Reconfigurable Modularization and Customer Engagement: Looking for a New PLM in an Age of Diversification and Personalization

Shuichi Fukuda

Reconfigurable modularization and customer engagement: Looking for a new PLM in an age of diversification and personalization

Shuichi Fukuda

System Design and Management,
Keio University, Yokohama, Japan
shufukuda@gmail.com

Abstract. It is pointed out in this paper that current PLM is built up on traditional hardware development style of fixed functions with fixed morphology. But changes of environments and situations are so frequent and extensive so more attention must be paid to adaptability. Reconfigurable Modulation (RM) is expected to be one of the most versatile and useful approach. It not only enhances adaptability, but it also enables customer engagement in design and manufacturing. Thus, although traditional hardware development is focused on one time product value, RM directs processes to generate values, which grow with time. Further, RM will bring forth win-win relations between experts and non-experts, between low and high technologies, and between advanced and developing countries.

Keywords: Reconfigurable Modularization, Product with Growing Value, Customer Engagement, Process Value, Self-Actualization

1 Introduction

This paper begins with comparison between hardware development and software development and observe how they are different. What is important in software development is that the functions and the values of their products grow with time and with customers. Thus, they are developing lifetime values. Hardware development has been focused on one time value of a final product and how we can adapt our products to changing environments and situations are not so much discussed or explored. Rather, efficiency and higher functions have attracted major attention in hardware development.

As environments and situations change so often and so extensively, hardware developer should learn a lesson from software developers. To achieve adaptability

and growing functions and values, it is pointed out in this paper that changeable modularization is a first step toward this goal and then we can go one step further to reconfigurable modularization. Then, hardware developers can also grow values of their products and secure lifetime customers.

2 Hardware and Software: Their Difference

Hardware is developed with fixed functions. It is developed to satisfy the design requirements and once it is completed and delivered, it starts to degrade as shown in Fig.1.

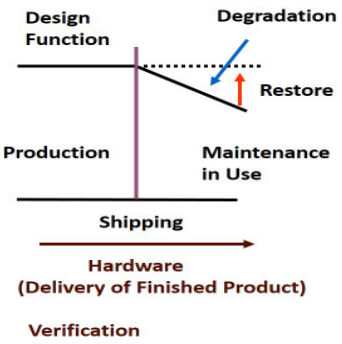


Fig. 1. Hardware development

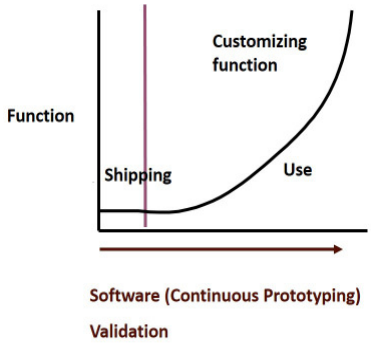


Fig. 2. Software development

Software used to be developed in the same way as hardware. But it was soon realized that software and hardware are completely different. Hardware is physical, but software is non-physical. And with the help of progress of software programming languages, software changed their development style to such a style as shown in Fig.2.

In this style, basic functions are provided first and when customers get used to the system and feel confident, little bit higher functions are added. Thus, functions grow or evolve with time and with customers. This style of development is called continuous prototyping, but this name is somewhat misleading. It is developing not prototypes, but products. So it would be better to call it growing or evolving product development. So this name will be used hereafter.

What is important in this growing or evolving development is that as customers feel confident, they put trust in the system. The more confident they become, the more trust they put in the system. If we look at the curve in Fig. 2, we will realize that this curve is nothing other than a learning curve. We learn to grow. And the more we learn, the more confident we become. Interestingly enough, confidence and trust are called by different names in English, but in German, both are called by the same name

Vertrauen. The basic nature of confidence and trust is the same. Confidence is toward yourself and trust is toward others.

And when we learn and grow together with the system, we are gradually attached to the system. The more time we spend with the system and the more we get used to it, the more attached we feel to the system. Thus, this development style changes our customers to lifetime ones. What is important in software development is that it is creating lifetime customers and with adequate additions of higher functions, customers trust increases and accordingly the value of the system increases. Thus software development style is in other words, value growing development.

On the other hand, hardware development is focused on a one time value at the time of delivery. And hardware is physical so that once it is completed, it immediately starts to deteriorate. In fact, if we Fig.2 upside down, it is nothing other than hardware development figure. So while the product value grows with time in software, it decreases in hardware. It is no exaggeration to say that all the benefits of software development correspond to demerits of hardware development.

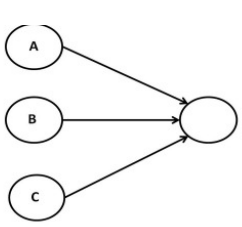
2. Value Growing Hardware PLM

Most PLM discussion about hardware are based upon the current style of hardware development. Then, aren't there any ways to turn the tide and make hardware development and its lifecycle management value growing activities? This is the issue which will be discussed here.

3. Modularization

If we compare hardware with software, hardware may be said to be developed in a unified manner. Or in other words, it is tree-structure based. This is because the set of final functions are pre-determined and all the efforts are paid to achieve this goal.

On the other hand, software development is network based. You combine different sub-systems and come up with a system you want. This difference may be described as convergent vs. divergent. Convergent approach is shown in Fig.3 and divergent approach in Fig.4. In a convergent approach, you apply all resources to achieve the goal. So it is goal-driven. A divergent approach, on the other hand, starts from where you. This is the way of thinking President Theodore Roosevelt said "Do what you can, with what you have, where you are". You look for a goal where you can reach with your current resources.



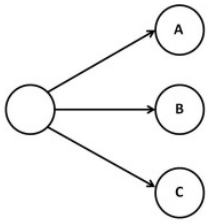


Fig. 3 Convergent approach Fig. 4. Divergent Approach

In hardware development, many different technologies are mobilized toward a fixed goal, i.e., to realize the final product that meet the design specifications. But in software development, there are many ways to connect subsystems so you can come up with different systems. Software is non-physical so it does not deteriorate. Therefore, how we can utilize legacy is a big issue. We cannot build a system every time from scratch, so how we utilize COTS (Commercial off the Shelf) is another big issue. So it has been important in software how a system can be modularized. Modularization in software may be compared to Lego. By putting different Lego pieces together, we can come up with a different toy. Lego is a typical divergent activity.

In hardware, too, modularization is attracting wide attention. But their objectives are different. They are looking for common parts to share among different product models to reduce cost, time and to increase efficiency. But there development style is still tree-structured.

4. Self-Reconfiguring Modular Robot

Then, how can we introduce an idea similar to software so that hardware products become more adaptive to the changes of environments and operating conditions? Self-Reconfiguring Modular Robots give us a hint. It is being developed to provide a robot with adaptability. Self-Reconfiguring Modular Robot may be defined in such a way as "Beyond conventional actuation, sensing and control typically found in fixed-morphology robots, self-reconfiguring robots are also able to deliberately change their own shape by rearranging the connectivity of their parts, in order to adapt to new circumstances, perform new tasks, or recover from damage" [1].

Most of hardware products, however, are fixed in morphology. And the same holds true with most machines. Their motions, sensing and control are fixed. Current PLM of hardware products are built upon this traditional hardware design. Although modularization is attracting wide attention in hardware industries, too, their focus is how they can share the components/parts among different product models to reduce cost and to increase productivity and it is not to adapt to the changing circumstances.

5. Reconfigurable Modularization (RM): Difference between Robots and PLM

But what happens if we introduce reconfigurable modularization into PLM? The big difference between robots and PLM is that in the case of a robot, all are automated. The robot itself reconfigures to be able to adapt to the environment to work there. Here the reconfiguration is carried out by us, engineers and by our customers. And in the case of a robot, in order to easily control and manipulate the modules, homogenous modules are used in most cases. i.e., all modules are identical

so that any shape can be produced by managing the configuration. But in PLM, units or elements are heterogeneous in most case.

6. Changeable Modularization (CM)

CM can be more easily understood if we see the new design of Daihatsu Copen (Fig. 5 and Fig.6)

Fig. 5. Daihatsu Copen

Fig. 6. Daihatsu Copen

It allows us to change doors, etc. as we like. Such changeable components would introduce Car Code just as we do with Dress Code. We can enjoy combining components or parts to best suit to the situation. CM has such a benefit of attracting customers, but it has other benefits, too.

One important one is that we do not have to produce too many final products, i.e. cars here. We can produce a common platform and at the same time produce a wide variety of option modules. What is good is that we can mass-produce common platforms and option modules, but we can come up with a wide variety of product models.

Hardware engineers have to been trying to produce a variety of final products as wide as possible in order to cope with diversifying requirements. But if we produce final products with integral structures, there is a great risk of large amount of unsold inventory of final products. But if the module are changeable, then inventory turnover will become far better because customers may take one module today but use another one tomorrow. So customers buy more modules than the current way of selling final product. They do not have to choose one out of many, but they can buy several modules for different situations and combine them appropriately or as they like. Therefore, the inventory turnover will be improved greatly and further, component producers, too, can mass-produce their products. Thus, their profitability goes up.

7. Benefits of CM

(1) We can separate the common modules and the adaptable modules. The common

modules are not only common among different product models, but it can be used for a long time. The same module can be used for all purposes. Thus, the common modules can be mass-produced so that the common module producers can make profits more easily.

(2) The changeable modules can easily deal with such problems as wear, deterioration, etc. which take place because hardware is physical. We can change the module with the new one or we can remanufacture the module so that it keeps the best working conditions. We should note that remanufacturing is attracting wide attention these days, because remanufacturing more often than not makes greater profits than manufacturing new products. Therefore, producers of changeable modules would enjoy making profits.

And what is more important is that in remanufacturing, they have to remanufacture item by item separately. But if we introduce CM, then they can manufacture such quickly deteriorating modules in mass to prepare for replacement.

(3) By separating common modules from changeable modules, we can design longer life for common modules and we can design appropriate length of life for changeable modules. In short, we can assign necessary or desirable length of life for each module.

Thus, product lifecycle management will become lifecycle managements of modules and it becomes important how we assign lifecycles to modules.

.

(4) Thus, CM will bring forth great benefits to the producer. But what is more important is that it will enable customers to get involved in design and manufacturing. Customers, as the word indicates, would like to customize their products. But currently what they can do is just to choose one from many that are offered. Customer requirements are diversifying, but it does not mean that the range of their choices are widening. What customers really want is to customize their products their way.

But currently they are regarded just as mere passive consumers or end-users. They are not allowed to be a player in the game.

If we introduce CM, we could design changeable modules to allow the customer involvement. Some modules are not so much important or crucial for product functions. So we can leave their design and manufacturing to our customers. They can enjoy designing and manufacturing them.

3D Printing Technology or Additive Technology can serve a great deal for them. They can really fabricate the module personally by themselves. This will increase the whole value of a product and they will be attached to the product and they will use it much longer.

This can be compared to the dresses and accessories. The same dress, but different accessories not only adapt to different situations, but the task of combining them appropriately gives the joy of creation on the part of the customer. This is because such actions create values. Hardware engineers have been discussing value only in terms of a final product, but this indicates processes create values no less than products [2], [3], [4].

(5) Thus, we can design our products in such a way that we leave long enduring modules to experts and we can assign short enduring modules to experts or non-experts according to its requirements.

And it should also be added that we can mix low technology and high technology in an appropriate manner so we can assemble products using local workforce in developing countries who are not so proficient. Thus we can increase employment opportunities in developing countries. And we can assign the production of some changeable modules, which do not need so much proficiency to developing countries. This will increase their employment, so that we can grow our market.

It should be stressed that the above discussion of CM is not related to configuration management in a straightforward manner. In most cases, configuration management focus on maintaining the initial design goals, but CM discussed here focus more on fast or quick adaptability to the changing environments and situations. Another big difference is that CM is expected to increase value of a product, or even more to grow value during product lifecycle with active engagement of customers. Configuration management is not.

8. Reconfigurable Modularization (RM)

RM goes one step further beyond CM. In CM, the basic morphology will stay the same all throughout product lifecycle. But RM changes its morphology in order to quickly adapt to the changing environment and situations.

Stewart Brand published a very interesting book “How Buildings Learn” [5]. He pointed out the buildings designed by very smart architects often do not survive, while those designed by less smart people more often than not survive for much longer time. He explains that because buildings designed by smart architects are often too much goal-driven and narrow-focused. Their ideas are excellent, but when the environments and situations change, their excellence sometimes changes into demerits. Those buildings developed by less smart people are more often than not so sharply focused. This introduces greater adaptability so they allow for wider and flexible use. He shows London Docklands (Photo 1) as an example. This is another good example of RM.

Photo. 1. London Docklands

Containers are designed and developed for another purpose, but if we combined them appropriately, we can use them as units of a building. And what is better still, we can reconfigure it whenever there is a need for change of morphology. This is the same idea as that of self-reconfiguring modular robot with homogenous units.

But we do not have to stick to the idea of homogenous units, although it might allow much flexibility and easiness for changing morphology. But our primary purpose is not to change morphology flexibly. Ours is to make PLM flexible to respond to the changes of environments and situations. And most hardware product cannot be easily divided into homogenous units, so we have to utilize heterogeneous units and combine them as needed.

9. Merits of Reconfigurable Modularization

Let us take Electric Vehicles (EV) for example. Unlike current automobiles, we can assemble parts and a motor as we like. Further if we design it as a frame structure, non-experts can build their own vehicles at their homes. Or even if it is a unified structure, such DIY technologies as 3D printing will help support personal fabrication. It is exactly the same as construction kits such as robots and toys. Although the parts are produced by experts, customers can feel they are manufacturing the product and this sense of involvement provides them with the joy of being a player in the game. So the product is evaluated much higher than when they receive the completed product. And just as in Lego, if many different morphologies can be generated by combing parts or components, customers feel like they are designing and manufacturing the product. This is the important benefit of customer engagement of RM.

Maslow proposed the hierarchy of human needs [6] as shown in Fig. 7.

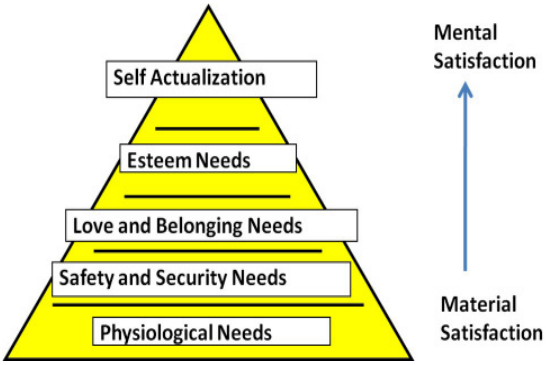


Fig. 7. Maslow's Hierarchy of Human Needs

At the lower level, we look for material satisfaction, but as we go up higher, our needs change from material to mental and at the top, we look for self-actualization. RM satisfies our needs for self-actualization. We have to remember that our needs changes from material to mental so that process value becomes increasingly important. RM changes our product value based hardware development to process value focused one.

10. Concluding Remark

This paper points out that we are now entering the age of mental satisfaction, so that PLM should be changed in this direction. Reconfigurable Modularization proposed here will be one of the most useful and versatile approaches to cater to this need and to adapt to the quickly and extensively changes.

References

1. http://en.wikipedia.org/wiki/Self-reconfiguring_modular_robot
2. Fukuda, S. (ed.): Emotional Engineering-Service Development, Springer, London (2011)
3. Fukuda, S. (ed.): Emotional Engineering, Vol.2, Springer, London (2013)
4. Fukuda, S. (ed.): Emotional Engineering, Vol.3, Springer, London (2014)
5. Brand, S.: How Buildings Learn: What Happens after They're Built, Penguin Books, London (1995)
6. Maslow, A.H.: A Theory of Human Motivation, Psychological Review, Vol.50, No.4, pp.370-396 (1943)