Embedded Systems and Open Source in Japan

- Current Status and Future Directions -

Tatsuo Nakajima
Distributed and Ubiquitous Computing Laboratory
Waseda University/
Japan Embedded Linux Consortium
tatsuo@dcl.info.waseda.ac.jp

Abstract:
In this short article, we introduce Japan Embedded Linux Consortium. Then, we show the impact of ubiquitous computing researches to embedded system industries, and show a brief overview of our research activities.

1. Linux and ITRON

In Japan, a lot of industrial embedded products has adopted the ITRON specification operating system(ITRON). ITRON is not an actual operating system implementation. It specifies the kernel interface, and many vendors has been implemented the specification for their products. Also, many RTOS vendors have sold products to implement the specification. The specification contains basic functionalities such as scheduling, thread managements, and simple inter process communication. Thus, respective companies have been implemented a lot for software on the operating systems. Currently, the operating systems have been adopted by many products such as digital televisions and cellular phones that are developed in Japan.

However, most of embedded systems have became very complex now. For example, a current cellular phone in Japan contains a Web browser, a Java virtual machine, e-mail software, and camera software. On ITRON, these software modules are runninng on a single address space. To implement these software modules in a robust way, we need a more powerful operating system. Linux currently supports various CPU architectures, and many kernel modules can be loaded dynamically. The characteristics are very suitable for embedded systems. Therefore, many industries have considered that using

Linux for their products provides big merits for them.

2. Japan Embedded Linux Consortium

Japan Embedded Linux Consortium(Emblix) has been established from July of 2000 to promote Linux in embedded areas in Japan. Currently, about 100 members have been involved in our consortium. The members include Sony, Panasonic, Toshiba, IBM, NEC and Fujitsu. In our consortium, the above members have been discussed various issues to adopt Linux for their embedded products. In the section, we present a background and current activities of Emblix, and show our future plan. We also would like to describe the relationship between ubiquitous computing and embedded systems, and our opinions to show the importance of ubiquitous computing in future embedded system communities.

2.1. Motivation of Emblix

As described in the previous section, many embedded system products have adopted ITRON in Japan. Therefore, there is a big barrier to adopt Linux for their products because there are a lot of differences between the operating systems. One of the most important roles of Emblix is to identify and solve problems to migrate ITRON to embedded Linux in a smooth way. For example, most developers expect to use existing tool chains to develop embedded systems even if they adopt embedded Linux. Also, we need to reuse a lot of software developed on ITRON because we need to develop actual products very quickly, and there is no time to rewrite entire programs on Linux completely. Moreover, many people do not know issues for using Linux in embedded areas such as legal issues and worst case response time.

It is very important to discuss these issues publicly to share a lot of knowledge about using Linux for embedded systems. Emblix presents various issues to adopt embedded Linux at many embedded system forums, and defines specifications about technologies related to the migration from ITRON to embedded Linux.

2.2. Current Activities of Emblix

Currently, Emblix organizes four working groups to discuss issues about embedded Linux. The first working group discusses legal issues. The group is especially talking

about issues about GPL and its impact to embedded products.

The second working group is working on the hybrid architecture. Especially, we are working on making a specification of ``Linux on ITRON''. Linux on ITRON defines a specification that executes Linux on ITRON. Thus, both ITRON-based software and Linux-based software can be executed on the same system. This makes the migration from ITRON to Linux very easy because we can still use existing ITRON-based software with Linux-based software to build Linux-based embedded systems.

The third working group is working on the development environments for embedded Linux. The group defines several specifications to use ITRON-based development tools for embedded Linux.

The fourth working group is working on the real-time capabilities of embedded Linux. Especially, the group is interested in the worst case response time of embedded Linux that reports in the paper. Also, the group is working on the differences among several real-time extensions of Linux.

2.3. Future Plan of Emblix

In Japan, currently several products that adopts embedded Linux has appeared in the markets, and many products based on Linux will be shipped next year. We believe that the first phase of our role is finished. However, we need to consider many issues for building future embedded systems. For example, we need to take into account security and privacy for Internet appliances. Also, we need to take into account educational issues since traditional embedded systems use various operating system platforms, and it is not easy to use them for educational purposes. We believe that Linux may solve many educational problems due to its openess.

Currently, the most important issue in Emblix is how to establish a open source community for embedded systems. We still need many features in Linux to build future embedded systems. Also, it is very important to provide various open source middleware on embedded Linux to reduce the development cost of future embedded systems. The open source middleware enables us to build various embedded systems in research communities, and it makes it very easy to create various research prototypes and make them actual products very quickly.

3. Embedded Systems and Ubiquitous Computing

In the future, ubiquitous computing environments[4][7] will change our lives dramatically. The vision of ubiquitous computing environments is to acquire information in our environments that are not available before by using sensor technologies[1]. Also, the environments will make it possible to control many everyday objects by embedding very small and cheap computers. One of the most important issues to realize ubiquitous computing is to integrate a real world and a cyber space in a seamless way. This makes it possible to merge bits and atoms[6]. Thus, software infrastructure for Ubiquitous computing should provide a world model that provides a model of our world, which can be accessed by a program, and an application can change its behavior and change the real world by accessing the model[2]. Also, a model in a cyber space can be manipulated by a physical object[3].

In ubiquitous computing environments, we need to consider ultra heterogeneity in various aspects such as hardware platforms, application's requirements, and environmental divergence. For example, it is not easy to develop software on respective hardware platforms. Once a program is written, the program should be executed on various platforms. However, it is very difficult to achieve the goal since respective requirements are surprisingly different. We believe that it is necessary to take into account physical environments such as resource constraints, distribution, and failure explicitly when designing programs. Also, we think considering the each component's assumption and the dependencies among components is very important to build ultra portable software.

In our projects, we are currently working on developing middleware for ubiquitous computing environments. We believe that our middleware will support various future ubiquitous computing applications such as smart space applications, entertainment applications such as robotics and game, and ad-hoc interpersonal communication applications. If there is no right middleware support, it is very hard to implement these applications.

We are currently organizing the following four projects: *Autonomic Components*, *Pervasive Servers*, *Universal Interaction*, and *Smart Materials*. The autonomic component project provides a component framework for ultra heterogeneous

environments for building ultra portable applications. The framework makes it possible to adapt a component according to application requirements, platform requirements and environment requirements by supporting dependencies among components and assumptions of each component explicitly. In the pervasive server project, a large
number of servers are embedded in our environments. Some servers contain sensors to provide information about the real world, and other servers allow us to control actual objects from programs. A mobile server that is wearable by each person integrates several pervasive servers near from the mobile server in an ad-hoc way. We are developing several applications on a mobile server to show the effectiveness of our approach. The universal interaction project allows us to use any interaction devices to access various services. For example, a control panel can be shown at the nearest display from a user, and the panel can be navigated by various input devices such as a PDA, a cellular phone, and a game console. Also, the structure of the presentation is changed according to a user's situation. If a user can use both a large display and a small display, a video will be shown on the large display, but a control panel is appeared on the small display. Also, in the project, we are working on how several services are composed without noticing multiple services from a user. In the smart material project, a world model is created by using various sensor technologies. Also, the project is developing a framework to accessing the world model. In the future, various sensors are contained in everyday materials. The goal of the project is to extract various information from everyday objects without a great effort.

One of the most important issues in our projects is that ubiquitous computing middleware will become future embedded system middleware, and offer additional values to current embedded systems. This means that embedded Linux is a candidate for an operating system for ubiquitous computing. Also, we need to cooperate with industries in embedded areas to achieve the vision of ubiquitous computing. For example, in our previous project, we have implemented HAVi on Linux[5]. Our experiences show that current embedded technologies are important foundations to realize ubiquitous computing. One of future roles of Emblix is to bridge the gap between embedded technologies and ubiquitous computing technologies to make our future lives more fruitful because various future appliances and services related to embedded systems will need ubiquitous computing technologies. We believe that merging between ubiquitous computing and embedded system is a key to improve Japanese economy.

References

[1] H.-W.Gellersen. A.Schmidt, and M.Beigl, ``Adding Some Smartness tp Devices and Everyday Things'', In Proceedings of the Third Workshop on Mobile Computing System and Applications, 2000.

[2] A. Harter, A. Hopper, P. Steggles, A. Ward, P. Webster, ``The Anatomy of a Context-Aware Application''. In ``Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking'', 1999.

[3] H. Ishii, B.Ullmer, ``Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms'', In Proceedings of Conference on Human Factors in Computing Systems,1997.

[4] T. Nakajima, et. al., ``Technology Challenges for Building Internet-Scale Ubiquitous Computing'', In Proceedings of the Seventh IEEE International Workshop on Object-oriented Real-time Dependable Systems, 2002.

[5] T.Nakajima, ``Experiences with Building Middleware for Audio and Visual Networked Home Appliances on Commodity Software'', In Proceedings of ACM Multimedia 2002, 2002.

[6] Nicholas Negroponte, "Being Digital", BeingVintage Books, 1996.

[7] Mark Weiser, ``The Computer for the 21st Century'',Scientific American, Vol. 265, No.3, 1991.