A PROPOSAL OF AUTONOMOUS DECENTRALISED SOFTWARE ARCHITECTURE FOR SAFETY-RELATED SYSTEM AND ITS APPLICATION TO RAILWAY SIGNALLING SYSTEM

Takashi Kunifuji, Manager, East Japan Railway Company  
Hiroshi Ito, East Japan Railway Company  
Masayuki Matsumoto, Corporate Officer, East Japan Railway Company

SUMMARY

Recently, social environment rapidly varies and it affects the trend of social infrastructure. Railway transportation which is typical social infrastructure is also affected by this trend. To meet this situation, the railway system needs a capability for gradual and short-term extension. To realize these requirements, an autonomous decentralised software architecture is proposed and its practical application Intelligent Signalling Equipment Controller (ISEC) which has been developed by East Japan Railway Company (JR-EAST). The online software maintenance function enabled by the architecture is introduced, and its feasibility is shown.

1 INTRODUCTION

Recently, social globalization has been progressing and individuals’ sense of values rapidly varies. In railway transportation, more advanced services has been required to meet customers’ needs as well as safe and reliable transport.

To meet this situation, railway system needs a capability for gradual and short-term extension. In a conventional signalling system, it needs to be stopped during the extension work to secure the safety. However, the signalling system secures safety not only for train operation but for maintenance works performed during the maintenance time at night, and thus the allowable time to stop the system is quite limited. So, long period is required to complete an expansion. Therefore, a technology that the old and new system needs to coexist and to be switched seamlessly with keeping operation and securing safety and reliability is required.

In this paper, autonomous decentralised software architecture based on autonomous decentralised system [1][2][3][4] and an online software maintenance technology that achieves the safe coexistence of different versions of software and seamless switch between the versions during online extension is proposed. Moreover, as a practical formation for these technologies, the Intelligent Signalling Equipment Controller (ISEC) as developed and put into practical use by JR-East, is introduced.

2 AUTONOMOUS DECENTRALISED SYSTEM (ADS)

The objectives of the ADS are realization of online expansion, fault tolerance and online maintenance that are called online property. In the ADS, these properties are assured by system architecture.

2.1 ADS Architecture

The ADS architecture is defined under the following standpoints.

- Being faulty is normal.
- System is result of integration of subsystems.

At first a subsystem exists. A total system cannot be previously defined. The system is the integration of the subsystems, some of which may be faulty or undergoing modification and repair.

2.2 Data Field

The system with the basic features of equality, locality and self-containment can be realised under a new system architecture, where there is no central operating or coordinating system. Each subsystem has its own

A proposal of autonomous decentralised software architecture for safety-related system and its application to railway signalling system
management system, the Autonomous Control Processor (ACP) to manage itself and coordinate with the others. The subsystem including its application software modules and ACP is an autonomous unit called “Atom”.

The self-contained subsystems including their respective ACP’s are integrated into a system. In the ADS, all of the subsystems are connected only through the Data Field (DF); all data is broadcasted into the DF and the data itself logically circulates in the DF. The data moves around the application module in the Atom and the DF in the Atom is called the Atom Data Field (ADF). In the DF, each data is attached with its “content code” which is uniquely defined with respect to the content of the data. The system structure may vary depending on the expansion, the reduction or the partial faults of system. Therefore, to protect the operation of the subsystems from variation in the system, each subsystem broadcasts a message containing the content code instead of the receivers address. A subsystem decides whether or not to accept a message on the basis of content code. The receiver selects to receive the data but the sender never points out the receiver. The necessary content codes for the Atom are determined dependently on the application functions within it.

This content code communication enables every subsystem to have autonomy in sending and receiving data. That is, each subsystem does not need to know the relation among the sources and the destinations. This feature of the content code communication ensures the locality of the information necessary for each subsystem.

2.3 Data-Driven Module

Each Atom selects to receive the data necessary for its application software modules on the basis of their content code. After all of the necessary data is received, the application software module in the Atom starts its execution. This feature indicates the autonomous “data-driven mechanism” of the application software module in the sense that no module ever drives the other nor directs them to receive and process the data. This mechanism makes the modules loosely coupled. No module or Atom controls the others but each independently judges and controls itself; that is, the Atoms are equal.

The necessary content codes for each application software module are assigned in the ACP of the Atom. The ACP can dynamically change to assign the content codes in it according to the change of the application software modules. Even if the content codes assigned to the ACP are changed, it is unnecessary for the Atom to inform any other Atom or ACP.

3 AUTONOMOUS DECENTRALISED SOFTWARE ARCHITECTURE

3.1 Safety Requirements

To realize the online property of the safety related applications in the autonomous sub-system, following safety requirements should be satisfied.

- If any application module fails, is repaired and / or is newly added, the other application modules can continue to manage themselves and to perform their own responsible functions safely.
- If any application module fails, is repaired and / or is newly added, the other application modules can coordinate their individual objectives among themselves and can operate in a coordinated fashion safely.
From these requirements, the concepts of software architecture listed below are derived.

- Dividing the process by the application and keeping its independence
- To limit the affected area to corresponding function due to upgrading or partial disorder, independence between functions needs to be established.
- To maintain the independence between the application modules, each application module is created as a task and is set unable to access directly to the other software module.
- By this protection, even if there is a bug in the software or some other disorders, the task with a problem cannot destroy the other program or data. In this way, the affection of disorders for other functions will be eliminated.

3.2 Autonomous Decentralised Software Architecture for Safety Related System

To realize the requirements listed above the autonomous decentralised software architecture for safety related system is proposed. In this architecture the sub-system consists of safety management function, safety related application and atom data-field.

![Autonomous Decentralised Software Architecture for Safety Related System](image)

3.2.1 System management function

This function is responsible for the safety on task scheduling, memory management, and fault detection.

In task scheduling, generally, a multi-task configuration utilizes CPU power effectively, so when executing task is waiting, next priority task will be called and started. However, in safety related applications, keeping the processing order is required for safety. To fulfil this requirement, system management function centrally manages the other tasks’ operation and keeps the processing orders. It also monitors not to start another task even when current process takes a wait.

In memory management, independent memory space is reserved for each task. System management function watches the memory violation and any task which attempts to violate the memory space reserved for other tasks is forcibly aborted before the contents of the memory can be destroyed.

In fault detection, hardware such as memory, CPU, electrical circuit, and so on are periodically monitored. When a fault is detected, the safety related applications which use the faulty hardware are forcibly aborted. Furthermore, the processing result of the safety related application is always verified. Generally, the safety related computer is configured as dual system, and equivalent application modules are processed on each system in synchronisation. The processing result of each system is verified with the other and if difference is detected the safety related application is forcibly aborted.
3.2.2 Safety related application

Safety related application is installed as a task and one application is surely consists of one task. Each task should be independent and assigned to dedicated memory space. Safety related application is also real-time one. The time constraint is defined and the timeout of processing is watched by system management function.

3.2.3 Atom data field

This data field is for sharing the data among safety related applications and configured on physical memory of the sub-system. The data field doesn’t assure the safety of data. So, the safety of the data should be assured by data exchange protocol between safety related applications. Generally, safety code which helps detecting of all sort of the fault occurred in the data field should be generated and checked by safety related application.

4 INTELLIGENT SIGNALLING EQUIPMENT CONTROLLER (ISEC)

4.1 Re-configuring of the Signalling Logic

As a practical application of autonomous decentralised software architecture, East Japan Railway Company has been developing Intelligent Signalling Equipment Controller (ISEC). In ISEC, all the signalling logic such as for interlocking, automatic train protection (ATP), level crossing, etc. are integrated into one high performance safety related computer. To improve the availability, expandability and maintainability of the signalling software, autonomous decentralised software architecture is applied.

Figure 3 shows the difference of current architecture and autonomous decentralised architecture. In current architecture, control logic of each signalling devices are centralised into one software module and exchange information directly via netlike set upped interface. So it is difficult to localize an influence when logic or function has changed. On the other side, in autonomous decentralised architecture, all information is exchanged via data-field. So it is able to localize the influence concerned changing of logic or function.

4.2 Detail of Re-configuring

4.2.1 Analysing of signal control logic

On re-configuring of the signalling logic we analysed the logic. The logic is mainly classified into interlock and automatic signalling. Interlock is logic to ensure the safety of operations in the station yard. This logic coordinates competitive operations such as train operation, shunting, maintenance, etc. Automatic signalling is a logic to control signalling device automatically based on train tracking such as automatic blocking, ATP, Level Crossing, etc. (Figure 4).

A proposal of autonomous decentralised software architecture for safety-related system and its application to railway signalling system
4.2.2 System model

Figure 5 shows the relation of signal control logic and external objects and information flow among them.

4.2.3 Application model of autonomous decentralised signalling logic

Figure 6 shows the application model of ISEC. An outline of this model is described below.

(1) Autonomous software module

In ISEC, each signalling function such as interlocking, ATP, level crossing, etc. is installed as an autonomous software module. An autonomous software module consists of many sub-modules which process signalling logic of abstract signalling devices. An abstract signalling device corresponds to one or multiple signalling devices. For example, route corresponds to colour light signal or shunting signal, train tracking corresponds to track circuit and so on. Each software module is configured as an independent task which encapsulates signalling logic for coordinating signalling devices and safety function, and it broadcasts the information of itself to the atom data field and receives other software module’s information from the atom data field. In this way, the software module is able to work autonomously.

(2) Atom data Field

Atom data-field consists of multiple sub-routes, each of which is abstract data defined by an area of train detection and switch direction. It maintains information about interlocking and train tracking. Each sub-route has unique ID and this is the content code which is used by the sub-systems to broadcast and accept messages.

(3) Interface

ISEC has 3 interfaces for external systems, that is operator, signalling device, and train or rolling stock.
4.3 Online Maintenance Function of ISEC

4.3.1 Improvement in Maintainability of the System

By configuring the logical processes with divided tasks by the function, it is possible to stop certain tasks for maintenance. That is, the system management function enables to stop a task by a software module in maintenance or partial malfunction. During the partial stop, the system management function records each subsystem’s operation status to the atom data field. The other running subsystems can detect the stopped software module by referring the database (Figure 7).
4.3.2 Realization of online maintenance of software

Signalling system usually exchanges control information or software version among multiple systems and the system will stop when detecting the mismatch for safety. Due to this feature, software upgrading is executed with the system stopped so as not to detect the mismatch. Once the interlocking function stops, its effect is very large. Therefore, online software maintenance is required.

As conventional fail safe system with 2 out of 3 system configuration secures a fail-safe condition by monitoring the control results to be the same, mismatch of software version among the multiple system is not permitted. On the other hand, ISEC’s dual duplex system can maintain the fail-safe condition only by one system. By utilizing this feature, realizing online software maintenance with “Maintenance Mode,” which allows temporal mismatch of the software version, is considered.

The definition of LC maintenance mode is as follows.

- Allows mismatch of the software versions between the systems
- Stops slave’s output for upper and lower devices
- Switches master and slave system without interruption
- Provides no reset control for the other system
- Does not allow the automatic switch of master and slave systems
- Carries over the set maintenance work between master and slave system
- Can be released only when the software versions match or only master system exists

4.3.3 Online software maintenance procedure

The flow of the non-stop upgrading by using the LC maintenance mode is shown in Figure 8.
In step 1, by setting LC maintenance mode, the slave’s output stops. After upgrading the slave’s software in step 2 and switching the master and slave in step 3, the output is now comes from the previous slave. At this time, the output is generated by the upgraded software. As the switch of master and slave is executed without any instantaneous interruption, the software upgrading can be done without interruption. However, when the upgrading involves some change of lines, the assignment of devices also needs to be changed and the takeover of data between master and slave is unable; this means the entire system needs to stop for such upgrading.

5 CONCLUSION

In this paper, at first changing of the user needs for railway system in accordance with rapidly changing social environment and system requirements which realize the user needs are clarified.

To meet these requirements, autonomous decentralised software architecture is proposed as the purpose of improving availability, expandability and maintainability of a safety related system. As a practical application of this architecture for railway signalling system, Intelligent Signalling Equipment Controller (ISEC) and online maintenance procedure for this system are shown.

In ISEC, signalling logic should be re-configured from control driven to data driven to apply autonomous decentralised software architecture. For this re-configuration, reliability, expandability and maintainability are improved.

ISEC is now in final development stage for practical use. This will be introduced for the first station in 2016.

6 REFERENCES


