Requisites of Embedded Operating System for Network Electronics
(Extended Abstract)

Yasuhiko Yokote
Sony Corporation
6-7-35, Kita-shinagawa, Shinagawa-ku,
Tokyo, 141-0001 JAPAN
Yasuhiko.Yokote@jp.sony.com

1. Introduction

Today, Windows®-based (and UNIX®-based) PCs have ability to access to the networks at anytime and anyplace. The benefit of such an “always-connected” environment is expanded to non-PC users with new consumer products such as Airboard and CoCoon, which we call the network electronics (or NE) products in this paper.

An NE product has nature both of “embedded” and “open-ended,” and partitioned into two. The “embedded” means the software is designed and implemented in traditional embedded manners and optimized for a product to differentiate it from others. The “open-ended” means the software is designed and implemented for complying with open standards and sometimes updated after product deployment. It also has ability to download an application from Internet and run it on an NE product. These two partitions are inherently conflicted and need to be intervened by an underlying operating system, which is considered as the first role of an embedded operating system (Role 1).

In addition, an NE product is facing the threat of product commoditization. The “value” of an NE product is created by the feature realized by the “embedded” partition, as well as the services on the networks accessed through the “open-ended” partition. Avoidance of the commoditization threat is to increase the “value” of an NE product. The second role of an embedded operating system is to provide facilities to realize it (Role 2).

The product “value” is also produced by interconnection of two or more NE products. A single application runs on a set of interconnected NE products to create a new experience to users. The last role of an embedded operating system is to manage such a distributed environment (Role 3).

This paper discusses on these three roles of an embedded operating systems in the context of an NE product development, based on the author’s experience of Aperios deployment in the product development. Then, the 7-layer model for the NE software is defined to clarify these 3 roles on the software layer. Finally the paper is concluded that three layers are identified to create product’s “value,” and should be addressed in the context of the new computing for the network electronics.

2. NE Software 7-Layer Model

The author and his group developed the Aperios operating system to implement the feature of “evolution” in a consumer product. Aperios is the first commercial operating system embedded a “reflection” [1] mechanism at the system software level. It has some novel features including object-oriented system structuring, object migration, base/meta separation implementing reflection, etc. There are many published papers on Aperios, some of them can be found in [2, 3].

The three roles mentioned in the above have been address by Aperios. For example, Aperios provided a set of API (as a meta-space) compatible with typical real-time operating system to address role 1, so that existing open standards are easily ported on Aperios. It also provided open implementation [4] to efficiently embed unique features of an NE product in software to address role 2.

Lessons from experience of Aperios deployment revealed difficulties in behind of commercialization of the new technology. In the beginning of the 21 century, open source development has become popular and well known in product development. The source of product’s “value” has moved from where Aperios provided. The new approach for developing software of an NE product needs to be investigated.

The software of an NE product can be divided into 7 layers. The lowest layer is located just above of the underlying hardware and is the one for managing and controlling hardware resources. The next (2nd) is the layer for managing and controlling program execution on the above of this layer. Linux belongs to this layer. The 3rd layer is the one for security and protection. The 4th is the layer for communication. It manages any communication between
processes and devices. The 5th layer is for representing exchangeable data/information between processes and devices. XML is a typical technology belonging to this layer. The 6th is the layer for providing any services to applications. Java, 3D graphics, and an application launcher are typical technologies at this layer. The highest layer is the one for applications. Internet browser, EPG, and IM applications are typical ones at this layer.

The three roles of an embedded operating system discussed are investigated in terms of this 7-layer model. In this paper, three layers of it are focused to produce the “value” of an NE product: “resource management and control” (1st), “security and protection” (3rd), and “data representation” (5th). Other layers are necessary for complete software of an NE product, but these can be provided by many software vendors and no direct link to “value” creation.

In role 1, independence of two partitions, “embedded” and “open-ended” ones, should be guaranteed. Since software running in the “open-ended” partition is updated and/or upgraded after NE product deployment as well as a new application is downloaded from the network, it should never interfere with “embedded” partition such as to avoid virus infection, avoid malicious access, and cause total failure. The “security and protection” layer is responsible for it. In Aperios, these partitions were divided by a meta-space mechanism. However, it had no mechanism to securely prevent interference between them.

In role 2, several devices such as sensors and controllers are implemented on a single chip, or a single system LSI. An NE product is differentiated by the chip, and confronted with its commoditization. It is facilitated by the digital technology in the nature, and software makes it accelerated. Working software tightly with the chip is the one way to avoid it. The “resource management and control” layer is responsible for it.

In role 3, interoperability among two or more NE products should be guaranteed without any special operations. For example, assignment of the network address and negotiation of communication protocols are automatically achieved by an NE product itself without involving users. In addition, environment created by a set of the NE products is heterogeneous and requires appropriate scheme to exchange information among them. The “data representation” layer is responsible for it.

Some technologies necessary for each layer have already been developed. However, these should be commercialized in a consumer product, where many unresolved problems are waiting for their resolution, which research area is called network electronics computing.

3. Network Electronics Computing

The most crucial feature of the consumer products is “dependability” on the products, in other words reliability, security, and safety. Consumers do not like to see the blue screen that PC users can see when Windows® crashes. Consumer people never allow us to reboot an NE product.

Since an NE product is purchased by our customers and owned by them, it is never trespassed by others. In addition, any contents played on an NE product should be protected from illegal copy when contents creators want. Because of the nature of “open-ended,” its probability is higher than a traditional consumer electronics product.

The XML technology is considered as optimal technology for guaranteeing interoperability among NE products. However, it is originally designed from the server and desktop computers. The special care should be taken to adopt the XML technology effectively in an NE product. For example, the heap area should be carefully managed for parsing an incoming text from the networks.

The product developers should always consider the following three factors upon designing an NE product: performance, memory usage, and power consumption. The power consumption is recently prioritized, even for stationary products. They budget for these factors with the single measurement unit, i.e. cost. Unlike software developed for desktop PCs, the NE product developers always have to work with slower processors with limited memory under low power consumption to balance between cost and product’s features. It is a crucial limitation impacting on the NE product development. Therefore it should be studied in some disciplined manners, which we name the network electronics computing.

4. Conclusion

This paper discussed three roles of an embedded operating system for an NE product in the context of the NE software 7-layer model. Three layers are identified from the product “value” creation point. The NE computing is the new research domain that we have to study to create new experience to users with an NE product.

References