TRIOLE Organic Computing Architecture

● Akira Katsuno ● Satoshi Tsuchiya ● Motomitsu Adachi

(Manuscript received April 25, 2007)

Autonomous mechanisms are expected to serve as foundations for stable, adaptable, and cost-efficient managed IT systems that can support social and economic activities. This paper describes an autonomous mechanism developed by Fujitsu called the TRIOLE organic computing architecture that integrates human activities and IT processes. Next, it describes a unified description scheme that defines a series of management processes for integrating human activities and IT processes and outlines a method for easily creating reliable processes by using policies and templates. Lastly, it describes a configuration management database that is based on system knowledge about this architecture.

1. Introduction

IT systems that support social and economic activities must have both stability to ensure continued operation and flexibility to rapidly provide new services while adapting to environmental changes. On the other hand, IT systems are becoming more and more complex due to the shift to larger scales and openness accompanying technological innovation and capability requirements. Therefore, the risk of system troubles such as failures and performance degradation is increasing, resulting in higher costs for operation management.

An autonomous function is a core technology for stabilizing the operation of an IT system and realizing flexible and efficient operation. An IT environment managed by an autonomous function is called an autonomous computing environment. Autonomous functions are similar to the autonomic nervous system of the human body, which has complicated functions for performing tasks such as temperature regulation. Additionally, people who have cold symptoms such as fever and a sore throat will consciously try to restrict their physical activities, sleep more, and take medicine so they can recover while the autonomic nervous system cures the symptoms. Fujitsu's TRIOLE organic computing architecture is a base of autonomous computing environments that integrates the management of both IT processes and conscious human activities such as the above (e.g., authorization, confirmation, installation of equipment, countermeasures against emergencies).

This paper describes the problems of current operation management. It then outlines the TRIOLE organic computing architecture and its common infrastructure, which is a configuration management database.

2. Problems in operation management

Research conducted by Fujitsu shows that approximately 80% of system troubles are caused by human errors such as missed input parameter settings, misunderstanding of specifications, and errors in operation. In one example, a missed setting caused a system slowdown and it took several days to discover the cause. These problems in operation management are caused by, for example, poor understanding of the system configuration, ambiguous modification procedures, and insufficient cooperation among software components. Under present circumstances, the rapid fixing of accidental troubles largely depends on the individual capabilities of skilled people. In the late 1980s, to exclude these risks in operation management and provide high-quality IT services, a British government agency established an integration of best practices of IT service management called the IT Infrastructure Library (ITIL).[1] ITIL describes the daily operation processes performed mainly by operators as service supports and describes the establishment and evaluation of mid-term and long-term enforcement projects as service deliveries. We think that autonomous computing is a fundamental technology for efficiently realizing high-quality operation management based on ITIL. An important point to note is that, even when an autonomous function is implemented, some operation processes might not be completely automatized without intervention by a system administrator. For example, the autonomous function should not include physical work such as the delivery and installation of servers; advanced judgments such as how to respond to unexpected accidents, natural disasters, and terrorism; confirmation proceedings; and auditing procedures. Therefore, an architecture that integrates human activities and IT processes is required. There are four major technical problems:

1) Establishing a common format for operation processes
2) Defining the skills and responsibilities of the human and IT systems
3) Automating operations
4) Accelerating the growth of systems

Fujitsu established the TRIOLE organic computing architecture as a basic software structure for meeting these problems.

3. Basic structure of TRIOLE organic computing architecture

This section describes the basic structure of the TRIOLE organic computing architecture and also outlines a database called the system knowledge that maintains information about various components and the relationships among them.

3.1 Basic structure

As shown in **Figure 1**, the basic structure of the operation management is divided into two

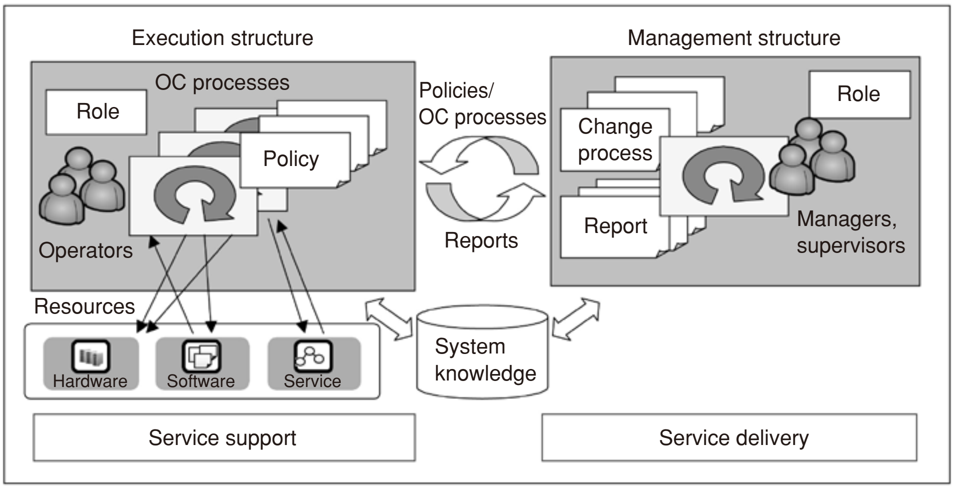

Figure 1
Overview of TRIOLE organic computing architecture.

substructures: a management structure and an execution structure. The management structure establishes and evaluates the policy of operation management from the mid-term and long-term viewpoints. On the other hand, the execution structure performs daily operations based on the management policy. The management structure mainly consists of the operations of system administrators. In the execution structure, the IT system plays a key role and operators only perform a minimum amount of work, for example, the authorization and verification processes. The management structure and execution structure correspond, respectively, to the Service Delivery and Service Support defined in ITIL. Every operation management procedure is described as a series of processes called an organic computing process (OC process). In the execution structure, many OC processes are executed simultaneously. Each OC process includes a policy for regulating the operation guidelines and autonomously operates to always match the given policy. Operation managers can control the IT system clearly and precisely just by issuing a policy that they can create easily. Additionally, human activities, for example, the activities of system administrators and operators are specified by their roles, which consist of their authorities and skills. The structure of human activities is the same as that of an IT system. Therefore, technical problems 1) and 2) described in the previous section are resolved by using the structure for specifying the roles of the human and IT systems.

3.2 System knowledge

As shown in **Figure 2**, the system knowledge consists of information — including the physical- and logical-configuration data of the system and the system's performance data — and knowledge, which formalizes the know-how of designers, operations managers, and SE/CEs. Knowledge includes various templates and information for troubleshooting. From the viewpoint

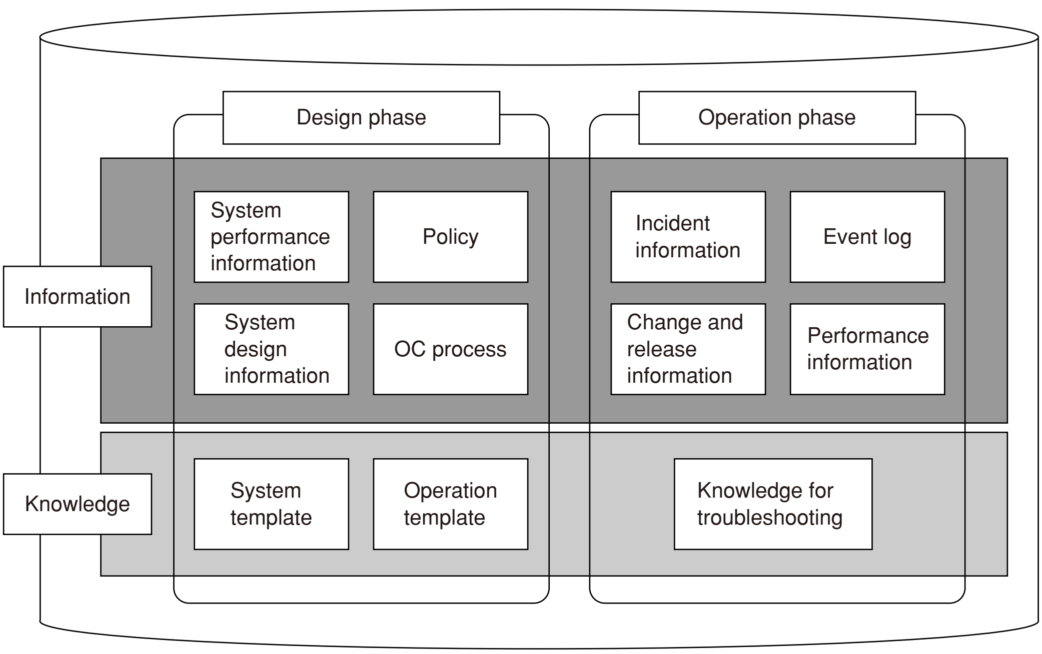

Figure 2
System knowledge.

of an IT system's lifecycle, the system knowledge is divided into the information/knowledge used and generated in the design phase and the information/knowledge used and generated in the operation phase. While the design phase part is generated in the management structure mentioned above, the operation phase part is generated in the operation structure and is related to the design phase part. What is important here is to build a mechanism not only for extracting knowledge from information and know-how but also for enhancing its quality. For example, there is a mechanism for removing old knowledge and increasing the priority of valuable knowledge by continuously evaluating and reflecting the indicators of the knowledge such as its effectiveness and freshness. This structure for system knowledge resolves technical problem 4).

4. Operation process

This section gives an overview of the OC process, which is a major component of this architecture.

4.1 Role and actor

The OC process is a set of processing to achieve the target of an operation management task (**Figure 3**). It consists of a mission description, including the goals and constraints on the process; the roles of the persons responsible for accomplishing the mission, which consist of their authorities and skills; and a detailed description of the activities for achieving the mission (e.g., the processing, procedures, and targets of operations). In the OC process, defined activities are performed by an actor with the authority and skill specified for the role. The actor is defined as the autonomous system that performs monitoring and operations or a person that represents individuals or groups. The OC process itself is not affected by the actor. In other words, the results are the same irrespective of which autonomous system performs the OC process or who executes it. By replacing the actor with an IT system in which humans are not engaged, it is easy to shift to autonomy without re-defining the OC process. However, if an accident occurs in the autonomous system (IT), operations can be performed based on the OC process by switching the actor from the autonomous system to the human system. Introducing this structure especially resolves problems 2) and 3).

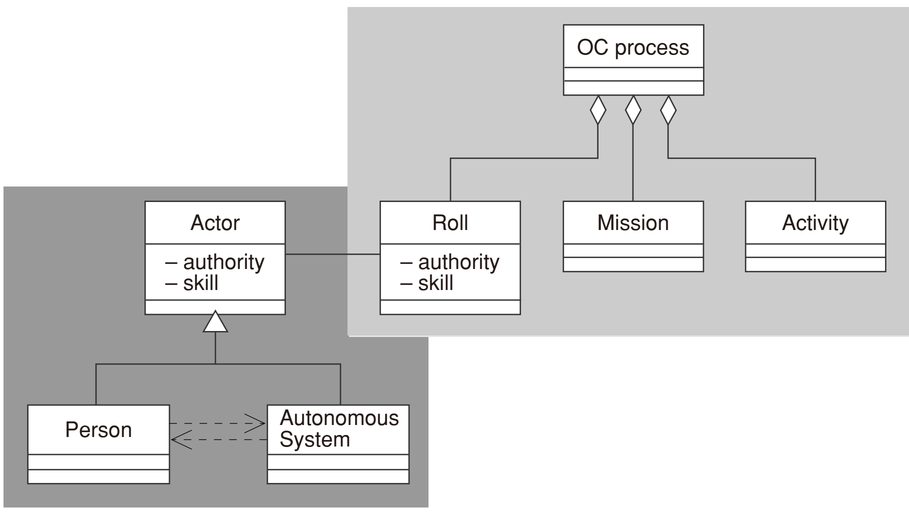

Figure 3
Class diagram of OC process.

4.2 Policy and template

A deep understanding of the architecture and many person-hours are required to describe the OC process from scratch. Therefore, we have been developing a technology for automatically generating the OC process in a simple manner by preparing an OC template that defines the common part of the operation management as a part and inputting a policy that specifies the operation guidelines. For example, when generating a server scale-out procedure in which a load balancer and additional servers are coupled, an OC template that describes the operation procedure by abstracting the commands according to each resource is selected. Then, a policy is created that includes resource information such as the manufacturers, model numbers, and OSs of the IT system components and the performance and availability requirements. Additionally, the template not only reduces the person-hours by using parts but also includes additional functions for enhancing the system reliability (i.e., functions for confirming procedure executions).

It is indispensable to install tools for verifying the operations of the generated OC process, detect conflicts between OC processes that are performed simultaneously, and detect redundancy and defects.

4.3 Verification based on case examples

We analyzed the patch applying procedure of an Internet provider to verify the structure and descriptive property of the OC process. We found that the patch applying procedure could be constituted from 18 OC processes with a four-level hierarchy and that a defined role and actor could be linked with an actual operator. Regarding the OC template, we found that about 30% of the operation work in the data center could be extracted as a common process.

5. Configuration management database

This section describes the configuration management database (CMDB), which is based on the system knowledge of the TRIOLE organic computing architecture. In ITIL, a CMDB is defined as a database having information required for operation management of an IT system. Fujitsu has developed a resource configuration management description language called RCXML as the base data model of the CMDB for managing configuration and design information.

5.1 Resource configuration management description language (RCXML)

This language can describe all the information required for managing an IT system throughout its lifecycle, from the budget management phase to the operation/maintenance phase, according to the concept of system knowledge. Conventionally, operation management software works independently and defines/maintains the contents and structure of managed data in its proprietary specifications. RCXML provides a unified content and structure for managed data, so data can be exchanged smoothly among the operation management software by using RCXML as a common data representation.

The major data processed by RCXML is classified into five types: physical resource information, logical resource information, design information, active service information, and user information (**Figure 4**). The physical resource information is about the system devices and is managed by dividing it into information about hardware and information about software. The logical resource information is about the virtualized physical resources and their correspondence to physical resource information. This information maintains the domain information (grouped logical resources) so only the required amount of resources are allocated from the resource pool. The active service information is about the system in the operation phase. The design information is about the resource configuration and operation policy required by each service. The user information is about the service providers and center administrators. Operation management software manages IT systems by using and linking these five types of data.

5.2 Configuration management database (CMDB)

The CMDB is constructed by integrating multiple distributed databases virtually. To achieve this integration, it must specify a common model and interface for the data that is exchanged among the operation management software and must develop the following fundamental technologies: a reconciliation technology for merging data from the same resource that is distributed among multiple databases, a discovery technology for automatically collecting the configurations of resources, and a Web service technology for exchanging data among heterogeneous platforms. We are now establishing an open industry specification for these technologies with other companies.

We developed a system infrastructure for troubleshooting by using the CMDB based on the TRIOLE organic computing architecture. In this system, existing know-how for troubleshooting is transformed into knowledge in the form of symptoms and accumulated in the CMDB. A symptom is a procedure for resolving a trouble that occurs in the target system and consists of two major processes and various minor processes. The two major processes are a process for identifying the cause of a problem (cause identification flow) and a process for deciding appropriate actions for the cause (action selection flow). These two major processes call appropriate minor processes if they need to obtain more specific information or analyze in more detail. For example, a log reference process can obtain event logs and associate a trouble with the related event logs. A diagnostic process identifies the root problem (e.g., the application server that caused the problem) by analyzing the target system comprehensively using other data such as the configuration data and performance data. This multilevel structure of symptoms classifies processes into common trouble action processes and system-specific processes, which allows them to be reused in other IT systems.

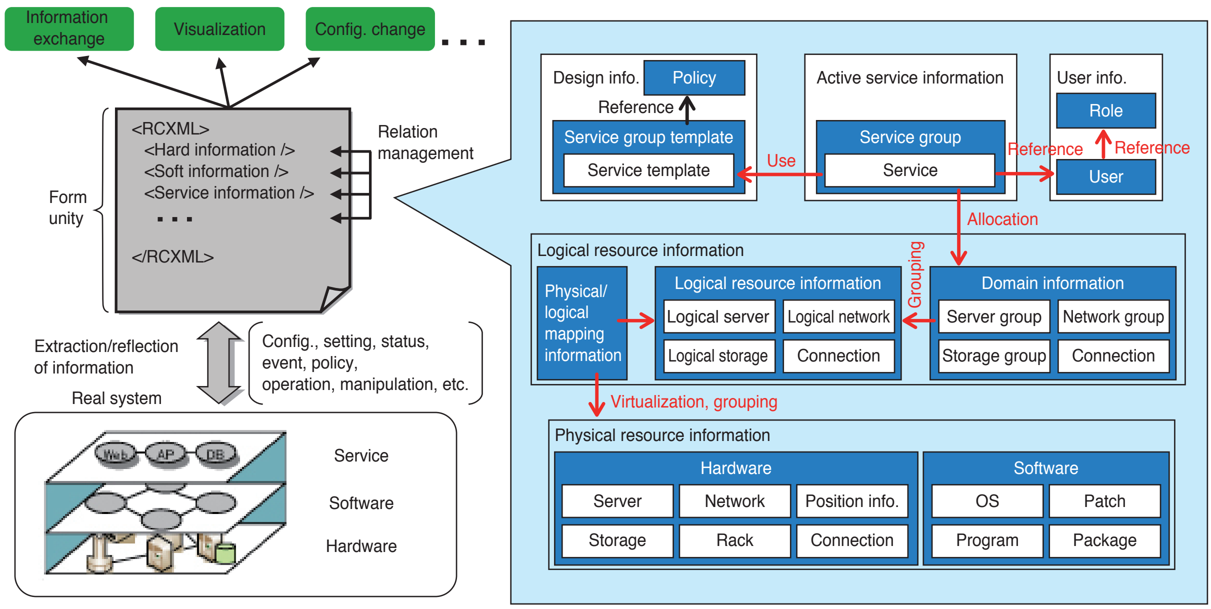

Figure 4
Data model of RCXML.

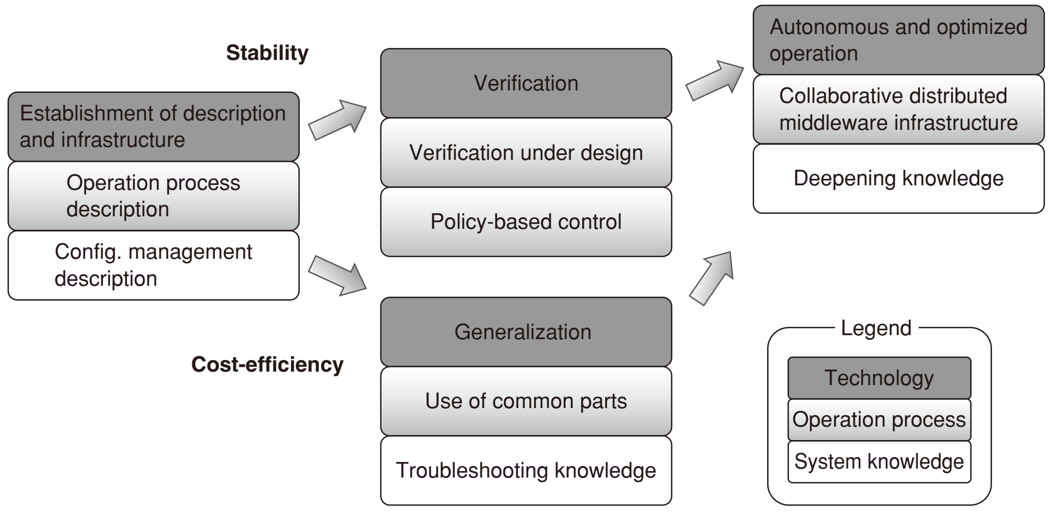

Figure 5
Roadmap.

6. Roadmap

Figure 5 shows the roadmap of the advantageous effect of the TRIOLE organic computing architecture. The operation infrastructure is considered to be the combination of the operation process (procedure) and system knowledge (data) for simultaneously pursuing stability and cost-efficiency. In the first step, as described above, we established the description and infrastructure for achieving the objective. Now, in the second step, we are developing verification technologies (verification under design and policy-based control) to achieve stability and are pursuing generalization (i.e., use of common parts and turning troubleshooting information into knowledge) to improve the cost-efficiency. In the final step, we will realize autonomous and optimized operation by using a collaborative distributed middleware infrastructure and technologies for deepening knowledge.

7. Conclusion

This paper outlined the basic structure of an IT autonomous infrastructure called the TRIOLE organic computing architecture that integrates human activities and IT processes, system knowledge supporting the architecture, and an organic computing process (OC process), which is one of the architecture's major components. It also introduced the configuration management database that forms the common infrastructure of the architecture and the roadmap of technologies for achieving stability and efficiency.

We will now develop a prototype of this architecture, promote demonstration experiments, and apply the architecture to operation management products that will be released in April 2008. It is indispensable that we work with other companies to improve the autonomous function of entire IT systems. Fujitsu is actively creating standards in this field in cooperation with IBM. Through these activities, we will make further contributions to the evolution of IT systems.

Reference

1) ITIL.
http://www.itsmf.org/

Akira Katsuno, *Fujitsu Laboratories Ltd.*
Mr. Katsuno received the B.S. degree in Applied Physics from Osaka University in 1985. He joined Fujitsu Laboratories Ltd. in 1985, where he was engaged in research on LSI design and technology. Since 1993, he has been working on the development of enterprise server systems. His research interests include computer architecture and management system.

Satoshi Tsuchiya, *Fujitsu Laboratories Ltd.*
Mr. Tsuchiya received the B.S. and M.S. degrees in Information Engineering from Tohoku University in 1989 and 1991, respectively. He joined Fujitsu Ltd. in 1991, where he was engaged in planning and business development of software. He moved to Fujitsu Laboratories in 1997, where he has been engaged in research and development of network computing. His current research interest is autonomic system management. He is a member of Information Processing Society of Japan (IPSJ).

Motomitsu Adachi, *Fujitsu Ltd.*
Mr. Adachi received the B.S. and M.S. degrees in Electronic Engineering from Waseda University, Tokyo, Japan in 1977 and 1979, respectively. In 1979, he joined Fujitsu Laboratories Ltd., Kawasaki, Japan, where he was engaged in research and development of communication services. From 1987 to 1989, he was engaged in research and development of heterogeneous communication network architectures and services as a visiting researcher at Bellcore, USA. He is currently developing enterprise management systems in Fujitsu Ltd. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan and the Communication Society of the IEEE.