Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

MANUFACTURING ADAPTER OF DISTRIBUTED SIMULATION SYSTEMS USING HLA

Hironori Hibino

Technical Research Institute
JSPMI
(Japan Society for the Promotion of Machine Industry)
1-1-12, Hachiman, Higashikurume, Tokyo, JAPAN

Yoshiro Fukuda

Faculty of Engineering
Hosei University
3-7-2 Kajino, Koganei, Tokyo, JAPAN

Yoshiyuki Yura

Engineering Department
Shimizu Corporation
1-2-3 Shibaura Minato, Tokyo, JAPAN

Keiji Mitsuyuki

Production Engineering Department
Denso Corporation
1-1 Shouwa, Kariya, Aichi, JAPAN

Kiyoshi Kaneda

R&D Department
Makino Milling Machine Corporation
359-3 Mimase, Aikawa, Kanagawa, JAPAN

ABSTRACT

In this research, the distributed simulation system to easily evaluate a very large manufacturing system by synchronizing several different simulators is developed. We designed the distributed simulation system using High Level Architecture(HLA) as IEEE1516 standard. A manufacturing adapter to connect manufacturing system simulators with HLA using a plug-in style is proposed. The developed distributed simulation system is confirmed using a case to evaluate a hypothetical manufacturing system which produces motors. Three major commercial based manufacturing system simulators as QUEST, SIMPLE++, and GAROPS are connected using the developed manufacturing adapter. The storage model method to connect simulation models and synchronize the simulators is used in the case study. The case study is then carried out to evaluate the performance of the cooperative work.

1 INTRODUCTION

A simulator plays an important role in designing new manufacturing systems (Fukuda, Hibino, Mitsuyuki, Kojima, Ikeda, Aratake, Tukada, and Suzuki 1994; Fujii, Kidani, Ogita, and Kaihara 1999; Hibino, Fukuda, Fujii, Kojima, Mitsuyuki, and Yura 1999; Hibino, Fukuda, Nakano, and Sato 1999; Hibino, Fukuda, and Nakano 2001; Kryssanov, Abramov, Hibino, and Fukuda 1998). As manufacturing systems are being created on larger and more complicated scales than ever before, it is increasingly necessary to design a manufacturing system with several persons concurrently. Recently, it is becoming harder to keep the high value of a product in the market as a long seller. Therefore industries need to design and make new products for the market in rapid succession. There is a need for short lead-time in design and evaluation activities.

Based on our past research results for analyzing manufacturing system designs, a manufacturing system is broken down into a number of subsystems based on various specifications required by the system (Hibino, Fukuda, Fujii, Kojima, Mitsuyuki, and Yura 1999; Hibino, Fukuda, and Nakano 2001). Concurrently with the separation of the subsystems, partial optimization in each subsystem is attempted. Each subsystem is modeled as a unit on a suitable simulator in response to the purposes required (Hibino, Fukuda, Fujii, Kojima, Mitsuyuki, and Yura 1999; Hibino, Fukuda, and Nakano 2001). After the subsystems are evaluated, it is necessary to evaluate the whole system. As simulators depend on a particular usage, the modeling method of each simulator varies. At present, it is very difficult to combine different models which are made by different simulators because there are no standard descriptions for simulation models and a standard language for simulation programs has not been developed. Therefore designers cannot cooperate easily when using different simulator models. Currently, it is difficult for designers to make and evaluate a whole system model by synchronizing all subsystem models. For these reasons, it is important to research a distributed simulation system. The distributed simulation system is defined as executing a simulation while connecting and synchronizing several different simulators (Fujimoto 1990; Fujimoto 1995). Figure 1 is a concept of concurrent engineering using the distributed simulation system.

Concerning the distributed simulation system, there are many research papers in academe (Das and Fujimoto 1997; Fujimoto 1990; Fujimoto 1995; Jones and Das 1998; Nicol, Johnson, and Yoshimura 1997; Tay, Teo, and Ayani 1998). But there are a few cases where industries have used the ideas presented in those papers. In order to use the ideas in actual designs of manufacturing systems, it was necessary to develop a distributed simulation system which is easy to use in various industries using commercial based manufacturing system simulators.

In this research, the distributed simulation system to easily evaluate a very large manufacturing system by synchronizing several different simulators is developed. We developed the distributed simulation system using High Level Architecture (HLA) as IEEE1516 standard (IEEE1516 2001). A manufacturing adapter to connect manufacturing system simulators with HLA using a plug-in style is proposed. The developed distributed simulation system is confirmed using a case to evaluate a hypothetical manufacturing system which produces motors. Three major commercial based manufacturing system simulators as QUEST, SIMPLE++, and GAROPS are connected using the developed manufacturing adapter (QUEST 2002; SIMPLE++ 1999; GAROPS 2002). The storage model method to connect simulation models and synchronize the simulators is used in the case study. The case study is then carried out to evaluate the performance of the cooperative work.

2 FUNDAMENTAL STRUCTURE OF DISTRIBUTED SIMULATION SYSTEM

In this research, our objective is to establish a distributed simulation system which can be used easily in the industries. HLA is only one standard for the distributed simulation system architecture as IEEE 1516. Figure 2 shows an outline of HLA. Currently there are several commercial based HLA/RTIs. However when we use the commercial based HLA/RTIs, we need to implement interface modules combing program methods based on the interface specifications by ourselves. Therefore it is difficult for manufacturing system designers to use HLA.

In order to solve the problem, a fundamental structure of the distributed simulation system using HLA is proposed. Figure 3 shows a proposal for a fundamental structure.

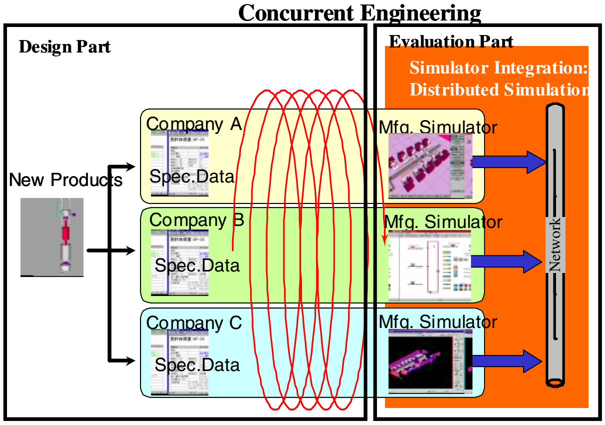


Figure 1: Concept of Concurrent Engineering Using Distributed Simulation System

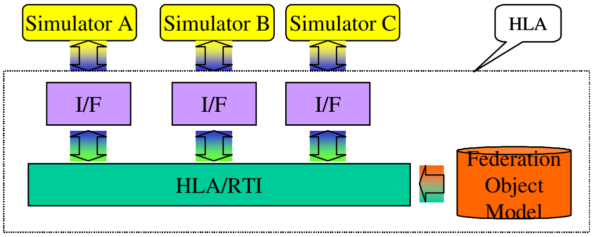


Figure 2: Outline of HLA

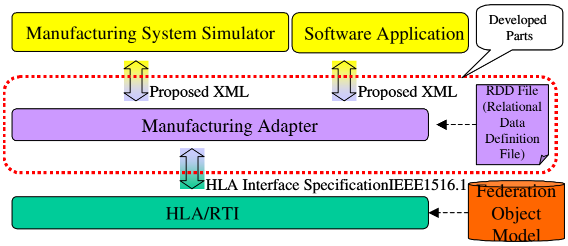


Figure 3: Proposed System Structure

An adapter intermediately between manufacturing simulators and the HLA Runtime Infrastructure (HLA/RTI) is developed. This adapter is called a manufacturing adapter. HLA/RTI provides many kinds of services which are defined in IEEE1516.1 interface specifications (Kuhl, Dahmann, and Weatherly 1999; IEEE1516 2001).

The manufacturing adapter can be connected with HLA/RTI and then the manufacturing adapter can access services which are used to evaluate the manufacturing system. Concerning communication between the manufacturing adapter and manufacturing system simulators, XML format messages are developed. The manufacturing adapter transforms an XML format message from a manufacturing simulator into an HLA interface specification message to HLA/RTI. The manufacturing adapter also transforms the message from HLA/RTI into an XML format message to the manufacturing system simulators. Using the manufacturing adapter, the manufacturing system simulator can access the services of HLA/RTI by only sending the XML format messages without implementing any program methods. Therefore the manufacturing system simulators can easily connect with HLA/RTI and receive many HLA/RTI services.

We also propose a Relational Data Definition (RDD) file which defines exchanged message contents between the simulators. We develop XML format definitions in the RDD file. The manufacturing adapter is developed to get the message exchange services by reading parameters of the RDD file when the manufacturing adapter starts and finishes. The contents of the RDD file describe exchanged message contents. Therefore the RDD file is important to connect with simulation models. We assume that the RDD file is generated by concurrent engineering support systems.

2.1 Manufacturing Adapter

The manufacturing adapter connects with HLA/RTI, the manufacturing system simulators, and software applications which are applications for the manufacturing system such as scheduling software, an equipment controller, and so on. First, functions of the manufacturing adapter to communicate with HLA/RTI are explained, then functions of the manufacturing adapter to communicate with the manufacturing system simulators are described.

2.1.1 Manufacturing Adapter between HLA/RTI

In order to get the services of HLA/RTI based on the interface specifications as IEEE1516.1, the manufacturing adapter was developed. The manufacturing adapter must be implemented in the following services to synchronize simulators. The services are a data distribution management service, a time management service, and a federation management service.

The data distribution management service has the roles of getting messages from a manufacturing system simulator or a software application and distributing the messages to designated simulators or designated software applications. One of the most important services on the distributed simulation system is to distribute messages with a time stamp to simulators which use the messages to synchronize with the other simulators. The data distribution management service provides this important service. In order to use the data distribution management service, the following functions are implemented in the manufacturing adapter.

1. A function to declare publishing and subscribing objects which are defined in Federate Object Model (FOM). (Exchanged data among manufacturing simulators are defined in FOM as objects and their attributes based on the object-oriented method).
2. A function to send a time stamp message concerning published objects using the Time-Stamp-Order (TSO) program method. (The TSO program method adds a time stamp to a message concerning published objects. Then the TSO program method sends the message to HLA/RTI.)
3. A function to receive a callback message concerning subscribed objects.

The time management service manages the distributed simulation time clock and each simulator's time clock. In order to use this service, the following functions are implemented in the manufacturing adapter.

1. A function to send a simulation clock from the manufacturing system simulator to HLA/RTI using the Time-Advance-Request (TAR) method and Next-Event-Request (NER) method.
2. A function to receive a callback message concerning the Logical Time which is the distributed simulation time clock.

The federation management service has the roles of joining and resigning a simulator or software application. In order to use this service, the following functions are implemented in the manufacturing adapter.

1. A function to join a simulator or a software application to HLA/RTI using the Join program method.
2. A function to resign a simulator or a software application from HLA/RTI using the Resign program method.

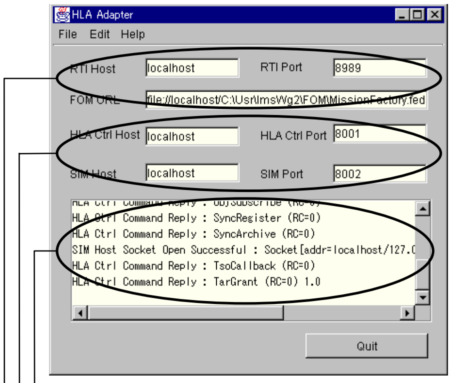
Concerning the data distribution management service and the federation management service, the initialization procedure to get these services is known. However in this procedure, it is necessary to set several kinds of parameters in HLA/RTI. Therefore these parameter types are arranged and the parameter types using XML are defined. We develop the manufacturing adapter to get the services by reading parameters using the developed XML definition when the manufacturing adapter starts and finishes. Then the parameters are set in a RDD file which is shown in Figure 3. Figure 4 shows the main contents of the developed XML definitions in the RDD file. Figure 5 shows the GUI of the developed manufacturing adapter.

2.1.2 Manufacturing Adapter between Manufacturing System Simulator

Concerning communication between the manufacturing adapter and the manufacturing system simulator, XML definitions are developed. Figure 6 shows the detailed contents of the XML definitions for the data distribution management service, the time management service, and the

```
<InitializationFile>
<RtiParameter>
   <RtiHost>Host Name</RtiHost>
   <RtiPort>Port No.</RtiPort>
   <FomUrl>FOM file name and URL</FomUrl>
   <ManagerSim>Exist or Nonexist</ManagerSim>
   <FederationName> Federation Name </FederationName>
   <FederateName> Federate Name </FederateName>
   <LogicalTime>Logical Time Value</LogicalTime>
   <Lookahead>Lookahead Value</Lookahead>
   <SynchLabel> Synchization Label Name</SynchLabel>
   <SynchronizationMechanism> Synchronization Mechanism Name</SynchronizationMechanism>
  </RtiParameter>
  <ObjPublish>
    <ObjClassName> Object Class Name
     <AttribName> Attribute Name </AttribName>
     <AttribName> Attribute Name </AttribName>
   </ObjClassName>
  </ObjPublish>
  <ObjSubscribe>
   <ObjClassName> Object Class Name
   <AttribName> Attribute Name </AttribName>
     <AttribName> Attribute Name </AttribName>
    </ObjClassName>
  </ObjSubscribe>
  <Properties>
   <DebugMode> Yes or NO </DebugMode>
  </Properties>
</InitializationFile>
```

Figure 4: RDD File Definition Using XML



RTI Host: localhost
RTI Port: 8989
FOM URL: file://localhost/C:\Usr\ImsWg2\FOM\MissionFactory.fed
HLA Ctrl Host: localhost
HLA Ctrl Port: 8001
SIM Host: localhost
SIM Port: 8002

```
HLA Ctrl Command Reply : ObjSubscribe (RC=0)
HLA Ctrl Command Reply : SyncRegister (RC=0)
HLA Ctrl Command Reply : SyncArchive (RC=0)
SIM Host Socket Open Successful : Socket[addr=localhost/127.0
HLA Ctrl Command Reply : TsoCallback (RC=0)
HLA Ctrl Command Reply : TarGrant (RC=0) 1.0
```

HLA-RTI Communication Log:
Federation management service function
Data distribution management service function
Time management service function

Network Information

FOM Address

Figure 5: Mfg-Adapter GUI

Data Distribution Management Service Function

```
Set up for Publish Objects
 [Indication Frame]
   <CtrlCommand>ObjPublish</CtrlCommand>
   <CommandData>
     <ObjClassName>Object Class Name</ObjClassName>
     <AttribName>Attribute Name</AttribName>
     <AttribName> Attribute Name </AttribName>
              …
   </CommandData>
[Return Code Frame]
   <CtrlCommand>ObjPublish</CtrlCommand>
   <ReturnCode>Return Code</ReturnCode>

Set up for Subscribe Objects
[Indication Frame]
   <CtrlCommand>ObjSubscribe</CtrlCommand>
   <CommandData>
     <ObjClassName>Object Class Name</ObjClassName>
     <AttribName>Attribute Name</AttribName>
     <AttribName> Attribute Name </AttribName>
              …
   </CommandData>
[Return Code Frame]
   <CtrlCommand>ObjSubscribe</CtrlCommand>
   <ReturnCode>Return Code</ReturnCode>

TSO(Time Stamp Order), TSO(Callback)
[Indication Frame]
  <CtrlCommand>TsoCallback</CtrlCommand>
  <CommandData>
   <ObjClassName>Object Class Name</ObjClassName>
   <AttribName>Attribute Name</AttribName>
   <AttribValue>Attribute Data Value</AttribValue>
   <LogicalTime>Logicaltime</LogicalTime>
  </CommandData>
[Return Code Frame]
  <CtrlCommand>TsoCallback</CtrlCommand>
  <ReturnCode>Return Code</ReturnCode>
[Callback Frame]
   <CtrlCommand>TarGrant</CtrlCommand>
   <CommandData>
     <TsoCallback>
       <ObjClassName>Object Class Name</ObjClassName>
       <AttribName>Attribute Name</AttribName>
       <AttribValue>Attribute Data Value</AttribValue>
              …
       <LogicalTime>Logicaltime</Logical Time>
              …
     </TsoCallback>
   </CommandData>
   <ReturnCode>Return Code</ReturnCode>
```

Time Management Service Function

```
TSO(Time Stamp Order)
[Indication Frame]
  <CtrlCommand>TsoCallback</CtrlCommand>
  <CommandData>
   <ObjClassName>Object Class Name</ObjClassName>
   <AttribName>Attribute Name</AttribName>
   <AttribValue>Attribute Data Value</AttribValue>
   <LogicalTime>Logicaltime</LogicalTime>
  </CommandData>
[Return Code Frame]
  <CtrlCommand>TsoCallback</CtrlCommand>
  <ReturnCode>Return Code</ReturnCode>

TAR(Time Advance Request), Grant(Callback)
[Indication Frame]
   <CtrlCommand>TarGrant</CTRLCommand>
   <CommandData>
     <LogicalTime>Logicaltime</LogicalTime>
   </CommandData>
[Callback Frame]
   <CtrlCommand>TarGrant</CtrlCommand>
   <CommandData>
   <TarGrant>
      <LogicalTime>Logicaltime</LogicalTime>
    </TarGrant>
   </CommandData>
   <ReturnCode>Return Code</ReturnCode>
```

Federation Management Service Function

```
Create Federation
[Indication Frame]
   <CtrlCommand>FedCreate</CtrlCommand>
   <CommandData>
     <FederationName>Federation name</FederationName>
     <FomUrl>FOM URL</FomUrl>
   </CommandData>
[Return Code Frame]
   <CtrlCommand>FedCreate</CtrlCommand>
   <ReturnCode>Return Code</ReturnCode>

Join Federate
[Indication Frame]
   <CtrlCommand>FedJoin</CtrlCommand>
   <CommandData>
     <FederationName>Federation Name</FederationName>
     <FederateName>Federate Name</FederateName>
     <LogicalTime>Logicaltime</LogicalTime>
     <Lookahead>Lookahead</Lookahead>
   </CommandData>
[Return Code Frame]
   <CtrlCommand>FedJoin</CtrlCommand>
   <ReturnCode>Return Code</ReturnCode>

Kill Federate
[Indication Frame]
   <CtrlCommand>FedKill</CtrlCommand>
   <CommandData>
   </CommandData>
[Return Code Frame]
   <CtrlCommand>FedKill</CtrlCommand>
   <ReturnCode>Return Code</ReturnCode>

Destroy Federation
[Indication Frame]
   <CtrlCommand>FedDestroy</CtrlCommand>
   <CommandData>
     <FederationName>Federation Name</FederationName>
   </CommandData>
[Return Code Frame]
   <CtrlCommand>FedDestroy</CtrlCommand>
   <ReturnCode>Return Code</ReturnCode>
```

Figure 6: Communication Definition between Manufacturing Adapter and Simulator Using XML

federation management service. Using the XML definitions, the manufacturing system simulator can access the services of HLA/RTI by only sending the XML format messages without implementing any program methods.

Qualifications of the manufacturing system simulators in this research require the following qualifications.

1. The manufacturing simulator has external program methods of simulation execution control

such as running control, resuming control, pausing control, and stopping control.

2. The manufacturing simulator has external program methods to refer to and write the internal data which are in the simulator.

3 SYSTEM STRUCTURE OF DISTRIBUTED SIMULATION SYSTEM

Figure 7 shows the developed system structure of the distributed simulation system.

Terms of federation and federate are defined in the IEEE1516 rule (IEEE1516 2001). A federation is a system model which is connected with several subsystems・models in a distributed simulation system. A federate is one of the connected components of the distributed simulation system.

The developed system structure of the distributed simulation system consists of three parts which are Simulation-Federates, one Manager-Federate, and HLA/RTI. In this research, Simulation-Federates and Manager-Federate are developed. A commercial based HLA/RTI which has been developed by Pitch company using the Java program language are used (pRTI 2001).

3.1 Simulation-Federate

The Simulation-Federate is a unit of simulation execution. The Simulation-Federate consists of a simulator, the manufacturing adapter, and an HLA controller. The Simulation-Federate is created in response to a simulation model which is a part of the manufacturing system. The simulator corresponds to the manufacturing system simulator. The HLA controller assigns objects for the data distribution management service to declare and join the adapter to the HLA/RTI and to resign the adapter from the HLA/RTI. Figure 8 shows the HLA controller GUI.

In this research, a communication protocol between the manufacturing adapter and the manufacturing system simulator or HLA controller is used the socket protocol.

3.2 Manager-Federate

The Manager-Federate has the roles of controlling execution of the distributed simulation system such as starting, resuming, pausing, and finishing. The Manager-Federate also has the roles of checking and repairing network troubles. The Manager-Federate consists of a HLA manager and the manufacturing adapter. Figure 9 shows the manager federate GUI.

4 CASE STUDY

The developed distributed simulation system is confirmed using a case to evaluate a hypothetical manufacturing system which produces motors. Figure 10 shows structure of the motors. The manufacturing system mainly consists of a upper housing line subsystem, an under housing line subsystem, a rotor line subsystem, a stator line subsystem, an assembly line subsystem, a storage facility which stores all parts including purchased parts, and a material handing subsystem which connects with all subsystems. With this manufacturing system, it is assumed that three companies design the manufacturing system separately and concurrently. Company B designs the upper housing line subsystem. Company C designs the assembly line subsystem. Company A designs the other subsystems which include

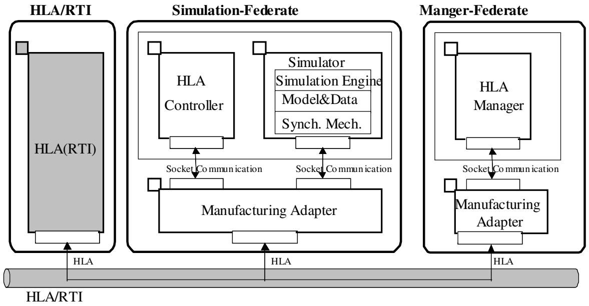
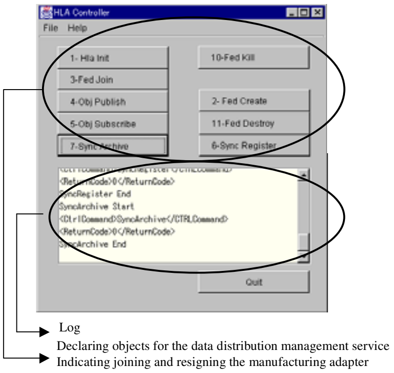


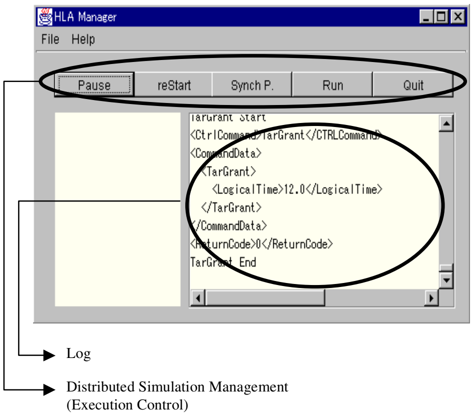
Figure 7: System Structure of Distributed Simulation System



```
<CtrlCommand>SyncRegister</CTRLCommand>
<ReturnCode>0</ReturnCode>
SyncRegister End
SyncArchive Start
<CtrlCommand>SyncArchive</CTRLCommand>
<ReturnCode>0</ReturnCode>
SyncArchive End
```

Log
Declaring objects for the data distribution management service
Indicating joining and resigning the manufacturing adapter

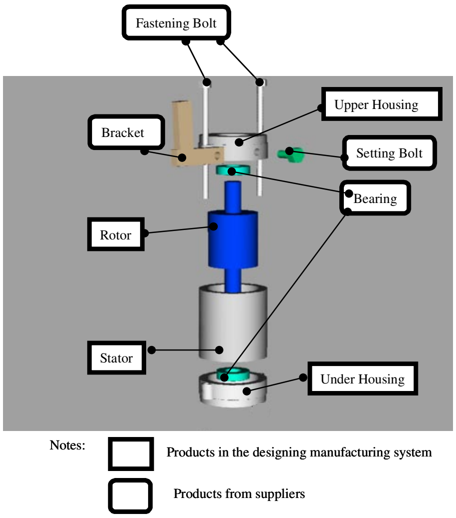
Figure 8: HLA Controller GUI



```
TarGrant Start
<CtrlCommand>TarGrant</CTRLCommand>
<CommandData>
  <TarGrant>
    <LogicalTime>12.0</LogicalTime>
  </TarGrant>
</CommandData>
<ReturnCode>0</ReturnCode>
TarGrant End
```

Log
Distributed Simulation Management
(Execution Control)

Figure 9: Manager Federate GUI

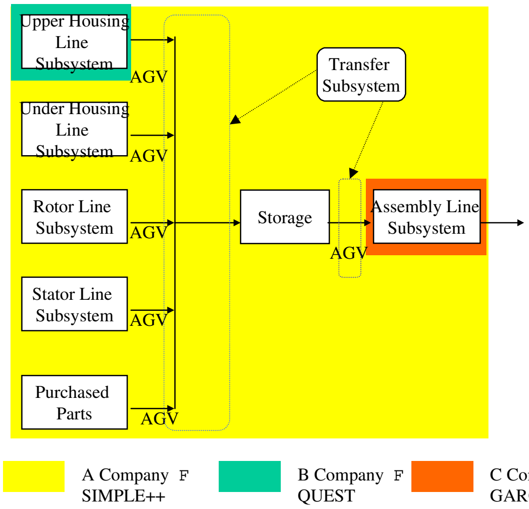


Notes: □ Products in the designing manufacturing system
▢ Products from suppliers

Figure 10: Structure of Motor

the storage and the material handing subsystem. Then Company A is modeled a simulation model of the assigned subsystems using the manufacturing simulator SIMPLE++ (SIMPLE++ 1999). The Company B is modeled a simulation model of the assigned subsystem using the manufacturing simulator QUEST (QUEST 2002). Company C is modeled a simulation model of the assigned subsystem using the manufacturing simulator GAROPS (GAROPS 2002). Relations among the subsystems are shown in Figure 11. The manufacturing system was evaluated by connecting one simulation model using SIMPLE++ and one simulation model using QUEST and one simulation model using GAROPS. The outline of the connection structure is shown in Figure 12.

Concerning a connection method between simulation models which are made on the different manufacturing simulators, we use the storage model method (Hibino, Fukuda, and Nakano 2001). Using the storage model method, one buffer as the physical state is divided into two buffer simulation models as the logical state (Hibino, Fukuda, and Nakano 2001). Each divided buffer simulation model is modeled separately in a different simulator. Figure 13 shows an outline of the buffer.



🟨 A Company F SIMPLE++ 🟩 B Company F QUEST 🟧 C Company F GAROPS

Figure 11: Relations among the Subsystems

Either an event message which means to store works into the logical buffer simulation model or an event message which means to ship works from the logical buffer simulation model is sent to the other logical buffer simulation model. Then the same events which are included in the message are made to occur in the simulation model which has received the logical buffer simulation model which is got the message. Figure 13 also shows the structure to connect simulation models using the logical buffers. One buffer for shipping the upper housing parts in the upper housing line subsystem and six buffers for storing parts which are assembled in the assembly line subsystem are defined as connection points of the storage model method. Figure 14 shows relationship between a manufacturing system layout and the connection method between simulation models using the storage model method. Figure 15 shows one of the contents of the RDD file in this case study. Figure 16 shows parts of the FOM contents used in this case study.

We confirmed through the case study that a manufacturing system could be easily created by connecting the separately developed subsystem models using the developed distributed simulation system. Then we confirmed the potential applicability for the distributed simulation system through the case study using three major commercial based manufacturing system simulators as QUEST, SIMPLE++, and GAROPS.

5 CONCLUSIONS

A distributed simulation system for manufacturing systems was developed.

The results were:

1. To propose a fundamental structure of a distributed simulation system for manufacturing systems.
2. To develop a manufacturing adapter which can connect with several kinds of manufacturing simulators.
3. To implement the system structure of a distributed simulation system for manufacturing systems.
4. To confirm through a case study that a manufacturing system can be easily created by connecting separately developed subsystem models using this developed distributed simulation system
5. To confirm the potential applicability for this developed distributed simulation system through a case study using three major commercial based manufacturing system simulators as QUEST, SIMPLE++, and GAROPS.

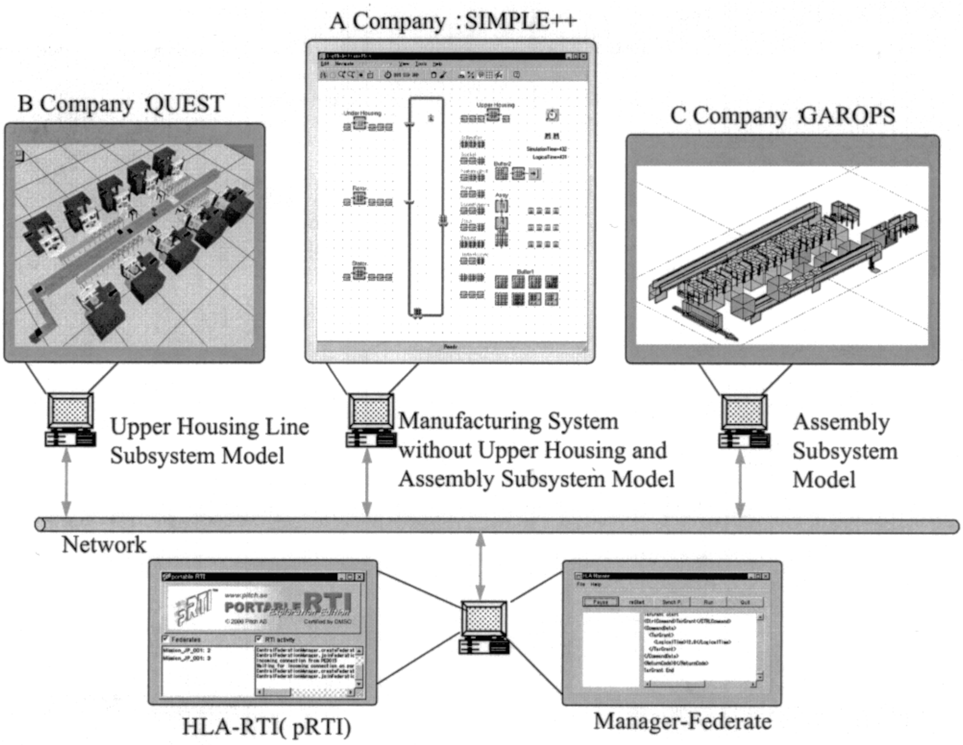


Figure 12: Outline of the Case Study

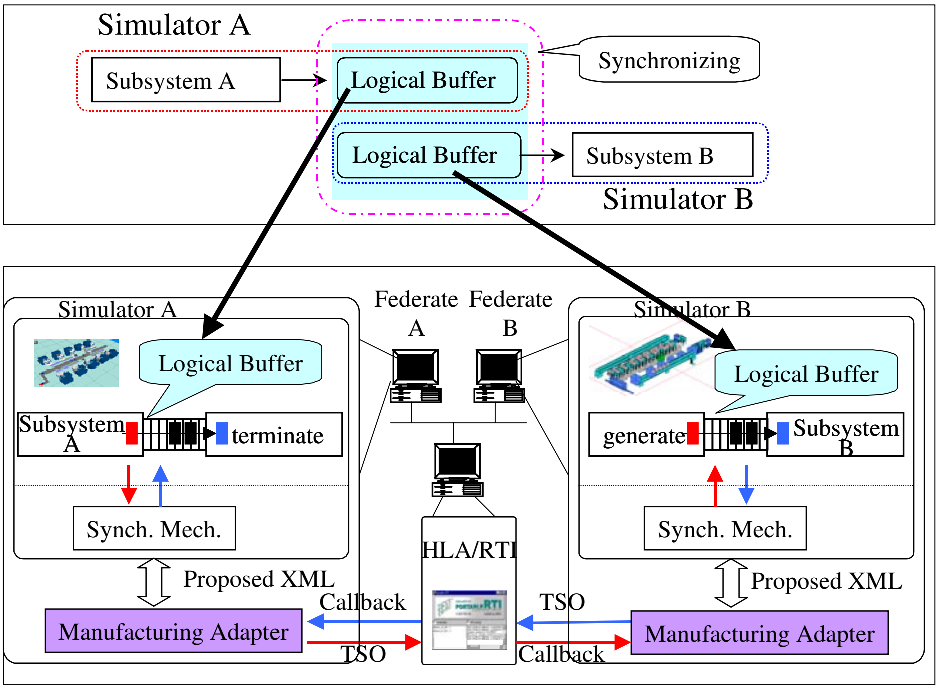


Figure 13: Storage Model Method: Structure to Connect with Simulation Models Using the Logical Buffers

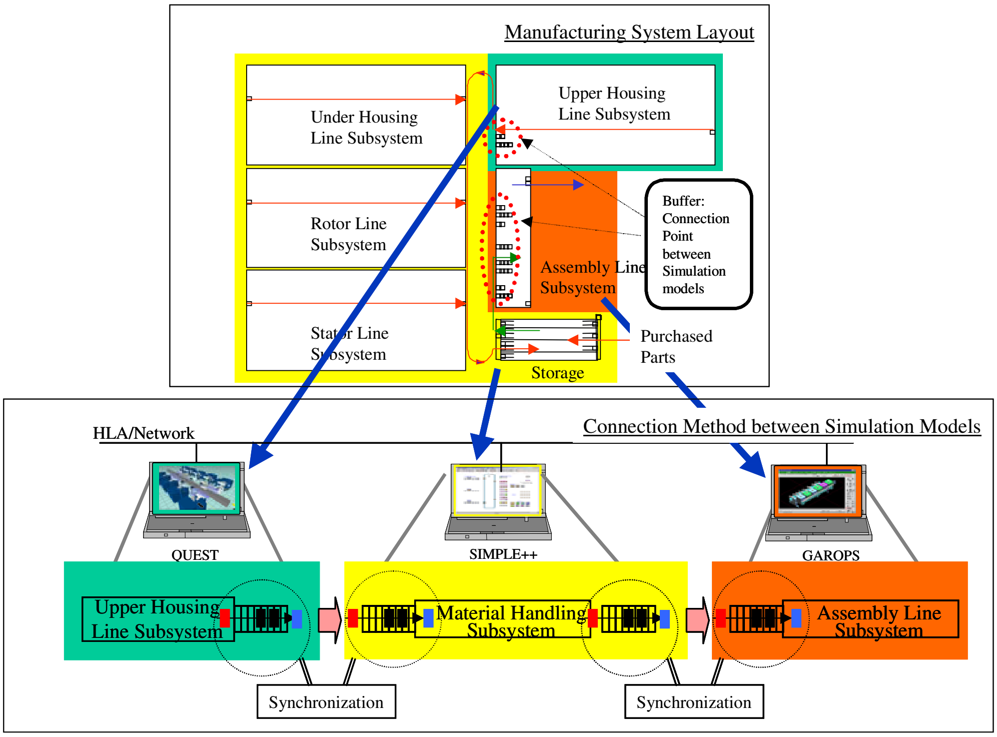

Manufacturing System Layout
Under Housing Line Subsystem
Rotor Line Subsystem
Stator Line Subsystem
Upper Housing Line Subsystem
Assembly Line Subsystem
Buffer: Connection Point between Simulation models
Storage
Purchased Parts
HLA/Network
Connection Method between Simulation Models
QUEST
SIMPLE++
GAROPS
Upper Housing Line Subsystem
Material Handling Subsystem
Assembly Line Subsystem
Synchronization
Synchronization

Figure 14: Relationship between the Layout and the Connection Method between Simulation Models

```
<InitializationFile>
  <Properties>
    <SimulationName>Upper Housing Sub-system</SimulationName>
  </Properties>
  <RtiParameter>
    <RtiHost>localhost</RtiHost>
    <RtiPort>8989</RtiPort>
    <FomUrl>file://localhost/C:¥ImsWG2¥HlaRti¥config¥MissionFactory.fed</FomUrl>
   <FederationName>MissionFactory</FederationName>
    <FederateName>Logistics</FederateName>
    <LogicalTime>0</LogicalTime>
    <Lookahead>1</Lookahead>
  </RtiParameter>
<ObjPublish>
    <ObjClassName>ObjectRoot.MissionFactorySim.MissionFactoryBuffer
      <AttribName> Former_Storage_Shipping_Volume </AttribName>
     <AttribName> Former_Storage_Shipping_type </AttribName>
      <AttribName> Former_Storage_Storin_Volume </AttribName>
      <AttribName> Former_Storage_Storing_type </AttribName>

    </ObjClassName>
  </ObjPublish>
  <ObjSubscribe>
    <ObjClassName>ObjectRoot.MissionFactorySim.MissionFactoryBuffer
      <AttribName> Latter_Storage_Shipping_Volume  </AttribName>
      <AttribName> Latter_Storage_Shipping_type  </AttribName>
      <AttribName> Latter_Storage_Storing_Volume  </AttribName>
      <AttribName> Latter_Storage_Storing_type  </AttribName>
    </ObjClassName>
  </ObjSubscribe>
</InitializationFile>
```

Figure 15: Contents of the RDD File in the Case Study

```
(FED
  (Federation MissionFactory) ;; we choose this tag
  (FEDversion v1.3)          ;; required; specifies RTI spec version
  (spaces                    ;; we define no routing spaces
  )
  (objects
   (class ObjectRoot         ;; required
     (attribute privilegeToDeleteObject reliable timestamp)
     (class RTIprivate)
     (class MissionFactorySim
       (attribute position reliable timestamp)
       (class MissionFactoryBuffer
         (attribute Former_Storage_Shipping_volume reliable timestamp)
         (attribute Former_Storage_Shipping_type reliable timestamp)
         (attribute Former_Storage_Storing_volume reliable timestamp)
         (attribute Former_Storage_Storing_type reliable timestamp)
         (attribute Latter_Storage_Shipping_volume  reliable timestamp)
         (attribute Latter_Storage_Shipping_type  reliable timestamp)
         (attribute Latter_Storage_Storing_volume  reliable timestamp)
         (attribute Latter_Storage_Storing_type  reliable timestamp)
       )
     )
```

Figure 16: Parts of the FOM Contents in the Case Study

ACKNOWLEDGMENTS

This research is a part of the "IMS0010:MISSION; Modeling and Simulation Environments for Design Planning and Operation of Globally Distributed Enterprises" research project of the IMS program and the "Study on Manufacturing System Using Open-Oriented Information Technologies" research project of JSPMI. The latter project is supported by funding from the Japan Keirin Association. This research is also one of the results of the Manufacturing System Research Institute in Hosei University.

REFERENCES

Das S, Fujimoto R. 1997. An Empirical Evaluation of Performance-Memory Trade-offs in Time Warp. *IEEE Transactions on Parallel and Distributed Systems* 8 (2): 210-224.

Kuhl F, Dahmann J, Weatherly R. 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Prentice Hall.

Fukuda Y, Hibino H, Mitsuyuki K, Kojima F, Ikeda Y, Aratake T, Tukada M, Suzuki K. 1994. Integrated Environment for Production System Design Simulations. In *Proceedings of Simulation for Manufacturing and Communications*, 434-438.

Fujii S, Kidani Y, Ogita A, Kaihara T. 1999. Synchronization Mechanisms for Integration of Distributed Manufacturing Simulation Systems. *International Journal of Simulation* 71 (3): 187-197.

Fujimoto R. 1990. Parallel Discrete Event Simulation. *Communication of ACM* 33 (10): 30-53.

Fujimoto R. 1995. Parallel and Distributed Simulation. In *Proceedings of the 1995 Winter Simulation Conference*, 118-125.

GAROPS.2002. <http://www.toyotamacs.co.jp/products/ catalog13/>.

Hibino H, Fukuda Y, Fujii S, Kojima F, Mitsuyuki K, Yura Y. 1999. The Development of an Object-Oriented Simulation System Based on the Thought Process of the Manufacturing System Design. *International Journal of Production Economics* 60-61: 343-351.

Hibino H, Fukuda Y, Nakano M, Sato S. 1999. The Development of a Method for Integration between Different Types of Simulators. *Global Production Management, Kluwer Academic Publishers*, 201-211.

Hibino H, Fukuda Y, Nakano M. 2001. A Synchronization Mechanism for Distributed Manufacturing Simulation Systems. In *Proceedings of the 16th International Conference of Production Research.*

IEEE 1516. 2001. <http://www.demso.mil/>.

Jones K, Das S. 1998. Combining Optimism Limiting Schemes in Time Warp Based Parallel Simulations. In *Proceedings of the 1998 Winter Simulation Conference*, 499-505.

Kryssanov V, Abramov V, Hibino H, Fukuda Y. 1998. A Framework for the Development of Manufacturing Simulators: Towards New Generation of Simulation Systems. In *Proceedings of 1998 Japan-U.S.A. Symposium on Flexible Automation*, 1307-1314.

Nicol D, Johnson M, Yoshimura A. 1997. The IDES Framework: A Case Study in Development of a Parallel Discrete-event Simulation System. In *Proceedings of the 1997 Winter Simulation Conference*, 93-99.

PRTI. 2001. <http://www.pitch.se/prti/>.

QUEST. 2002. <http://www.delmia.com/>.

Simple++. 1999. <http://www.tecnomatix.com/>.

Tay S, Teo Y, Ayani R. 1998. Performance Analysis of Time Warp Simulation with Cascading Rollbacks. In *Proceedings of 12th ACM/IEEE/SCE Workshop on Parallel and Distributed Simulation*, 30-37.

AUTHOR BIOGRAPHIES

HIRONORI HIBINO is a Researcher of Technical Research Institute of JSPMI (Japan Society for the Promotion of Machine Industry) which is a subsidiary of the Ministry of Economy, Trade and Industry in Japan. He is also a Visiting Lecturer in Toyo University and a Guest Researcher of Manufacturing System Research Institute in Hosei University. He is a member of JSME, JIMA, and JSPE. His e-mail address is <hibino@tri.jspmi.or.jp>, and his Institute's Web page is <http://www.tri.jspmi.or.jp>.

YOSHIRO FUKUDA is Professor of Industrial and Systems Engineering Department and Head of Manufacturing System Research Institute in Hosei University. He is a fellow of JSME. He is a member of JSME, JIMA, JSPE and IFIP5.12. He is a sector board member in IEC. His e-mail address is <fukuda@k.hosei.ac.jp>, and his Institute's Web page is <http://www.mslab.k.hosei.ac.jp>.

YOSHIYUKI YURA is an Engineer of Engineering Division in SHIMIZU CORPORATION which was the prime contractor of the IMS-MISSION project. His e-mail address is <yura@shimz.co.jp>, and his company's Web page is <http://www.shimz.co.jp>.

KEIJI MITSUYUKI is a Project Manager of the Production Engineering Department in DENSO CORPORATION. He is a member of JSME and JSPE. His e-mail address is <mitsu@prd.denso.co.jp>, and his company's Web page is <http://www.denso.co.jp>.

KIYOSHI KANEDA joined MAKINO MILLING MACHINE CORPORATION in 1975. He has worked in the areas of NC software and Simulation for the machining system. He is currently engaged in the Integration of the Distributed Manufacturing Simulation. His e-mail address is < kaneda@makino.co.jp >, and his company's Web page is <http://www.makino.co.jp>.