Journal of Manufacturing Systems
Vol. 21/No. 1
2002

Design of Distributed Information Systems for Agile Manufacturing Virtual Enterprises Using CORBA and STEP Standards

Liangyu Zhou and Rakesh Nagi, Dept. of Industrial Engineering, State University of New York at Buffalo, Buffalo, New York. E-mail: nagi@buffalo.edu

Abstract

In the global market environment, agile manufacturing companies are forced to cooperate with each other and form temporary opportunistic alliances called virtual enterprises. Each partner may have unique software applications and hardware systems as well as policies and procedures that govern the flow of work. This paper presents a distributed information system architecture for agile manufacturing enterprises such that the heterogeneous partners are seamlessly integrated and, more importantly, their internal processes remain unaffected, while enterprise-wide information is kept consistent. The architecture is related to the three-tier information communication model proposed by Song and Nagi (1997). This architecture is advanced for partner heterogeneity by incorporating two standards: (1) Common Object Request Broker Architecture (CORBA) for information exchange, and (2) International Standard for the Exchange of Product Data (STEP) for information modeling and mapping. Further proposed, for policy integration, is a workflow manager and a systematic approach to develop composite workflows for the virtual enterprise by wrapping the legacy partner workflows. Petri nets and meta data are used to facilitate the workflow simulation and workflow description, respectively. A prototype system has been developed for demonstration purposes.

Keywords: *Agile Manufacturing, Information Systems, Workflow Management, Common Object Request Broker Architecture (CORBA), Standard for the Exchange of Product Data (STEP)*

1. Introduction

Within the new global market environment, customer requirements are not limited to traditional issues, such as higher quality, lower price, and so on, but extend to the speed of delivery and variability of products. Correspondingly, the profitable lifetime of products is shortening drastically. Therefore, enterprises must be capable of delivering new products efficiently to the customer-driven market to remain competitive, that is, to maintain "agility." However, it is extremely difficult for an individual manufacturer to adopt the strategy supporting speed to market for new products and the ability to satisfy individual customer preferences while also being responsive to intensified public concern about manufacturing's social and environmental impacts (Iacocca 1991). Under this consideration, the cooperation among different organizations becomes almost unavoidable.

The second half of the 1990s saw the emergence of a new form of commercial organization, the virtual enterprise (Ball 1997). The virtual enterprise offers a solution to the abovementioned problems. Partners from different locations and with different backgrounds can link their complementary core competencies to react swiftly to various rapidly changing customer wishes, which are far beyond the capabilities of individual companies. Judged by the practice in the last several years, this new organizational format can work more efficiently than large organizations based on managerial hierarchies and deep organization charts.

At present, there is still no authoritative definition for a virtual enterprise. A good attempt from the National Industrial Information Infrastructure Protocols (NIIIP) Consortium is: "A temporary consortium of independent member companies that come together to quickly exploit fast-changing worldwide product manufacturing opportunities." Virtual enterprises assemble themselves based on cost effectiveness and product uniqueness regardless of organization size, geographic location, computing environments, technologies deployed, or processes implemented. Virtual enterprise companies share costs, skills, and core competencies, which collectively enable them to access global markets with world-class solutions that cannot be provided individually (NIIIP 1998).

To achieve the agile manufacturing virtual enterprise visions, ubiquitous communication and information are the central, critical, and fundamental parts of the technical elements (Goldman, Nagel, Preiss 1995). The managerially independent busi-

nesses in a virtual enterprise must exchange information efficiently within and across companies. The data to be exchanged concerns many aspects of the business, including product data management, manufacturing resource planning, procurement, human resources, and finance. But in many cases, the existing information systems of the partners in the virtual enterprise are placed at different geographical locations, installed on different operation systems, and organized in different data structures. Therefore, virtual enterprises increasingly rely on the concept of distributed information systems, which can work independently across these boundaries.

The purpose of this paper is to develop a systematic approach for agile manufacturing virtual enterprise partners to construct distributed information systems using emerging information modeling and exchange standards. Partner policy and workflow modeling, verification, construction, and reuse are included to address the dynamic knowledge management needs in agile manufacturing. By adopting the proposed distributed information system (DIS) architecture, partners should be able to conduct workflow and share information efficiently within the virtual enterprise, while maintaining data, application, and policy autonomy.

2. Literature Review

In this section, the classical approaches to constructing on enterprise information system are reviewed. Then, the discussion is extended to several previous efforts that are closely related to the information system architecture for virtual enterprises, including (i) National Industrial Information Infrastructure Protocols (NIIIP); (ii) Agile Manufacturing Information System (AMIS); and (iii) Workflow Management System (WfMS). Some related Internet technologies and their potential use in virtual enterprises are also presented.

2.1 Information Integration Architecture

Traditionally, the information exchange spans the range of a single company with a relatively static policy on conducting business processes. Therefore, the information integration architectures are static. One unified application package can be developed to manage all types of data and data exchanges (Flatau 1988). For instance, a well-developed application package can catch the modification of CAD files, extract the related CAPP and BOM data through a database management system, make corresponding changes, save the changes, and then notify relevant personnel if needed. As an attempt to facilitate the data sharing between various database management systems, Krishnamurthy et al. (1988) propose a common interface that can connect the existing database management systems and applications. This common interface can be extended to adopt new application packages. But the common interface is so closely coupled with the database management system that it is difficult to integrate it into the heterogeneous virtual enterprise environment.

For a more recent account of broader issues in information system integration for Next Generation Manufacturing (NGM), the reader is referred to Chapter 17 of Jordan and Michel (2000). In 1995, the National Science Foundation initiated the Next Generation Manufacturing Project (Jordan and Michel 2000) as a follow-on to the Agile Manufacturing Project. The objective was to develop a framework for action that U.S. manufacturers could adopt to excel in an increasingly complex global business environment. This reference describes four integration frameworks: SEMATECH CIM application framework specification, the DoD Common Operating Environment as it might be applied to manufacturing, the NIIIP architecture (which is elaborated in section 2.3), and the National Advanced Manufacturing Testbed. Many of the information integration issues raised in the NGM project are in the focus of the current paper.

2.2 Prototype Implementation of Virtual Information System

To meet the needs for a mechanism to manage and control information flow among collaborating partners, Song and Nagi (1997) propose the design and implementation of an agile manufacturing information system (AMIS). In this AMIS model, partners are represented with modules connected by a network. In addition, each of the modules can be further broken down into seven submodules: local database, database management system, workflow manager, knowledge base, partner client, and partner server. In this architecture, the server/client layer stands between the database management system on partner sites and conducts the information exchange

under the guidance of the workflow manager. The open-ended structure of the server/client layer makes it possible to connect various database management systems to a unified structure. Moreover, the rules described in the workflow manager reflect the partner policies for information exchange. Modifications of these rules change the behavior of the information system. To define data and workflow hierarchy, the authors develop the partner query language and the partner workflow language. These tools facilitate the construction of this proposed architecture by unifying the format of information models and their manipulations. Song and Nagi's model adopts object-oriented technologies successfully and layers the distributed information systems naturally. The model provides an efficient basic structure to develop the information system for virtual enterprises.

2.3 National Industrial Information Infrastructure Protocols (NIIIP)

The NIIIP Consortium was formed under a cooperative development agreement with the U.S. government. Layering on the International Standard for the Exchange of Product Data (STEP), the NIIIP focuses on enabling the effective interoperation of manufacturers and suppliers who utilize a wide range of computer systems, operating environments, and business processes.

In NIIIP specifications, the information protocols are explicitly divided into three layers to meet the requirement of an information system for virtual enterprises (Hardwick et al. 1997). The bottom layer consists of communication protocols that deal with the communication and data stream between the clients and servers. The second layer, organization protocols, lies upon the communication protocols. The organization protocols organize the data, which are processed by communication protocols, into models and then define the relationships of the data. The top level of the NIIIP data protocols is the management protocols, which present the operations that can be applied to the product data. Based on the protocols, the NIIIP Consortium built the NIIIP reference model (NIIIP 1998). This model has been tested under many scenarios by member companies.

The NIIIP addresses these challenges of STEP integration (fine-granularity of STEP data models and the integration of STEP information) by defining STEP data objects. These data objects are analogous to the application objects identified in the application reference model (ARM) of each STEP model. The NIIIP also defines application-level services that can be applied to the data objects. The NIIIP provides four levels of implementation for its STEP integration: (0) legacy applications (for example, IGES-CAD); (1) applications with STEP Part 21 file I/O; (2) interoperability of ARM-level objects by standard data access interface (SDAI) application programmers interface (API); and (3) application-interpreted model (AIM)-level, where an interface definition language (IDL) interface manages access to STEP data in the CORBA environment (see Section 3 for details on these acronyms).

While the NIIIP details an extensive work and knowledge management view of the reference architecture (NIIIP 1998), the main difficulty is in the definition and management of this knowledge. In addition, agile manufacturing enterprises might be very short-lived, and there is a need for expedient methods and reuse of policy knowledge. This is where this paper makes a contribution.

2.4 Workflow and Workflow Management System

Workflow is the automation of a business process, in whole or part, during which documents, information, or tasks are passed from one participant to another for action according to a set of procedural rules. Correspondingly, a workflow management system provides procedural automation of a business process by managing the sequence of work activities and the invocation of appropriate human and/or information technology (IT) resources associated with the various activity steps (WfMC 1993).

A comprehensive description of workflow and the workflow management system, including the systematic design and implementation approaches for both build time and run time, has been provided by the literature to encompass function, operation, behavior, information, and organization perspectives of workflow (Jablonski and Bussler 1996). A good number of tools and technologies have been developed for or applied to workflow modeling, including mobile script language, Petri nets, middleware, and object-oriented technology.

The development of IT and the enlargement of

company scale generate the requirement of integrating workflow management systems into distributed computing environments. Within a highly distributed environment, several workflow management systems must interoperate with each other to enforce a distributed workflow. The communication and interoperation have become popular topics, covered abundant literature. Many architectures and frameworks have been proposed during the past decade to enable the distribution of workflow management systems. Based on these works, the Workflow Management Coalition tries to standardize workflow process and specification using the meta model approach and workflow process definition language (WfMC 1999).

Despite the enriched literature library on workflow management systems, little research has been done on their integration in the virtual organization setting. Reducing network load and resolving conflicts between the internal workflow of an individual partner and the interorganizational workflow are major concerns in this field. Some empirical approaches (Sengupta and Zhao 1998) and a framework (Subramanian 1998) for reengineering the workflow and workflow management systems to meet the requirement of virtual enterprises have been undertaken as initial steps in this field. In this paper, workflow management system concepts have been applied to agile manufacturing virtual enterprises with full consideration of partner autonomy and network load.

2.5 Emerging Information Technologies

With the explosive expansion of the Internet, communication in the business world increasingly relies on the emerging information technologies, such as network programming, component-based architectures, and so on. Many of these technologies have the potential of being employed by the information systems of virtual enterprises because the nature of the Internet is to establish connection between various computing environments.

Among the connections, the most influential one to the information systems of virtual enterprises is middleware. As the name suggests, middleware is a software layer residing between the application and the network infrastructure of multiple protocols. The basic concept of middleware is to centralize the communication mechanism and shield applications from the details of interprocess communications. Therefore, the implementation of middleware can drastically simplify interfaces within the information system and facilitate the communication with the legacy system (Vinoski 1997). There are several influential middleware products on the software market, such as Distributed Component Object Model (DCOM) by Microsoft and Digital Equipment, Remote Method Invocation (RMI) (JavaSoft 1997), Simple Object Access Protocol (SOAP) (SOAP 2000), and CORBA (OMG 1995b). In this paper, CORBA is used as the communication model. The reason for this choice as well as the structure of CORBA is discussed in the next section. For detailed information on DCOM and RMI, readers are encouraged to refer to the previous references.

Aglet is another fast-developing information technology that may be beneficial to the construction of the information systems of virtual enterprises. Originally, Aglet was developed by IBM Japan as a mobile extension of Java™ applets. It carries programs that can be dispatched from one computer and transported to a remote computer for execution. Once arriving at the remote computer, Aglet shows its credentials and then obtains access to local data. The major advantage of Aglet is to reduce the heavy network load caused by data exchange because it is slim applications instead of bulky data that travel on the network (Lange and Oshima 1998). The tentative works of integrating Aglet into information systems of virtual enterprises can be found in Papaioannou and Edwards (1999).

2.6 Summary

While the formation of virtual enterprises is becoming prevalent in manufacturing industry, there is a strong need for the development of a clear and easy-to-install information system framework by which partner companies can share desired information within a certain scope. The related topics, including distributed information infrastructure and workflow management, have been covered in a wide range of different techniques and by a large number of researchers. However, the importance and possibility of combining these techniques and applying them to agile manufacturing have only been recently noticed. This paper builds on these results to propose important contributions to a distributed information system architecture for agile manufacturing virtual enterprises.

3. Methodologies Applied in the Distributed Information Systems for Virtual Enterprises

"Information system support for the agile activities in Virtual Enterprises requires both a flexible database system and distributed communications software support" (DeVor 1997). The flexibility refers to the conformance of the information system to the heterogeneous data structures that exist on different clients, both geographically and organizationally. Therefore, a communication mechanism through the Internet or an intranet to accomplish the data-sharing goals between different autonomic individual plants in a virtual enterprise must be established. This task can be simplified by installation of unified data standards and implementation of middleware.

In the distributed information system (DIS) architecture presented in this paper, the backbones of data conformance and communication mechanisms are the STEP standard by the International Organization for Standardization (ISO) and CORBA by the Object Management Group (OMG) (Needham, MA). In this section, the concepts and integration of these technologies in the DIS framework will be discussed.

3.1 Production Data Exchange and Modeling Using STEP/EXPRESS

In this section, background information of STEP and EXPRESS will be presented (see STEP standards (ISO-10303) (Nell 1999) for details). In a typical virtual enterprise environment, numerous different software applications and computing platforms can be involved. Similar to other research initiatives, STEP is naturally adopted as a common and neutral way to model and exchange data in virtual enterprises.

3.1.1 International Standard for the Exchange of Product Data (STEP)

The International Standard for the Exchange of Product Data, generally known as STEP (ISO 10303), currently offers the desired neutral specification for product model data with its open and extensible structures. STEP data files are built on the foundation of specific standardized languages or format. These files can then be read directly by STEP processors or applications or through STEP access interfaces, allowing better integration of applications based on companies' disparate systems. To ensure the full utilization of modern object-oriented technologies, STEP is designed to be a modular standard that can be grouped into the following infrastructure components and industry-specific information models (STEP Tools Inc. 1999c):

- Product information description methods (parts 11-13), focusing on the information modeling language—EXPRESS.
- EXPRESS-driven data exchange structures and access specification (parts 21-26), including text encoding (part 21) and standard data access interface (SDAI) with bindings to the C, C++, and interface definition languages (IDLs), etc. (parts 22-26).
- Conformance testing methodology and framework (parts 31-35).
- Library of general-purpose information models for things like geometry, topology, product identification, dates, times, and so on (the 40-series parts).
- Industry-specific application protocols (AP) that are built from the library of the general model (the 200-series parts): Explicit Drafting (AP 201), Associative Drafting (AP 202), Configuration Controlled 3D Assemblies (AP 203), etc. All of these application protocols are developed by various industries, instead of by the software vendor or a third-party organization. Moreover, the application protocols are open ended in STEP; that is, any protocols in the product data domain can be added to the STEP family.

As support for STEP is increasing, numerous efforts have been devoted to the explicit conversion between various data forms and STEP. Many converters are available for functional mappings between various production data forms and STEP data specifications. STEP Tools Inc. (Troy, NY) offers online translation services between IGES, DXF, and STEP (STEP Tools Inc. 1999b). Many CAD vendors, such as Pro/ENGINEER®, AutoCAD, and Solid Edges, also have the STEP format as one of their built-in file import/export formats. These facilities provide an easy means of mapping legacy data to upcoming STEP APs and provide a homogeneous data framework for the information system of virtual enterprises (see also Bhandarkar and Nagi

1999, Bhandarkar et al. 1999). Therefore, it is prudent to focus on the STEP standard while constructing distributed information systems of virtual enterprises and then handling the legacy data with the help of corresponding converter packages.

3.1.2 An Information Modeling Language: EXPRESS

EXPRESS was originally developed to specify an interface between design and manufacturing for product definitions in 1982 (Wilson 1987). After revisions and modifications, EXPRESS has gathered a rich collection of supports and mappings to the implementation program languages, such as C, C++, and Smalltalk. At present, EXPRESS is an ISO standard as a formal language to describe language-neutral information models (ISO 1994a).

EXPRESS represents information by schema, which are complicated data objects. A typical schema contains all or some of the following components: entity definitions, type definitions, correctness rules, and algorithmic definitions. With this particular structure, EXPRESS is capable of describing complex objects, functions, and algorithmic constraints. As one of the most influential data modeling languages in design/manufacturing, EXPRESS is employed to represent the information models of STEP. ISO is working on the mapping of data defined by various EXPRESS models, named EXPRESS-X (STEP Tools Inc. 1998).

In this work and prototypical implementation, EXPRESS is applied to construct the explicit structure of engineering data and management information that are in conformance with STEP.

3.2 Common Object Request Broker Architecture (CORBA)

To meet the requirement of interoperation between the numerous software and hardware in the market, various middlewares were developed in the last decade. Common Object Request Broker Architecture (CORBA) by the Object Management Group (OMG) is becoming one of the most influential among them because of its platform independence and interoperability properties. In 1995, the OMG announced the latest version, CORBA 2.0 specification (OMG 1995b). The capability of CORBA 2.0 has been proved by numerous successful implementations.

The most important feature of CORBA is interoperability, which allows intelligent agents to discover each other and exchange information automatically by introducing the mechanisms of object request brokers (ORBs). In addition, the OMG also specifies an extensive set of bus-related services (OMG 1998) and common facilities (OMG 1995a) for object manipulation, database storage arrangement, and object status externalization.

To facilitate the interoperability between various implementations, the OMG explicates a purely declarative language—the Interface Definition Language (IDL). The IDL can be applied to define the application programming interface (API) neutrally. Moreover, the interface defined in the IDL can be translated into implementation languages (C++/C, Smalltalk, or Java), following the OMG's mapping specifications. The generated codes are called skeletons on the server side and stubs on client side. Then, the server and client develop applications independently, based on skeletons and stubs. At last, the CORBA communication model will automate the communication between the server and client implementations.

In the basic CORBA communication model, the client applications invoke client requests. These requests include handles of the modules that will be applied to perform the corresponding operation. They will then be sent to the ORB. As the router between local and remote objects, the ORB is responsible for tracing relevant components through a naming service (locating the objects by names) or trading service (locating objects by properties). After the appropriate server machine is located, the request will be sent to the server-side ORB, possibly via Internet inter-ORB protocols (IIOPs). The server will activate the requested object. At the same time, an object adapter creates a socket endpoint, the address of the endpoint, and the interoperable object reference (IOR) that contains the machine name, port number, and object handles. Then, the object adapter registers the IOR with the ORB. When the IOR is returned to the client side, a proxy will construct a socket connection to the server with the endpoint address. At this stage, the communication will be established successfully.

3.3 Comparison of CORBA and Simple Object Access Protocol (SOAP)

In the fast-moving IT world, new object and exchange paradigms emerge rapidly. One such

recent development that contends with CORBA is the Simple Object Access Protocol (SOAP), which is an XML-based lightweight protocol for information exchange. SOAP consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses (SOAP 2000). SOAP codifies the existing practice of using XML and HTTP as a method-invocation mechanism. More details about SOAP can be found at www.w3.org (SOAP 2000).

CORBA is not a markup language like SOAP. CORBA works as a go-between, making the connection and communication between heterogeneous distributed and object-oriented applications possible. CORBA offers a solid and comprehensive development platform for large-scale projects in which managing complexity and ensuring reliability are major concerns. It also ensures excellent transaction rates and fail-over capabilities. Although CORBA is powerful, it is difficult to develop and deploy. The choice of CORBA for small projects may not justify its performance benefits because it requires considerable code development.

In contrast, SOAP is built from two simple and well-known standards, HTTP and XML, and is furnished with the smallest possible XML vocabulary to package the essential data for RPC. It is easy to learn and implement. The use of the two textual protocols makes network data transfer heavy, requiring more bandwidth, but testing and debugging become easier.

SOAP is a wire protocol. SOAP lacks the activation elements, security, and state management that CORBA provides. The XML data inside the SOAP envelope must be extracted and parsed, thereby degrading the performance. In comparison to CORBA's binary data, SOAP's text-based data consume significantly more bandwidth.

Finally, the Object Management Group has taken important steps to ensure compatibility between CORBA and new communication protocols. There are a number of bridges, including open-source projects, capable of translating SOAP requests into CORBA invocations. Thus, by utilizing SOAP, CORBA can still remain as first choice for large-scale projects where performance and reliability are critically important.

3.4 Integration of STEP and CORBA

STEP offers virtual enterprises the homogeneous data structure and CORBA provides the transparent communication between heterogeneous computing environment. They work hand in hand to solve the interoperability problem, which is critical for virtual enterprise information systems. Recently, there have been a number of papers devoted to this topic. Hardwick et al. (1996) discuss the issue of sharing manufacturing information in virtual enterprises and describe a prototype information infrastructure that combines the Internet, STEP, and CORBA. Segarra (1999) discusses the advanced IT innovation roadmap based on the CIMOSA and ESPRIT projects. This paper delegates the technological integration to emerging standards such as STEP, CORBA, Java, and so on. However, these works are conceptual expositions and their focus is not in workflow management.

There are two implementation methods, physical file and standard data access interface (SDAI), in STEP to manipulate information represented by application protocols. In the physical file implementation, information in EXPRESS-defined data sets is stored as physical files, such as plain ASCII files. Part 21 (ISO 1994b) defines the exchange file format. This method is straightforward. The client and server applications in the distributed information system can manipulate the physical files of EXPRESS in the same manner as any ordinary data file.

The second implementation method—SDAI—defines protocols to access the EXPRESS database. The server applications can manipulate data in an EXPRESS database through implementations of SDAI. In Part 22 (ISO 1995a), STEP offers two different methods for data distribution, namely early binding and late binding. They both can cooperate with CORBA standards. The remainder of this section discusses these two implementations in detail and then elaborates on their usage in the proposed architecture.

In the integration of early binding (see *Figure 1*), each application schema defined by EXPRESS is translated into functions and attributes in the implementation language on the server. For the translations to C++ and C, ISO defines Part 23 (ISO 1995b) and Part 24 (ISO 1995c), respectively. The server applications then retrieve the data for the entities through generated functions in the binding to accomplish the SDAI implementations.

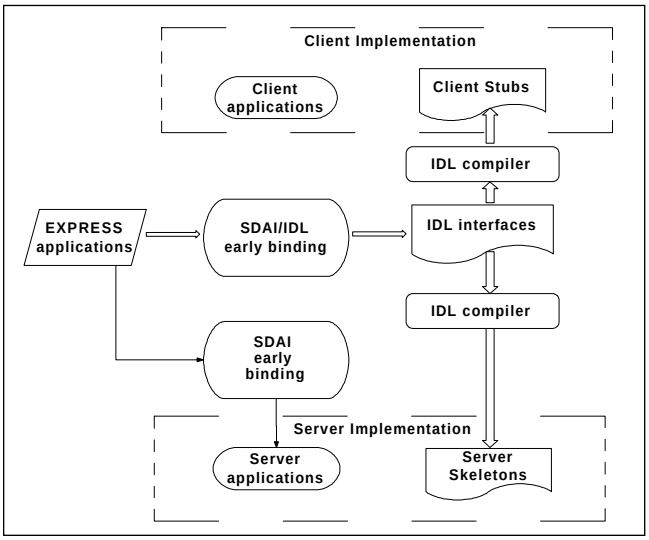

Figure 1
Early Binding of EXPRESS and CORBA

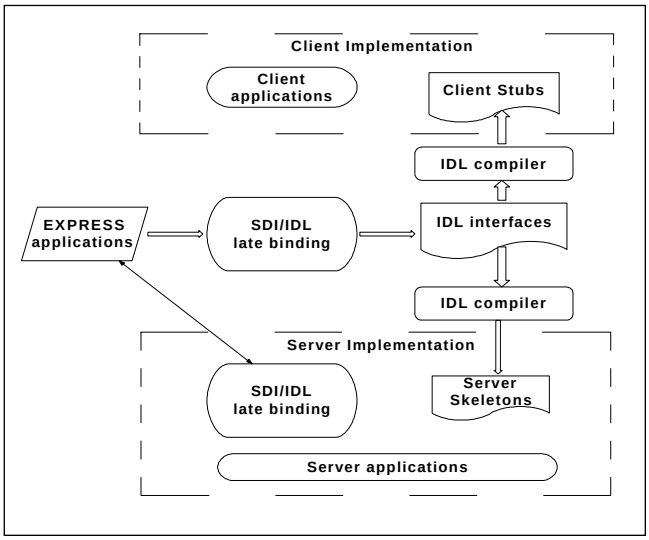

Figure 2
Late Binding of EXPRESS and CORBA

On the other hand, the EXPRESS data model should be mapped to CORBA IDL files, following STEP Part 26 (ISO 1996). The generated IDL files are used to produce the skeletons for the server and the stubs for the client. At last, the skeletons and stubs can be linked to the implementations from early binding to connect the communication channel between client and server.

In the case of late binding (see *Figure 2*), the EXPRESS data models are converted into CORBA IDL and form the dictionary interface repository. The generated interfaces must be translated into server skeletons and client stubs by the IDL compiler before their use by applications. The server applications can manipulate the information defined in EXPRESS data models through instantiating the corresponding dictionary interfaces. Then the CORBA communication schema helps the client applications to retrieve the data through the server. The early binding approach is application oriented; that is, the data managed by CORBA are decided by the corresponding EXPRESS entities. This approach relies on the direct generation of IDL interfaces from EXPRESS schema that define the objects to be distributed. Examples of this kind of direct generation can be found in ST-Developer from STEP Tools, Inc. (1999a). On the contrary, the late binding approach depends on the specified distribution of the STEP meta data. Each of the EXPRESS schemas has an instance counterpart in the IDL interfaces dictionary, referring to STEP Part 26.

To keep the real data interoperability between applications, the late binding approach is preferred to integrate virtual enterprise information into the distributed information system architecture (see Zhou 1999 for reasons). The architecture of the distributed information system for virtual enterprises, enabled by the CORBA communication standard and these binding approaches, is presented next.

4. Distributed Information System Architecture for Virtual Enterprises

According to the requirements of virtual enterprises, a distributed information system (DIS) architecture is proposed based on standards for information representation and exchange. It is related to the previous work of Song and Nagi (1997) on the agile manufacturing information system but incorporates workflow management for information control (see section 5). In this section, the structure and components of this DIS prototype and its characteristics will be discussed.

4.1 Structure of DIS Architecture

Usually, a functional virtual enterprise organization consists of several autonomous partners who collaborate on one-of projects. The main task of that project is then partitioned into several subtasks, according to various capabilities and fortes of individual partners. For instance, a particular product's realization tasks can be divided into design, process planning, manufacturing resource planning, and NC machining, and each of these subtasks can be

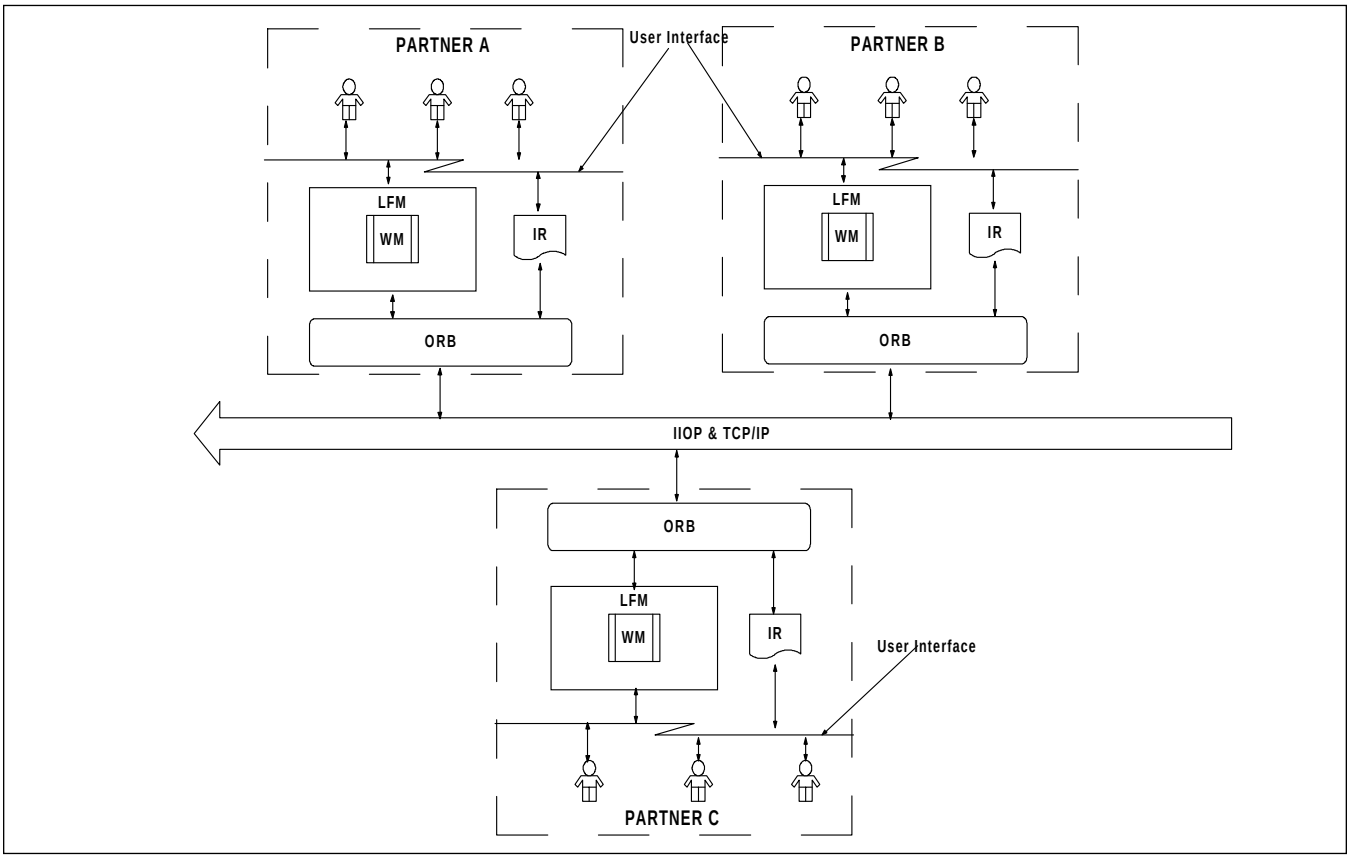

Figure 2
Late Binding of EXPRESS and CORBA

assigned to one of the partners of the virtual enterprise. To meet this specific requirement, each partner is represented as a separate functional module in this architecture (see *Figure 3*). The communications between the modules are enabled by an Internet inter-ORB protocol (IIOP) and TCP/IP on the network.

Zooming into the functional modules, it is easy to see that each of them is composed of five interconnected components: local function module (LFM), workflow manager, interface repository (IR), user interface (UI), and object request broker (ORB).

The local function module represents the partners' local system application packages, which are committed to execute the subtasks of each partner site in the virtual enterprise. It may include server/client implementations and supporting function modules, such as database management system, and CAD, CAPP packages, based on the manufacturing subtask of the designated partner.

The workflow manager is responsible for implementing business logic, partner policies, and application sequences. As the control panel of information exchange in the DIS, the workflow manager reflects the partner policies on information flow for every business/manufacturing transaction. The application packages in the LFM must follow the guidance of the workflow manager strictly to keep the DIS data consistency and perform business functions properly. The construction of the workflow control systems is sophisticated but extremely important to avoid the loss of control on the virtual enterprise business processes (McCusker 1993). Section 5 will be dedicated to extend the discussion on the systematic construction of the workflow management approach.

The interface repository is a specific form of database for the object interfaces, as its name suggests. The interfaces registered in the IR are in the form of meta data; that is, they include self-describing information that is ready to use for other partner sites. The owner of the interfaces publishes those relevant to the virtual enterprise tasks into the local IR, which is transparent to every other partner within the virtual enterprise. Partners can look up and retrieve the corresponding interfaces and then devel-

op applications based on the generated IDL stubs and the information included in the meta data of interfaces. This transparency of IRs and usage of meta data opens the partner forums throughout the organizational structure, enabling large-scale data sharing and distribution in the virtual enterprise effectively.

The basic function of the UI is to provide users with a tool to communicate with the information system. It can be in any format, from the MS DOS or UNIX command line to the user-friendly graphic user interface (GUI). The choice of UI purely depends on the preference of the partner site and the ability of application developing tools. It will not affect the overall structure of the DIS architecture.

ORBs are in charge of resolving requests for object references, tracing proper object locations, enabling application components, and establishing connections to proper partner sites. In other words, ORBs carry all of the information exchange within the virtual enterprise as a public channel. ORBs also define the connection mechanism throughout the DIS. The functional applications in LFM can apply this mechanism by simply imbedding ORB calls in the context of user-specified programs.

4.2 Characteristics of DIS Architecture

Corresponding to the basic virtual enterprise objectives, this DIS architecture has the following characteristics:

1. **The system is scalable and easy to install.** The communication of the DIS is enabled by IIOPs and the Internet, which has become an indispensable tool for industries. Therefore, an industry partner can simply join the virtual enterprise information environment by installing CORBA and IIOPs, without any further overhead considerations. The connection will be automated by ORB communication schema and Internet protocols. This plug-and-play feature enables the fast and flexible establishment of a virtual enterprise information system to meet the short product cycle timing requirement. This installation does not affect the settings and operations of legacy information systems at partner sites. In addition, partners can join or quit the virtual enterprise at any point in its life span without damaging the remaining information system structure.
2. **Partners in this architecture are autonomous.** In this DIS architecture, the specifications of CORBA guarantee object location transparency, while the IR in meta data format enables the transparency of application interfaces. Therefore, partners should be able to develop and deploy application packages to manipulate data under the guidance of their own organizational policies. No interference from other members is possible because only information that flows in or out of partner sites is of concern to the other members of the virtual enterprise. In addition, client and server functions are unified into the individual partner sites. This feature gives partners more flexibility while designing their local virtual enterprise logic implementations. Any legacy applications can be delegated to fulfill server or client functions according to the preferences and policies of partners.
3. **Network load is reduced.** This DIS architecture adopts the three-tier communication concept. Clients issue the requests for information and wait for the incoming responses from the corresponding servers, instead of accessing and manipulating data on server sites directly. Therefore, the data flow through the network is cut down significantly. Only the inevitable data exchanges, such as the data stream for establishing appropriate communications and the processed objects required by client applications, instead of the bulky raw data file transference occur on the network. The communication schema of this DIS architecture supports multithread objects. Once a server application has been started, it can also wait for incoming client requests and dispatch one thread for each invocation. There is no need to instantiate one server process for each client call. Therefore, the network resource and computing time for server initiation and instantiation are spared.
4. **The components of this architecture are reusable.** All the components of this architecture are designed and implemented to minimize duplication of functionality. Therefore, after dissolution of a virtual enterprise, partners are able to join or organize another one with only minor modifications on the existing system and applications. For instance, the generated SDAI late binding dictionary is general purpose. Therefore, the application codes are reusable

and can be linked to the skeletons or stubs for new interfaces. The contents in the rulebase of the workflow manager are also partially reusable and very helpful as a reference for the construction of another virtual enterprise information system. Because a virtual enterprise is a temporary organization in most cases, this feature can bring both dramatic cost and time savings while organizing various virtual enterprise information environments.

5. **Workflow manager reflects the dynamic aspect of a virtual enterprise.** To address the dynamic characteristics of the contemporary market-driven environment, partner policies and business logic in a virtual enterprise are far from static. They are always subject to modifications under the unanimous consent of virtual enterprise partners. Once policies are modified or appended, the affected partners need to update their workflow manager rulebases according to the new policies. In addition, the business processes of this DIS architecture will be changed dynamically by retrieving and following the new rules defined in the workflow manager.

According to the specific requirement of distributed information systems for virtual enterprises, the DIS architecture proposed in this paper helps overcome limitations of the previous work of Song and Nagi (1997). Constructed on standards, CORBA and STEP, this architecture can be compatible with the legacy systems on virtual enterprise partner sites, and the construction process of this architecture can be shortened by the existing mapping protocols between the standards. As the engine of information flow in the virtual enterprise, the workflow manager is very important for the proper functioning of this DIS architecture. The next section presents a systematic approach for workflow manager construction. A basic implementation of this prototype architecture with a simple workflow manager is developed and discussed in Section 6.

5. Workflow Manager for Distributed Information Systems Architecture

Construction of the workflow manager is a process of mapping the business logic and partner policies of the virtual enterprise into computer-recognizable scripts. It is an extremely important step in the DIS framework because the workflow manager is critical to the overall physical and logical performance. This section describes the basic building blocks of workflows in the virtual enterprise. The authors emphasize the composition of virtual enterprise workflows and their relation to legacy workflows, which exist at individual partner sites. Based on this structure, a systematic approach for virtual enterprise workflow manager construction is developed to enable the concurrent operations of virtual enterprise business logic.

5.1 Workflow Hierarchy of Virtual Enterprises

The workflows of virtual enterprises can be divided naturally into two layers—virtual enterprise organizational workflows and local workflows. The local workflow layer can be further categorized into two groups—virtual enterprise local workflow and local legacy workflow. Virtual enterprise organizational workflow represents joint policies, such as the role of each partner, how to allocate joint tasks, how to collaborate with each other, and their information dependence and flow. Virtual enterprise local workflow deals with the business processes on individual partner sites to fulfill the corresponding responsibilities required by the organizational level workflow. To keep the partner autonomy, virtual enterprise local workflow is treated as a “black box” at the virtual enterprise organizational level. The communication between the two workflow levels includes no more than information input and output. Moreover, the upper level should not specify the implementation methods of individual partner responsibility. This vision is enabled in the proposed DIS architecture

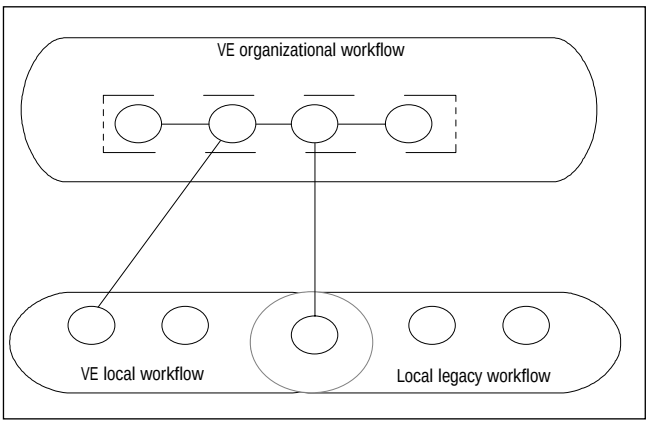

Figure 4
Hierarchy of Workflow Manager in Virtual Enterprise Vision

by the transparence of ORB and CORBA services. The virtual enterprise local workflow model should also be compatible with the local legacy workflow, which has already been implemented on a partner site before the formation of the current virtual enterprise. *Figure 4* illustrates an abstract relationship among virtual enterprise organizational, virtual enterprise local, and local legacy workflow.

The remainder of this section focuses on the organizational level of virtual enterprise workflows and elaborates on a systematic approach for workflow manager construction. This approach is illustrated by an example for easier understanding.

5.2 Systematic Approach for Workflow Manager Construction

Based on the agile manufacturing information system (Song and Nagi 1997) and a prototype of workflow manager (Subramanian 1998), a systematic approach for workflow manager construction is developed. As illustrated in *Figure 5*, this approach includes four major steps—policy description, policy modeling, policy verification, and policy construction—to ensure the validity and performance of the generated rules.

Policy Description

The primary goal of this step is to document the virtual enterprise partner policies and workflow specifications, which are essential to fulfill the business logic of the partner or the virtual enterprise. These specifications include a list of transactions, potential data flow between transactions, partner agents involved in the transactions, and specifications of the virtual enterprise environment. Tasks in this step should cover the following aspects:

- Identify and name the required transactions under the business logic of the partner or virtual enterprise.
- Trace the route of workflow and data flow to accomplish these transactions. This involves specification of preconditions and post-conditions to formulate the network of transactions.
- Determine the roles of partners involved, including what kind of decisions they can make and to what extent they can take control of the workflow. Define every applicable business process outside the virtual enterprise environment, such as interface with raw material suppliers, etc.
- Prioritize the workflow transactions to be performed and issue passwords to the partner processes involved. The priorities and access privileges will be further delegated to local agents or processes on the corresponding partner site, according to the specifications of the virtual enterprise local workflows.

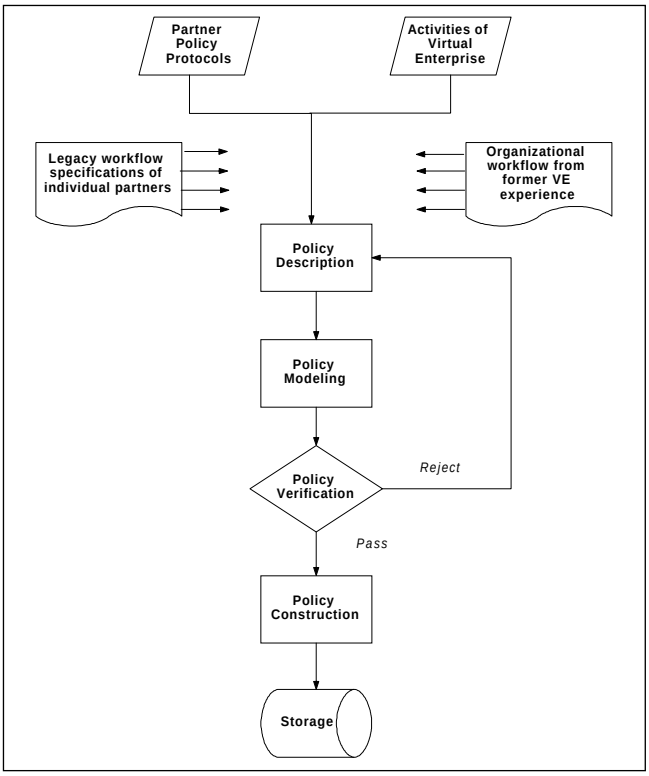

Figure 5
Flowchart for Workflow Manager Construction

Each partner in the virtual enterprise must be involved in this description process to ensure the validity and comprehensiveness of the generated policies. In addition, the local workflow system and former organizational workflow experiences are also highly valuable references in this enumeration. Those formerly constructed policies and workflow are still applicable with modifications, due to the reusability of object-oriented technology. The output of this step is the comprehensively documented transaction network.

As an example, a virtual enterprise scenario consisting of three partners is simulated. The requirement for the product data update has been identified. The partner policy here is that all the engineering changes should be initiated and committed by the

Transaction ID		EC 1.0
Transaction name		Engineering change
Partner tasks	Owner of data	• Initiate request of engineering change • Request related partners to vote on engineering change • Commit or cancel engineering change according to votes
	Other related partner of data	• Vote on engineering change after requested, either positive or negative

Figure 6
Workflow Description of Engineering Change

Workflow ID		EC 1.0-sub(A)
Workflow name		Engineering change review
Tasks	Designer	• Review engineering change on request • Issue positive vote to deliver EC request to supervisor or negative vote to terminate workflow
	Supervisor	• Review engineering change on request • Issue positive or negative vote to terminate workflow

Figure 7
Local Workflow Description of Engineering Change

owner of the product data with unanimous positive votes from the partner sites related to the product data. The generated document for this example workflow in virtual enterprise organizational level is shown in *Figure 6*. As mentioned before, the local workflows that deal with the vote process can be different from one partner site to another partner site. A sample local workflow on a partner site, which involves the designer and supervisor on this site, is described in *Figure 7*.

Policy Modeling

Certainly, it is imprudent to expect the descriptions of business logic and partner workflow, which are generated in the first step, to be flawless and complete. What needs to be done next is to reduce the complexity and redundancy of the prescribed procedures, check the validity of the policies, and locate the conflicts within the workflow if any exist. To facilitate this goal, the documented partner policies and workflow in the architecture are proposed to be mapped into Petri net models.

At the beginning of 1960s, the Petri net was designed by Carl Adam Petri as a formal and graphically appealing language to model discrete parallel systems (Petri 1962). Since then, various extensions of Petri nets, such as colored Petri net (Jensen 1994), timed Petri net (Ramchandani 1974), and object Petri net (Lakos 1994), have been developed for more complex system modeling. Petri nets provide the proposed information system with their capability of representing concurrent interactions between multiple operators and multiple systems. Depending on the level of modeling and analysis capability, an appropriate Petri net version can be employed.

Policy Verification

The purpose of this step is to check the validity of the constructed models for policies, to detect the redundancy in the policy model, and to identify any potential conflicts. If the models fail in this verification, one should go back to the first step and make necessary adjustments or corrections.

Corresponding to the workflow manager structure that was described previously, the Petri net (PN) modeling of connections between organizational and local virtual enterprise workflows should conform to the connection rules of free-choice Petri nets (Baccelli 1996). Each arc from a place is restricted to being either the unique output of that place or the unique input to a transition. By this, it is meant that the conflicts can be localized so that a place can decide for itself where to send its tokens. Each place and transition can also be viewed as a process that could be considered separately, the arcs between them as message streams, and a token as a message. This feature is desirable to facilitate the generic construction and maintenance of virtual enterprise workflows. Each place in the organizational level workflow model can be considered as a workflow in the local level workflow. In this case, any modifications of the lower level workflow do not affect the function of an organizational workflow and the integrity of the overall virtual enterprise workflow

system, as long as the interfaces between them remain the same.

Simulation tools are committed to validate Petri net models of the rules. A major advantage is that the animation results provide powerful visualization tools that can be shown to every partner who should collaborate during a transaction.

According to the description of engineering change transactions in the last subsection, the primitive model for this organizational workflow is built with place and transition Petri nets. Then, with help of visual object net (Drath 1998), the structure analysis to the model constructed in the previous step is performed. To resolve structural conflicts, redundancies, and misfirings that are detected by the simulation process, one should go back to the first step and recheck the description of the appropriate workflows. Based on this analysis, modifications are made in this transaction model. The final version of the Petri net model is shown in *Figure 8*. Following the PN model convention, circles stand for places, black bars for transitions, black dots for tokens, and arrows indicate the directions of information flow. In this model, P5 and P7 represent the positive (existence) and negative (nonexistence) response to the request, respectively. They share the same token resource. Therefore, they are not allowed to be occupied by tokens at the same time. The same happens to the place pairs, (P11,P18), (P22,P23), and (P24,P25). In addition, this model can be extended to fit into a scenario with more partners by simply copying the setting and connections of existing partners.

This model passes the structural validation, and the simulation result reflects how this workflow can be executed accurately. On the other hand, this model conforms to the virtual enterprise workflow

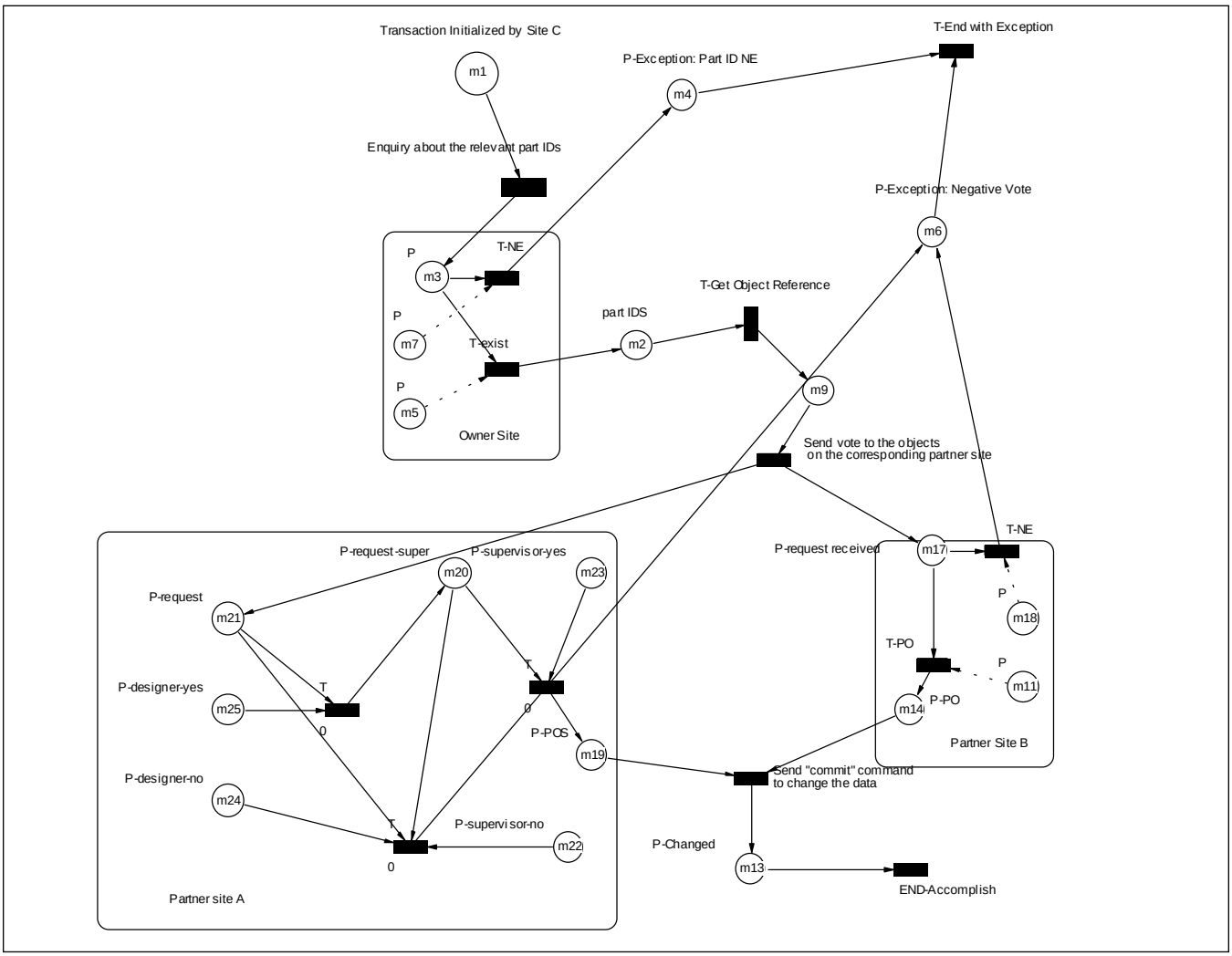

Figure 8
Refined Petri Net Model of Engineering Change Transaction

```
MODEL                "Engineering Change"

     NAME
     MODEL_VERSION          "1.0 Beta"
     AUTHOR                 "IE Student in UB"
     CREATED                1999-6-4
     DESCRIPTION            "This is a sample to illustrate the
                            construction of workflow in the DIS for virtue
                            enterprise"
     STATUS                 DRAFT

     //Workflow participant list
     PARTICIPANT            "Owner"
          NAME              "Owner"
          DESCRIPTION       "Owner of the product data to be changed"
          TYPE              ORGANIZATIONAL_UNIT
          STATUS            Default
     END_PARTICIPANT

     PARTICIPANT            "Partner A"
          NAME              "Partner A"
          DESCRIPTION       "User of the product data to be changed"
          TYPE              ORGANIZATIONAL_UNIT
     END_PARTICIPANT

     PARTICIPANT            "Partner B"
          NAME              "Partner B"
          DESCRIPTION       "User of the product data to be changed"
          TYPE              ORGANIZATIONAL_UNIT
     END_PARTICIPANT

     //Workflow application description
     WORKFLOW               Initiate request of engineering
                       change on product data

     ACTIVITY               "Initiate request of engineering
                    change on product data"
          NAME              "Initiate request"
          IMPLEMENTATION    ATOMIC
          EXCEPTION         NO
     END_ACTIVITY
```

Figure 9
Meta Data Model of Engineering Change Transaction (partial)

manager architecture described in *Figure 4*. Each place or transition can be delegated into the virtual enterprise local workflow, which should be elaborated within the range of the local partner. For illustration purposes, the local workflow described in *Figure 7* is incorporated in this model (*Figure 8*).

Policy Construction

Based on the constructed PN model, expert rules are defined in this step. The rules are presented in the format of computer-processable pseudo-codes, known as process definitions. These definitions can be interpreted by workflow engines of workflow management system software at run time to initiate and automate the corresponding work process. The generated expert rules are stored into the rule base of the workflow manager, which is usually called the knowledge base (Subramanian 1998). The knowledge base is scalable and subject to additions, revisions, and deletions to reflect the dynamic characteristics of virtual enterprise business practices. However, any modifications of the knowledge base will dynamically change the performance of the virtual enterprise. Therefore, any modifications of the knowledge base should be made with the unanimous agreement of every partner in the virtual enterprise and should be modeled into Petri nets to ensure the validity and compatibility with the existing policies. *Figure 9* shows part of the sample process definition derived from the engineering change transaction model. This is generated automatically from the PN model.

6. Prototype Implementation of a Distributed Information System for a Virtual Enterprise

Based on the proposed system architecture, a prototype system is implemented for illustrative purposes. It starts from a sample EXPRESS data model (see “picture.exp” below) and demonstrates the construction process of partner server and partner client. The sample workflow presented in the previous section is also incorporated in this prototype to present the function of workflow manager.

This EXPRESS model is then compiled by the *exp2idl* compiler provided by STEP Tools Inc. (1999a). In this step, the data model is mapped into a file, namely “picture.idl,” defined by the interface definition language (IDL) from the Object Management Group. After some minor customizations, the IDL file is ready to be used as input to the CORBA communication schema.

```
//picture.exp
SCHEMA picture;
ENTITY Line;
enda : Point;
endb : Point;
END_ENTITY;

ENTITY Circle;
radius : REAL;
center : Point;
END_ENTITY;
END_SCHEMA;
```

```
//picture.idl
module PictureORB{
.................
//define circle model
struct circleStruct
{
short CircleNo;
float Cir_cen_x;
float Cir_cen_y;
float Cir_radius;
string Status;
};
```

With the help of *idl2java* and *idl2cpp* compilers from Visigenic’s Visibroker software, the CORBA stubs and skeletons are generated. Based on server skeletons, a Java partner server is developed. This prototype uses a JDBC-ODBC bridge to access the data that are prestored in a FoxPro database. Based on stubs, the partner clients are developed in Java and C++, respectively, to illustrate the heterogeneous virtual enterprise computing environment.

Once started, the server program will try to contact the local ORB agent, establish connection, and register the "Picture" objects. The ORB agent will listen to the valid incoming requests continuously. At this stage, the server is ready to be launched. Then, partners in the virtual enterprise can access "Picture" objects through client programs without specifying the destination server. CORBA will locate the objects and establish the communication automatically. The data in the server database also can be manipulated through this interconnection. The server and client codes can be found in Zhou (1999). A screen shot of the Java client's main interface is shown in *Figure 10*.

To illustrate the incorporation of workflow in this DIS architecture, the engineering change workflow example defined in Section 5.2 is also implemented. Based on the computerized meta data of the workflow in *Figure 9*, the electronic mail system is employed as the means to send engineering change requests and collect votes.

The graphic user interface (GUI) is shown in *Figure 11*. Users can click the "view" button to browse the circle data in the "Picture" database on the server and send requests of engineering changes by clicking the "Edit" button. An e-mail will be sent to designated partners. The program will listen to the vote from the partner side. If a positive response is received, the "Picture" database will be modified according to the request. If a negative feedback is received, the workflow will be terminated with no further action.

Starting from the same EXPRESS data model and CORBA interface, programming tasks of individual partners in this prototype implementation are independent with each other. In addition, many coding tasks are simplified by the automatic mapping between well-defined protocols. Therefore, the development span for this system is significantly shortened.

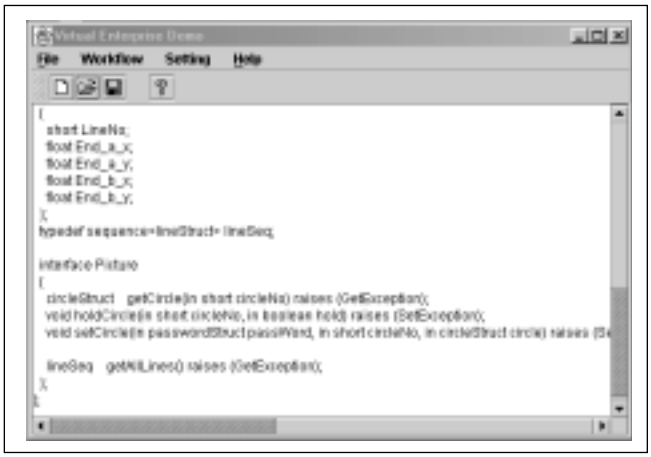

Figure 10
Main Frame of Java™ Client Program

Figure 11
Graphic User Interface for Engineering Change Workflow Implementation

Conclusions and Recommendations

Today's global market environment generates a new format of organizations, referred to as virtual enterprises. Manufacturers share capabilities and advantages with each other in a virtual enterprise to meet the requirements of fast delivery of customized products and quick response to dynamic market demands. In this paper, a systematic approach to design and implementation of a distributed information system for virtual enterprises was presented to solve information communication problems in the heterogeneous computing environment of the virtual enterprise.

After identifying the key features of information sharing within the virtual enterprise scope, this paper proposed a DIS architecture to enable the data autonomy and dynamic workflow automation while also considering the reusability of the DIS components and the reduction of network load. In this DIS architecture, the Common Object Request Broker Architecture (CORBA) and the International Standard for the Exchange of Product Data (STEP) serve the purpose of a transparent communication channel and a uniform data model format, respec-

tively, to overcome the heterogeneity and promote standardization. To assist the functions of the DIS, a four-step systematic approach is proposed to construct the control panel of the virtual enterprise, namely the workflow manager. Through policy specification, verification/simulation, and automation, this approach can synergize the organizational and local workflows in virtual enterprises.

In summary, this research proposes a distribution information system architecture that employs the CORBA and STEP standards for information exchange and representation and a workflow management based solution for information control in virtual enterprises. Future work in this area can focus on the supervisory capabilities for scheduling and monitoring of complicated workflows, such as to prioritize workflows or track the progress of workflows. In addition, the solution proposed in this paper can also be extended by developing interface protocols that can communicate with third-party utility software, such as commercial workflow software, CAD, and CAPP packages.

Acknowledgments

The authors acknowledge the support of the National Science Foundation under grant DMI-9624309.

References

Baccelli, F. (1996). “Free-choice Petri nets - An algebraic approach.” *IEEE Trans. on Automatic Control* (v41, n12), pp1751-1778.

Ball, S. (1997). “Enterprise enablement for Java applications.” Technical Report. XDB Systems.

Bhandarkar, M.P. and Nagi, R. (2000). “STEP-based feature extraction from step geometry for agile manufacturing.” *Computers in Industry* (v41), pp3-24.

Bhandarkar, M.P.; Downie, B.; Hardwick, M.; and Nagi, R. (2000). “Migration from IGES to STEP: One-to-one translation of IGES drawing to STEP drafting data.” *Computers in Industry* (v41), pp261-277.

DeVor, R. (1997). “Agile manufacturing research: accomplishments and opportunities.” . IIE Trans. (v29), pp813-823.

Drath, R. (1998). “Visual object oriented Petri net based engineering tool.” Technical Report. Technical Univ. of Ilmenau.

Flatau, U. (1988). *Design and Analysis of Integrated Manufacturing Systems*. Washington, DC: National Academy Press.

Goldman, S.L.; Nagel, R.N.; and Preiss, K. (1995). *Agile Competition and Virtual Organizations: Strategies for Enriching Customers*. New York: Van Nostrand Reinhold.

Hardwick, M.; Spooner, D.-L.; Rando, T.; and Morris, K.C. (1996). “Sharing manufacturing information in virtual enterprises.” *Communications of the ACM* (v39, n2), pp6-54.

Hardwick, M.; Spooner, D.L.; Rando, T.; and Morris, K.C. (1997). “Data protocols for the industrial virtual enterprise.” *IEEE Internet Computing* (v1, n1), pp20-29.

Iacocca, L. (1991). “21st century manufacturing enterprise strategy.” Technical Report. Bethlehem, PA: Lehigh Univ.

ISO (1994a). “Industrial automation systems and integration - Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.” Technical Report ISO 10303-11:1994(E).

ISO (1994b). “Industrial automation systems and integration - Product data representation and exchange—Part 21: Implementation methods: Clear text encoding of the exchange structure.” Technical Report ISO 10303-21. Geneva, Switzerland: ISO.

ISO (1995a). “Industrial automation systems and integration - Product data representation and exchange—Part 22: STEP data access interface.” Technical Report ISO Document TC184/SC4 WG7 N392. Geneva, Switzerland: ISO.

ISO (1995b). “Industrial automation systems and integration - Product data representation and exchange—Part 23: C++ language binding to the standard data access interface specification.” Technical Report ISO Document TC184/SC4 WG7 N393. Geneva, Switzerland: ISO.

ISO (1995c). “Industrial automation systems and integration - Product data representation and exchange—Part 24: Standard data access interface — c language late binding.” Technical Report ISO Document TC184/SC4 WG7 N394. Geneva, Switzerland: ISO.

ISO (1996). “Industrial automation systems - Product data representation and exchange—Part 26, IDL binding for the standard data access interface. Technical Report ISO 10303-26. Geneva, Switzerland: ISO.

Jablonski, S. and Bussler, C. (1996). *Workflow Management Modeling Concepts, Architecture and Implementation*. Int’l Thomson Computer Press.

JavaSoft (1997). “Java remote method invocation-distributed computing for Java.” Technical Report. JavaSoft.

Jensen, C.K. (1994). “An introduction to the theoretical aspects of colored Petri nets.” Technical Report. Denmark: Aarhus Univ., Computer Science Dept.

Jordan Jr., J.A. and Michel, F.J. (2000). *Next Generation Manufacturing: Methods and Techniques* New York: John Wiley & Sons.

Krishnamurthy, V.; Su, Y.W.; Lam, H.; Mitchell, M.; and Barkrneyer, E. (1988). “IMDAS-an integrated manufacturing data administration system.” *Data Knowledge Engineering* (v3), pp109-131.

Lakos, C.A. (1994). “Object Petri nets - Definition and relationship to coloured nets.” Technical Report. Univ. of Tasmania, Dept. of Computer Science.

Lange, D.B. and Oshima, M. (1998). *Programming and Deploying Java(tm) Mobile Agents with Aglets*. Reading, MA: Addison Wesley Computer and Engg. Publishing Group.

McCusker, T. (1993). “Workflow takes on the enterprise.” *Datamation* (v39, n88).

Nell, J. (1999). “STEP on a page.” Technical Report. Gaithersburg, MD: Gaithersburg, MD, 1999.

NIIIP (National Industrial Information Infrastructure Protocols) (1998). “NIIIP Consortium Reference Architecture.” Technical Report. NIIIP Consortium.

OMG (1995a). “Common facilities architecture.” Technical Report. Needham, MA: Object Management Group.

OMG (1995b). “The common object request broker: Architecture and specification. Technical Report. Needham, MA: Object Management Group.

OMG (1998). “CORBA services: Common object services specification.” Technical Report. Needham, MA: Object Management Group.

Papaioannou, T. and Edwards, J. (1999). “Using mobile agents to improve the alignment between manufacturing and its support systems.” *Journal of Robotics and Autonomous Systems* (v27, n45), p57.

Petri, G. (1962). “Kommunication mit automaten.” *Technical Report*. Bonn, Germany: Univ. of Bonn.

Ramchandani, C. (1974). “Analysis of asynchronous concurrent systems by Petri nets.” Technical Report. Cambridge, MA: MIT.

Segarra, G. (1999). “Advanced information technology innovation roadmap.” *Computers in Industry* (v40, n2), pp185-195.

Sengupta, K. and Zhao, Z.L. (1998). “Designing workflow management systems for virtual organizations: An empirically grounded approach.” *Proc.* of 31st Hawaii Int’l Conf. on System Sciences (HICSS’98).

SOAP (2000). “Simple Object Access Protocol (SOAP) 1.1.” Technical Report, W3C.

Song, L. and Nagi, R. (1997). “Design and implementation of a virtual information system for agile manufacturing.” *IIE Trans. on Design and*

Mfg. (v29, n10), pp839-857.
STEP Tools Inc. (1998). "EXPRESS-X manual." Technical Report ISO TC184/SC4/WG11 N066. Troy, NY: STEP Tools Inc.
STEP Tools Inc. (1999a). "ST-Developer v7 online manuals." Technical Report. Troy, NY: STEP Tools Inc.
STEP Tools Inc. (1999b). "STEP translation service." Technical Report. Troy, NY: STEP Tools Inc.
STEP Tools Inc. (1999c). "The ISO STEP standards." Technical report. Troy, NY: STEP Tools Inc.
Subramanian, L. (1998). "A workflow management system for agile manufacturing enterprises." Master's thesis. Buffalo, NY: Univ. at Buffalo (SUNY).
Vinoski, S. (1997). "CORBA: Integrating diverse applications within distributed heterogeneous environments." *IEEE Communications* (v35, n2), pp46-55.
WfMC (1993). "The workflow reference model." Technical Report. Workflow Management Coalition.
WfMC (1999). "Process definition interchange process model." Technical Report. Workflow Management Coalition.
Wilson, P.R. (1987). "A short history of CAD data transfer standards." *IEEE Computer Graphics and Applications* (v7, n6), pp64-67.
Zhou, L. (1999). "Design of distributed information system for virtual enterprise using CORBA. Master's thesis. Buffalo, NY: Univ. at Buffalo (SUNY).

Authors' Biographies

Liangyu Zhou is a software engineer in Columbus, OH. He received his MS in industrial engineering from the University at Buffalo (SUNY) in 1999.

Rakesh Nagi is an associate professor of industrial engineering at the University at Buffalo (SUNY). He received his PhD (1991) and MS (1989) degrees in mechanical engineering from the University of Maryland at College Park while he worked at the Institute for Systems Research and INRIA (France). He is a recipient of SME's Milton C. Shaw Outstanding Young Manufacturing Engineer Award (1999), IIE's Outstanding Young Industrial Engineer Award in Academia (1999), and a National Science Foundation CAREER Award (1996). His papers have been published in journals including *IIE Transactions, International Journal of Production Research, Journal of Manufacturing Systems, International Journal of Flexible Manufacturing Systems, Journal of Intelligent Manufacturing, Computers in Industry, Computer Integrated Manufacturing Systems, Operations Research, Annals of Operations Research, Computers and Operations Research,* and *Computers and Industrial Engineering.* Dr. Nagi's major research thrust is in the area of production systems. His recent research interests are in agile enterprises, information-based manufacturing, and logistics and supply chain management.