Networked Embedded System Architecture for Controlling

Teruaki Ito

Information Technology R & D Center,
Mitsubishi Electric Corporation

5-1-1 Ofuna, Kamakura, Kanagawa,
247-8501 Japan
Ito.Teruaki@eb.MitsubishiElectric.co.jp

Masahiro Inoue

College of Systems Engineering,
Shibaura Institute of Technology

307 Fukasaku, Minuma-ku, Saitama-city
Saitama, 337-8570 Japan
inouem@sic.shibaura-it.ac.jp

Abstract. Embedded systems are used in broad variety of areas. As increasing processing power, being networked and having more multiple and complicated functionality, the embedded systems used in control systems have more issues to be resolved. In many cases, design constraints are applied to the entire life-cycle of the embedded systems. There are stakeholders involved in the systems through the life-cycle and they play their roles in each stage of the life-cycle. However, concerns of all of the stakeholders are less taken care of. Challenges to develop networked embedded systems for control systems are to maximize all of the stakeholders' satisfactions through the entire life-cycle in addition to satisfy several constraints. This paper discusses architectures of networked embedded systems from a perspective of stakeholders' satisfactions. We present the stakeholders and their standpoints, composition of networked embedded systems and architectures, then pros and cons of the architectures from each stakeholder's view.

Introduction

As improving processing power and system resources, embedded systems are gaining its functionality so far. There is a recent accumulating trend towards networked embedded systems composed of embedded systems interconnected one another via network medium. Networked embedded systems which often used for control application have more issues to be resolved. Design constraints are often applied to the entire life-cycle of the networked embedded systems. There are participants involved in the lifecycle of the systems. However, concerns of all of the participants are less taken care of. Challenges to develop and maintain networked embedded systems for control systems are to maximize all of their satisfactions through the entire life-cycle in addition to satisfy several constraints.

Networked embedded systems are in the diversity of applications. In vehicle control systems and industrial automation systems, networked embedded systems play more and more important roles. Fundamental functions and more comfort purpose functions in vehicles are realized by electronic control unit (ECU) and their networks. Applications for controlling, supervising, and configuration are implemented by using programmable logic controllers (PLC) and networks in manufacturing systems.

In many cases, networks have replaced traditional point-to-point wiring. Reducing volume of wiring with networked embedded systems is also simple but most important advantage in manufacturing systems (Moyne 2007). Decreasing the number of physical connectors and wires results in gaining more reliability. Networked systems have advantages in construction and maintenance. The fewer connectors and wires device has, the easier technicians install and replace such device.

Automotive industry has similar motivations. An answer toward the problems of weight, cost, and reliability caused by increasing point-to-point links among ECUs and between ECU and sensors/actuators is the replacement of the links with network. Networks also have power to enable new and complex functions by means of integrating functions (Axelsson 2003). The demand toward network in vehicle comes from efficient diagnostics.

Design issues and requirements in development of embedded systems are discussed (Koopman 1996) (Hänninen 2006) (Axelsson 2003). Koopman addresses life-cycle support as one of the design constraints for embedded systems. Axelsson et al present architectures from three different automotive manufacturer points of view.

Participants playing a role during product life cycle are not only final commercial product's manufacturer. Figure 1 shows one view of embedded system lifecycle with the participants in lifecycle stages for automotive industries. Arrowed circle shows an embedded system lifecycle (Koopman 1996). Semicircle shows duration for participants of the lifecycle to play some roles in stages. In this lifecycle, a vehicle manufacturer, suppliers, dealers are shown as participants. They have their own stages to be participated. An automotive manufacturer is in a lifecycle from the start of it. Suppliers to provide embedded system components join at a product design stage. Car dealers start to participate at a deployment stage. As Axelsson et al shows the architecture for each manufacturer due to the differences of demands and constraints, demands and constraints of participants other than final product manufacturers may cause the different architecture from that of manufacturers.

In this article we first introduce issues on networked embedded systems for control applications and participants playing a role during product life cycle. Then we propose and compare three architectures. Finally we evaluate the architectures from our issue-player-architecture perspective.

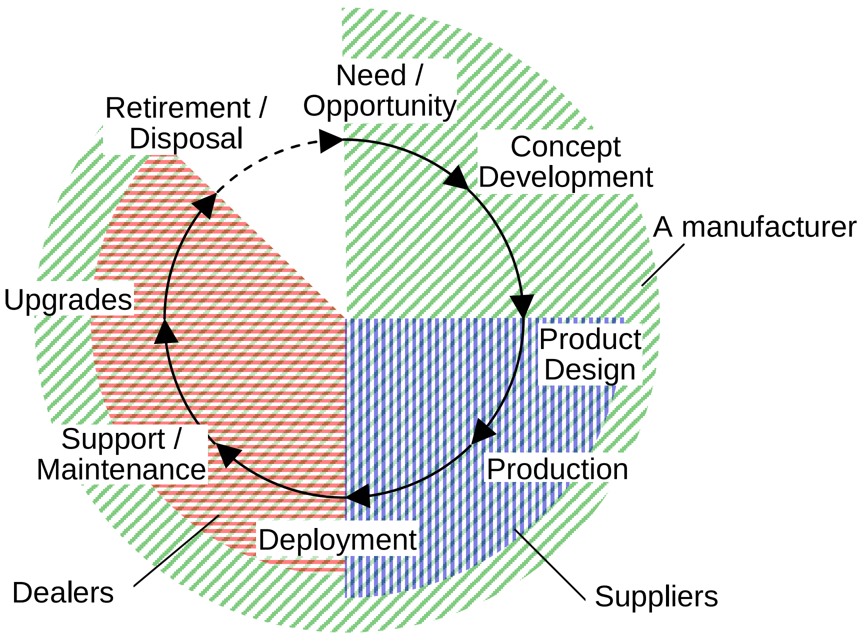


Figure 1 An embedded system lifecycle and participants in automotive (taken by Koopman and modified)

Issues

There are researches discussing concerns of the development of embedded system and networked embedded systems. Koopman presents computer design requirements, system level design, life-cycle support, business model, and design culture in development of embedded systems (Koopman 1996). The requirements are shown in Table 1. Hänninen *et al* presents concerns in development of embedded systems in the automotive domain. The concerns are safety, maintainability, testability, reliability, portability, reusability (Hänninen 2006).

Table 1 Requirements (Koopman 1996)

Category	Requirement
Computer design	Real-time characteristics
	Packaging
	Safety and reliability
	Harsh environment
	Cost sensitivity
System level	End-product utility
	System safety and reliability
	Controlling physical systems
	Power management
Life-cycle support	Component acquisition
	System certification
	Logistics and repair
	Upgrades
	Long-term component availability

This paper focuses on the following issues and discussed architectures based on the issues later.

Real-time characteristics. Networked embedded systems used in control systems needs to be satisfied with some real-time constraints. The constraints are depends on the applications' requirements. There are several real-time characteristics, such as delay, delay variances, response time, and deterministic.

Cost. Cost is divided into production cost and development cost. The production cost is simply the summation of hardware parts and assembly cost. For software development, the development cost is period which it takes to implement required functionality.

Reliability. It is considered that reliability is one of the most important features in development of networked embedded system. From hardware point of view reliability depends on error rate of components consisted in the hardware. In case of software, failure may be caused by incorrect specifications or unexpected situations that the specification does not address. Incorrect implementation also may cause failure.

Maintainability. After the completion of a system development, the system has to keep requirements in use. Systems occasionally will be modified for keeping correctness or improvement.

Reusability. In most cases, a product may have a function similar to that developed for another product. It is neither cost effective nor efficient development to create both in each time. Reusability helps to reduce cost and development period.

Portability. When hardware or lower level software is changed, application or function must be adapted for the change of the lower. Those situations come from modification of requirements, or specification, improvement products, or maintaining compatibility among product series. Portability also mitigates an impact caused by unavailability of some electronic components and migration to the substitution of the component. As the same of reusability, portability helps to reduce cost and development period.

Scalability. If a hardware component, software module, or software application has scalability, these can suit to the wide scale of requirements. This effects cost and development period reduction.

Testability / Diagnosisability. Testability and diagnosis ability are important to verify and diagnose achievement of functional requirements. The complexity of networked embedded systems is increasing today. Testing these complex functions in development and diagnosing error and cause at dealers efficiently are indispensable to cost reduction and development efficiency.

Customizability. It is often that a developer has to create functions or components whose requirements are almost same but slightly different. This case happens when a developer sells a function to a customer then sells the same function to another customer with modification on demand of the second customer. Easy to modify or customize to in order to satisfy customer demands gains not only customer satisfaction but also cost and development efficiency.

Issues mentioned above are related one another under the purpose to meet market demands. The relation is shown in Figure 2.

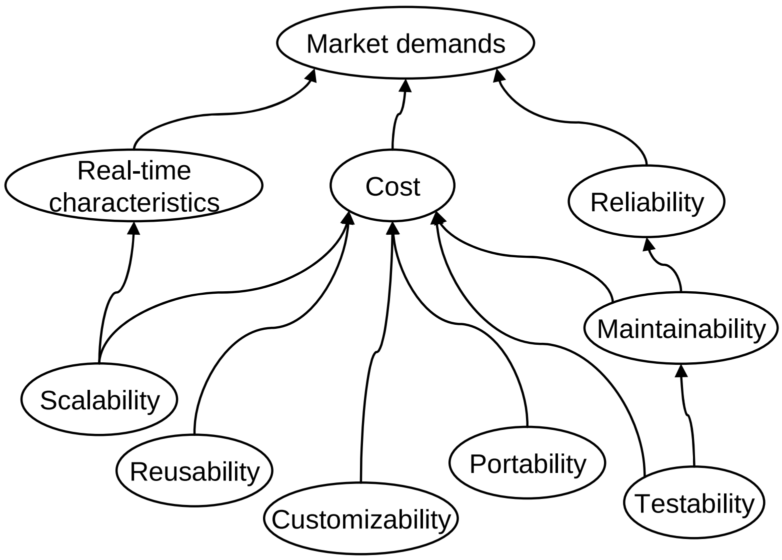


Figure 2 Issue tree

Role model

As shown in Figure 1, networked embedded systems have their own product lifecycle and there are participants who play their role to the systems in the stages of the lifecycle stakeholders. We present two role models of the participants to systems. The role models are for automotive industry and for industrial automation systems. In automotive industry, the participants are vehicle manufacturers, suppliers, and vehicle dealers. A manufacturer is an integrator of vehicle which has networked embedded systems and decides a specification of the entire systems. A supplier provides components of the networked embedded systems. A dealer is a maintainer of the networked embedded systems. In industrial automation system, the participants are operators, integrators, maintainers, and suppliers. A factory owner decides a specification of a networked embedded system for automation and is an operator of the system. The factory owner may be also an integrator and/or a maintainer of the system. System engineers are integrators of the system, who build an automation system based on demands of the system owner. A maintainer does maintenance of the system to keep required functions. A supplier provides components of automation systems.

System architectures

In most cases, embedded systems are composed of hardware and software. A simplified standalone embedded system is shown in Figure 3. Processing units, memory, special purpose hardware logic such as ASIC or FPGA, sensors and actuators are included in hardware. Real-time operating system and application are software. An embedded system realizes more than one application.

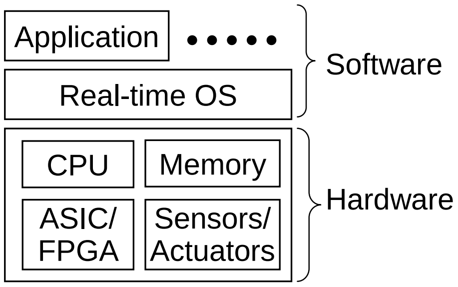


Figure 3 Standalone embedded system

A networked embedded system has network connectivity in addition to the components of an embedded system. Software construction of networked embedded systems is slightly different from that of embedded systems. In the case of networked, application may be realized by means of functions resided on several nodes spreading in a network. Software is composed in layered manner. Complex and integrated applications may also be executed in a network by integrating several applications in the network. This is shown in Figure 4.

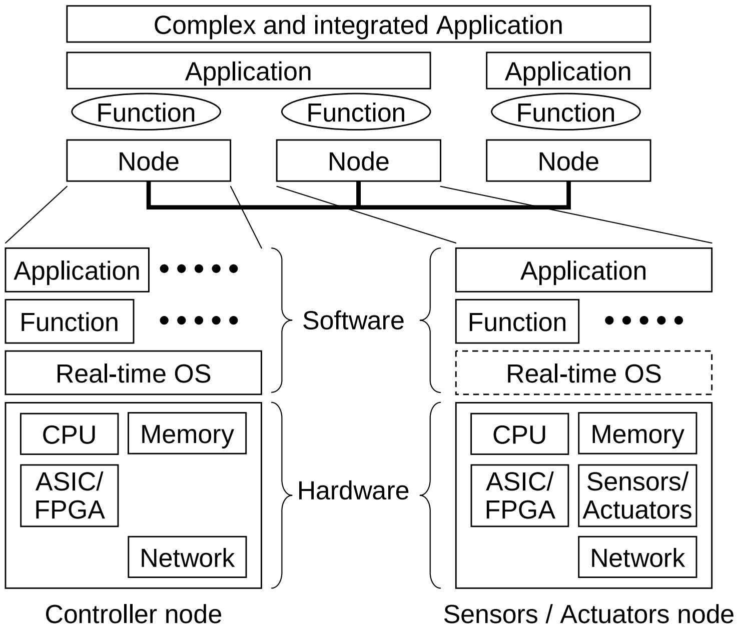


Figure 4 Networked embedded systems

Real-time operating system used in networked embedded systems for controlling must have ability to handle network hardware component. There are several approaches to coordinate among nodes interconnected by network medium. Classification of the approaches is based on residence of network coordinators and network awareness at applications. Network coordinators helps for applications communicate with one another. Network awareness means whether or not application needs to know the existence of network or remote nodes.

Three architectures are derived from the location of the coordinators. These are shown in Figure 5. For the first architecture the coordinators are included in an application itself or function which an application uses. The second architecture introduces middleware for coordination between application/function layer and real-time OS. Wang presents the second architecture; Common Object Request Broker Architecture (CORBA) based middleware for distributed embedded real-time systems (Wang 2005). The third one uses real-time OS with coordinator function.

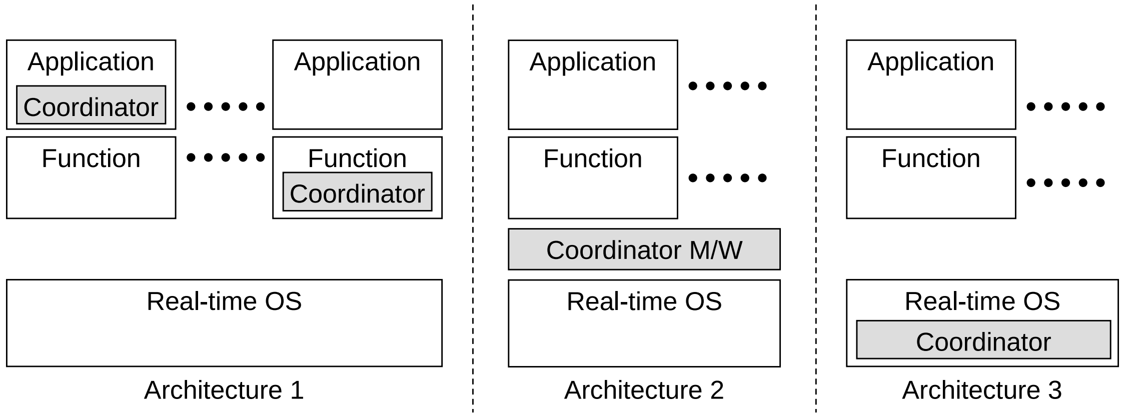
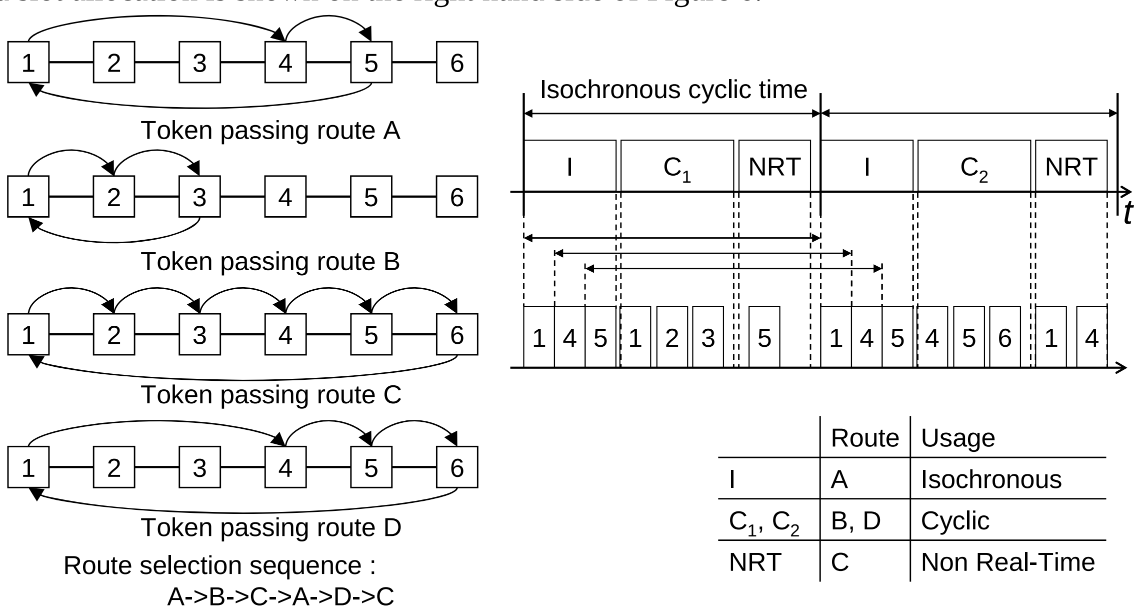


Figure 5 Residences of network coordinators

For the underlying mechanisms to coordinate nodes token-passing base coordination can be applied (Ito 2008). Figure 6 shows the coordination proposed by Ito. On the left hand side of Figure 6, some token passing routes are depicted. The numbered squares are shown as nodes, line as connectivity between nodes, and arrowed lines as route of token. There is one token manager in the network. The token manager selects a route from the routes decided in advance. Routes are selected to satisfy with requirements on the allocation of time slots. A time chart and slot allocation is shown on the right hand side of Figure 6.



	Route	Usage
I	A	Isochronous
C_1, C_2	B, D	Cyclic
NRT	C	Non Real-Time

Figure 6 Token passing base coordination (Ito 2008)

In the second and third architecture, applications do not need to know the existence of network explicitly. In industrial automation systems, controllers access sensors and actuators via I/O mapping model. By extending naturally, common memory model is introduced and sensors and actuators are mapped on the common memory. Application does not need to know whether sensors and actuators are located in the same node or in a remote node.

Evaluation

Each participant has different concerns about the issues. We classify participants and discuss the issues from their point of views. Participants are classified into three roles, operator / maintainer, integrator, and supplier.

Operator / Maintainer. They do daily operational or maintenance work of a system. They usually do those works according to the manuals provided by an integrator and do not know the details of the system, such as what kind of networks are used in the system, how many controllers are in the system, or how applications communicate with each other. Their main concerns are achievement of their requirements in real-time characteristics, cost, and reliability. They also emphasize maintainability and diagnosisability to mitigate their work load. Customizability may be necessary when they want to modify or adjust a system. They consider those issues but do not care about the internal architecture of a system even though their concerns affect the architecture.

Integrator. Embedded systems are interconnected with network and integrated into a system which is required to fulfil demands. An integrator decides the system composition of embedded systems as components, specifications of them, and interactions among the embedded systems. Applications using functions and applications provided by the embedded systems are made and stacked on the entire system. In order to do these activities over and over, an integrator concerns reusability, portability and scalability as well as requirements for real-time characteristics, total cost, and reliability to be satisfied with. The comparison of three architectures from integrator's point of view is shown in Table 2. The number of X's indicates relative importance.

Table 2 Comparison from integrators' view

	Application/Function	Middleware	Real-time OS
Real-time characteristics		X	XX
Cost		X	X
Reliability		X	X
Maintainability	XX	X	X
Reusability		X	X
Portability		X	X
Scalability		X	X
Testability / Diagnosisability	XX	X	X
Customizability	XX	X	X

Supplier. Specification of an embedded system provided by a supplier is decided by an integrator or the supplier itself. The main concern of the supplier is achievement of the balance among cost, performance requirements (real-time characteristics), and reliability. The larger production quantity, the more effective cost reduction of electronic parts even though it is small amount for each pieces. CPU which has less powerful processing performance may have to be adopted because of reducing hardware cost. As increasing lines of software code and importance of software for reliability and avoiding defects, suppliers' concerns about software rapidly raise. All of maintainability, reusability, portability, scalability, testability / diagnosisability and customizability are their concerns. The comparison of three architectures from supplier's point of view is shown in Table 3. Again, the number of X's indicates relative importance.

Table 3 Comparison from suppliers' view

	Application/Function	Middleware	Real-time OS
Real-time characteristics		X	XX
Cost		X	XX
Reliability		X	XX
Maintainability	X	X	
Reusability	X	XX	X
Portability	X	XX	X
Scalability	X	X	
Testability / Diagnosisability	XX	X	X
Customizability	XX	X	X

Under the situation where several participants exists and issues for them are different each other, it is a challenge to decide a system architecture. We expect that the third architecture tightly related to real-time OS is a good choice according to comparisons shown in Table 2 and Table 3. In order to verify that, we will use the Analytic Hierarchy process (AHP) for verification. The AHP is a powerful tool to deal with complex decisions.

AHP hierarchy to decide the system architecture of the networked embedded systems is shown in Figure 7. Architectures are corresponding to those shown in Figure 5. Architecture 2' and 3' are variants of Architecture 2 and 3, respectively. As mentioned in System Architectures, applications may not know that the underlying system uses networks when a model such as common memory is adopted. In the architecture 2' the location of the coordinator is the same as that of architecture 2, but an underlying model that application does not need to know the existence of networks is used.

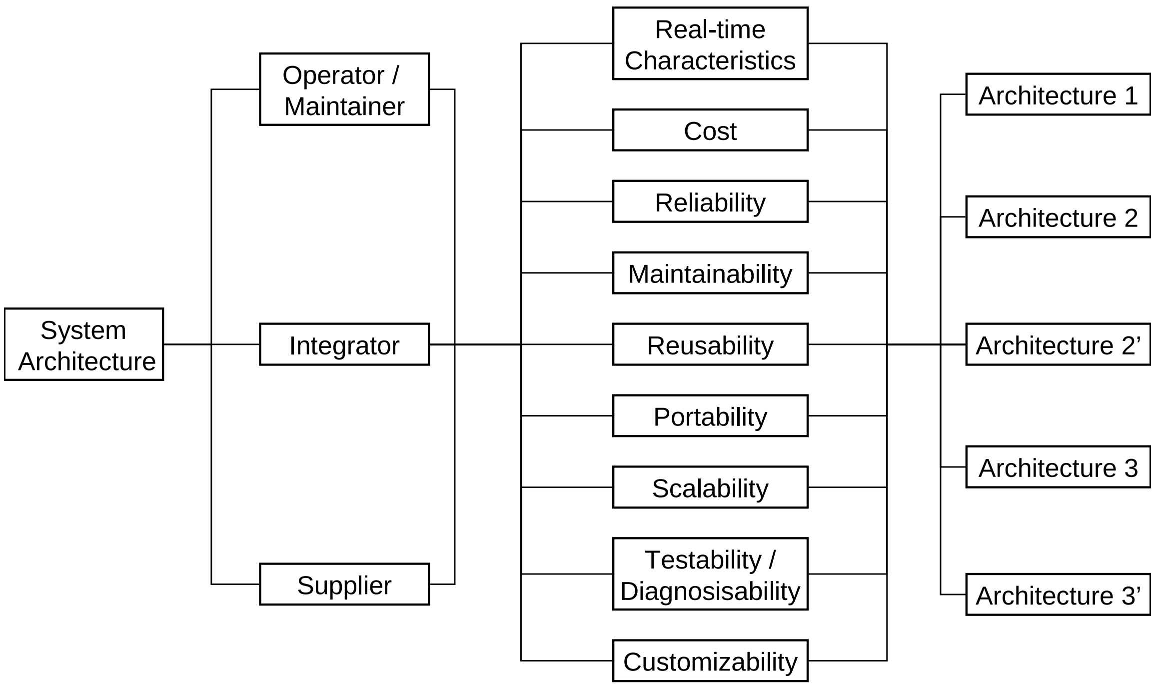


Figure 7 AHP hierarchy

Conclusion and future work

There are issues on networked embedded systems to be resolved as embedded systems gain more processing power, interconnectivity via network, and multiple and complicated application. Real-time characteristics, cost, and reliability are the first points that market demands are reflected. Several other issues are related to the three issues.

We present participants involving with networked embedded system and their roles during a product lifecycle of the system. Main roles are an operator / a maintainer, an integrator, and a supplier.

In networked embedded system, a coordinator of the networks is necessary. According to the residence of the coordinator, three architectures are introduced and discussed. Two of them have variants derived from a model of network awareness.

The issues and the system architectures are discussed and evaluated from the roles of the participants. We expect that the architecture tightly related to real-time OS is a good choice for the networked embedded systems.

Next step of our work is to make a verification of the advantage of the selected architecture with the AHP. It is for our future work to find out whether this choice is common in the area where the network embedded systems are use, or not.

References

Axelsson, Jakob, Joakim Fröberg, Hans Hansson, Christer Norström, Kristian Sandtröm, Björn Villing. 2003. A Comparative Case Study of Distributed Network Architectures for Different Automotive Applications. MRTC report ISSN 1404-3041 ISRN MDH-MRTC-69/2003-1-SE, `http://www.mrtc.mdh.se/index.php?choice=publications&id=0478` (accessed November 14, 2008)

Hännsen, Kaj, Jukka Mäki-Turja, Mikael Nolin. 2006. Present and future requirements in developing industrial embedded real-time systems - interviews with designers in the vehicle domain. In Proceedings of the 13th annual IEEE international symposium and workshop on engineering of computer based systems.

Ito, Teruaki, Masahiro Inoue. 2008. A real-time communication network based on token passing. *IEICE Technical Report* 108 (203): 111-14.

Koopman, Phillip. 1996. Embedded system design issues (the rest of the story). In Proceedings of 1996 IEEE International Conference on Computer Design: VLSI in Computers and Processors.

Moyne, James R., Down M. Tilbury. 2007. The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data. *Proceedings of the IEEE* 95 (1): 29-47.

Navet, Nicolas, YeQiong Song, Françoise Simonot-Lion, Cédric Wilwert. 2005. Trends in Automotive Communication Systems. *Proceedings of the IEEE* 93 (6): 1204-23.

Wang, Lingfeng. 2005. Middleware for implementing distributed embedded real-time measurement and control systems. Autotestcon, IEEE.

BIOGRAPHY

Teruaki Ito is a researcher in Information Technology R&D Center, Mitsubishi Electric Corporation. His research interests include real-time networks and real-time embedded systems. He received his B.S. and M.S. in Engineering from Keio University, and M.S. in information networking from Carnegie Mellon University.

Dr. Masahiro Inoue is a professor of College of Systems Engineering, Shibaura Institute of Technology. His field of research includes embedded system, ubiquitous network, systems engineering, and project management. He received his B.S. and M.S degree in Applied Physics from Waseda University and Ph. D degree in Computer Science from Shizuoka University. He was engaged in research and development at Mitsubishi Electric Corporation. During 1990-1991, He worked at the University of Michigan as a visiting researcher. In 2005, He became a professor of Shibaura Institute of Technology. He is a Professional Engineer in Japan and a Project Management Professional.