

# Recent trends of new product development and production management in Japanese electronics industries

*Toshio Itoh*

*Mitsubishi Electric Corp.*

*Akasaka Minatoku, Tokyo, Japan*

*Phone 03-5573-4635*

## **Abstract**

This paper describes a new technology for market-driven and rapid development which is carried out by making full use of advanced information technologies such as intranets.

## **Keywords**

Abduction, service-oriented technology, concept engineering, concurrent engineering, collective genius, intranet

## 1 INTRODUCTION

Today's rapidly changing markets and technologies require rapid and market-driven development of new products. This paper suggests a new technology addressing this need. In this technology, development is carried out by a collective genius linked over long distances across organizational boundaries with new information technologies. Specialists within the collective genius devise creative ideas by abduction and carry out developments of new products integrating these ideas by service-oriented technology, concept engineering, and concurrent engineering.

Needless to say, the effective utilization of information technologies is indispensable for today's product development and production management. It is the author's opinion that, among these technologies, those of intranets are most important by the following reasons:

- They cause a paradigm shift in the distribution of knowledge in an enterprise.
- They change the development process of new products and the ways of production management.

## 2 ABDUCTION

### 2.1 What is abduction ?

Abduction is a way of thinking. The concept of abduction was introduced by C. S. Peirce (1839-1914), the founder of American pragmatism. It differs fundamentally from the conventional approaches which are carried out by using mainly deduction and induction. Different from conventional approaches that depend mainly on new combinations of existing knowledge, it consists of establishing original objectives, devising several hypothetical methods of the form "X leads to Y" to attain these objectives, and verifying or disproving these hypotheses.

Since induction and deduction depend mainly on new combinations of existing knowledge, they are suited to dealing with known facts and are largely independent of personal experience. On the other hands, abduction is suited to dealing with unknown problems, being largely based on personal experience and setting many transcendental hypotheses, as explained in the following section.

### 2.2 Transcendental hypotheses which are core of abduction

Contrary to the conventional approaches which uses validated hypotheses, abduction frequently uses transcendental hypotheses. There are three types of transcendental hypothesis:

- (1) Hypotheses that have not yet been validated, but for which a process of validation can be devised.
- (2) Hypotheses that are not susceptible to direct observation even in theory.
- (3) Hypotheses that assume the existence of things that have not yet been developed although they may be in the future.

Note : Types (1) and (3) are used in new-product development.

### 2.3 Abduction for product development

In applying abduction for product development, it is important to execute the following:

- to identify the functions desirable in the new product from the standpoint of marketing.
- to adopt transcendental hypotheses even if their prior non-existence has been demonstrated.
- to verify the feasibility of the transcendental hypotheses theoretically and experimentally through the development process.

### 2.4 Salient features of abduction

- (1) Abduction is unique among thought processes in that it supports creative exploration of the unknown. Neither deduction, with its strict logical pro-

gression, nor induction, with its requirement for a large body of evidence to support a hypothesis, allow a thinker to freely explore the unknown.

- (2) Abduction starts with easily available reference materials without waiting for a thorough elucidation of unknown factors or even for all the relevant facts to come in. Naturally, the larger the pool of available knowledge, the better this process will operate. Information technology supports abduction because it greatly facilitates the use of available information resources.
- (3) Abduction can proceed earlier and faster than induction because it does not require a large body of verified facts. Please note that this makes it an ideally suited to today's fast-changing Internet/intranet environment.
- (4) The disadvantage of abduction is its vulnerability to error compared with other approaches where facts are painstakingly assembled beforehand. However, the disadvantage of time loss by the other approaches is greater than the vulnerability of abduction.
- (5) Abduction constitutes a progressive, open-ended process giving the thinker latitude to incorporate numerous different points of view.
- (6) Abduction can lead to more useful hypotheses if we try to find them in other fields of academic and industrial importance. Please note that by using intranets, multiple individual specialists engaging in abduction can be linked across divisions of organization, discipline and culture, creating a space for the phenomenon of collective genius to emerge.

### 3 SERVICE-ORIENTED TECHNOLOGY ?

#### 3.1 What is a service-set ?

To promote the market-driven approach, development of a product should be oriented to the preferences of the individual consumer. For ease of this development, we are introducing a new technology called "service-oriented technology" which satisfies the following conditions:

- (1) To be suitable to design new products which include many semiconductors.
- (2) To be suitable for both hardware and software development.
- (3) To be comprehensible to all those involved, from top management to the individual researcher.

Service-oriented technology regards any activities like the above in terms of three vital elements: an object which performs a requested service, the request (order or direction) / events which invoke the service, and the contents of the service. We call these "object," "message" and "service," and the set of {receiver, message, service} a "service-set." Please note that the object in the service-set has the same concept as the object in object-oriented technology and is subject to many notions related to the object in the technology such as class, inheritance, polymorphism, framework, and so on for the reuse of existing service-sets/objects.

In service-oriented technology, we start to design a new product by describing

the service-sets of the product which are visible to its users. We call these service-sets “ visible service-sets” because users should understand them when they use the product.

In the development of a compact disk player, for example, we start to design a CD player by its visible service-sets as follows:

The service-sets of a CD player

- ① {**CDPlayer *PowerOn*** ^Power turns on}
- ② {**CDPlayer *MotorOn*** ^Disk rotates without sensible wow}
- ③ {**CDPlayer *SelectTrackNumber*** ^Selects track only by key selection}
- ④ {**CDPlayer *Play*** ^Play without scratch noise in dynamic range of 95dB}
- ⑤ {**CDPlayer *VolumeUp*** ^Amplify sound signals to the specified level without sensible distortions of phase and amplitude}
- ⑥ {**CDPlayer *Stop*** ^Stop disk rotation automatically}
- ⑦ {**CDPlayer *PowerOff*** ^Turn off power supply}

In the service-set of {**CDPlayer *PowerOn*** ^Turns on power supply}, for example,

- ‘**CDPlayer**’ is an object
- ‘***PowerOn***’ described by bold Italic letters is a message
- ‘**^Turn on power supply**’ following the caret mark describes in verb-object combination the service that the object executes.

### 3.2 Hierarchical structure of service-sets

While the service-sets from ① to ⑦ are all visible to users, they are also implemented by a number of other service-sets not shown which are invisible to users. We call these service-sets “invisible service-sets.”

Now, the visible service-sets of the CD player are performed through the coordinated activity of its parts/sub-systems such as a remote control unit, microprocessor, power supply, disk, pickup, and audio system. For example, the service-set ⑥ is realized by the coordination of the service-sets of the many parts such as the remote control unit, the micro-processor, the pickup, the disk and the power supply. Therefore it is described hierarchically as follows:

- ⑥ ⊃ {**CDPlayer *Stop*** ^Stop disk rotation}
  - {**Remote *Stop*** ^Send infrared signal to stop play}
  - {**MicroProcessor *Stop*** ^Send signal to move pickup to track 0}
  - {**Pickup *MoveToTrack0*** ^Seek and move to track 0}
  - {**Disk *RotationStop&InformPowerSupply*** ^Stop disk rotation}
  - {**PowerSupply *RotationStop*** ^Tell PowerSupply disk rotation stop}

Note that while the object of visible service-sets is the product itself, those of invisible service-sets are its parts / sub-systems.

Generally, the invisible service-sets themselves are performed through the coordinated activity of their parts/sub-systems too. This means that service-sets

comprise a hierarchical structure with visible service-sets at the top. We refer to this as the service-set hierarchy; a single service-set made up of multiple service-sets is called a superset, and the sets that together comprise a superset are called subsets.

## 4 CONCEPT ENGINEERING

### 4.1 External functions of a product

Determining the services of visible service-sets of a product is the most important part of concept engineering, because users do not buy a product for its hardware or software but for the service it returns in response to the request (message) of the user. Note that it is *not* the particular hardware or software but the *service* that they need. They buy a telephone because it returns the service of enabling conversations with those geographically remote, a washing machine because it returns the service of cleaned clothes, etc. In the case of a product, its services take the form of the functions it provides to the user and we call these “external functions.”

### 4.2 Discontent analysis

The most effective method for analyzing the external functions is discontent analysis. Every existing product has external functions with which its users are discontented. For example, I could list my feelings about the past LP audio system.

Stylus replacement is troublesome.

Record surface noise is too high.

Unacceptable distortion occurs in the amplification process.

Stereo channel separation is poor.

Selection of individual musical items on a disk is difficult.

The dynamic range is too restricted.

In the analysis, the analysts determine discontent with existing products that are similar to the intended product using one or more of the following methods:

- They may engage in direct dialogue with customers.
- They may try to derive the users' discontent by becoming virtual users themselves. (Consumer goods such as home appliances and personal computers are often used by the analyst in person. This can be effective provided the analyst successfully emulates a true user, ignorant of the product's technological aspects.)
- Questionnaires surveying customers and dealers can also reveal discontent with existing products. And finally, traveling the sales routes can reveal discontent from conversations between the analyst, dealers and salespersons.

### 4.3 Design of external functions

Generally, discontents cannot be derived systematically. This makes it necessary to arrange them systematically before we can use them in the design of visible service-sets with external functions without the discontents. I would like to suggest how external functions can be arranged systematically from the discontents.

Before attempting a systematic arrangement, all discontents should be restated as the converse ideal functions. Then, analysts can collaborate with design engineers to describe the external functions thus derived in a systematic way. It is recommended to express functions in a verb-object combination, occasionally modified by phrases that specify the function more precisely. Analysts must incorporate their own ideals for the functions of the intended product, or the subsequent phase of designing functions will amount to no more than an improvement to an existing product. Some examples follow of such descriptions of external functions.

#### A Classified Description of the External Functions of a Compact Disk

##### ① ⊣ Recorded Sound

○ Record signals in dynamic range higher than 95dB ← Dynamic range is too restricted.

##### ② ⊣ Music Selection

○ Select pressing a button ← Selection of music items on a disk is difficult.

##### ③ ⊣ Sound Reproduction

○ Detect signals without physical contact ← Surface noise is too high.

○ Amplify signals without phase distortion ← Channel separation is poor.

○ Amplify signals linearly ← Distortion occurs in reproduction.

In the above table, the discontents are given to the right of the arrows ←, and the ideal functions derived by restating them are described on the left.

Once we arrange functions of a CD player systematically as above, it is easy to design its service-sets as shown in section 3.1.

### 4.4 Architecture design

In the next stage, design engineers strive to devise the invisible service-sets whose objects are various parts/sub-systems which realize the external functions by implementing their services/functions collaboratively. These invisible service-sets, too, are collaborative combinations of activities of their sub-sets and themselves. In doing this, we should strive to reuse the service-sets of existing parts. It is important to be persuaded of the effectiveness of the reuse of existing parts. If we could not find reusable service-sets, we should use abduction to create a new service-set setting the realization of the function of its superset as its objective and devising transcendental hypotheses. We call this activity “architecture design” of a product.

Thus, in the architecture design, the visible service-sets are expanded hierarchically down to either the existing service-sets or the new service-sets which will be

developed by the concept engineering explained in the following section.

Through the above processes, the concept of a new product is visualized.

#### 4.5 Value analysis of product concepts

To evaluate the value of product concepts developed in these ways, follow the steps given in the following, then re-examine the architecture design in the light of your findings. This process should be iterated until a conclusion can be reached.

(1) First, suggest a sales strategy and estimate total system costs. This process is broken down into the following steps:

- Cite examples of similar products.
- Perform market segmentation for these products.
- Identify the market segment of the intended product.
- Determine the sales promotion appropriate for this segment.
- Estimate market share in this segment and forecast the sales amount.

The estimates of sales amount are particularly important for electronic products with custom-made ASICs, because the cost of the ASICs decreases in direct proportion to the number of items produced.

(2) Next we should estimate a viable sales price. This process breaks down into the following steps:

- Investigate the prices of existing products with external functions similar to those of the intended product.
- Estimate a viable price from the prices surveyed, taking into account the fact that the viability of a price is not affected by internal functions hidden from users nor by external functions in which users have no interest.

(3) To estimate the function cost is important as the third step. The function cost is the lowest cost for which a product with given external functions (visible service-sets) can be produced and sold. Needless to say, given the functions, we will go for the lowest cost, and it is the function cost that determine the lowest possible price at which we can sell.

To get the function cost of a new product, iterate the following process against the architecture designs to obtain the function cost until the function cost is lower than the viable sales price (minus target profit and the cost of sales).

Estimate the cost of reusable objects first. Then estimate the cost of new objects to be developed. As they may include transcendental hypotheses, estimation is difficult. However, rough estimates may be made using the following method.

For hardware, use conventional value analysis (VA) methods to conceive a production method and determine the relationship between the number of items produced and the production cost, particularly in the case of mass-produced products.

- Estimate the cost of materials and purchases.
- Estimate development cost and divide by the number of items to be sold.

For software, estimate development costs and divide by the number of units

to be sold.

## 5 CONCURRENT ENGINEERING

In the next step, we should carry out the following activities:

- To design the method of the service in the service-set of a product whose service and messages have been designed in the concept engineering.
- To develop parts and/or subsystems for the hardware and programming for the software.
- To assess the validity of the many transcendental hypotheses conceived during concept engineering, often giving rise to further new hypotheses.

To complete these activities as speedily as possible, we are promoting concurrent engineering. Concurrent engineering is a technology for rapid development in which we divide a development project into a number of sub-tasks and perform them simultaneously to the maximum extent permitted by available resources.

Please note here that it is vital to select the more important themes from among many proposed before implementing concurrent engineering so that as many people as possible may co-operate in their development. Without allocating finite human resources on the basis of careful selection, the effectiveness of concurrent engineering is seriously impaired.

### 5.1 Partitioning the product to identify sub-tasks

After the selection, you should begin to make scheduling of development and allocate necessary resources. This involves identifying the various super-tasks to be performed, and breaking these down into sub-tasks. It also means identifying the milestones that mark completion of each super-task. Only then is it possible to allocate the resources necessary to complete the major tasks. Here, a task is any development activity that produces a clearly identifiable result.

Partitioning the product is required to identify sub-tasks before determining super-tasks. Generally, the hardware configuration and/or the system architecture of a product can be divided into a number of partitions reflecting the parts/subsystems and technologies to be utilized in the product. In this context, a partition can be any set of functions that delineate a sub-task. The process of partitioning the project enables sub-tasks to be identified and set up corresponding to the partitions. Partitions for a CDPlayer follow:

**A CDPlayer:** Remote controller, Microcomputer, Signal pick-up head, Audio amplifier, Power source, Drive motor, Speaker, etc.

### 5.2 Splitting a super-task into sub-tasks

In concurrent engineering, sub-tasks are identified by developing a set of partitions, as above, and are performed according to the overall schedule. Concurrent engi-



neering is distinguished by performing these sub-tasks simultaneously and in parallel to the maximum extent permitted by available resources.

Before concurrent engineering begins, the project leader divides each super-task into a number of sub-tasks for its effective implementation of concurrent engineering, based on the foregoing partitioning. The project leader then co-ordinates the tasks, allocating resources for each individual sub-task and assigning due dates for completion. For certain sub-tasks, it is important to obtain the participation of other companies who are the leading experts in the technologies involved in those sub-tasks. Such co-operative development with other companies in this way forms what has been called a "virtual corporation."

### **5.3 The process of development**

Development using concurrent engineering is composed of both iterative and incremental development, as explained below.

#### **(1) Iterative Development**

Iterative development is used in the validation of the transcendental hypotheses associated with the sub-task: the results are reviewed and the hypothesis modified as necessary, with this process being reiterated until a conclusion is reached.

When the tasks of iterative development are performed simultaneously, in parallel, the transcendental hypotheses can be rapidly validated and errors eliminated while getting feedback from different experts and users and making improvements.

We have two sayings that encourage the use of this pragmatic approach: "The road to the truth is lined with mistakes," and "If you don't make mistakes, you won't make anything."

The word iterative is used to legitimize the abandonment of any rigidly pre-determined order in performing tasks. It is recommended that the order should be changed flexibly as dictated by the current situation.

#### **(2) Incremental Development**

As its name suggests, incremental development proceeds in small increments to produce effective results quickly. Here, too, there are sayings that suggest its applicability: "Do not attempt to perform a difficult, complex task in a single leap. Take it one step at a time, and check your progress every step of the way."

For incremental development, the partitions must be defined so clearly that as each associated sub-task is completed, the partition itself can be tested and validated, along with its interface with other, already completed, partitions. Note that the assessments of each step of the development are bound to have an effect on the later sub-tasks.

## 5.4 Prototyping

Prototyping, within the development process, is the creation of a preliminary, incomplete and perhaps scaled-down version of the intended product. It is an important means of showing users and sometimes colleagues working concurrently on key parts of the system and getting their feedback before development is complete.

In the era of analogue circuitry on printed-circuit boards, the lead time for prototyping electronic circuits determined the lead time of the whole development, which typically took from one to two years. Recently, it has become possible to create a prototype with much shorter lead times by making full use of computer systems.

Moreover, you can speed up prototyping by reusing existing service-sets/objects provided you use service-oriented technology in an appropriate environment.

Be aware that teams get excited if the first prototype works well, but are more likely to make a mistake in the next phase. Moreover, since speed is the key element of rapid prototyping, it is not always necessary for the prototype to be of high enough quality to ensure good long-term maintainability.

## 6 COLLECTIVE GENIUS

### 6.1 What is collective genius?

Organizations with collective genius are those in which the top specialists meld their individual expertise to fulfill their shared objectives. The concept of collective genius has a long history in the United States. The many specialists who shared NASA's goal of putting a man on the moon achieved spectacular success, and form a representative model of collective genius. We do well to note that a project quite impossible for an individual of genius becomes possible when the top specialists of many fields cooperate.

Traditional specialists have certain weaknesses. In most enterprises and research institutes, the technologies are divided up into too many tiny specialities, and each specialist functions as an individual "star."

A collective genius cannot be created just by collecting these stars and encouraging them to adopt a better attitude! The first prerequisite is to master the process of abduction using transcendental hypotheses. When those who form the collective genius have mastered abduction, you no longer get the defensive responses of the over-specialized experts of the past. Collective genius inspires aggressive, positive, independent, autonomous comments that are made from a wide perspective. Within the collective genius, the specialists treat each other's goals as transcendental hypotheses, and you can expect a synergistic effect from their interaction.

Another prerequisite for creating collective genius is to give the experts the

weapons they need—the cooperative working environments using advanced information technologies such as PCs linked by intranets. Deprived of weapons, they will be like the Japanese armies towards the end of the Second World War.

If, on the other hand, you use an intranet to link together specialists across organizations and over long distances, you *can* create a collective genius. Those who participate in the collective genius share a common database, use service-oriented technology as their common language, and effectively pursue common objectives. Service-oriented technology provides a language that promotes not only dialogue between the specialists from different fields who constitute the collective genius, but also communication with the manager and others responsible for direction. The workplaces that provide this kind of cooperative working environment are what we call information-oriented offices.

One final requirement for the creation of collective genius is a flat organizational structure. Only in this kind of organization can each specialist work unhindered towards an ideal. This is impossible in traditional hierarchical structures. The flat organization is like an orchestra in which top musicians perform under the leadership of a single conductor. The orchestra-type organization is the topic of the next section.

## 6.2 A flat, orchestra-type organization

In an orchestra, each member's personal contribution is of critical importance. The overall performance of an orchestra is a delicate mixture of superb, exquisitely refined performances by many different instruments. The conductor makes the whole greater than the sum of the parts, although a poor performance by a single player will seriously impair the whole.

The ideal organization for the collective genius that challenges the unknown is flat, like a symphony orchestra.

- (1) It pursues a common goal, and is not merely technology driven.
- (2) The project leader gives the lead in creating a new product, like the conductor of the orchestra, for he combines the individual abilities of his team members to create something that is greater than the sum of its parts.
- (3) Evaluation is based on actual performance, not potential.

The success or failure of a flat organization depends on how well the means of information transmission and exchange are implemented. The environment must encourage not only formal reports but informal exchanges of information about development activities and active dialog between experts. This kind of environment is found in information-oriented offices linked by the intranet, for it can collect these kinds of information and transmit them quickly and correctly. Moreover, it speeds senior management decision-making by increasing the speed of information transfer.

## 7 CONCLUSION

The ultimate benefit of intranets will be to harness the power of collective genius while on a practical level providing a powerful means to publicize the goals and targets of a project and to solicit broader participation causing the following shifts towards democratization:

- (1) From commands and controls towards self-determination of central issues and collaboration with others
- (2) From authority based on position to authority based on knowledge
- (3) From vertical communication to horizontal communication
- (4) From mistrust and obedience to trust and sincerity

This democratization engenders the collective genius and it is this collective genius that creates ideas by abduction and develops new products rapidly integrating these ideas by concept engineering, concurrent engineering by making full use of advanced information technologies.

## 8 REFERENCES

- W. H. Davis (1972), *Perice's Epistemology*, Martinus Nijhoff, The Hague.  
Toshio Itoh (1996), *A New Approach to Future Enterprises*, Ohmsha Tokyo.  
Toshio Itoh, Teruhisa Ichikawa, Hitoshi Ogata, Tadatoshi Yamada (1996), *Technology for Information-Oriented Society*.

## 9 BIOGRAPHY

Toshio Itoh received B.S and Doctor degrees in engineering from Kyoto University, Kyoto, Japan, in 1954 and 1962, respectively. Since 1954, he has worked for Mitsubishi Electric Corp.. He was a director from 1986 to 1996 during which he became successively the roles of General Manager of Manufacturing Development Laboratory, Central Research Laboratory, Corporate Engineering, and of Corporate R&D. Presently he is an adviser of Mitsubishi Electric Corp.