34

# A STUDY ON DATA HANDLING MECHANISM OF A DISTRIBUTED VIRTUAL FACTORY

Kentarou Sashio[1], Susumu Fujii[2], and Toshiya Kaihara[1]
[1]*Graduate School of Science and Technology, Kobe University, Japan*
[2]*Faculty of Engineering, Kobe University, Japan*
*e-mail: sashio@kaede.cs.kobe-u.ac.jp*

Abstract: To cope with diversified consumers' needs, recent manufacturing systems are required to accommodate the agility in manufacturing by shortening the lead time, reducing the indirect costs and so on. To evaluate the performance of the total manufacturing system taking the complicated information flow as well as the material flow among areas into consideration, Distributed Virtual Factory (DVF) has been proposed. To construct DVF by integrating area level simulators, Communication, Synchronization and Data Handling Mechanism are required. The Data Handling Mechanism supplies external information required by area level simulators. In this paper, a Database Interface is developed to integrate databases into a DVF and the Data Handling Mechanism is implemented by utilizing database.

Key words: simulation, Distributed Virtual Factory, manufacturing, Database

## 1. INTRODUCTION

To deal with the diversification of demands in the market, manufacturing systems are strongly required to accommodate agility in manufacturing by shortening the lead time and reducing the indirect costs. Through utilizing rapidly developing information technology (IT), computer networks connect separate sections of a manufacturing industry as well as related industries and the time lags in the communication among them are reduced considerably. The effectiveness of simulations for evaluating manufacturing systems has been widely recognized and simulations at the area level, such as processing area, assembly area and so on, have been performed independently. The reduction of the time lags in communication, however, made the relationship among areas or sections closer and the mutual effect in the material flow among areas became difficult to neglect. This suggests the importance to model the total manufacturing system for the simulation, in addition to the independent simulations of areas.

To solve the difficulty to model a large manufacturing system, a Distributed Virtual Factory (DVF) has been proposed, which integrates existing area level simulators via computer networks [1,2]. Each simulator of an area level is executed on the computer of that area and all simulators cooperate with each other based on the concept of distributed simulation[3]. A large scale simulation system can be constructed efficiently by reusing existing area level simulators, and efficient execution is obtained by distributing loads of computing. To deal with the changes in manufacturing systems, only the reconstruction of the corresponding area level simulator in the DVF is needed, and thus the high flexibility for the changes in manufacturing systems can be realized by the DVF.

To develop DVF of real manufacturing systems, it is necessary to consider the difference of objectives and the difference of developing environments of area level simulators. In the former research work[4], the communication system among the simulators has been reported enabling to integrate the simulators developed under various OSs and by various simulation languages at the communication level. In this study, we focus to the difference of accuracy and information description (e.g. material name) among area level simulators and a data handling mechanism which supplies information appropriately is developed.

## 2. DISTRIBUTED VIRTUAL FACTORY

### 2.1 The Concept of the DVF

The DVF is developed by integrating existing area level simulators via computer networks and utilizing the concept of distributed simulation. In addition to area level simulators, a simulator of transportation system is required to represent the material flow among areas and a factory management system is required for the decision making for the whole manufacturing system, such as for the production control.

To utilize area level simulators as subsystems of the DVF, a mechanism for communicating with other simulators via computer network is required. Since the difference of simulation time among area level simulators arises during the process of simulation, a synchronization mechanism[5] which copes with the difference of simulation time is also necessary. In addition, to supply the external information which required by area level simulators for simulation of DVF, data handling mechanism is required. In this study, a set of these three functions is named as the communication interface.

In this study, we employ the Time Bucket Method for the synchronization[6] and the communication system by utilizing TCP/IP[4] which realizes data exchange between simulators developed on various OSs and simulation languages.

## 2.2 Problems on Integrating Area Level Simulators

Area level simulators are developed independently and the description of products and the detailness in the modelling are different. Each simulator assumes that the material arrival and the production orders to the area are externally supplied or are randomly generated at a source node based on the prespecified distribution and that the completed products are simply terminated at a sink node in the simulation after recording the production statistics as shown in Figure 1. In reality, the products terminated at an area A represented by the simulator A are transported to an area B and become source material there.

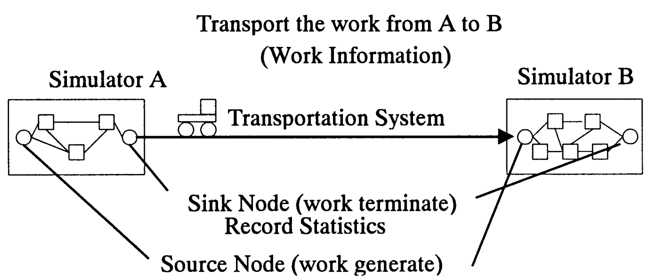


*Figure 1.* Construction of Area Level Simulators and Concept of DVF

In a DVF, this implies that the completed products need to have the information of the destination and the name in the destination area level simulator and to request the transportation to the transportation system. To supply these information, we develop a data handling mechanism to utilize the databases already developed in a real manufacturing system, such as those in MES, POP, PDM and so on. For simplicity, this study concentrates on only two kinds of information required for area level simulators in DVF. One is the destination area of material after all processes are completed in an area and its name in the simulator at the destination area. Other information includes the processing sequence and time, and so on, if they are not prepared in the simulation at the succeeding area, i.e., the destination area.

Data handling mechanism supplies the information by the following procedures.

For the preparation of simulation, a master database which includes information required for the data handling mechanism is developed by utilizing the existing database of the real manufacturing system. The master database consists of six tables i.e., the Production Instruction Table, the Process Information Table, Process Sequence Table, The Bill Of Materials Table, the Transportation O-D Table and the Rename table, as shown in Figure 2 including the following information.

- production schedules
- process information
- transportation information
- Bill Of Materials (BOM)

Since the local names of materials in each area level simulator may not be included in the databases of real manufacturing system, the Rename Table is required to be carefully developed by checking the material flow from simulator to simulator.

When the production instruction is received by an area level simulator, following procedures are required to obtain the process information from the master database.

1. By the instruction ID, identify a record of instruction from the Production Instruction Table.
2. Identify the process information from the Process Information Table with the Process ID (Arrow ① in Figure 2).
3. Repeat the arrows of ②,③ and ④ for number of processes in an area.

The destination area of the transportation is obtained from the Process Sequence Table by following procedures when a product becomes ready to transport after completing all processes in the area.

1. Identify the Transportation ID from the Process Sequence Table with the Material Name and the Area shown as arrow ⑦ in Figure 2.
2. The destination area is identified from the Transportation O-D Table with the Transportation ID.

The material name in the destination area level simulator is obtained by following procedures.

1. When all processes of a material finished in an area, the material changes into a part and its name also changes into a part name. To obtain the part name, the Bill Of Materials table is utilized shown as arrow ⑤ and ⑥ in Figure 2.
2. From the Rename Table, the local name of the part in the destination area is identified by procedures discussed above and the part name shown as arrow ⑧ and ⑨ in Figure 2.

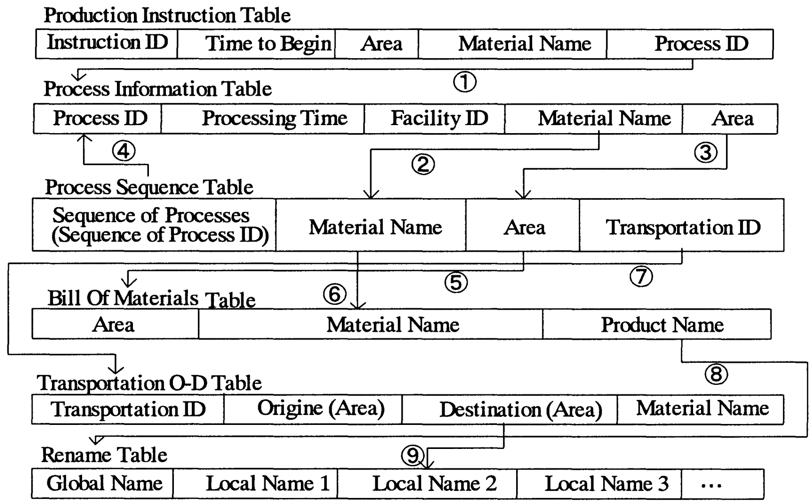


*Figure 2.* Structure of the Master Database for the DVF

# 3. INTEGRATION OF DATABASES

In this study, various databases are integrated into a DVF by developing an interface for Relational Database Management Systems (RDBMS) of widely used databases such as, Oracle, SQL Server and ACCESS.

## 3.1 Database Interface

The storing schemas of different RDBMSs are different from each other. However, interfaces of input and output are based on a common specification called ODBC (Open Data Base Connectivity). Thus in this study, databases are integrated through ODBC to utilize a variety of RDBMSs. To integrate databases into a DVF, both the interfaces for RDBMSs and the interface for the DVF are required. The module, which has these functions, is called a database interface in this study. The database interface is developed by the following processes and the concept of database interface is shown in Figure 3.

- **Interface for RDBMS**

  ActiveX Data Object (ADO) and Visual Basic are employed to develop an interface for RDBMS. ADO provides a unified procedure to connect RDBMS. Visual Basic enables to construct graphical user interface easily.

- **Interface for the DVF**

For integrating databases into the DVF, several configurations are considered (detail is discussed in section 3.2). To implement them, following two methods of mounting databases are required.

(1) Database is mounted on a computer apart from simulator.

(2) Database is mounted as a subsystem of area level simulator.

By utilizing communication interface, method (1) can be realized. Method (2) can be realized by connecting an area level simulator and the database interface on a computer with the call back IP address of the local host and is generally specified as 127.0.0.1. To fit the objective system, these methods can be combined.

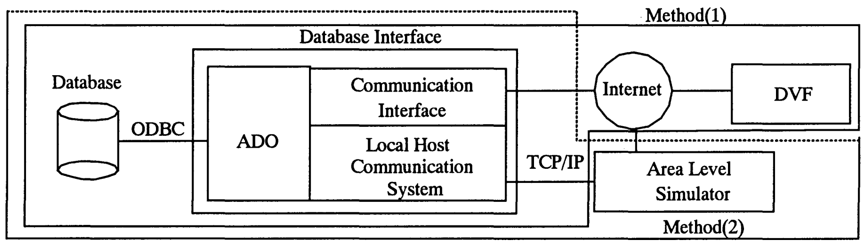


*Figure 3.* Database Interface

## 3.2 Configurations of the Database

On integrating the database, following three different configurations of the database are considered in this study.

Type 1. Only one master database is developed in the DVF for integration.

All area level simulators access the master database to obtain information. In this configuration, each area level simulator needs to access the master database every time when a product is completed, even if the required information has been obtained before.

Type 2. Each area level simulator provides a full copy of the master database.

When the information required by an area level simulator, the area level simulator refers to its copied database. To implement this configuration, copies of the master database for area level simulators are required at the preparation stage.

Type 3. Master database is developed in the factory management system and empty databases are prepared in area level simulators.

When the information required by an area level simulator, the area level simulator refers to its own database at first. If the required information is not stored in the database, the area level simulator accesses to the master database to obtain the necessary information. The obtained information is stored in the database of the area level simulator.

When two or more databases are provided in DVF as configurations of Type 2 and Type 3 above, the contents of the databases are required to be the same and consistent. In this study, databases are updated at a certain time span by informing the changes of contents among the databases.

## 4. EXPERIMENTAL SYSTEM

Based on the concepts discussed above, experimental implementation of the data handling mechanism is performed. The configuration of the objective DVF is shown in Figure 4. In this paper, we explain the system based on the configuration Type 2 although Type 1 and 3 were also implemented. By updating databases, production instructions are distributed among area level simulators. By receiving production instructions from the Factory Management System at storage 1, materials which are stored in the Storage 1 is transported to the Processing A or B by the transportation system. According to the instructions, materials are processed in processing areas and transported between processing areas. After completing the all processes in processing areas, materials are transported to the Storage 2 and terminated.

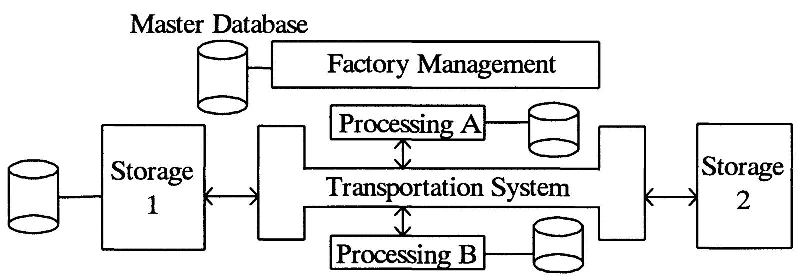


*Figure 4.* Configuration of the Objective DVF

Communications among the area level simulators are realized by exchanging messages in the DVF. Thus, the messages of transportation request and database updating are defined before execution of simulation. Message format of transportation request is shown in Figure 5(a) as serial arrangement of a set of time, area name, message type, material name, destination of the transportation, instruction ID and number of finished areas. Message format of the updating message is shown in Figure 5(b).

*Time : Area Name / MessageType / MaterialName / Destination / InstructionID / Number of Finished Areas*

(a) Transportation Request

*Time : Area Name / MessageType / MaterialName / InstructionID / Sequence of Areas / Process ID / Time to Begin*

(b) Database Updating

*Figure 5.* Message Formats

# 5. CONCLUSION

In this paper, the data handling mechanism integrating databases is discussed on constructing a DVF by integrating area level simulators. To realize the data handling mechanism, the database interface is developed. The data handling mechanism is implemented in a simplified DVF for verifying its performance. In this study, we assumed that the information required for area level simulators is only the material name at the destination area and the destination of transportation. However, the same procedure is considered to be applicable for other information which resembles in characteristics, such as facility name, process information and so on.

In this study, the database interface for the WINDOWS system is developed, where one for UNIX system is left for the future research. Further study is also necessary to clarify the characteristics of information which can be supplied by the data handler.

# REFERENCES

1. Susumu FUJII, Haruhisa TSUNODA, Atsusi OGITA, Yasusi KIDANI (1994). DISTRIBUTED SIMULATION MODEL FOR COMPUTER INTEGRATED MANUFACTURING,*Proceedings of the 1994 Winter Simulation Conference,*pp.946-953
2. Susumu FUJII, Yoko HIRASHIMA, Haruhisa TSUNODA, Mahiro YAMANE, Masaru HIRANO (1994). A BASIC STUDY ON A DISTRIBUTED SIMULATION FOR VIRTUAL MANUFACTURING, *New Directions in Simulation for Manufacturing and Communications*, pp.277-281
3. RICHARD M. FUJIMOTO. (1990). PARALLEL DISCRETE EVENT SIMULATION, *COMMUNICATIONS OF THE ACM.*,33,10,pp.30-53
4. Susumu FUJII, Toshiya KAIHARA, Kentarou Sashio. (2001).A Development of Distributed Virtual Factory –Integration of Area Simulators Modeled by Various Simulation Softwares-, proceedings of the 3rd Manufacturing & Machine Tool Conference, pp.37-38 , in Japanese
5. DAVID R. JEFFERSON (1985). Virtual Time, *ACM Transactions on Programming Languages and Systems* , 7,3, pp.404-425
6. Susumu FUJII, Atsushi OGITA, Yasushi KIDANI, Toshiya KAIHARA. (1999). Synchronization Mechanisms for Integration of Distributed Manufacturing Simulation Systems, *SIMULATION*, MARCH, 72, 3, pp.187-197