I. Introduction

The improvements of computer and network technology enable the control systems to handle massive information in the distributed computing environments like other computing applications. To develop such a distributed control system, a middleware that provides the various services such as ORB (Object Request Broker) or event services is required. Using the middleware reduces the development cost and provides the compatibility between various control systems [1]. Even this kind of middleware provides various services to the developers, it is difficult to write domain-specific applications because the ordinary commercial middlewares are designed for the general Internet applications. With these basic middleware services, it is really hard to maintain the compatibility between the softwares from different vendors. To achieve the openness within the control domain, there are a couple of consortiums. OPC (OLE for Process Control) is a good example to use OLE-based middleware for the control domain [2]. It, however, defines only interfaces between the components rather than provides the runtime environments such as an event delivery service. There are some well-known basic middlewares currently available; CORBA (Common Object Request Broker Architecture) from OMG (Object Management Group), EJB (Enterprise JavaBeans) from Sun and COM/DCOM (Component Object Model/Distributed COM) from Microsoft [4,5,6].

This paper proposes a control domain specific middleware based on the Microsoft’s COM/DCOM, because it is integrated very well with Microsoft Windows 2000 platform that is used in this paper, and it provides more services than other commercial middlewares.

In this paper, the middleware that provides the enhanced services is introduced in the control system. It consists of the specification of equipment, distributed event service, the data format, and communication scenario. And this paper includes the implementation example to demonstrate its openness and flexibility. The implemented domain is the semiconductor-manufacturing domain based on the SEMI’s GEM protocol.

Chapter 2 introduces the middleware for components and communications. Chapter 3 explains the services for the control domain and the configuration of whole system. And the implementation of the prototype for GEM protocol is in chapter 4.

II. Control-domain Middleware

2.1 Component-base Middleware

The object-oriented methods have lots of problems such as insufficient reusability, code exposure, versioning difficulty, and deploying without the component-based development methodology. Furthermore, its white-box model inherently has the limitation of compatibility. To overcome these
problems, the component-based methodology uses the black-box model (interface based model). By the same reason, modern middlewares is designed based on the component-based methodology [3].

The lower layer API’s such as socket or RPC (Remote Procedure Call) have been used to develop distributed systems in the past. But these procedural methods require more time and cost to develop and maintain programs than object-oriented methods. Since the most of current component models support RMI, the component itself can be distributed over the network environment. This means that developers can write distributed control application more easily using these middlewares. Current trend is that these middlewares support services such as load balancing, resource management, message queuing, and event service [7].

2.2 XML-based Middleware

XML (eXtended Markup Language) is meta-language to describe other languages [8]. Since the under-lying philosophy of XML aims the flexibility and portability, it can be used in any platform. Thanks to its flexibility, the platform-independent RMI such as SOAP (Simple Object Access Protocol) appears recently. The most of platforms provide the parser for XML and SAX (Simple API for XML) engine [9,10].

XML has ideal characteristics to model the hierarchical data structure that is usually hard to model using 2-dimension database table. Its structure is suitable to construct configuration data of certain systems; therefore this paper uses XML to model configuration of systems and payload format of data packet.

III. System Configuration

3.1 The abstraction of equipment

The important requirements for the control systems are specifications to describe characteristics of the equipments. But it is very hard to model all the equipments with single specification. To get rid of this difficulty, instead of modeling all equipments with single specification, each function of equipment is modeled with one specification, and, further, an equipment is modeled as a group of functions.

In this paper, these specifications are implemented as the interfaces in COM/DCOM. Figure 1 shows the diagram of interface hierarchy for the read/write features [11].

![Figure 1 Device interfaces hierarchy diagram](image)

The most of equipments have four basic functions : (1) initialize, (2) configure, (3) set, and (4) acquire data; and these basic functions can be modeled as IBasicDevice interface. A component that represents an equipment can support these functions by implementing interfaces. And data consumer can use these functions as a query.

![Figure 2 Component that support interfaces of category](image)

Querying all interfaces to know whether a component implements the specific interface is very inefficient. To minimize the querying, a component category is used. Component category is a standard mechanism to advertise interfaces that is available in components in COM/DCOM environment. The component that describes specific equipment can tell its interfaces to users by registering to specific category. Users can use this information dynamically. Figure 2 shows examples of components that represent specific category.

3.2 Data format

There are two points to decide data format in proposed system. The first is the format for transferring data packet, and the second is one for configuration of equipment. Since the first data is likely to be transferred in network frequently, using XML for this format has the unnecessary large overhead. To reduce this overhead, a method that packs data into binary format is required. Packing method doesn’t, however, exist in XML standard. Although Microsoft partially uses ADTG (Advanced Data Tablegram) with ADO (Active Data Object), it is not adequate for these situations because it is developed to use with Recordset object [12]. Instead, transferring a group of data in one XML packet reduces this disadvantage. XML give the ability to write module that manages data with standardized manner in heterogeneous environment such as RPTID of GEM and PER format of ASN.1.
At the same time, data configuration is transferred only at the instance of initialization or modification of configuration data. Since it requires hierarchical structure, XML is suitable for these purposes. Figure 3 shows an example of configuration.

3.3 Event service

Event service is a mechanism that users can receive the signal change from the data producer. The simplest event service is polling. This method uses a querying mechanism periodically. This strategy is very simple to code, but it is terrible idea because it wastes much more CPU time and has long latency between the time that event is occurred and the time that user is notified. Therefore this strategy is not suitable for most of systems.

Figure 3  Example of device configuration XML

COM/DCOM infrastructure provides two mechanisms for event services. They are Connection Point mechanism and COM+ Event Service [13]. Connection Point is not suitable to be used in network environment because it is originally developed to be used within single process or machine boundary. And it requires the overlapping of execution between client and server. Since it, however, is compatible with automation mechanism such as script language, it can be used within single process or machine boundary. Though COM+ Event Service provides various services such as transaction and queuing, these services become overhead to be used in control system. With considering these natures of event services, this paper proposes a simple event service that has lightweight overhead. It uses TCP-based protocol between machines and uses Connection Point within machine boundary.

It can be used with various RTOS(real-time operating system) because it uses TCP and is easily used by scripting environment with Connection Point. Figure 4 shows the one-to-one matching between data supplier and user.

For example, CEID of GEM uses communication scenario that sends the data registered by users whenever it occurs. The protocol scenario like this can be ported into proposed event system, easily.

Generally speaking, it is difficult to apply location transparency of object in control system domain because specific component must be located in specific machine that has specific equipment. Mapping table between logical equipment name and physical location can be use to provide location transparency. Figure 5 shows the relation between equipments and host systems. The users do not have to know physical location of equipment.

3.4 System clustering

It is common to use the duplication of systems for stability. The clustering technology can be used to acquire the stability like duplication. The stability can be acquired by making a cluster with systems. And based on this cluster, it can provide various stable services by attaching service modules. The proposed system uses NLB (Network Load Balancing) feature of Windows 2000 platform [14]. Figure 6 shows the configuration of cluster.
IV. Implementation

To demonstrate the feasibility of the proposed architecture, it is implemented for a Samsung’s chip mount control system that is an example of distributed control system. To adopt high openness, the implemented system uses the standard protocol in the semiconductor industry, GEM.

SEMI (Semiconductor Equipment and Material International Inc.) introduced SECS (SEMI Equipment Communication Standard) protocol to reduce the cost and to improve communication ability between hosts and equipments. This protocol consists of SECS-I for message communication and SECS-II for message format. SECS-I has used point-to-point protocol such as RS-232. Recently, HSMS (High-Speed SECS Service) based on TCP/IP is used widely.

GEM (Generic Model for Communications and Control of SEMI Equipment) is the protocol that is used between semiconductor equipment and host computer on the top of SECS-I, II and HSMS. Since GEM uses semiconductor industry specific format, it is difficult to connect to general-purpose tools that is more cheap and convenience to use. And because GEM uses hierarchical data structure against flat data, it cannot be applied to general tools. In this paper, we apply GEM protocols to the designed system to verify its openness. We use Samsung chip-mounter that is used in semiconductor manufacturing as a testbed. With this implementation, we can reset or refresh the equipment and query various status variable. We can receive the current event of it, too.

The implemented system consists of two servers and some components. Two servers are able to initialize themselves with XML-formatted configuration data and also use XML for event delivery. The component for Samsung chip-mounter uses CEID for event delivery, and it can read or write status variables. And RPTID can be delivered using XML format. It can be verified by using Web browser and Web service module.

Although the prototype has one-to-one configuration between equipment and host, it can be many-to-many configuration. Attaching various service modules will increase the utilization of system.

V. Conclusion

In this paper, we design and implement an middleware that is designed for the distributed control domain. Using the proposed interfaces and categories increases the compatibility between the heterogeneous control applications from multiple vendors, and, as a result, modeling the heterogeneous equipments can be standardized. The event services implemented in this paper provide ease of use and lightweight infrastructure. Combining the proposed event services with real-time network and/or QoS (Quality of Service), developers are able to get these advantages transparently.

Since it is important to guarantee the compatibility in heterogeneous environment like Internet, the proposed system provides XML-based protocol and Web service modules for the communications between the different platforms. And it also uses NLB of Windows 2000 for the stable operation

References