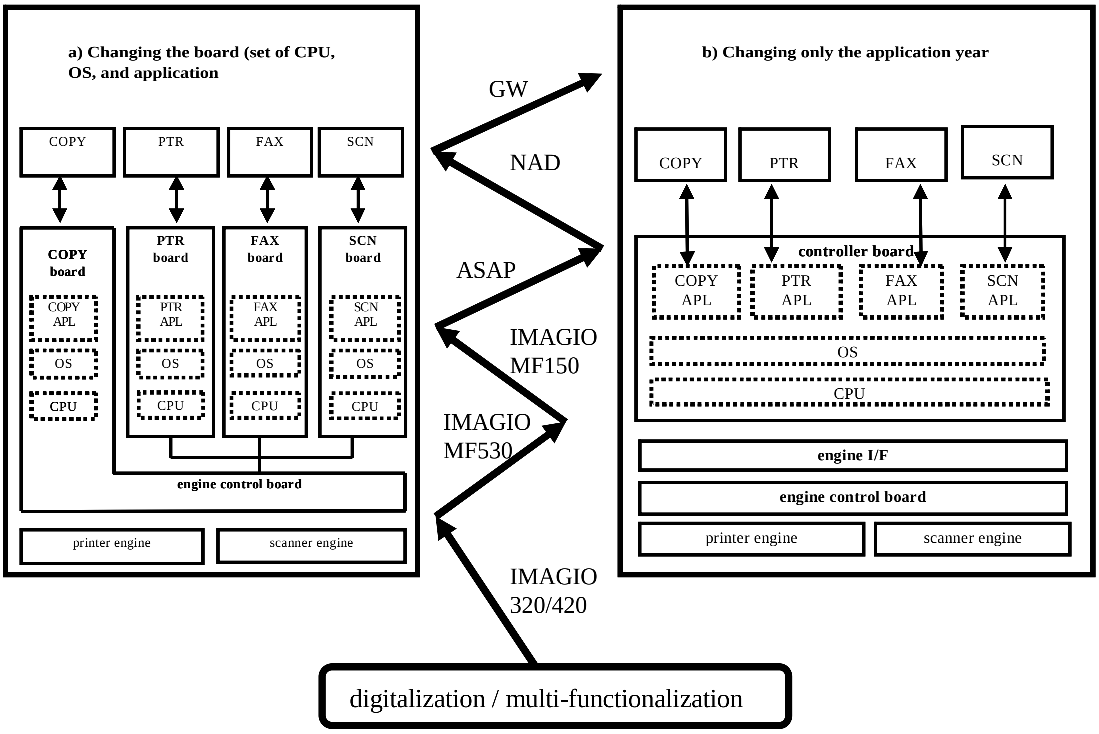
When multi-functionalization was pushed forward in the development of Ricoh's digital MFP, the approach wavered between two different firmware architectures (Figure 4). The first uses a circuit board where an application software, an OS, and a CPU are mounted, where new functions are realized by the addition of a board. The other method copes with an addition or deletion of a function by manipulating the software program in the application layer on a single OS. The former intends partial optimization of each function, so the demand on the processing speed of the CPU is low and the performance of each function is optimized, but it is so redundant that the parts cost rises and is unsuitable for realizing "a unified function" as a digital MFP (Figure 4a). The latter has the goal of total optimization of each function. It is suitable for realizing a unified function as a digital MFP and the parts cost can be reduced, but it is also necessary to provide a high-power CPU and to develop each firmware component under close

Table 1. Comparison of the development activities between the ASAP and GW

	ASAP architecture	GW architecture
resource	little	huge
organization	small team in a division	GW project team
project leader	printer	copier
development of OS	developed by a person original OS original debugging tool	developed by group (at least ten people) UNIX OS public debugging tool
information sharing (interdepartmental coordination)	Delayed →failure of standardization between functions →failure of same timing release	Facilitated →success of standardization between functions →success of standardization between models

Source: Author.

Figure 4. Wavering of firmware architecture of digital MFP



Source: Author.

mutual coordination between functions (Figure 4b). In the ASAP and GW architectures, the modularization of the firmware architecture is advanced by the interdependency between the components (i.e., construction and rearranging of the OS, middleware, and the application interface).

Wavering between two kinds of architectures as in this case does not suggest whether a product architecture is classified as the integral type or the modular type, but suggests the importance to pay attention to the difference in viewpoint and orientation when determining the interdependency between the components to realize a modular type architecture. We can examine three factors that affected the choice between the architectures: CPU performance, product concept, and the level of interdepartmental coordination in the organization.

3.4.1. Performance of the CPU

Because at first the clock speed of the controller CPU used for the digital MFP was merely tens of megahertz, it was impossible to conduct parallel processing between multiple firmware components while realizing the target performance. Later, in the GW architecture, the CPU clock speed improved up to several hundred megahertz allowing parallel processing between large-scale software which realizes multiple functions.

However, it cannot necessarily be said that the improvement in the processing speed of the CPU converge into a particular type of firmware architecture. This is because the performance of the CPU used in the product may not be good enough for the increasing amount of information that the product engineer requires. And even if the CPU did meet the requirements, the project is likely to fail if the organizational capability was lacking.

3.4.2. Product concept

Through the product development activities in Ricoh, the product concept for the digital MFP evolved from “a copier that can do other things” (IMAGIO 320/420, IMAGIO MF530, IMAGIO MF150, and NAD architecture) to “an office appliance equipped with functions to process various information in an office” (the ASAP architecture and the GW architecture). With IMAGIO 320/420, two functions were offered in one machine. This was the first step to perform “multi-functionalization,” but it was viewed as attaching an additional function to “a copier.” Following this, with IMAGIO MF530 and IMAGIO MF150, multi-functionalization was pushed forward to a greater extent, but there was still no fundamental change in the product concept. On the other hand, in the ASAP architecture, “the fusion” of each function was intended. This was a change in product concept from “a copier that can do other things” to “a multifunctional office appliance.” In place of this, with the NAD architecture, although multi-functionalization was intended as in the IMAGIO 320/420, the IMAGIO MF530, and the IMAGIO MF150, the product concept reverted to “a copier that can do other things.” In the GW architecture, the product concept of the ASAP architecture was more straightforwardly accomplished, and more integrated functions was realized.

Thus, even though the term “digital MFP” can be applied to each of these products, they are different in terms of the functions integrated into one machine and under what kind of design concept. This appears as a difference in the firmware architecture, that is, whether it is a copy-function centric type or a parallel type that emphasizes integrated functions.

3.4.3. Interdepartmental coordination in product development organization

Conventionally, firmware was developed individually at each functional division such as a copier (an MFP) division, fax division, printer division, or scanner division. However, a specialized organization was later established to develop the firmware architecture under close mutual coordination among different divisions. This organizational shift took considerable amount of time because there had been little mutual coordination between groups until then, and the importance of the firmware architecture development was not sufficiently recognized at Ricoh. These problems occurred for the following reasons: a) because the product development organization was hardware-centered and b) because a section of the existing core business (i.e., the copier team) developed the firmware architecture, and it little occurred to bring in a total optimization point of view into the design.[8] Expressing the former plainly, a hierarchy dominated the office that ranked the mechanical engineer at the top, followed by the electronics engineer and the software engineer. The firmware development team was regarded as a minor group and was allowed little development resources. Generally, the development cost of the firmware was not accounted for in the product development cost, and even if it was, it was treated as part of the cost for the ROM installed in the product.

Because the hardware and copier section were the center of the corporate organization at Ricoh, IMAGIO 320/420, IMAGIO MF530, IMAGIO MF150, and NAD architecture were developed with a focus on the copying function under the concept of "a copier that can do other things." On the other hand, the ASAP architecture and the GW architecture were designed as parallel architectures under the product concept of "a multifunctional office appliance," within organizations that are not copy-function centered (i.e., the project was lead by the printer division in the case of ASAP architecture and an individual project team in the case of GW architecture). This suggests that because the core product function and concept design are formed within the given organization, the product architecture is built to best suit the organizational benefits as well. Consequently it is necessary to change the organization in order to shift to a different product architecture.

4. Discussion

4.1. A product architecture generation and selection process

Because the firmware increased in scale and complexity as a result of digitalized multi-functionalization and integration, most of the interdependency issues between functions and components concentrated on the firmware development stage. This made the selection of the firmware architecture synonymous with the selection of the product architecture. We consider that the wavering between the two firmware architectures, as described in the Ricoh case, was a series of processes seeking for the most fit product architecture for the digital MFP. In the case of the ASAP architecture, although the CPU capacity was insufficient, an architecture in which multiple applications operate on one OS was selected. To fully enable this architecture, it was necessary to sort out firmware components that could be standardized by close mutual coordination between functional divisions, but the attempt failed. Then in the following NAD architecture, the design reverted to an architecture that excluded close mutual coordination between divisions. We can say that these facts suggest that the degree of "the differentiation" between groups was high, but the

[8] Mutual coordination between groups were lacking, and meaningless source codes were generated. Furthermore, because many man-hours were necessary to execute the integration tests between the firmware components, people were inclined to seek a way of not having to change the architecture that has proved good in the market place. Therefore, the product architecture has not been reviewed thoroughly.

degree of “the integration” was low (Lawrence & Lorsch, 1967). In other words, when the ASAP architecture was developed, the organization at Ricoh was fixed to the traditional differentiation and integration that were copy-function centered. Within that organization, they were incapable to make necessary mutual coordination between functional divisions that was required to realize the intended product architecture. In contrast, when the following NAD architecture was developed, a product architecture that reflects the existing organizational structure was chosen. Taking a lesson from these failures, the GW architecture finally succeeded in realizing an architecture fit for multi-functionalization. This, involved a major update in CPU capacity that enabled parallel operation of multiple applications, and an organizational shift in which an independent team produces close mutual coordination between functional divisions.

The case discussed in this article can be interpreted as follows: Although there were already two architectures that Ricoh could have adopted, a lack in hardware performance made only one of them technically possible. However, the following interpretation seems more likely to explain what was really taking place behind the decision making at that time: The wavering between two firmware architectures was observed because in the days when each architecture was developed, there were no clear standards to decide which was good. Thus, they could but seek the answer through trial and error until they found the ideal organizational structure that realizes coordination between different groups. Therefore, the product architecture that an organization “should choose” reflects the organizational structure, product concept, and available resource at that time. This selection process cannot be fully explained within the framework of the existing argument that “an organization adapts itself to a given change in the product architecture.”

4.2. Lessons from the case of Ricoh

The copier had been the main business of Ricoh for a long time that maintained a high market share since the analog copier, and development organizations were lead by the copier division. As a result, a copy-function centric architecture was conceived. It seems that many companies face the same problem when they try to develop a new product architecture suited for multi-functionalization, because the ongoing core business limit their way of thinking. What kind of countermeasures should these companies adopt?

From this case study, it is suggested that a team of engineers specialized in each function needs to coordinate between groups in developing a firmware architecture suitable for multi-functionalization. To this end, first, it is necessary to spend resources to develop the firmware architecture in large quantities. Second, the product concept must be reworked without being bound by the philosophy of the core business. Finally, a hardware development centric organization needs to be rejected, and engineers (both hardware and firmware) with technical knowledge about each function should work in close coordination.

5. Conclusion

5.1. Summary

In this article, we investigated a case involving the firmware development of a digital MFP. By digitalization and introduction of the firmware, the original single-function analog copier evolved into an appliance that added several functions to mere copier, and then to the sophisticated digital MFP offering variety of functions. During the process, the firmware architecture went back and forth between two opposed conceptions. The case revealed the process for a generation and selection of a product architecture and the difficulties faced by a company in the development of the firmware. Furthermore, we were able to present a solution to overcome the problem. As for the distinct problems in the development of a digitalized product, some adverse effects were shown from the hardware-centric nature of the development organization, along with the importance of action to integrate the knowledge that lay scattered through the organization, which is necessary for the development of a firmware.

Because a common application interface was built at the OS and middleware level, and the independency of each functional elements was maintained, the development activity for the ASAP architecture and the GW architecture can be regarded as a kind of modularization. The important point is that to drive modularization with an unconventional orientation, it is necessary to create a new specific component that manages the interdependencies between the components. And also the organizational architecture must be an integral type with a knowledge (integrated knowledge) to grasp the interdependencies from the total optimization point of view. This presents the following dilemma: if an organization is going to run a development activity and operation under the existing modular type architecture, it is necessary for the organization to become modular (Aoshima and Takeishi, 2001; Fine, 1998; Fujimoto, 2001; Kusunoki and Chesbrough, 2001; Sanchez and Mahoney, 1996; von Hippel, 1990). However, to advance into a new form of modularization, the integrated knowledge held by an integrated organization is necessary. Organizations face this dilemma not because the product architecture changes, but because organizations change the product architecture at their will.

5.2. Toward competition based on the capability of firmware architecture development

In the product area affected by digitization, the competitive focus of companies has shifted from the hardware to the firmware in regards of architectural excellence. Conventionally, the makeup of an organization was based mainly on the hardware development, and the firmware development was treated as trivial. Now, the firmware grew in capacity and sophistication, so that the firmware design decides the whole product architecture. It is important to integrate the know-how and knowledge accumulated by different development groups in the firmware design, and an organizational framework that enables this is crucial. This will become more difficult when the area of the integration activity spreads across different companies. Furthermore, it is necessary to not only cope with the observed changes in the product architecture, but to also build a competitive strategy to develop an efficient product architecture. A firm should create a competitive circumstance that is advantageous to the company by designing product architecture proactively.

5.3. Limitations and future research

The argument in this article is based on a single case study so it is unclear to what extent the results should be applied to other companies and industries. Thus, in future research, it is necessary to conduct comparative studies of competing firms in the same industry to investigate the differences in firmware architectures, development organizations, and product strategies between companies, and the factors that produce these differences. In addition, comparative studies across industries, as for example in the automobile industry and the copier industry, will be necessary. Furthermore, we wish to investigate the relationship between the firmware architecture and the product development performance of manufacturers in future research.

Acknowledgements

The author would like to thank Junjiro Shintaku, Hirofumi Tatsumoto, Koichi Ogawa, and Tetsuo Yoshimoto for comments on earlier drafts. The author would also like to extend sincere gratitude to engineers at Ricoh Ltd. for research cooperation.

This article is a translated version of an original article published in Japanese as Fukuzawa, M. (2008) Seihin akitekucha no sentaku purosesu, *Soshiki Kagaku, 41* (3), 55–67, with minor revision.

References

Aoshima, Y., & Takeishi, A. (2001). Akitekucha to iu kangaekata [The concept of product architecture]. In T. Fujimoto, A. Takeishi, & Y. Aoshima (Eds.), *Bijinesu akitekucha* [Business architecture: Strategic planning of product, process, and organization] (pp. 27–70). Tokyo: Yuhikaku. (In Japanese)

Baldwin, C. Y., & Clark, K. B. (2000). *Design rules, Vol.1: The power of modularity.* Cambridge, MA: MIT Press.

Fine, C. H. (1998). *Clock speed.* Cambridge, MA: Basic Books.

Fujimoto, T. (2001). Akitekucha no sangyoron [The architecture based analysis of industry]. In T. Fujimoto, A. Takeishi, & Y. Aoshima (Eds.), *Bijinesu akitekucha* [Business architecture: Strategic planning of product, process, and organization] (pp. 3–26). Tokyo: Yuhikaku. (In

Japanese)

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*(1), 9–30.

Kato, H. (2002). Mojurariti doraiba [The modularity driver]. *Akamon Management Review*, *1*(8), 633–641. http://www.gbrc.jp/journal/amr/AMR1-8.html (In Japanese)

Kusunoki, T., & Chesbrough, H. W. (2001). Seihin akitekucha no dainamikku shifuto: Bacharu soshiki no otoshiana [The dynamic shift of product architecture: The problem of a virtual organization]. In T. Fujimoto, A. Takeishi, & Y. Aoshima (Eds.), *Bijinesu akitekucha* [Business architecture: Strategic planning of product, process, and organization] (pp. 263–285). Tokyo: Yuhikaku. (In Japanese)

Lawrence, P. R., & Lorsch, J. W. (1967). *Organization and environment: Managing differentiation and integration.* Boston: Harvard University Press.

Ricoh Corporate History Editing Committee (Eds.). (1996). *Road to IPS Ricoh: 60 years of technology*. Tokyo: Ricoh Ltd.

Shibata, T., Genba, K., & Kodama, F. (2002). *Seihin akitekucha no shinkaron* [Evolution of product architecture]. Tokyo: Hakutoshobo. (In Japanese)

Shintaku, J., Ogawa, K., & Yoshimoto, T. (2006). Hikaridisuku sangyo no kyousouto kokusaiteki kyougyou moderu [Competition in the optical disk industry and international collaboration model] In K. Sakakibara, & S. Koyama (Eds.), *Inobeshon to kyousouyuui* [Innovation and competitive advantage] (pp. 82–121). Tokyo: NTT Press. (In Japanese)

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal*, *17*(S1), 63–76.

Tatsumoto, H. (2002). Sofuto kaihatu purosesu moderu to seihin zokusei [The process model of software development and product attribute]. *Akamon Management Review*, *1*(4), 309–336. http://www.gbrc.jp/journal/amr/AMR1-4.html (In Japanese)

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, *24*(3), 419–440.

von Hippel, E. (1990). Task partitioning: An innovation process variable. *Research Policy*, *19*(5), 407–418.

[Received September 29, 2008; accepted October 6, 2008]

ABAS: Annals of Business Administrative Science
Vol. 7 December 2008

ABAS: Annals of Business Administrative Science is published annually by Global Business Research Center, Marunouchi, Chiyoda-ku, Tokyo, JAPAN. http://www.gbrc.jp