통제 아키텍처를 이용한 정보자원 관리

김정욱[†]

세종대학교 경영대학

Information Resource Management Using by Integrated Control Architecture

Jeong-Wook Kim[†]

School of Business Administration, Sejong University

Key Words : Control Architecture, Information Resource Management

Abstract

Since management of information resources is getting more complicated in the distributed, heterogeneous computing environment, the capability of monitoring and controlling the dispersed information resources is perceived as a critical success factor for the effective enterprise-wide information resource management. Integrated Control Architecture(ICA) provides that capability. Utilizing such architecture, we can manage corporate information resources more efficiently, perform impact analysis for changes in information resources, and alleviate the human effort by automating the monitoring of critical information resources. In this paper, we propose a conceptual framework and metamodel of ICA.

1. Introduction

Efficient management of information resources becomes more difficult in the distributed computing environment as individual information resources such as data and applications are distributed across the increasingly complex web of heterogeneous hardware and networks. Information Resource management (IRM) originated from the idea of perceiving information and information-producing mechanism such as hardware and software as an important corporate resource. In general, IRM is understood as the set of activities that maximize the enterprise-wide utilization of information resources by effectively combining the current information technology with proper management policies and techniques. March and Kim [1988] define IRM activities as “diverse activities including strategic planning, capacity planning, information system development, project management, hardware and software acquisition, and data administration”.

For successful implementation of IRM activities, understanding of the target information resource is necessary. Information systems architecture (ISA) facilitates this understanding by providing IS managers with various graphical models of information resources and metadata handling capability through powerful repositories. Building an IS architecture was the number one issue among the ten most critical IS management issues as perceived by the IS executives [Niederman, Brancheau, and Wetherbe, 1991].

---

† 세종대학교 e-business 연구소 jukim@sejong.ac.kr

Zachman [1987] first formally introduced the framework of IS architecture as the two-way matrix consisting of different views (e.g., owner's view, architecture's or designer's view, builder's view) and different target resources (e.g., data, fun ction, network). Since then, various approaches and perspectives were taken by other researchers to build and implement alternative IS architecture frameworks covering both the traditional and distributed computing environments [Sheer, 1991; Sowa and Zachman, 1992; Umar 1993, Kim and Everest, 1994; Bauer et al., 1994; Everden, 1996].

Most of these architectural frameworks and related commercial tools, however, focus on optimizing individual components of the IS architecture such as data, application, hardware, network and lack the fundamental capability of controlling these information resources in an integrated way. In this paper, we propose a conceptual framework for Integrated Control Architecture(ICA) that provides such capability. Utilizing such architecture, we can manage corporate information resources more efficiently, perform impact analysis for changes in information resources, and alleviate the human effort by automating the process of monitoring information resources dynamically. ICA, when successfully implemented, will help manage distributed information resources on heterogeneous platforms as if they are centralized on a homogeneous platform.

The rest of the paper is organized as follows: first, literature review section will examine various conceptual frameworks for IS architecture and compare them on a set of given criteria. Second, one of the IS architecture frameworks is chosen and refined in the control perspective. Third, the rationale for building a control architecture is discussed in terms of its objectives and benefits. Fourth, the model of integrated control architecture, at two levels (logical and physical), is proposed. Fifth, the currently available IRM tools are analyzed from the integrated control perspective. Finally, the conclusion and future research directions are discussed.

2. Literature Review

2.1 Analysis and Critique of Current IS Architecture Frameworks

The primary purpose of IS Architecture is to provide the direction of planning, designing, implementing, and managing the individual information resources. Table 1 shows the comparison of various IS Architectures frameworks.

Sheer [1991] suggested the Architecture of Integrated Information System (ARIS) consisting of organization view, data view, control view, function view, and resource view to analyze and develop the new enterprise model. He utilized the Entity Relationship(ER) diagram to show the interrelationships among individual views, with focus on the control view. What the control view pursues is to integrate the management component of individual views. The control view communicates with other views through exchange of control information over organization unit, information object, functions and information technology resources. His control view, however, neither shows the dynamic states of information resources nor supports the impact analysis for changes in information resources.

Sowa and Zachman [1992], based on Zachman [1987], suggested an IS architecture framework viewing an information system from five different perspectives in six dimensions of the real world. Control structure is mentioned as time dimension of the technology model composed of time and cycle. Their architecture is too broad and complex to be fully implemented because it consists of thirty independent cells to be filled. It takes non- trivial amount of effort to create and maintain all the diagrams and metadata for each level. It has also limitations on two of its major dimensions : function dimension includes only information processing without considering cross functional business processes, while network dimension does not include processing components such as client or server but only defines network components such as node and link. The control architecture as a monitoring and management function is not supported, preventing the impact analysis over the related information resources.

Umar [1993] proposed a distributed computing reference model built from the functional layers of enterprise systems, application systems, and platforms. The enterprise system represents the business process and functional activities across the organization. Application system consists of a user database, programs to access and manipulate the database, and user interfaces to execute the programs. Platform corresponds to the technology architecture of other architecture frameworks. Management and support component is concerned with organizational management as well as the technical tools and techniques needed to administer distributed computing services, which is similar to the control architecture as proposed by Kim and Everest [1994]. His model does not provide the detailed information of control architecture because it exclusively focuses on the network management function without considering the other information resources such as database and applications. It covers only the planning level of information design, not connected with the implementation level. It is also not capable of performing the impact analysis over related information resources.

Kim and Everest [1994] framework includes the cross reference matrices between one architecture and another architecture to facilitate the integration of diverse information resources. They expanded Zachmans [1987] network stream to a more broad technology architecture and also added the temporal control aspects of all information resources to support the dynamic IRM environment. They added the control architecture, which represents business dynamics over time, to the existing IS Architecture. While they used the cross reference matrices to express the relationship among information resources, the matrices may not explain sufficiently the detailed components of control architecture. Since it only expressed the logical level of control architecture, there is no ground to implement the control architecture and a bridge between the logical and physical levels of the control architecture is nessary.

Bauer et al. [1994] suggested an integrated management architecture for distributed systems. This architecture defined all three classes including net

<Table 1> The comparison of various IS Architectures

	Scheer	Sowa & Zachman	Umar	Kim & Everest	Bauer	Everden
Component	Data, Function, Organization, Resource, Control	Data, Function, Network, People, Time, Motivation	Data, Application Technology, Control	Data Process Technology, Control	Network Operating System Application	Organizatio, Business, Technical
Levels of Details	Requirement Designing Implementation	Planner, Owner, Designer, Builder Sub-Contractor	Planner, Owner, Designer,	Planner, Owner, Designer,	Builder	Deconstruction Composition Operation Bound
Design Level	Planning Designing Implementing	Planning Designing Implementing	Planning	Planning Designing	Implementing	Planning Designing Implementing
Expression of Control	Control View	None	Management & Support	Cross Reference Matrix	Management tools Services & Agents	Version Control
Control Concept	Control View With Levels	Control structure in Time	Management	Control Architecture with Temporal View	Integrated Management	Transformation over Time
Impact Analysis	No	No	No	Yes	No	No

work services and devices, operating systems services and resources, and user applications. Its major components are management tools, management services, and management agents. The management services as a key component are classified into the following four subsystems: configuration, monitoring, control, and management information repository subsystems. Their architecture does not include the logical model for people who have the responsibility to integrate the individual resources. A bridge is needed to connect the logical and physical model of control architecture so that the change of business environment can be accommodated easily by the system managers. It controls the individual information resources separately by using management services but does not explicitly manage the relationships among individual information resources.

Everden [1996] suggested an information framework which is broader in objective and scope than Zachmans architecture. Including the six columns and five perspectives of Zachmans information system architecture, the information framework defined three views based on types of information such as organization view, business view, and technical view, and three levels of constraint such as deconstruction level, composition level, and implementation level. Each view is classified with ten columns, each level is also subgrouped with five rows. To represent architectural perspectives using different parts of the architecture, he suggested six dimensions; types of information, levels of constraint, content, transformation over time, ownership, methodology chains or routemaps. It can not manage information resources in an integrated way although it represents them with versioning feature of the time dimension. Therefore, the impact analysis is not available in this framework and only the individual states of the six dimensions can be traced with version control.

2.2 Refined Framework of IS Architecture

From Table 1, we chose Kim and Everests [1994] IS architecture as a basis for our integrated control architecture since it supports the concept of control architecture more directly. As shown in Figure 1, the refined framework of IS architecture is characterized by the integration of individual architectures, centered on control architecture. It represents the control architecture at two levels : logical level and physical level. Control architecture consists of two constructs : control objects (individual information resources) and control relationships. Examples of control relationships are relationships among homogeneous information resources such as data-data, process-process, technology- technology and relationships among heterogeneous information resources such as data-process, data-technology, process-technology.

Logical views of IS architecture deal with data, process, and technology architecture as shown in Kim and Everest [1994]. Data architecture provides a high-level, global view of organization data resources, actual or planned. Data architecture includes the important metadata types such as entity condition, entity, instance, attribute, domain, key attribute, non-key attribute and identifier. Process architecture focuses on the ways how major business processes of an organization relate to each other. Process architecture has the metadata types such as enterprise goal, strategy, business problem, business process, business function, business unit, and information requirements. Technology architecture specifies the organizations computing and communication technology standards and shows how the computer hardware and communication network facilities configure, interconnect, and integrate toward open systems.

Physical views of IS architecture deal with database architecture, application architecture, and platform architecture. While the logical view deals with the business domain, the physical view is more closely associated with the computing domain. Database architecture consists of actual entities of database such as data table, record, field, database key, value, index, domain and it is used by the system developers and database administrator (DBA). Application architecture is the physical implementation of the process. It consists of entities such as IS application, module, module program library, where each module is classified further into data module, process module, and user interface module. Platform architecture specifies the physical components of technology architecture. It has entities such as physical network, network device, client, server, and system software.

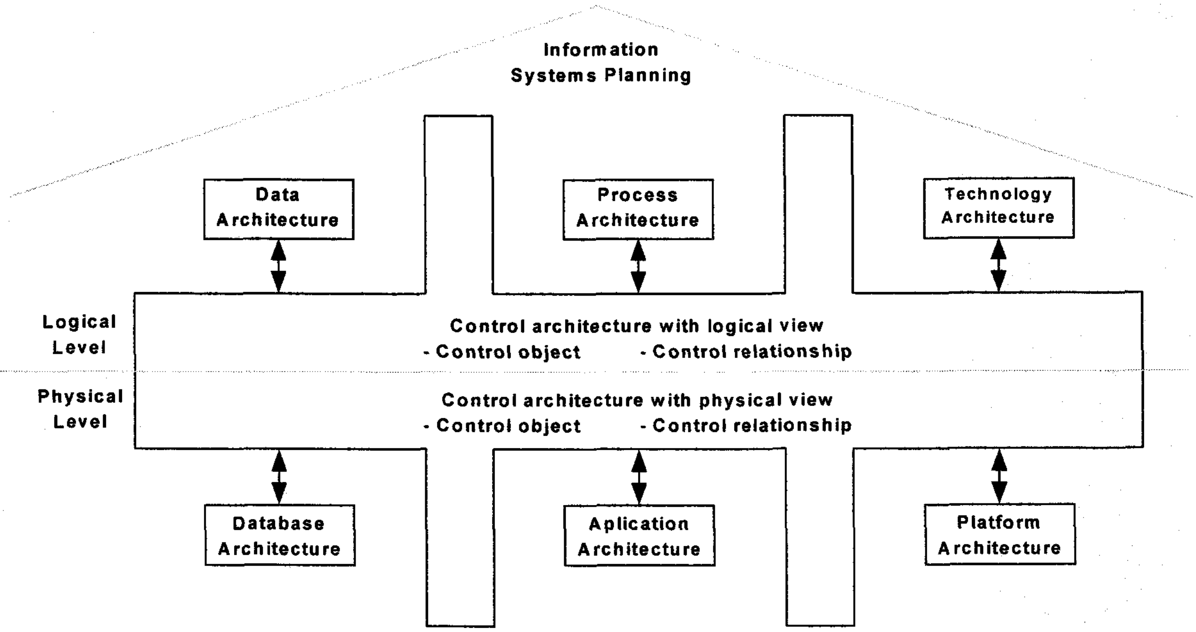
3. META-Model of Integrated Control Architecture

3.1 Components of Control Architecture

The metamodel of control architecture consists of two subarchitectures : control object architecture to monitor and manage individual information resources, and control relationship architecture focusing on the impact analysis between information resources of the same or different types. These two subarchitectures deal with static control using the blueprint of information resources at one point in time, and dynamic control continuously monitoring the transitive states of information resources over time. The following subsections will describe the logical and physical views of the control object architecture and control relationship architecture, respectively. Teorey, Yang, Frys [1986] extended entity-relationship modeling formalism was used to develop the metamodel of the control architecture constructs.

3.2 Control Object Architecture : Logical View

Control object architecture consists of two dimensions : the first dimension represents the goals, strategies, and types of control to be implemented while the second dimension models the target information resources to be controlled. As shown in Figure 2, the left-half portion of the model represents the control methods and the right-half reveals the target information resources. This figure shows which information resources are managed by which control methods. CONTROL OBJECT has subtypes such as DATA ENTITY, BUSINESS PROCESS, and TECHNOLOGY that are the target in-

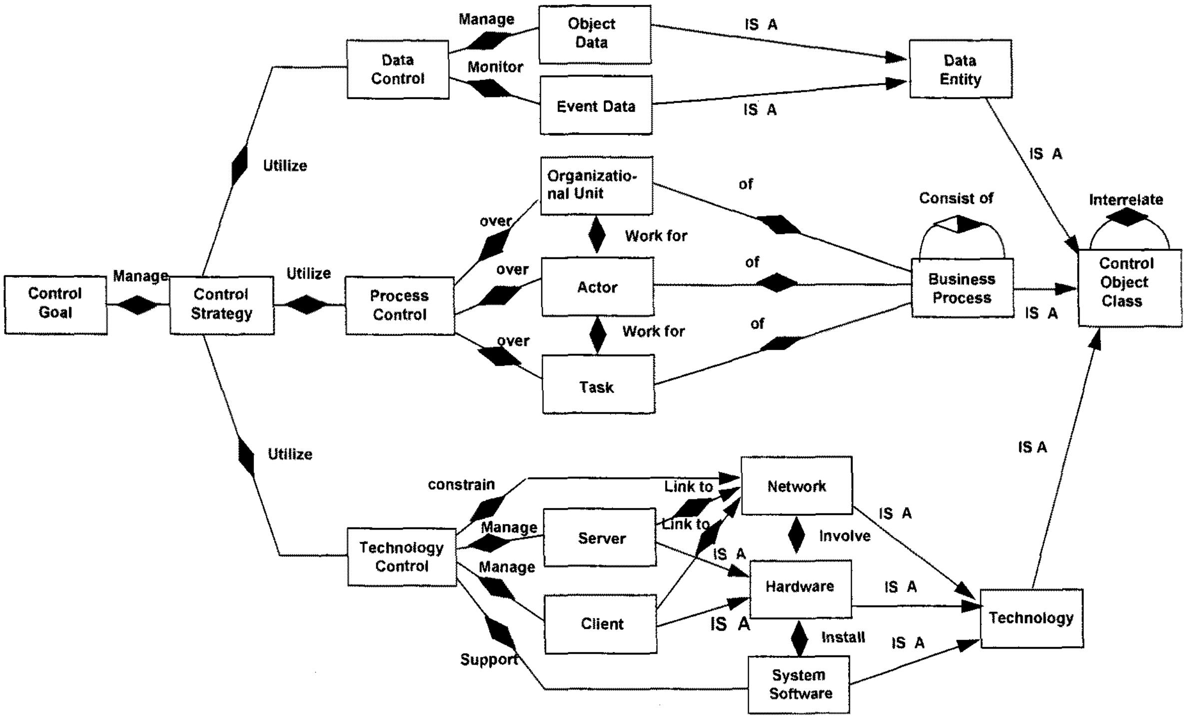


<Figure 1> A Refined IS Architecture Framework

formation resources of the control architecture. DATA ENTITY is categorized into either OBJECT DATA or EVENT DATA. OBJECT DATA such as customer or product is stable over time without frequent changes. EVENT DATA such as customer arrival or order status is unstable and changes its state frequently over time. Each OBJECT DATA and EVENT DATA are monitored and managed by the DATA CONTROL tools and techniques such as DBMS, data dictionary, normalization, and reorganization.

BUSINESS PROCESS is mostly cross-functional in nature and consists of one or more SUB-BUSINESS PROCESSes at lower levels. Lowest level BUSINESS PROCESSes consist of one or more functional tasks performed by various types of process actors (customer, owner, manager, executive, etc.) working for different functional units. PROCESS CONTROL over ORGANIZATIONAL UNITS may include monitoring of the units performance and conformance to the companys internal rules and procedures regarding the execution of the related BUSINESS PROCESSes. PROCESS CONTROL over ACTORs may involve their fitness of process role as well as their performance. Examples of PROCESS CONTROL over TASKs are coordination efficiency and effectiveness as well as individual task performance.

TECHNOLOGY entity has subtypes of NETWORK, HARDWARE, and SYSTEM SOFTWARE. A NETWORK involves HARDWARE components such as CLIENT or SERVER. Each of which can participate in more than one NETWORK. Each NETWORK has one or more SYSTEM SOFTWAREs installed, while the same kind of SYSTEM SOFTWARE can be installed at more than one NETWORK. Each NETWORK operates under the TECHNOLOGY CONTROL in the form of constraints or checklists. TECHNOLOGY CONTROL constraint is a criterion or threshold value to manage NETWORKs properly. CLIENT and SER VER also operate within their limits of performance and capacity as specified and monitored by the TECHNOLOGY CONTROL mechanism. SYSTEM SOFTWAREs such as operating systems, DBMS, and network ma



<Figure 2> Control Object Architecture : Logical View

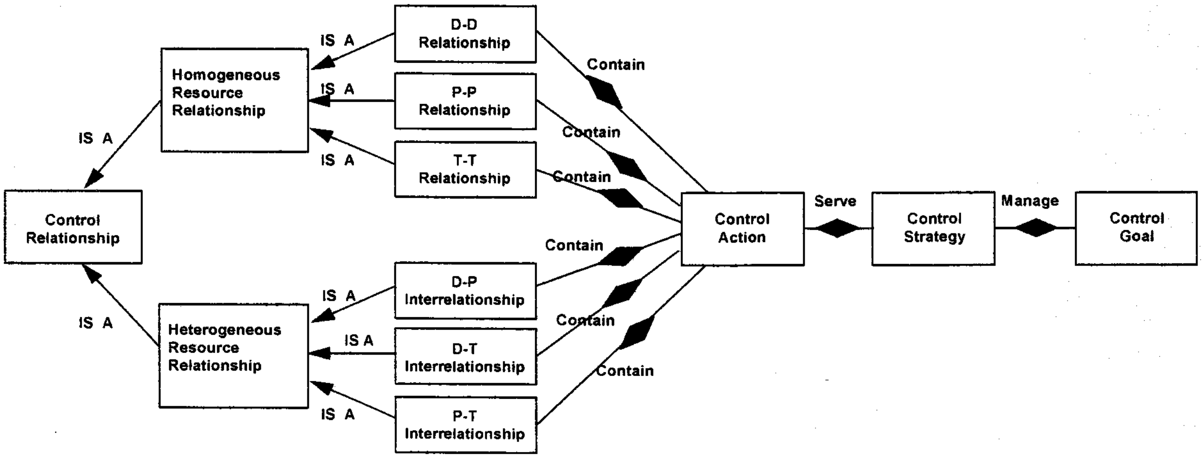
nagement systems are supported by the TECHNOLOGY CONTROL aspects of standardization, version control, and configuration management. At the same time, they provide TECHNOLOGY CONTROL with performance monitoring and integrity checking to manage other network components.

Enterprise CONTROL STRATEGY utilizes the above control mechanism such as DATA CONTROL, PROCESS CONTROL, and TECHNOLOGY CONTROL tools and techniques. These CONTROL STRATEGies are derived from and support the enterprise-wide CONTROL GOALs.

3.3 Control Relationship Architecture : Logical View

CONTROL RELATIONSHIP specifies how one information resource is related to another information resource. The Control Relationship Architecture consists of two important dimensions : the first dimension represents the relationship between/among information resources, while the second dimension shows the goals, strategies, and actions of control in managing information resources. The control relationship has two subtypes : HOMOGENEOUS RESOURCE RELATIONSHIP and HETEROGENEOUS RESOURCE RELATIONSHIP. The HOMOGENEOUS RESOURCE RELATIONSHIP is composed of three main components such as D-D RELATIONSHIP, PD-D RELATIONSHIP is the relationship between data resources. For example, if an order is deleted, the related order items should also be automatically deleted (cascaded delete) or be requested for deletion to guarantee the referential integrity. In another example, if a certain primary key is deleted, the corresponding field value has to be requested for deletion to guarantee entity integrity. P-P RELATIONSHIP is the relationship between business processes. The nature of such relationship may be sequential if execution of those process follows a certain sequence, selective if the process flow changes based on a conditional variable, and repetitive if the same process is executed repeatedly. T-T RELATIONSHIP is a relationship between technology components such as client and server. For example, if a client requests certain action to its server, the server answers the client in the form of a “response”.

The HETEROGENEOUS RESOURCE RELATIONSHIP has three subtype relationships such as D-P RELATIONSHIP, D-T RELATIONSHIP, and P-T RELATIONSHIP. D-P RELATIONSHIP is the relationship between data and business process. Mutual relationships between data architecture and process architecture are organized by the operations of four commands such as “Create, Read, Update, and Delete” [Martin,1990]. “C (Create)” means that the identified process produce the data class, while R, U, D



<Figure 3> Control Relationship Architecture : Logical View

means read, update, and delete, respectively. D-T RELATIONSHIP is the relationship between data and technology. It can explain which client or server influence which data, and what degrees of dependence each component of client or server has. Unique database suggests that only one copy is stored in the central server. Replicated database means that copies of a database are stored in multiple clients or servers. Partitioned database implies that partitions of a database are stored at different servers or clients. The criteria of determining whether the network traffic is high or low depends on the hardware and application type. P-T RELATIONSHIP is the relationship between business process and technology. It is organized according to which process utilizes which technology components as a resource. "S(Store)" means that a component of process architecture is stored at the corresponding technology component. "E(Execute)" means that the target application processed at the technology component such as processor or system software. "SE(Store and Execute)" suggest which components of technology architecture are used store and execute which process.

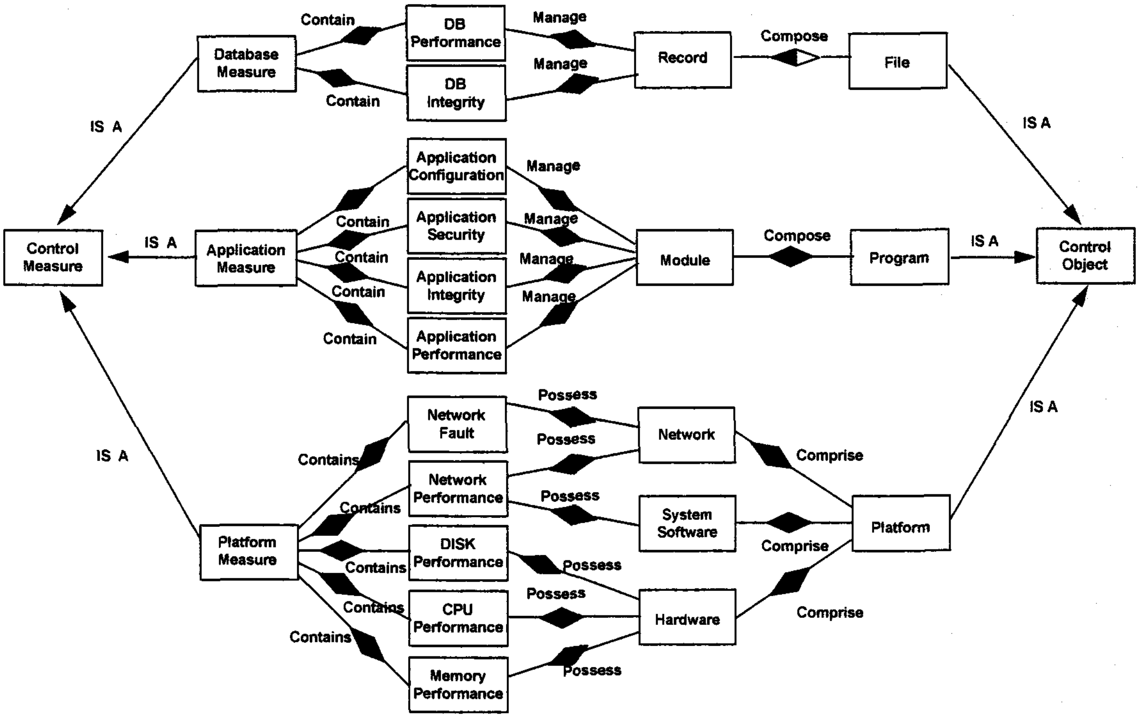
As shown in Figure 3, each RELATIONSHIP contains one or more CONTROL ACTIONs such as repository functions and monitoring functions. Each CONTROL ACTION can participate in more than one RELATIONSHIP. For example, in the case of D-D RELATIONSHIP, the CONTROL ACTION on referential integrity is applied to all data entity pairs where dependency exists between them(e.g., CUSTOMER and ORDER, ORDER and INVOICE, etc.). Each CONTROL ACTION serves one or more CONTROL STRATEGIES. For example, CONTROL ACTIONs to monitor and implement the database referential integrity are derived by the specific CONTROL STRATEGY such as cascade delete. Each CONTROL STRATEGY is the policy or procedure to achieve one or more CONTROL GOALs. For example, a CONTROL GOAL of "maintaining 95% data integrity" requires the CONTROL STRATEGY of cascade delete.

3.4 Control Object Architecture : Physical View

As shown in Figure 4, physical view of a control object architecture consists of two dimensions : the first dimension represents the control measure, while the second dimension models the target information resource at the physical level (e.g., application, database, platform). The main difference between logical view and physical view is the difference in the level of abstraction with logical view constructs more abstract and general and physical view constructs more concrete and computerized. If the control object and control policy are fixed, CONTROL MEASUREs may be expressed as deterministic factors. CONTROL MEASURE has three subtypes of DATABASE MEASURE, APPLICATION MEASURE, and PLATFORM MEASURE.

For example, the DATABASE MEASURE for performance includes the database environment parameters such as maximum process, block size, table space, segment size, caches, fragmentation size, locks, and disk allocation, statistics, fragmentation of table, etc. Another main category of DATABASE MEASURE is DB INTEGRITY. Each DATABASE MEASURE can be related to one or more DB INTEGRITY constraints such as entity integrity constraint or referential integrity constraint. The major control factors of the overall database system performance include throughput, response time, error rate, ease of use, ease of maintenance. DB PERFORMANCE and DB INTEGRITY control mechanisms manage physical files and records.

An APPLICATION MEASURE is related to one or more APPLICATION CONFIGURATION variables such as application component, application version, version hierarchy, parameter, dependency between application systems. APPLICATION MEASUREs for APPLICATION PERFORMANCE include response time, SQL usage, data structure efficiency, etc. APPLICATION MEASUREs for APPLICATION SECURITY are used to manage access priority, audit trail, user authorization, etc. APPLICATION MEAS



<Figure 4> Control Object Architecture : Physical View

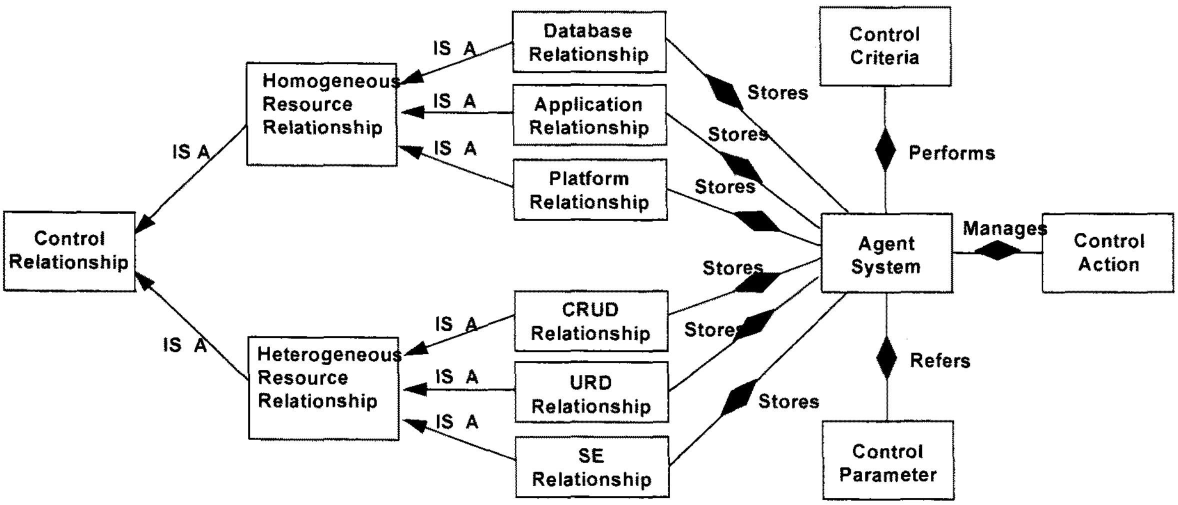
UREs for APPLICATION INTEGRITY include measures related to fault detection, fault isolation and resolution using mechanisms such as error or warning messages.

These control measures on CONFIGURATION, SECURITY, PERFORMANCE, and INTEGRITY are implemented on PROGRAM MODULEs which are basic building blocks of IS applications. PLATFORM MEASUREs contain measures to monitor and control network fault, network performance, disk performance, and memory performance. PLATFORM MEASUREs for NETWORK FAULT deal with fault identification, fault diagnosis, fault notice and resolution. PLATFORM MEASURE for NETWORK PERFORMANCE include information on network traffic, network load, number of users, average response time. PLATFORM MEASUREs are also used to manage DISK PERFORMANCE (e.g., utilization, access rate, input/output speed), CPU PERFORMANCE (e.g., average utilization, peak utilization, parallel processing capacity), and MEMORY PERFORMANCE (e.g., available memory, paging/swapping rate, memory size). Diverse sets of NETWORK, SYSTEM SOFTWARE, HARDWARE components can form different PLATFORMS where each instance of the physical CONTROL OBJECTs are governed by the technology architecture standards of the logical level.

3.5 Control Relationship Architecture : Physical View

Physical views of the control relationship deal with the same target relationships as in Figure 3 only at a lower abstraction level. At this level implementation tools such as the agent-based control system may be used to monitor and manipulate the homogeneous or heterogeneous relationships. Such system compares control data periodically, checks against the prespecified thresholds, and if necessary, produces event or warning statements.

In Figure 5, the left-half entities of the model are the same as in Figure 7. In the right-alf, CONTROL CRITERIA are put into memory for the AGENT SYSTEMs. For example, response time of a client is defined as an important CONTROL CRITERIA to meet the end-user requirement. CONTROL



<Figure 5> Control Relationship Architecture : Physical View

PARAMETERs are used by the AGENT SYSTEMs. For example, five seconds as a parameter for the client response time is needed to implement the control relationship between the client and server. Each AGENT SYSTEM performs one or more CON TROL ACTIONs (for example, if the response time is over five seconds, the agent system sends the alarm signal to the operator).

4. Conclusion and Future Directions

To manage diverse information resources such as data, software, hardware, and network more effectively, the Integrated Control Architecture (ICA) was proposed in this paper. ICA is based on and extends the active stream of research in developing and managing enterprise-wide information syst ems architecture. Features of ICA that make it dis tinct from the existing architectures are its use of two level (logical and physical) IS architecture and adoption of control as the central mechanism to manage and support the other information resource dimensions. By differentiating the logical, business level architectures (data, process, technology) from the physical, implementation level architect ures (d atabase, application, platform), ICA facilitates the effective management of not only computerized information resources but also various relat ed business resources. By emphasizing the control over relationships among diverse information resources over time, ICA addresses the issues of pro viding dynamic and integrated control of informa tion res ources. None of the current crop of commercial sy stems management tools provides such capability. In the future, ICA framework will be ext ended to include the intelligent agent services and an experimental prototype system will be implem ented to demonstrate the capability and features of ICA discussed in this paper.

References

[1] Amjad Umar, Distributed Computing and Client-Server Systems, Prentice-Hall, 1993

[2] August-Wilhelm Scheer, Architecture of Integrated Information Systems : Foundations of Enterprise Modeling, Springer-Verlag,1992

[3] Bauer, et al. "Reference Architecture for Distributed Systems Management", IBM Systems Journal, Vol.33, No.3, 1994, pp.428-444

[4] C.J. Date, An Introduction to Database Systems, Addison-Wesly, Sixth Edition, 1995

[5] E.G. Flamholtz, T.K. Dasand, A.S. Tsui, "Toward an Integrative Framework of Oganizational Control," Accounting Organizations and Society, Vol. 10, No. 1, 1985, pp. 35-40

[6] F. Niederman, J. C. Brancheau and J. C. Wetherbe "Information System Management Issues in the 1990s," MIS Quarterly, December 1991, pp. 475-500

[7] G.J. Myers, The Art of Software Testing, Wiley, 1979

[8] H.C. Lefkovits, IBM's Repository Manager/MVS Concepts, Facilities, and Capabilities, QED Technical Publishing Group, 1991

[9] IBM Corporation, "Business Systems Planning ; Information Systems Planning Guide," Publication No. GE 20-0527-4, Armonk, New York, 1978

[10] J.A. Zachman, "A Framework for Information Systems Architecture," IBM Systems Journal, Vol.26, No.3, 1987,pp. 276-292

[11] J.F.Sowa, J.A. Zachman, "Extending and formalizing the framework for information systems architecture," IBM Systems Journal, Vol.31, No.3, April 1992, pp. 590-616

[12] J. Martin, Information Engineering , Vol. II : Planning and Analysis, Englewood Cliffs, 1990

[13] J.P. Thompson, Organizations in Action, MaGraw-Hill, New-York, 1967

[14] K. Jones, Managing Diverse Environment with CA-UNICENTER for UNIX, 1995, pp.138-156

[15] K.M. Eisenhardt, "Control : Organizational Economic Approaches," Management Science, Vol. 31, No 2, 1985, pp.134-149

[16] M. Frank, "Modeling Transaction Integrity," DBMS, January, 1993, pp.62-67

[17] R. Everden, The Information Framework, IBM Systems Journal, Vol.35, No.1, 1996, pp.37-68

[18] Salvatore T. March and Young-Gul Kim, "Information Resource Management : A Matadata Perspective," Journal of Management Information Systems, Vol.5, No.3, Winter 1988-89, pp. 5-18

[19] T.J. Teory, D. Yang and J. P. Fry, A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model, Computing Surveys, Vol.18. No..2, 1986, pp.197-222

[20] T.K. Reeves and J. Woodward, "The Study of Managerial Control," Industrial Organizations : Behavior and Control, Oxford University, London, England, 1970, pp. 37-56

[21] Weber, The Theory of Social and Economic Organization, Free, Glencoe, IL, 1947

[22] W.G. Ouchi, "The Relationship between Organizational Structure and Organizational Control," Administration Science Quarterly, Vol.22, No.1, 1977, pp. 95-113

[23] W.R.King, "Strategic Planning for Management Information Systems," MIS Quarterly, Vol.2, No.1, 1978, pp. 27-3

[24] Young-Gul Kim and Gordon C. Everest, "Building an IS Architecture : Collective wisdom from the field," Information and Management, Vol. 26, No.1, 1994, pp. 1-11

2010년 1월 13일 접수, 2010년 1월 25일 수정, 2010년 1월 28일 채택