Concurrent Engineering
http://cer.sagepub.com/

A Systematic Approach of Virtual Enterprising Through Knowledge Management Techniques

Yuh-Min Chen, Chia-Ching Liao and Biren Prasad
Concurrent Engineering 1998 6: 225
DOI: 10.1177/1063293X9800600305

The online version of this article can be found at:
http://cer.sagepub.com/content/6/3/225

Published by:
SAGE
http://www.sagepublications.com

CONCURRENT ENGINEERING: Research and Applications

A Systematic Approach of Virtual Enterprising Through Knowledge Management Techniques

Yuh-Min Chen* and Chia-Ching Liao

Institute of Manufacturing Engineering, National Cheng Kung University, Tainan, Taiwan 70101 ROC

Biren Prasad

KBE Product Line Manager, Unigraphics Solutions, P.O. Box 250254, West Bloomfield, MI 48325-0254 USA

Abstract: Virtual enterprising is viewed as one of the most promising business strategies for manufacturing industries to meet global competition. The essence of virtual enterprising is to integrate the processes, activities, and resources from different enterprises through enterprise alliance and thereby quickly respond to customer expectations. The management and sharing of knowledge across enterprises is the basis for virtual enterprising. A development of a knowledge management and sharing system involves not only the technologies of communication, information, collaborative workflow and software engineering but also managing the characteristics and processes of a satellite set of allied-enterprises. The development of an effective knowledge management system is still a technical challenge for virtual enterprising. In this context, information and data sharing are considered parts of a knowledge management (KM) function.

This paper presents a systematic approach to the development of a prototype knowledge management (KM) system that is able to support the intricacies of virtual enterprising. The prototype KM system development is done in two parts. The first part focuses on the characterization, definition, and modeling of virtual enterprises. The second part concentrates on design and modeling of an information sharing system. The output of the second part is a knowledge management (KM) system. A prototype version of KM is developed in this case to test the underlying two-part approach proposed here. The proposed approach is however, quite general and is directed towards realizing most of the concepts of virtual enterprise, and thus achieving the goals of virtual enterprising.

Key Words: information sharing, data management, integrated product development, knowledge management, virtual enterprise, enterprise modeling.

1. Introduction

Global manufacturing has introduced to industries a number of new competitive challenges. These challenges include traits like simultaneously meeting customer requirements, reducing time-to-market, decreasing costs and increasing product quality while designing and developing manufacturable products. These challenges have forced manufacturing industries to rethink their mode of doing business and adept to new product design, engineering, manufacturing, and management strategies. Virtual enterprise is viewed as one of the most promising business strategies, which can address global competition [7]. Virtual enterprise is the integration of business activities and resources from different business units to satisfy quickly and efficiently customer needs. The changing customer needs emerge from a growing worldwide dynamic and competitive markets. The implementation of virtual enterprise requires firms to restructure their corporate organization, and reengineer business processes [18]. It also involves the use of management, reengineering process, system architectures, collaborative and information technologies. Among the applicable technologies, knowledge management (KM) is the most fundamental and challenging. This is because the integration of activities and resources in a virtual enterprise relies heavily on integrating and sharing knowledge not merely the data or information.

Many research efforts have been made on enterprise modeling and integration [1, 6, 9, 10, 26]. There is also a large number of useful works dealing with the development of information sharing systems and collaborative tools to support intra-organizational processes and teamwork [4, 5, 12–14, 16, 19, 23–25, 27]. However, only a few works addresses supporting virtual enterprise activities and processes in the true KM context. The development of knowledge management or information sharing systems for virtual enterprising is highly dependent on the characteristics of virtual enterprise processes and the involved technologies of knowledge

*Author to whom correspondence should be addressed.

management and information sharing. KM as such has been and still is a major technical challenge.

This paper presents a systematic approach to the development of a knowledge management (KM) system that is able to support the required intricacies of virtual enterprising. In this context, the definition of KM includes all aspects of information sharing, data management, and distribution channels. In order to ensure the applicability to many firms, this KM system development ought to be first guided by the results of virtual enterprise characterization and modeling. Consequently, the authors of this paper have laid-down a conceptual development of this KM system in two parts: a virtual enterprise characterization (a part one) and a KM system development (a part two).

The first part focuses on characterizing virtual enterprises to define (or clone) *virtual enterprising*. This part of development involves: (a) characterization and definitions and (b) virtual enterprise modeling. An enhanced IDEF0 technique (Integrated Definition for Function Modeling) [15] is proposed for modeling the processes of virtual enterprises.

The second part concentrates on the development of a KM system to support the practice of virtual enterprising. Most commonly, a KM system consists of a communication infrastructure and an information repository. Often, it also includes functions for inter-enterprise information management and inter-enterprise product data management, information distribution and control, and communication and information transmission. Establishing a full-fledged KM system involves the following five steps:

1. Identification of elements involved in sharing data, information, process and knowledge within a virtual enterprise
2. Identification of interactions and information/data shared in the interactions
3. Analysis of functional requirements
4. Design of a KM system framework and a software architecture
5. Modeling and implementation of a KM system

2. Overview of the Concept

This section presents an overview of the proposed concept to implementing a knowledge management (KM) system to support and clone virtual enterprising. The concept is based on a two-part developmental approach. Part one is a virtual enterprise characterization and a part two is actually a KM system development. Each part involves several steps of activities as shown in Figure 1.

Part 1: Virtual Enterprise Characterization

During this part, first the characteristics of virtual enterprise are defined and an analysis is then performed. Since virtual enterprise is often process oriented, it is modeled as a set of virtual processes, which, in turn, represent a logical temporal sequence of activities. Each activity is therefore seen as

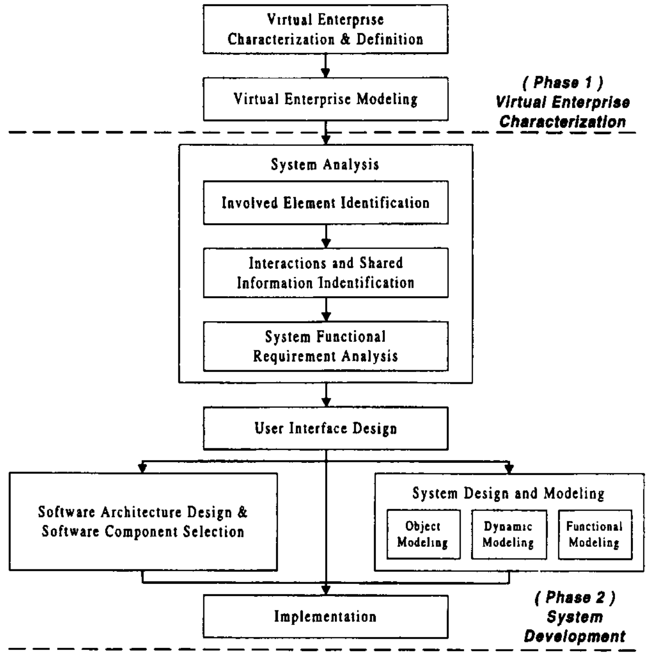

Figure 1. The steps of the proposed two-part KM concept.

the basic construct for virtual process modeling. To properly model the virtual processes, activities are analyzed and categorized based on the roles they play in the process definition, and an activity diagram is devised by enhancing a functional model using IDEF0. A virtual process model is established by connecting the inputs, outputs and coordination of the activities. Based on the virtual process model, a virtual enterprise information flow model is developed. This serves as the basis for system functional requirements' analysis by highlighting the information-related items in the virtual process model.

Part 2: KM System Development

During the second part, the elements involved in the information management and sharing of virtual enterprising and their interactions are analyzed to identify a KM system's functional requirements. Based on these functional requirements, a KM system framework and a user interface are designed. A software architecture is designed and its software components are then selected for KM system modeling and implementation. The object-oriented modeling methodology is employed to model the KM system. Later, the entire KM framework is implemented in an environment that simulates the heterogeneity of virtual enterprises.

3. Virtual Enterprise Characterization and Modeling

In this section, the characteristics of a virtual enterprise are identified. Based on the identified characteristics, a virtual enterprise model is developed for functional requirements' analysis. Later, the virtual enterprise model is used for a KM system development.

3.1 Characterization and Definition

According to Davidow and Malone [7], the term "virtual" means that *power of one thing comes from the other*. Therefore, the concept of virtual enterprise is that of an enterprise mostly made of functions provided by other supporting enterprises or a set of satellite allied-enterprises. In other words, a virtual enterprise can easily gain and integrate more resources through external integration and cooperation with other allied or supporting enterprises. This could provide quick responses to customer expectations within a rapidly changing business environment. Such concept is commonly realized by pooling a set of enabling capabilities for managing the following products' life-cycle variability:

1. Product's requirements' variability: This consists of capability of dealing with product requirement variability and to overcome changing market trends.
2. Production volume variability: This consists of capability for dealing with production volume variability with manufacturing chains to gain optimum production efficiency and throughput effectiveness.
3. Supply-chain utilization: This includes a capability of quick integration and optimal use of resources with supply chains.
4. Integrated value-chain process: This includes a capability for quickly integrating value-added activities from allied-enterprises into a value chain process. Through this value-chain process, it would be easier to reflect possible changes occurring in product design and manufacturing processes.
5. Organizational restructuring: This includes capability for quickly altering or restructuring organizational traits [18] to reflect a technological change or an orientation.

In summary, a virtual enterprise can be defined as follows: Virtual enterprise is an integrated product or service delivery strategy that emphasizes quick response to customer expectations and embodies the values of trust, cooperation, and resource sharing through strategic, tactical or operational alliances with other virtual enterprises.

In recent years, the process-oriented business concept has been widely accepted for business performance improvement [8, 11]. It is believed that business must be viewed not in terms of functions, departments, divisions, or products, but in terms of core processes [8]. Authors in this paper, therefore, have focused on the dynamic aspect of virtual enterprises and called it virtual enterprising with a definition as follows: Virtual enterprising is a systematic approach to integrated product or service delivery that emphasizes quick response to dynamically changing customer expectations and embodies the values of trust, cooperation and sharing through strategic, tactical or operational alliances with other virtual enterprises.

3.2 Virtual Enterprise Modeling

Enterprise modeling is to develop a set of interrelated models for describing various facets of an enterprise. These models let users perform activities with the help of appropriate tools and methods in order to address some desired modeling constructs [1, 9, 10, 26]. Similar to ordinary enterprises, a virtual enterprise can be defined in terms of many aspects such as organization, functionality, process, activity, information and data [6, 10, 21, 26].

3.2.1 ELEMENT IDENTIFICATION AND CLASSIFICATION OF VIRTUAL ENTERPRISE

As discussed earlier, part one of KM development focuses on virtual enterprise modeling. During part one, elements involved in virtual enterprising and their relationships are first identified and represented in terms of Rumbaugh's notations [22] as shown in Figure 2. The key elements in virtual enterprise include eight perspectives: organization, allied-enterprises, processes, resources, activities, products, information, and data. In Figure 2, bold blocks represent the elements, the connecting lines indicate the associations between elements, and 1-N icons indicate that more than one element involves in an association. Aggregation is drawn like association, except a small diamond icon indicates the assembly end of the relationships. The generalization is indicated by a triangular shape-icon connecting a super-class to its subclasses.

From an organizational perspective, a virtual enterprise can be seen as an aggregation of inter-related institutions, which themselves can be virtual enterprises or individual allied-enterprises. An allied-enterprise in turn can be a leading enterprise, a client enterprise, or a server enterprise. A leading enterprise is the enterprise that owns the end product and is in charge of the underlying virtual process. A client enterprise outsources or subcontracts one or more activities of a virtual process. A leading enterprise must be a client enterprise but a client enterprise may not be a leading enterprise. A client enterprise is also a server enterprise if it is not the leading enterprise. A server enterprise can be one of the following:

- A supplier that provides components or raw materials.
- A subcontractor that performs activities outsourced or subcontracted from a client enterprise.
- A consultant company that works with the client enterprise on a certain activity as a member of the virtual team.

A server enterprise can be a client enterprise if it outsources or subcontracts its activities.

From the process standpoint, a virtual enterprise can be also viewed as a large collection of business processes executed by a set of functional entities across enterprises to contribute towards their business objectives. A business process is a logical temporal sequence of the life cycle activities belonging to an end product. In other words, the business process produces the end product. An end product can be a physical product or a component, an electronic model or a file, a document or a drawing. A process is virtual, if one or more activities of the process are performed by different en-

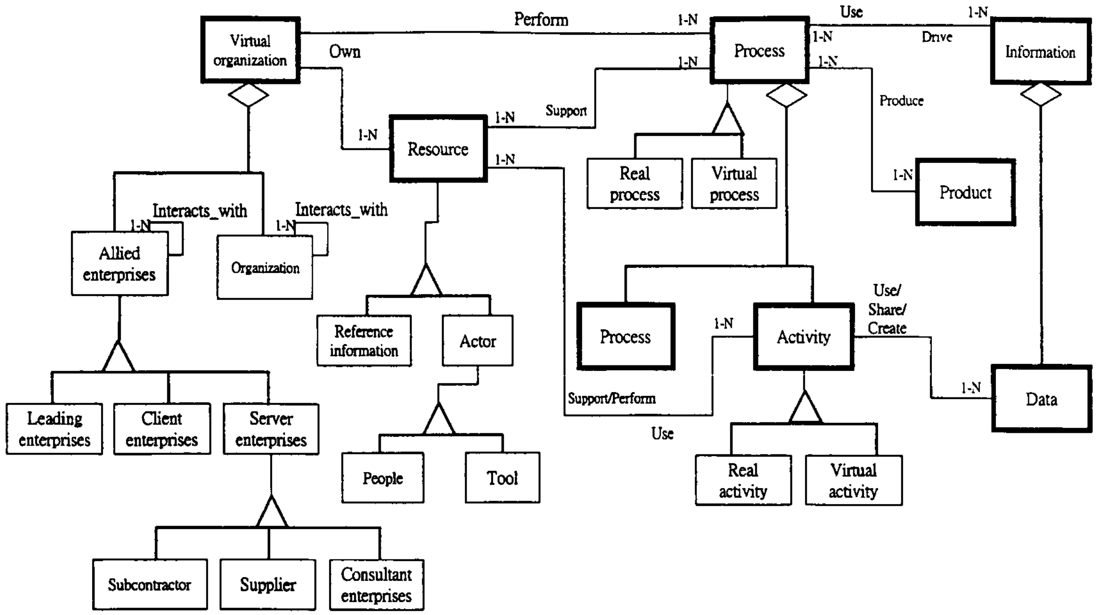

Figure 2. Elements in a virtual enterprise.

terprises and their specifications—(inputs, requirements and constraints) associated with one or more activities of the process—have come from a different enterprise unit. A virtual process itself can be part of another virtual process.

An activity is a series of elementary operations performed to realize a certain task within a process. An operation requires resources (requirements, and constraints) for its execution and transforms its inputs into outputs. An activity is real, if it is performed by a leading enterprise. An activity is virtual, if it is performed by the allied-enterprises or if it is outsourced or subcontracted. For the purpose of virtual enterprise modeling, virtual activities can be further classified into the following types:

1. Virtual activities completely controlled by a leading enterprise.
2. Virtual activities controlled by a leading enterprise but followed by other virtual activities that are outsourced and controlled by the server enterprise.
3. Virtual activities entirely controlled by a server enterprise. This may include activities that are outsourced by a server enterprise. In such cases, real activities can be distinguished by their actors. An actor can be an individual actor, a team from a leading enterprise or a team from an allied-enterprise.

3.2.2 STRUCTURE OF VIRTUAL ENTERPRISES

A virtual enterprise can be defined in terms of a hierarchical structure as shown in Figure 3. A virtual enterprise is the aggregation of allied-organizations, including a leading enterprise, which form the first layer of the structure. Each of the organizations can be an allied virtual-enterprise—that is responsible to perform a decomposable higher order functions—or an individual server enterprise that is responsible for performing a lower order function. The allied virtual-enterprises are in turn the aggregations of allied-organizations, which form the structure of this allied virtual-enterprise. This hierarchical structure of virtual enterprise also reveals the hierarchical, distributed, cooperative, and dynamic natures of virtual enterprising.

3.2.3 VIRTUAL ENTERPRISE PROCESS MODEL

To properly define virtual enterprise processes (virtual processes for short), an enhanced IDEF0 activity diagram has been used as the construct for process modeling. The graphical representation of activities is shown in Figure 4, which makes use of a box with six legs, I ^{2}CORS. The solid-lined box in the center stands for a real activity, while the dash-lined box stands for a virtual activity. The first "I" stands for an input to an activity, which is provided by a supplier. The second "I" at the upper side indicates interactions between activities. The "C" represents an enterprise control element to control an activity. The "O" is an output from an activity to a consumer. The "R" denotes the resources from information repository used to perform an activity and, the "S" indicates a server enterprise that performs an activity.

Inputs to an activity can be materials or information that need to be transformed by that activity. Outputs of an activity are the results of this transformation. The output can be a physical product or component, an electronic file, or documents. Control to an activity indicates "by whom" the activity is controlled. The interactions between activities can be classified based on the actors and application systems (e.g., the interactions among development team members, or the interactions between shop floor control system and CNC machines). An activity may interact with none or any number of other activities. Similarly, an activity may have none or any number of server enterprises. An activity is real if the number

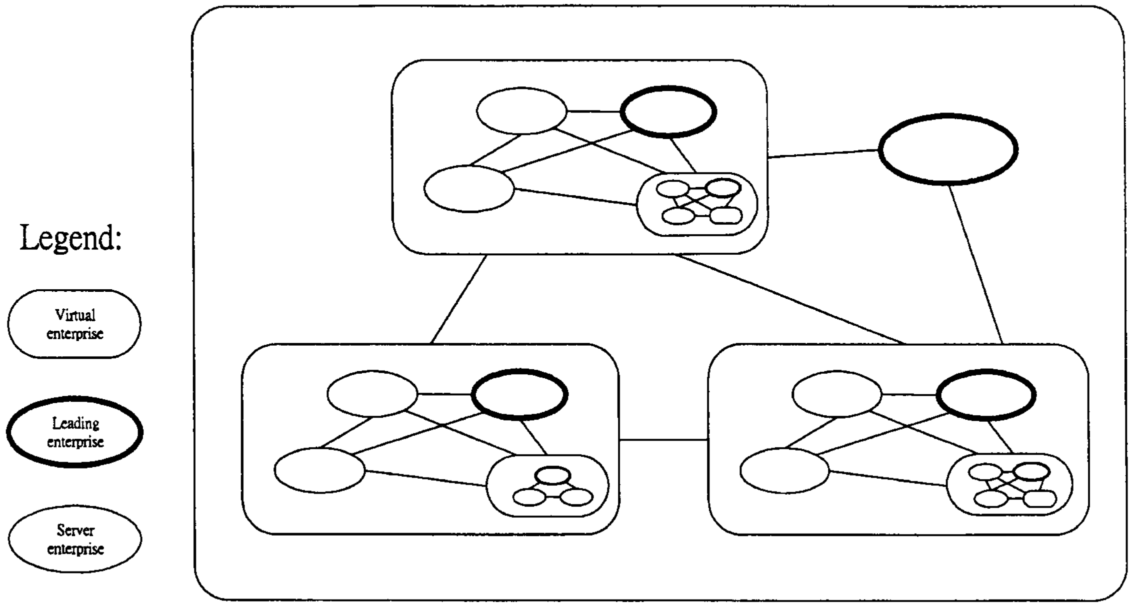

Figure 3. Hierarchical structure of virtual enterprises.

of its server enterprises is zero; otherwise, the activity is virtual. The resources to support an activity can be tools, teams or reference information.

Relationships between Activities

A virtual process model is established by connecting the inputs, outputs and interactions of the activities. The relationships between activities are classified as sequential, concurrent or coupled [18].

Dependent Tasks: The sequential relationship between activities defines a strict ordering relation between two activities, where one activity should be fully executed before the other can be executed.

Independent Tasks: Activities with concurrent relationships are both the successors of a certain activity and have no interactions between each other. They can, therefore, be performed concurrently.

Coupled Tasks: Activities with coupling relationships should be performed cooperatively or collaboratively through information and knowledge sharing as a result of their strong interactions.

3.2.4 VIRTUAL ENTERPRISE INFORMATION FLOW MODEL

Figure 5 shows an example of the virtual process model. By highlighting the information-related items from the virtual process model, a virtual enterprise information flow model is developed. The information flow model facilitates the identification of following two items:

1. Elements involved in the information sharing of virtual enterprising as well as their interactions and relationships.
2. Types of information as well as their characteristics and behaviors in virtual enterprising, so as to help the functional requirement analysis for a virtual enterprise KM system.

As shown in Figure 6, similar to the virtual process model, the virtual enterprise information flow model consists of levels of information flows depending on the levels of virtual enterprise structure. The basic construct of the information flow model is an activity information diagram that indicates the input information, output information, interaction information, reference information, teams, acting enterprise, and tools used to create and consume the information of an activity. An activity information diagram is derived by highlighting the above items in the activity diagram of a virtual process model. The information flow is built by connecting the inputs, outputs, and interaction information of the activity information diagrams as did in the virtual process modeling. An activity that is performed by a server virtual enterprise can be further decomposed into lower order activities that form the information flow of its next level.

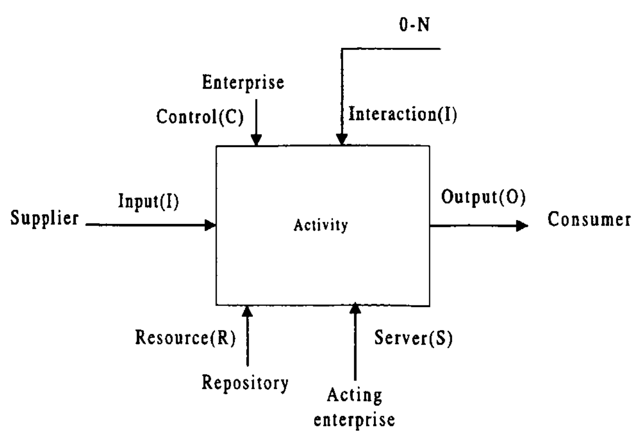

Figure 4. Activity diagram.

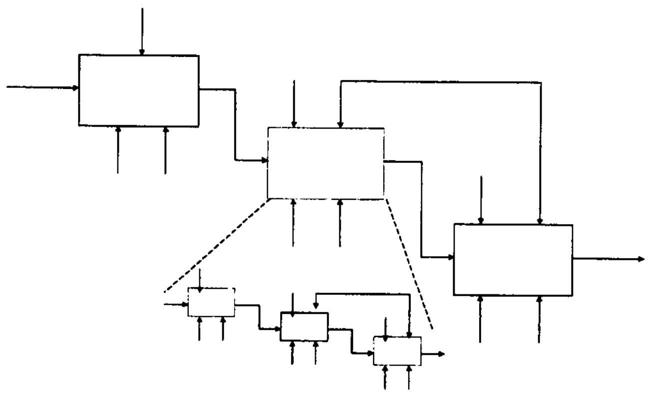

Figure 5. An example of virtual process model.

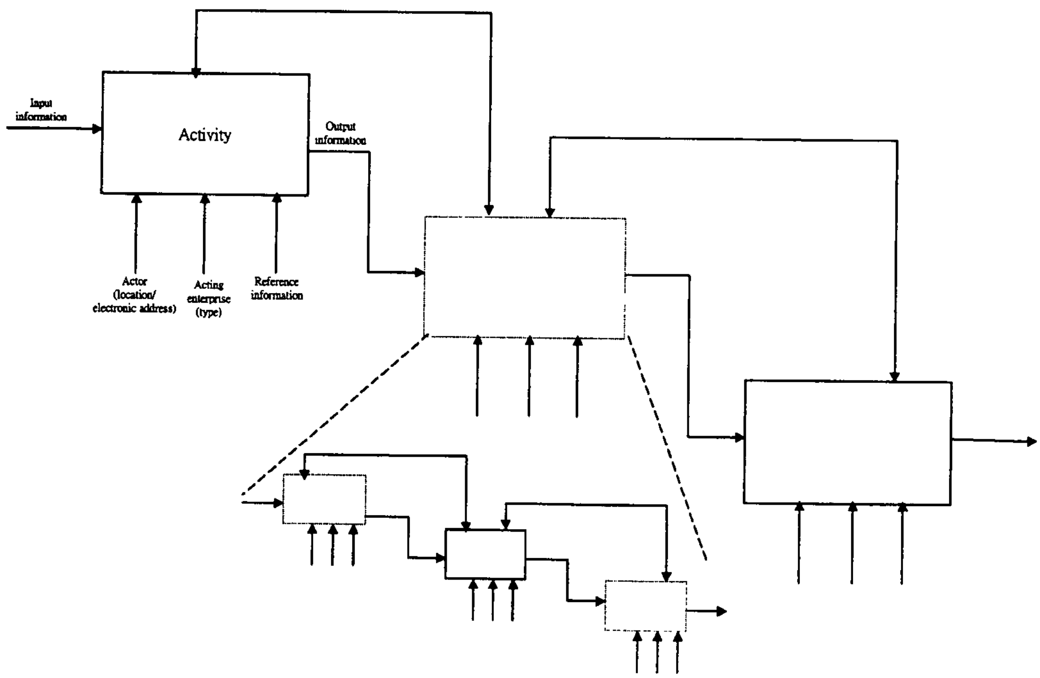

Figure 6. Information flow model of a virtual enterprise.

4. Functional Requirement Analysis

The success of virtual enterprising depends on the successful integration and management of processes, activities, and resources from a satellite set of allied-enterprises. To support virtual enterprising, the enabling functions or core technologies, such as communication of information and knowledge sharing, process control and coordination, and supply chain management are required. Such enabling functions or core technologies represent the functional requirements from the aspects of information, process, data and resources of a virtual enterprise. Communication of information and sharing of knowledge are the cores because the integration and management of resources, activities, and processes heavily depends on the integration and sharing of information, data and processes.

This section focuses on the functional requirement analysis of a KM system that is able to support virtual enterprising based on the results of virtual enterprise characterization and modeling. According to the spirit of virtual enterprising, authors first specify the goal of a KM system as "to provide the right knowledge (information, data or a process) to the right place at right time in the right format throughout the entire virtual process."

To achieve this goal, several 5W1H issues [18] have to be addressed. They are

1. What to share (types of knowledge—an information, a data or a process).
2. Who to share the knowledge with (actors of activities, i.e., team and application tools).
3. Why to share knowledge (design rationales or rules for sharing knowledge).
4. When to share knowledge (timing aspects — when an actor will execute a task).
5. Where to get or send the information (task levels and classification of actors).
6. How to share knowledge (possible interactions among tasks and actors).

To answer these 5W1H issues and consequently identify a KM system's functional requirements, the key players such as information (tasks, processes and actors) in virtual enterprise should be further classified. The interactions among information elements (tasks, processes, and actors) can then be identified for the analysis of a KM system's functionality.

4.1 Identification and Classification of KM Elements

From the information flow model, authors know that information and data sharing mostly occurs among actors when a process or a task is being performed. An actor can be a person, a computer program, or an application tool that performs an activity. People, information, application tools, and activities therefore play major roles in knowledge management.

People can be further classified into allied team-members, other company employees, customers, product development teams (PDTs), consultants, and guests (Figure 7). The allied team-members are those who perform the outsourced or subcontracted activities. The other company employees indicate teams from other related departments such as the marketing department, accounting department or the higher-level executive/management team. Company employees are members from the leading enterprise who work on the process. The company team members include process or project leaders; product and process developers; and staff to support the process activities. Customers are those who have placed or are willing to place an order for the end product. A consultant can be anyone, who is interested in sharing information related to the company or its products. A guest can be anyone who is interested in the general information of the company or its products.

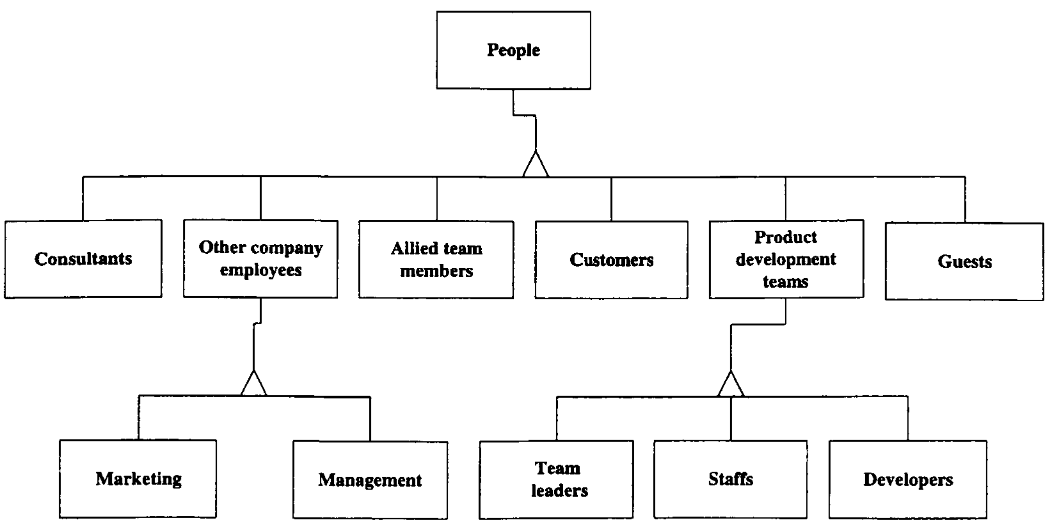

Figure 7. Classification of teams involved in virtual enterprise information sharing.

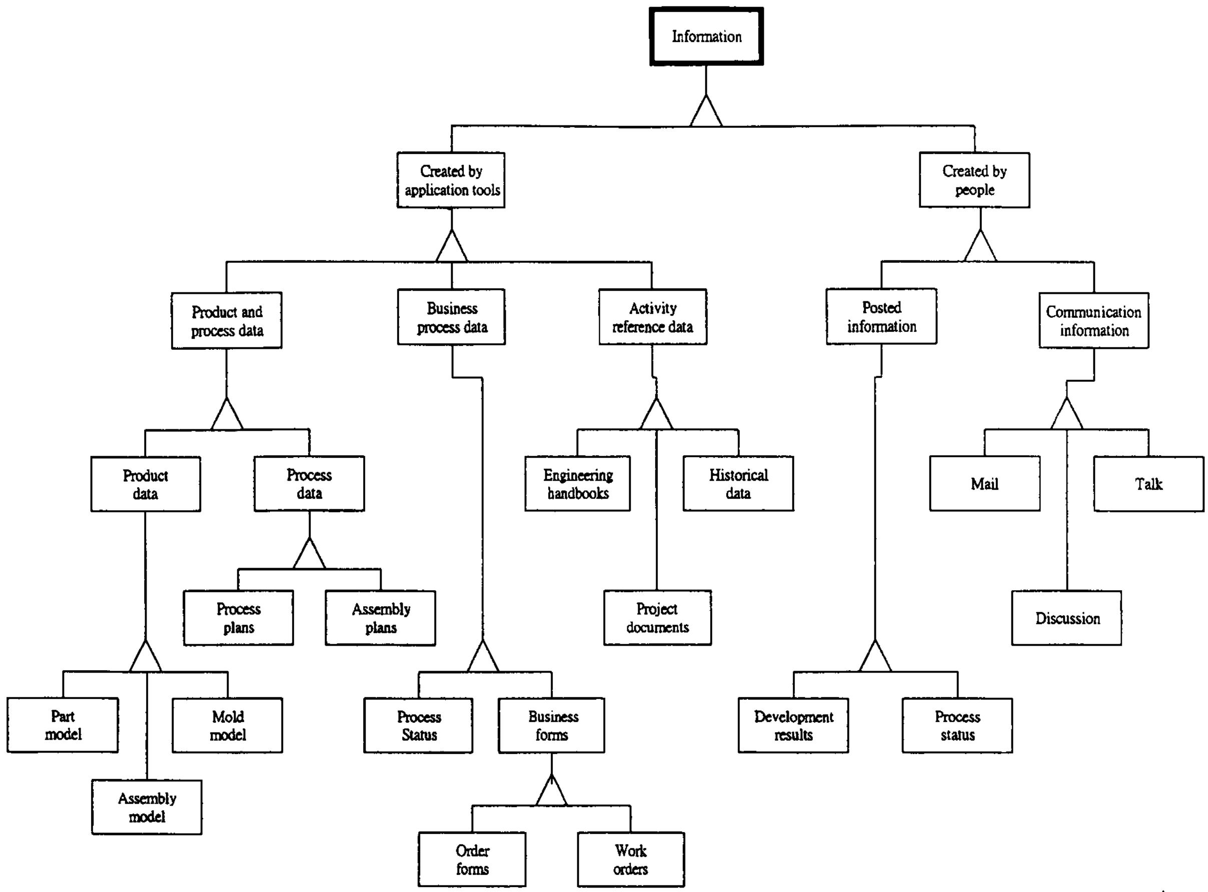

Figure 8. Classification of information involved in virtual enterprise information sharing.

On the basis of information sources (or "where to get information"), information can be grouped into two types: those that came from teams and those that were produced from application tools. The former are mostly interaction information, while the latter are basically derived information (input, output and reference information) as indicated in the activity information diagram. Both of them can be further classified into product and process data, business process data, activity reference data, posted information, and on-line communication information (Figure 8).

Product and process data are basically the outputs of the process activities, such as product design models, mold models, process plans, assembly plans, and engineering drawings. Business process data are those that flow along the product development process and drive the engineering process. Typical examples of business process data are business forms such as purchasing order forms, service work-order forms and request for proposals. The process data may also include information related to transaction forms, request forms and process status reporting. Activity reference data are those that are referenced for performing an activity. Examples of reference data are material data, machineability data, historical design data, project documents, policy documents, and operational standards. Posted information is basically public information that is posted in public domains and available to the public, such as new product lines and general company information, process status, and work results. On-line communication information is the knowledge shared during interactions among activities or actors. They are transmitted in the forms of mails, talks and discussions.

Classification of activities is done during a virtual enterprise modeling stage. Application tools are mostly activity dependent, therefore, there is no need of their further classifications.

4.2 Interaction Analysis

According to the way an actor interacts with others, the interactions can be classified into three levels, as follows:

1. Interactions between classes of actors (Figure 9).
2. Interaction between application tools. This occurs during the transmission of product and process data between application systems being used for product and process development.
3. Interactions between teams and application tools. The typical examples of this case include:
   (a) Engineers querying reference information from data-bases via database management systems.
   (b) Product developers getting product models from a

common repository through a product data management (PDM) system.

(c) A project manager checking the process status from a process control or a project management system.

If considered from the levels of information shared in the interactions, the interactions themselves can be classified into four levels. They are (1) on-line communication with or between people outside the development team, (2) on-line communication with or between members throughout the product life cycle, (3) off-line broadcasting, and (4) interactions throughout the product life cycle.

4.3 Scenario Analysis

Interaction analysis provides details of interactions that occur in a typical virtual enterprising session. The functional details of the interactions can then be identified through a scenario analysis. A scenario is a sequence of events that occurs during an interaction. An event is something that happens at a point in time, such as a customer sends a request for proposal or a project leader distributes work orders. The result of scenario analysis consists of a set of analysis documents (or forms) that include interacting agent name, the interaction scenario, a description, the name of the actors and types of information involved (Figure 9). The ultimate goal of scenario analysis is to identify the functions required to support an interaction. The scenario analysis can also be used to help design user interface. Figure 10 illustrates an example of scenario analysis of a "Check_work_status."

4.4 Functional Requirement Identification

A KM system that is capable of supporting concurrent engineering should provide facilities to deal with storage, management and manipulation of all sorts of knowledge and its

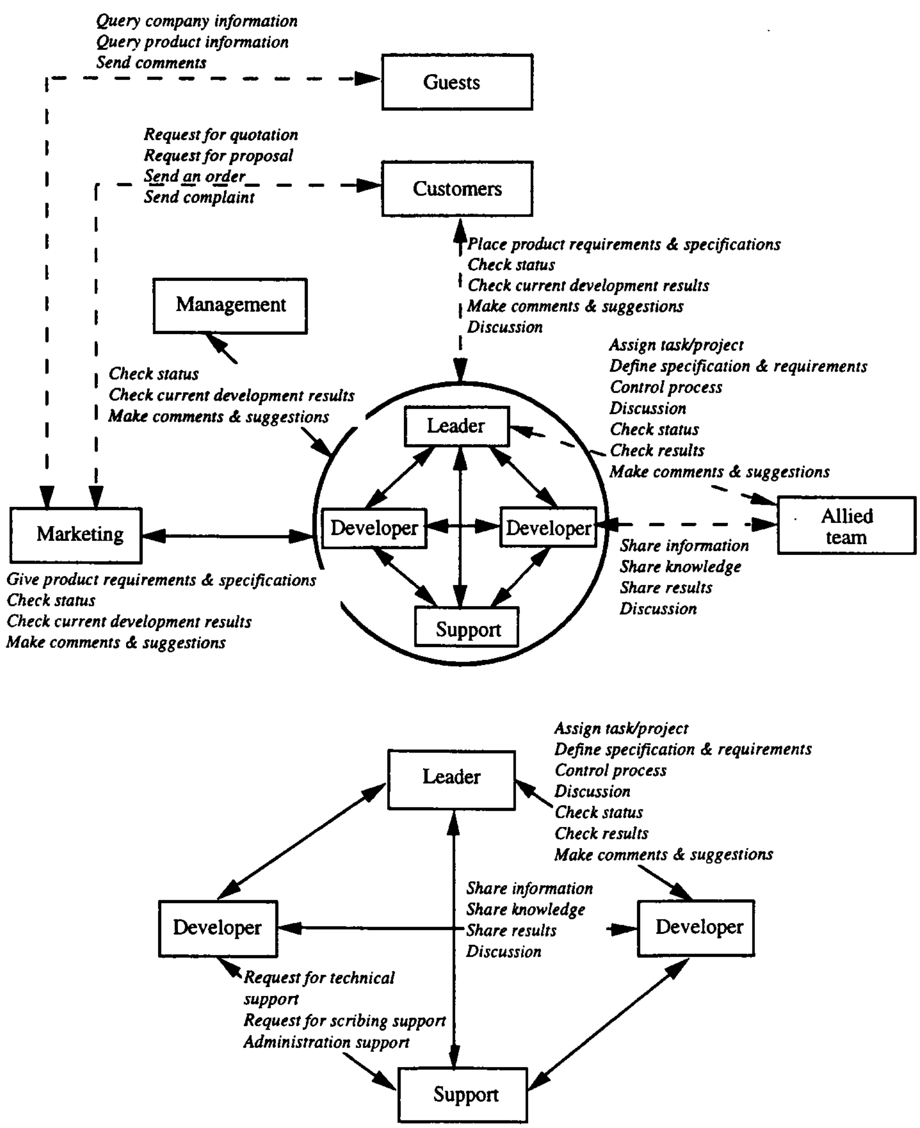

Figure 9. Interactions among people in a virtual process.

Interaction Analysis		
Interaction Name: *Check_work_status*		
Description: *Actor_1 Checks the current work status of Actor_3*		
Scenario: Actor_1 uses W3 browser to log into the Information Sharing System via the Internet. Actor_1 enters name, ID and password. System checks name, ID and password ,and displays user interface if they are valid. Actor_1 selects the function of "Work Status Query." System displays the list of "Work Status." Actor_1 selects one item from the list. System displays the work status of Actor_3		
Functional Analysis		
Actor_1: Team leader Team members Allied team members Management team	**Actor_2:** Information sharing system	**Actor_3:** Team members Allied team members
Information: User name, ID, password	**Information:** User interface the lists of "Work Status" the work status of Actor_3	**Information:**
Event: 1. *Log-In*System 2. *Enter*User name, ID, password) 5. *Select*the function of "Work Status Query") 7. *Select*the one of lists	**Event:** 3. *Check-validity*User name, ID, password 4. *Display*User interface 6. *Display*the lists of "Work Status" 8. *Display*the work status of Actor_3	**Event:**
Function: Log-In Check-In Display Selection Check-Out Enter		

Figure 10. Interaction scenario form.

maturity levels. These knowledge maturity levels within a KM system ought to be shared throughout an entire development cycle both to support on-line communications and off-line broadcasting. In such a KM environment, information can be distributed through various knowledge-bases and information repositories such as a product library, bill of processes, technical memory, bill of materials, and databases. Other distribution channels include mail and message libraries; and file and document libraries in the form of texts, graphics, images, formulas, and CAD multimedia data.

1) *Storage and management:* According to the characteristics of shared information, two types of storage are required to support knowledge sharing. One is a distributed database and the other is an inter-enterprise common repositories. The former is used for storing business data such as statistical tables, spreadsheets, and forms, while the latter is for product and process information such as documents, files and product models.

The management of shared information is therefore the management of databases and the management of common repositories. The former can be done by an ordinary database management system, while the latter requires functions for common repository access and data integrity maintenance.

A set of common repositories contains the product and process data that are shared throughout the allied concurrent engineering process. Besides providing accessibility control to different teams, the capabilities to maintain security, consistency and associativity among product and process data are also required. In practice, repository management should be supported by a set of underlying functions such as release management, change management and notification, and product structure management [3].

2) *Information distribution and control:* Information distribution and control deals with (a) distributing product and process data to and from workflows and controlling them throughout a product and process development process, (b) providing right information from data storage to inquiring activities or application systems, and (c) coordinating information sharing in between application systems.

3) *Communication facility:* This facility includes (a) functions for on-line two-way communication, (b) functions for coordinating group discussions among team members, (c) functions for accessing different levels of information, and (d) functions for sending information to a specific person or group of people, system, or destination.

4) *Data exchange and information abstraction:* Allied concurrent engineering is conducted in a highly heterogeneous environment. In CE, an effective means for sharing and exchanging product data among applications and organizations are required.

The objective of data abstraction is to provide degree of granularity (levels of information in accordance with levels of privileges). For example, the core PDT members can share the product and process data in the most detailed manner. The allied team-members can access only a portion related to their work. The guests, however, can only see pictures of the products at most. Similarly, the project leader can access the entire project plan and schedule. People outside of the product development team (e.g., consultants) are allowed only to query the project or work-process status.

5. System Design and Modeling

This section presents the design and modeling of a KM system that accommodates the functions identified in the previous section.

5.1 System Framework

Based on the functional requirements identified in the previous section, the system framework of a KM system is proposed (Figure 11). The main functionality of the KM system includes:

1. *Communication infrastructure:* The intranet for intra-enterprise communication and the Internet for inter-enterprise communication.
2. *Common information repository and knowledge-bases:* The storage for information shared in virtual enterprises. It includes distributed databases, allied team-libraries and knowledge-bases. The distributed database system consists of internal databases and allied-databases, is used for storing information that is suitable for being represented in forms of tables or objects. The allied team library con-

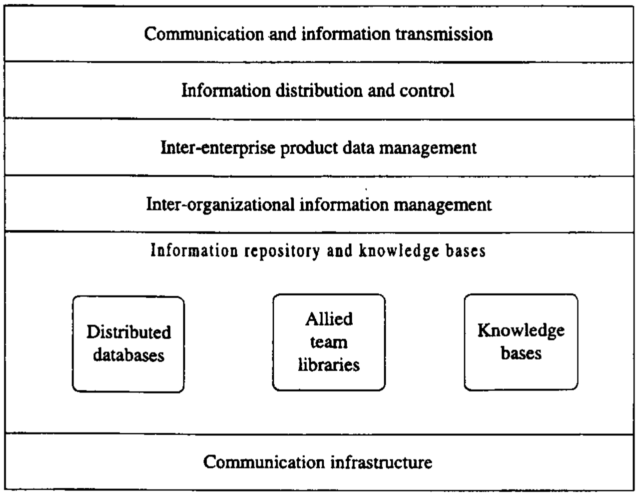

Figure 11. Framework of a KM system.

sists of an enterprise team-library and allied-enterprises' product and process data-libraries. The team-library is a group storage area managed by mechanisms of library management for the purpose of sharing product and process data among members of the product development team (PDT). Knowledge-bases are available for supporting product and process development activities such as product design, mold design, mold manufacturing process planning and molding process design.

3. *Inter-organizational information management:* The inter-organizational information management provides the following three types of functions: (a) database management, including storing and retrieving data to and from virtual enterprise distributed databases, (b) the transmission of data retrieved from the distributed databases to inquiring actors, and (c) coordinating the information sharing between application systems as an information broker.
4. *Inter-enterprise product data management:* It provides mechanisms for allied team library access and maintenance, which are supported by a set of underlying functions for design release management, change management and notification, and product structure management. Design release management provides functions such as reference, check-in, check-out, and copy functions for team members to access a team library. In addition, functions for data exchange are employed to ensure product data interchangeability in a heterogeneous environment.
5. *Information distribution and control:* Two classes of functions are available for information distribution and control. The first class provides functions for posting information to a public domain or to transmit information to individuals, a group of people or to a specific destination. The second class provides functions for information distribution and control among development activities at a process level.
6. *Communication and information transmission:* The functions for communication and information transmission include: information browsing and querying, facilities for on-line communication through mail, talk and discussion, and mechanisms for information transmission and posting.

5.2 Software Architecture

To support the KM system functionality, the main system components and their relationships are defined in Figure 12. Figure 12 constitutes the software architecture of the information sharing and KM system.

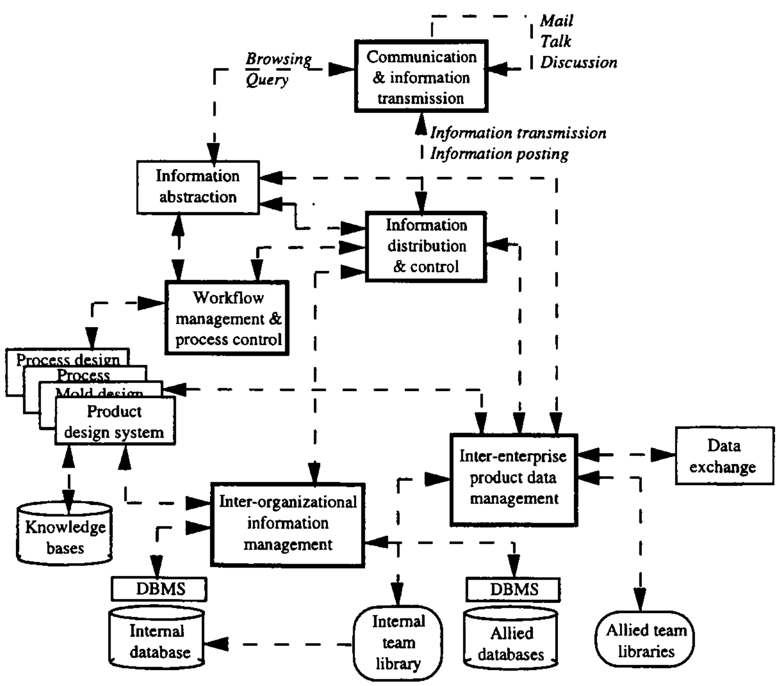

Figure 12. Software architecture of the KM system.

The internal-databases and allied-databases (that form a federated database system) are managed by an inter-organizational information management module. Similarly, team-libraries and allied team-libraries (that form the virtual enterprise repository) are managed by an inter-enterprise product data management module. In cooperation with a data exchange system, the product data management supports interchange of product models between resident and other CAD/CAM systems.

Both of the information management and product data management modules can be used directly by users when they are performing the product development activities. They can also be used as support functions such as information distribution and control and for providing timely information to the teams.

Since workflow is closely related to the information flow, the workflow management and control module generally works together with the functions of information distribution and control [20]. Information abstraction is a convenient method to provide process status, information location, and product development results to the work-groups and concurrent teams. Concurrent teams can get results from workflow management modules, information distribution modules or product data management modules. These modules can be used for browsing and querying as well by filtering out the details.

5.3 System Modeling

Object-oriented techniques provide a new way of thinking about problems when models are organized around real-world concepts. The fundamental construct is the object, in which data and associated operations that are normally performed on that data are encapsulated in a single entity. Therefore, instead of passing data to procedures and having these procedures operate on the data, the objects can be invoked to perform operations upon themselves.

Based on Rumbaugh's Object Modeling Technique (OMT) [22], the proposed framework is modeled in terms of three related models: object model, dynamic model, and functional model. In this paper, authors cite modeling of functions for communication and information transmission as an example. For the rest of the functions readers may refer to References [2, 3].

5.3.1 OBJECT MODELING

The result of object modeling is an object diagram that defines the static, structural, and data aspects of the framework in terms of objects and relationships, which correspond to elements in the domain of information sharing.

In Rumbaugh's notation, an object-class is indicated by a rectangular box with three regions: class names, list of attributes, and list of operations. An object is denoted by a rectangular box with rounded corners. An association between classes is drawn as a line, with which a verb in a problem statement is associated. Similarly, the OMT notation for a link is a line between objects. Aggregation is drawn like association, except a small diamond indicates an assembly end of the relationships. The generalization is indicated by a triangle connecting a super-class to its sub-classes.

Figure 13 illustrates the object model of the communication and information transmission. The information sharing system is the aggregation of classes of users, key server, mail server, other servers, information, databases, and repository etc. The user-class is specialized as classes of guests and product development team (PDT) members. The server-class is the aggregation of classes of information distribution and control, communication and information transmission, workflow management and process control, product data management, information management, and knowledge management.

The communication and information transmission class itself is the aggregation of a Web server-class and a set of browsers. The Web server-class is again the aggregation of a communication-class, an information distribution-class and a browsing-class. Each of the classes is composed of a set of lower level functional class as shown in this figure. And, each of the functional classes is interfaced to the users in terms of form classes such as guest, customer, employee, corresponding to the levels of users. The sub-classes of information are the same as those shown in Figure 7.

5.3.2 DYNAMIC MODELING

In the previous section, authors examined the static structure of the proposed framework by identifying the structure of the objects in it and their relationships. In this section, the dynamic modeling techniques of OMT are employed to describe changes to the objects and their relationships over time. The main construct of the dynamic modeling is the control information, including the sequence of events, states, and operations that occur within a system of objects.

1. *Events and states:* The object model presented in the previous section described the possible patterns of objects, attributes, and links that can exist in the proposed framework. The attribute values and links of an object at a single moment in time are called its "state." Over time, the user may perform operations or the objects may stimulate each other, which results in a series of changes to their states. An individual stimulus from outside of the system (i.e., the user performs operations) or from one object to another is called an "event." The response to an event depends on the state of the object receiving it. Response can include a change of state or the sending of another event to the original sender or to a third object. Following are some of the events from the user to the information sharing system:

   Enter the Web server
   Key in password
   Send a mail
   Talk to a person
   Discuss with the development team

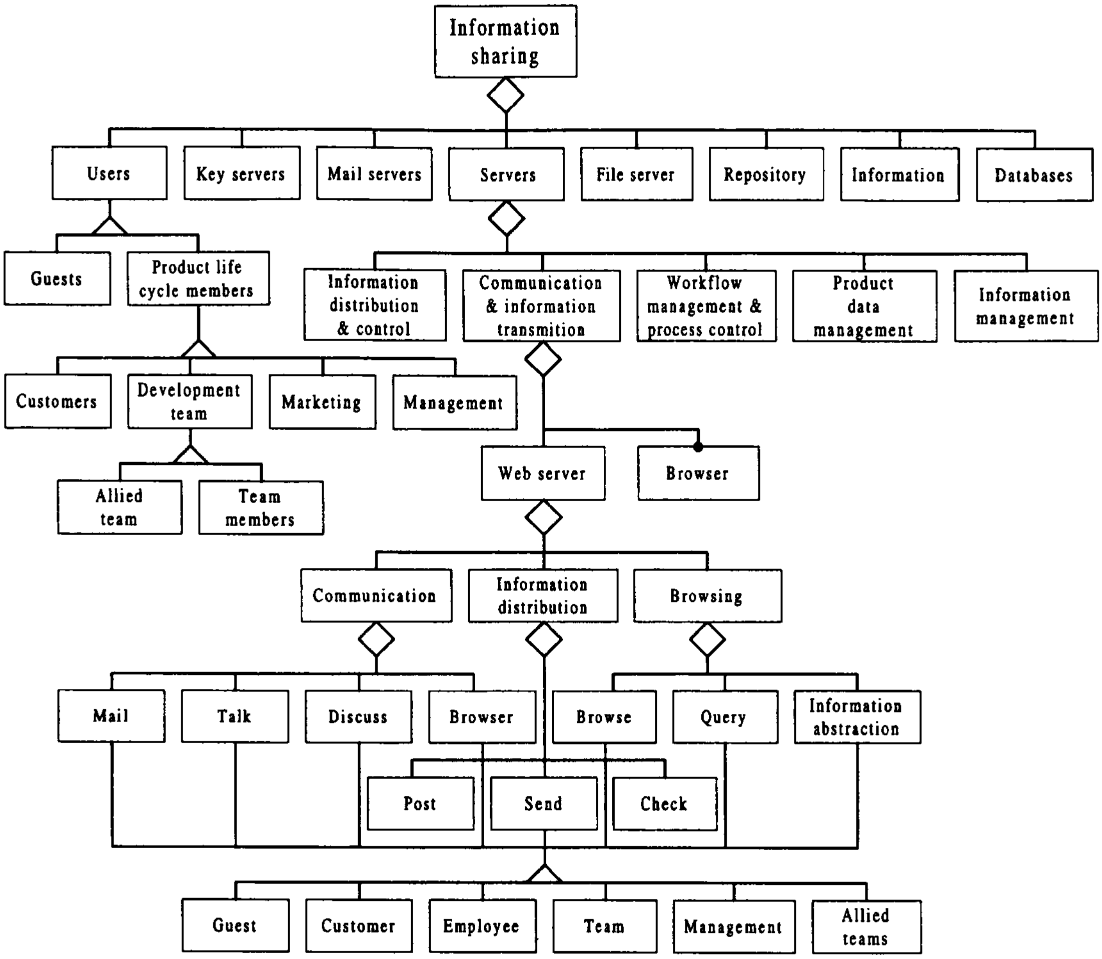

Figure 13. Object model of a KM system.

Distribute work orders

Receive a Request-For-Proposal (RFP)

2. *State diagrams:* A state diagram describes the behavior of a single class of objects. A state diagram is a graph whose nodes are states and whose directed arcs are transitions labeled by event names. A state is drawn as a rounded box containing an optional name. A transition is drawn as an arrow from the receiving-state to the target-state; the label on the arrow is the name of the event causing the transition. All the transitions leaving a state must correspond to different events.

Figure 14 shows the state diagram of the work order class as an example. An empty work order object will be instantiated once an "open work order" method in the information distribution object is activated. The empty work order object may transit to the state of (1) work order with destination but without message or (2) work order with message but without destination by specifying destination or work order message. Both of the two states can transit to state of work order with message and with destination.

After creation of a work order, the work order will be stored as "unread" in a database, which can be retrieved by the original creator or read by the actor of the specified destination. The work order is "read" if it is read or retrieved from the database. The work order can be deleted at the creation stage or from the database, that terminates the state of a work order object.

5.3.3 FUNCTIONAL MODELING

Functional models are employed to specify the meaning of operations and constraints in the object model and actions in the dynamic model. They consist of multiple data flow diagrams (DFDs) that specify the meaning of operations, constraints and actions. A DFD shows the functional relationships of the values computed by a system, including input values, output values, and internal data structures. A DFD contains processes that transfer data, data flows that move data, actor objects that produce and consume data, and data store objects that store data passively.

Figure 15 illustrates an example of a functional model of a "log-in" method in the Guest class. The method includes operations for username validity check, password validity check, user status updating, and user interface prompting. Once a user enters his or her username, the method will check the username with a pre-defined user table. If valid, the user is then required

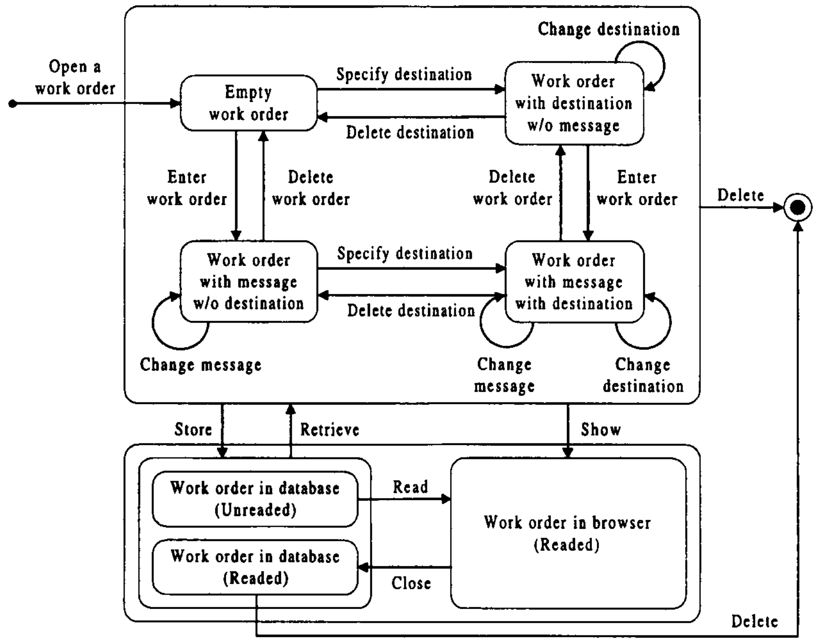

Figure 14. The state diagram of work order class.

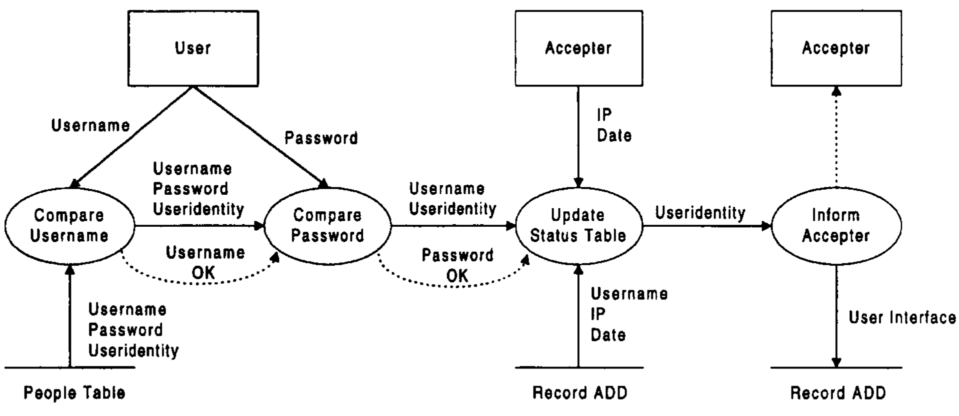

Figure 15. The functional model of "Log-In" method.

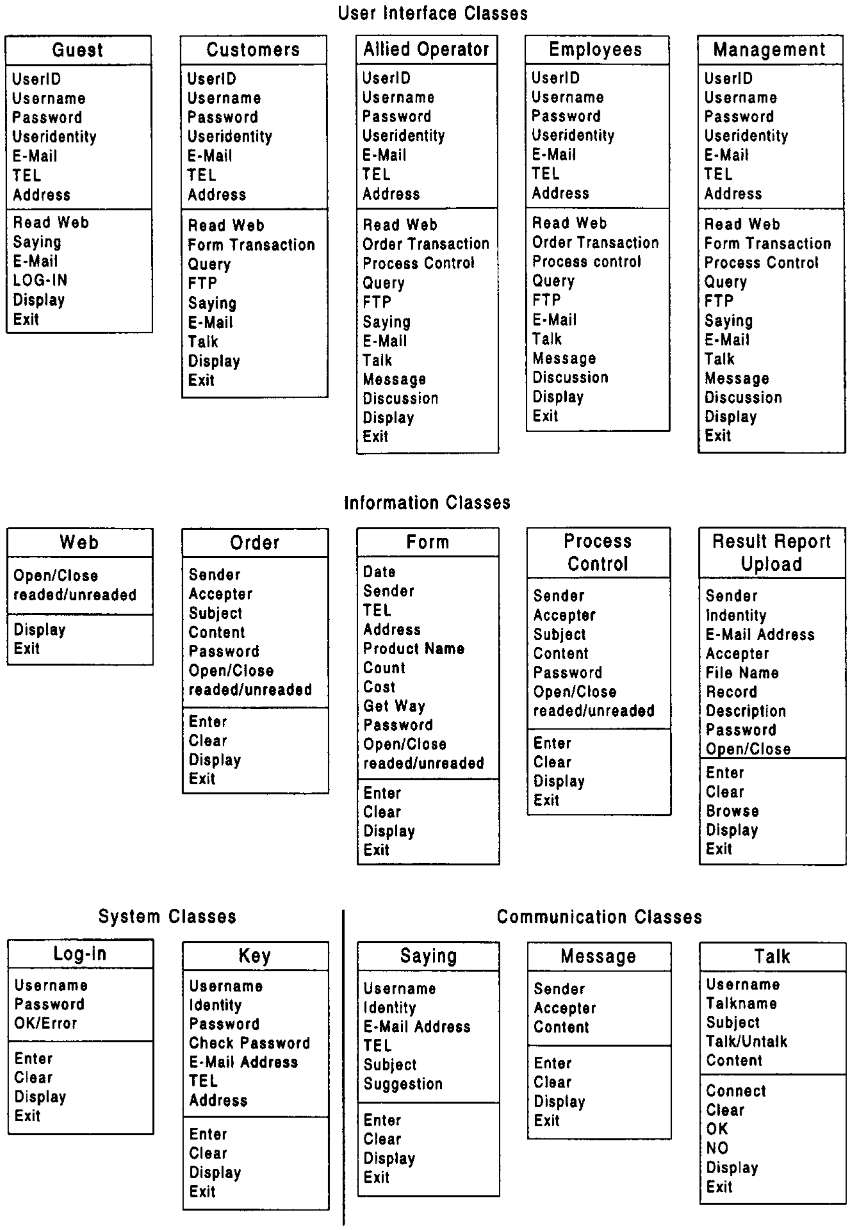

Figure 16. Part of the classes of a KM system.

to enter his or her password. If valid again, the status of the user will be updated and stored in a status table. A user interface that corresponds to the privilege level of the user will be prompted to the user for further operations.

5.3.4 OBJECT MODEL DESIGN

According to the object-oriented modeling, the classes of the KM system can be defined in terms of class names, class attributes and class methods. Figure 16 shows part of the classes.

6. System Implementation

Based on the results of system design and modeling, a prototype information sharing system has been developed for a trial implementation. This section presents the system configuration, software components used and the results of this trail implementation.

6.1 System Configuration

Based on the proposed system design and modeling, a prototype information sharing and KM environment has been implemented at two universities in Taiwan, ROC. This constituted a virtual concurrent engineering environment between the Computer-Aided Concurrent Engineering Research Lab at National Cheng Kung University and the Virtual Enterprise Resource Planning Research Lab at National Kaohsiung Institute of Technology, Taiwan, ROC. The Computer-Aided Concurrent Engineering Research Lab at National Cheng Kung University is equipped with Acer ALTOS 9000 PC™ servers and

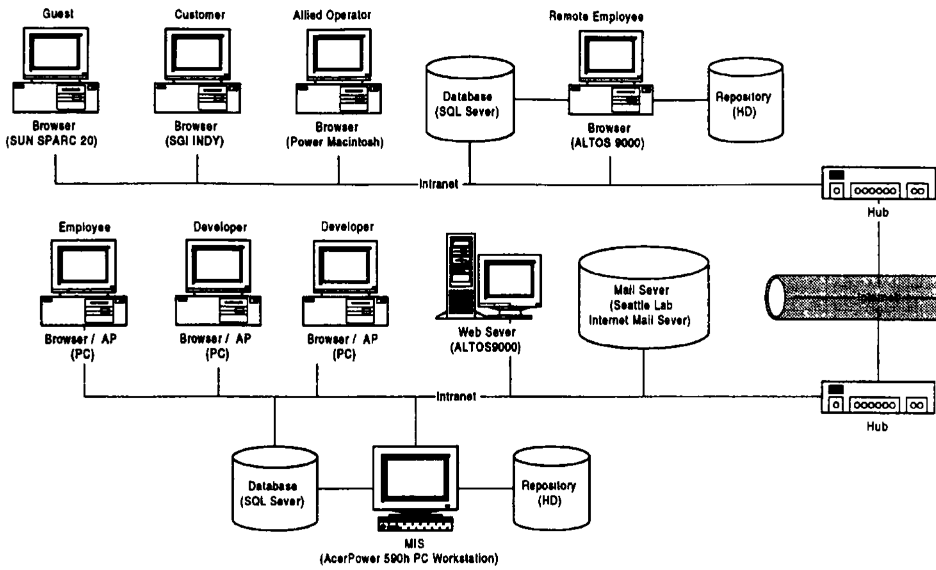

Figure 17. System configuration.

an Acer Power 590h PC™ workstation networked with six PC clients under Windows-NT™ environment. This plays the role of a leading enterprise. The Virtual Enterprise Resource Planning Research Lab at National Kaohsiung Institute of Technology is equipped with a SUN Sparc 20™ workstation, a Silicon Graphics INDY™ workstation, a Power Macintosh 6100/60™, and an Acer ALTOS 9000 PC. This plays the roles of guests, customers, remote employee, and allied-teams (Figure 17).

The software components employed in the implementation of the information sharing system include the Visual Basic™ program development tool, HTML programming language, CGI technology, Access database, Web server, and Web browser. The Web server and other information sharing functions run on an Acer ALTOS 9000 server and a Micro Soft SQL™ Server 6.5 and Micro Access database 7.0 run on the Acer Power 590h PC workstation as the data server and file server (Figure 18).

6.2 Implementation Results

Figures 19–22 show part of the user interfaces of a KM system. To enter the KM system, the user is required to enter

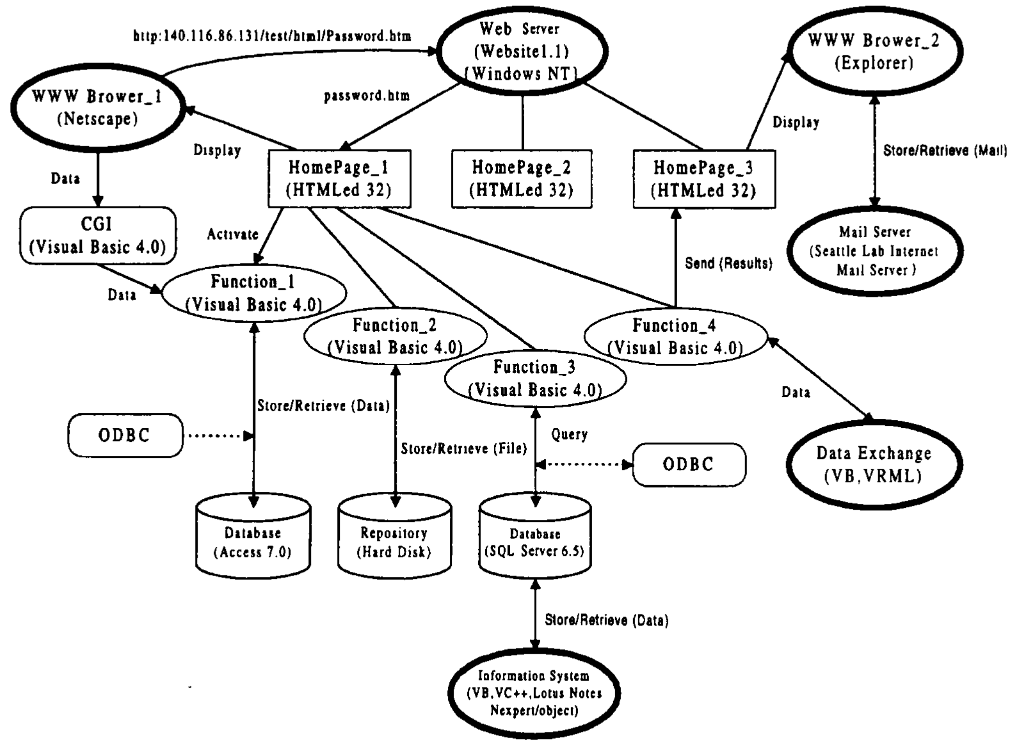

Figure 18. The software component architecture.

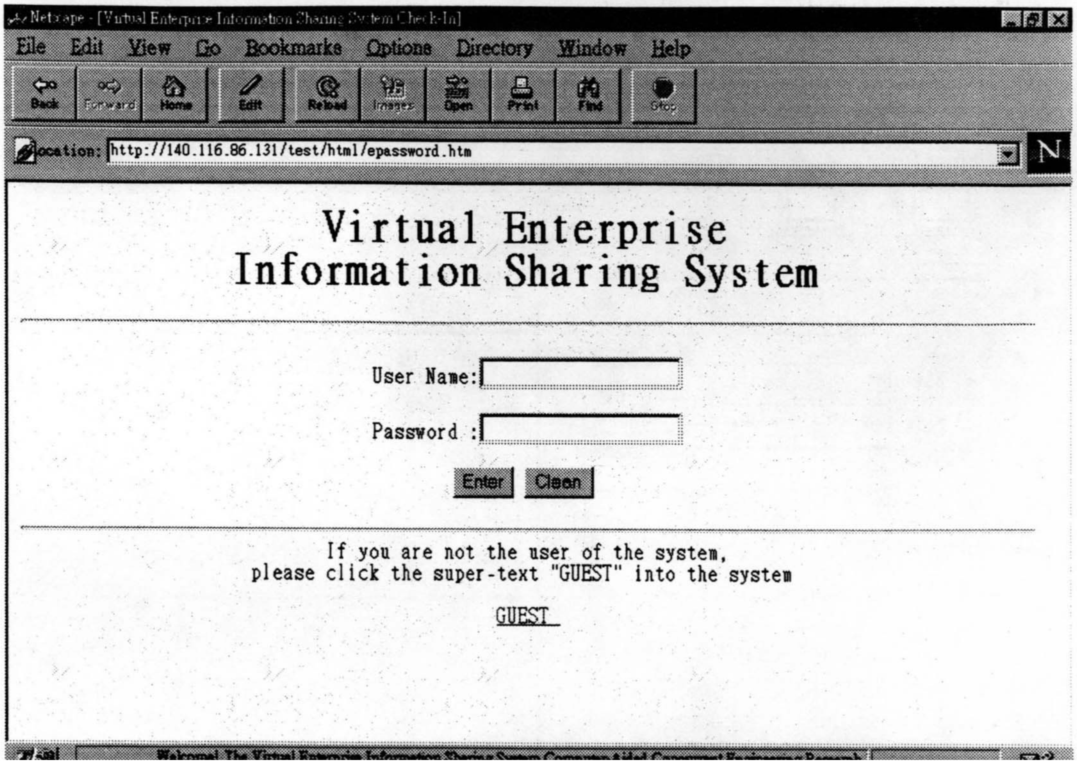

Figure 19. System log-in.

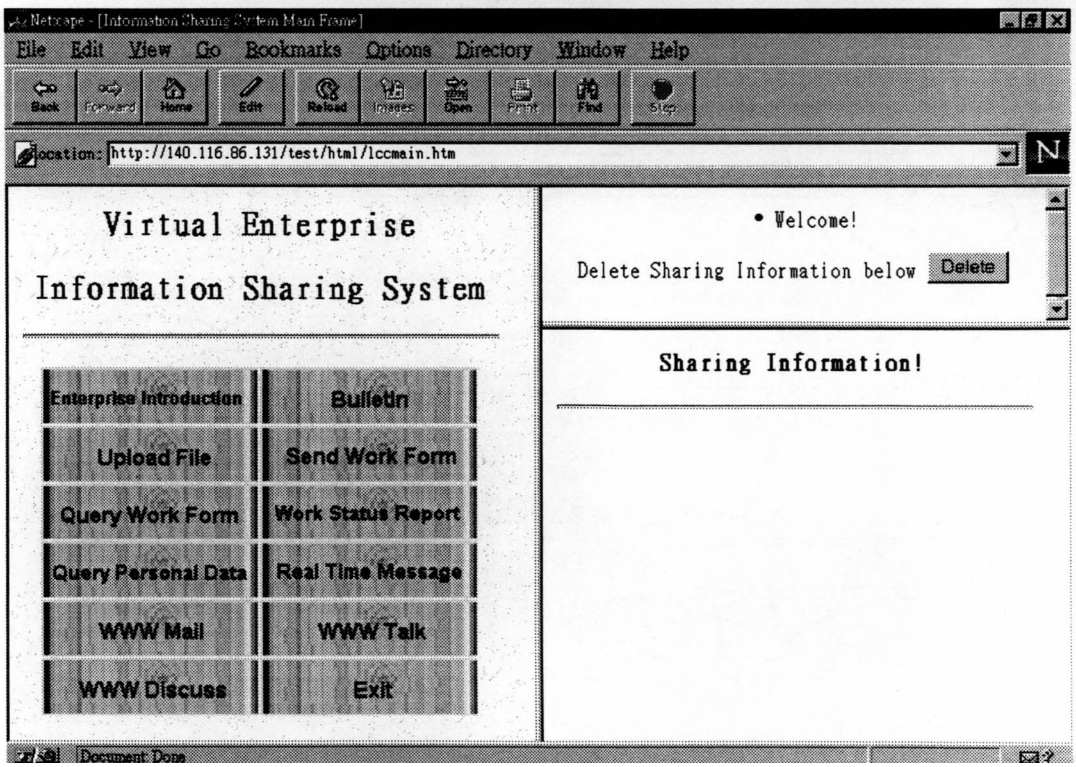

Figure 20. System main screen for employees.

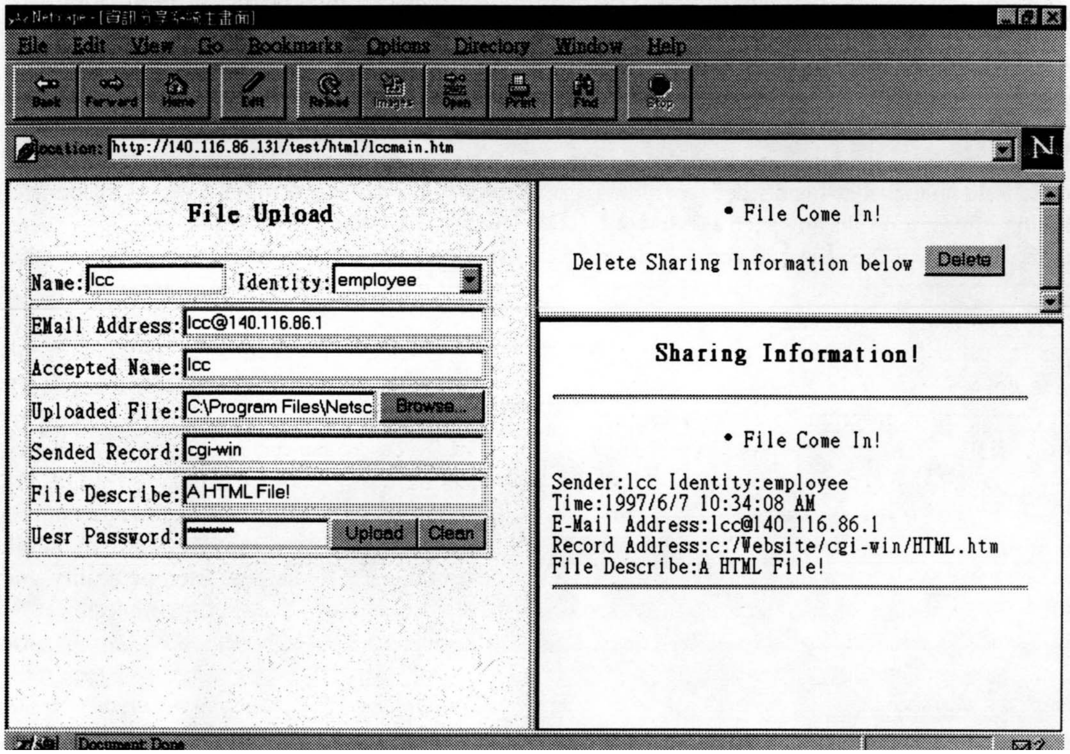

Figure 21. Interface for file up-loading.

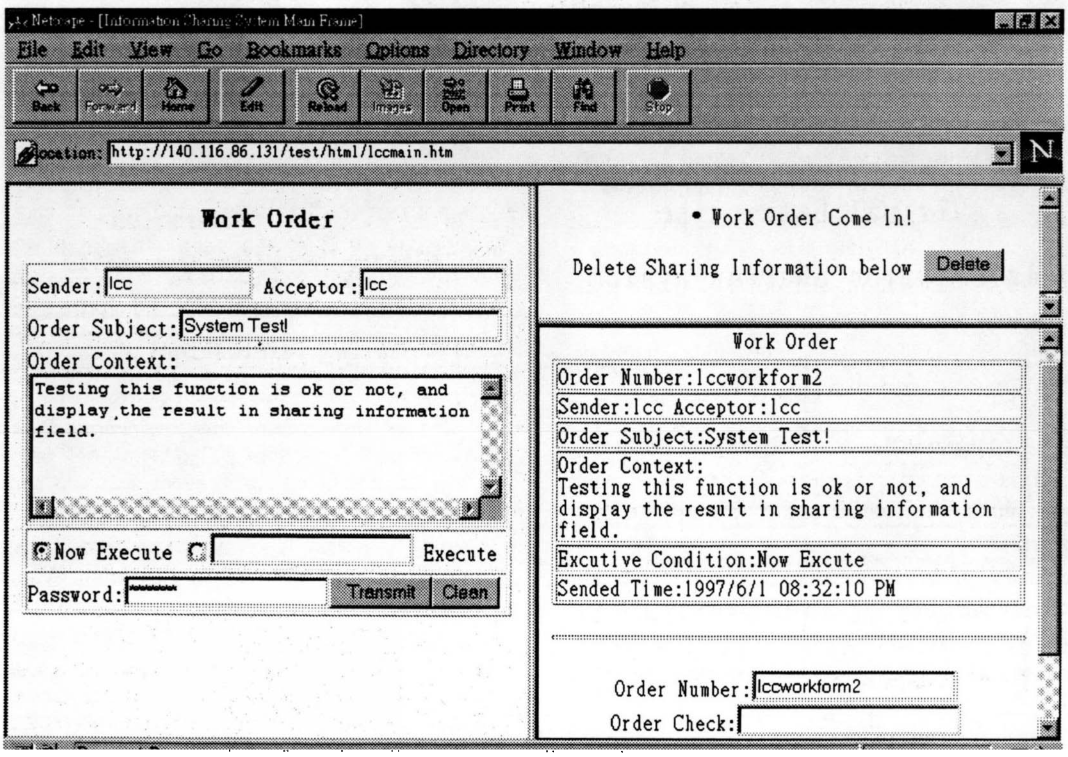

Figure 22. Interface for work order distribution.

username and password, while a guest can browse the system through hypertext linkage without username and password. After entering user name and password from the interface shown in Figure 18, the user is prompted with a main screen, depending on the privilege level of the user. Figure 19 shows the main screen for the employee. The items listed on the left-hand side of the screen are the functions provided for the employee, and the field on the right-hand side is for showing messages regarding information sharing such as the sender's name, address, the content of the message and the message arrival time.

The "Enterprise Introduction" contains the general information of the enterprise and the "Bulletin" provides information to all the employees. The "File upload" is for transmitting files, such as text files, program files and model files. Figure 21 denotes the screen for file uploading. This file-upload function is similar to FTP; however, it provides a security check and browser for file searching.

Figure 22 shows the screen for work order distribution. The information on the right-hand side is a received work order and the form on the left-hand side is the work order form for distribution. An work order contains information of work order number, the sender name, the receiver name, subject and the content of the work order.

Besides the above functions, an employee is allowed to query work order forms and personal data, report work status, send messages, and talk to other people through the screen.

7. Conclusion

In this paper, authors have presented a systematic approach to developing a knowledge management (KM) system that is able to support the intricacies of a virtual enterprise. The development includes a part of virtual enterprise characterization, definition and modeling and a part of KM system's design, modeling, and implementation. The KM system consists of a communication infrastructure, an inter-enterprise data repository, functions for inter-enterprise product data management, mechanisms for information distribution and control, and a utility for communication and information transmission.

In summary, this paper:

- Provides a definition of virtual enterprising to help clarify the concept of virtual enterprise and thus facilitate their implementation.
- Proposes a virtual process modeling method that facilitates the analysis and modeling of virtual enterprises.
- Describes a KM system that supports the practice of virtual enterprising and thus helps achieve the goals of virtual enterprise.
- Presents a systematic approach, from virtual enterprise process modeling through functional requirement analysis, system design and modeling, to system implementation in order to facilitate the development of virtual enterprise related computer-based systems.

This paper has focused on the modeling of a virtual enterprise mainly from process and information aspects. There are other aspects, one may consider such as organization, function, data, resource for defining an enterprise. In addition, this paper has concentrated on the horizontal integration of enterprises through the integration of processes. However, there is a need to vertically integrate virtual enterprises from strategic planning through process reengineering and integration to system development to ensure the applicability of the integration. Therefore, more research efforts on an integrated modeling methodology are required.

How to move from an ordinary enterprise to a virtual enterprise is another research issue. It involves the methods and technologies of organization restructuring, workflow management, process reengineering [18], and information management. To ensure the success of a virtual enterprise-based reengineering, a methodology similar to IFLOW [18] is required.

Information sharing in KM requires interoperability between activities, processes and systems in a heterogeneous environment. To increase interoperability, moving the current system to a CORBA (Common Object Request Broker Architecture)-based system will be a desirable extension [17]. KM requires information exchange between information systems or database systems on an evolutionary basis. A method for managing the evolution of information exchange and a global unified model, which dynamically accommodates different resources of knowledge (information, data and process) are required.

Acknowledgment

This paper reports the results of a research supported by National Science Council Grant No. NSC 87-2218 E-006-003.

References

1. Bernus, P. and L. Nemes, 1996. "Enterprise Integration—Engineering Tools for Designing Enterprise," In *Modeling and Methodologies for Enterprise Integration*, (ed. by P. Bernus and L. Nemes). Chapman and Hall London.
2. Chen, Yuh-Min, Chengter Ho and Tean-Heng Zhao, 1997, "An Approach to Implementing Engineering Data Management," The Fourth International Conference on Concurrent Engineering: Research and Applications, Oakland University, August 20–22.
3. Chen, Yuh-Min, Hsiao, Yun-Tao, 1997, A Collaborative Data Management Framework for Concurrent Team-Oriented Product and Process Development. I. J. Computer Integrated Manufacturing, Vol. 10, No. 6, pp. 446–469.
4. Cleetus, K. J., 1992, Modeling Evolving Product Data for Concurrent Engineering. CERC *Technical Report,* ERC-TR-RN-92-016. Concurrent Engineering Research Center, West Virginia University.
5. Cleetus, K. J., 1993, Virtual Team Framework and Support Technology. CERC *Technical Report,* ERC-TR-RN-92-016. Concurrent Engineering Research Center, West Virginia University.

6. Curtis, B., Lellner, M. and Over, J., 1992, Process Modeling. Communication of the ACM, Vol. 35, No. 9, pp. 75–90.
7. Davidow, William H. and Malone, Michael S., 1992, The Virtual Corporation, Harper Collins Publishers, USA.
8. Davenport, Thomas H., 1993, *Process Innovation: Reengineering Work through Information Technology*, Boston, Massachusetts: Harvard Business School Press.
9. Fox, M., 1993, Issues in Enterprise Modeling, Proc. of the IEEE Conference on Systems, Man, and Cybernetic, Le Touquet, France.
10. Gruhn, Volker, 1995, Business Process Modeling and Workflow Management. I. J. of Cooperative Information Systems, Vol. 4, Nos. 2 & 3, pp. 145–164.
11. Johansson, Henry H., Mchugh, Patrick, Pendlebury, A. John, Wheeler, William A., 1993, *Business Process Reengineering*, John Wiley & Sons, New York.
12. Karinthi, R., Jaganna, V., Montan, V., Petro, J., Raman, R., and Trapp, G., 1992, Promoting Concurrent Engineering Through Information Sharing. *Proceedings of ASME Winter Annual Meeting*, Anaheim, California, 8–13 November, 1992.
13. Karinthi, R., Jaganna, V., Montan, V., Petro, J., Sobolewski, R., Raman, R., and Trapp, G., 1992, Promoting Concurrent Engineering Through Information Sharing. *Proceedings of ASME Winter Annual Meeting*, Anaheim, California, 8–13 November, 1992.
14. McIntosh, Kenneth G., 1995, Engineering Data Management: A Guide to Successful Implementation, McGraw-Hill Book Company, New York.
15. Mayer, R.J., Paintec, M. K., and Dewitte, P. S., 1994, IDEF Family of Method for Concurrent Engineering and Business Reengineering Applications, Technical Report. Knowledge Based Systems, Inc.
16. Miller, E., 1993, Managing Engineering Data: An Inside Look at PDM. *Computer-Aided Engineering*. July, SR1 - SR8.
17. Orfali, Robert, Harkey, Dan and Edwards, Jeri, 1996, *The Essential Distributed Objects: Survival Guide*. John Wiley & Sons, Inc. New York.
18. Prasad, B., 1996. "*Concurrent Engineering Fundamentals*, Volume I: Integrated Product and Process Organization," New Jersey: Prentice Hall PTR, 1996.
19. Prasad, B., 1997. "*Concurrent Engineering Fundamentals*, Volume II: Integrated Product Development," New Jersey: Prentice Hall PTR, 1997.
20. Prasad, B., F. Wang, and J. Deng, 1997. "Towards a Computer-supported Cooperative Environment for Concurrent Engineering," Concurrent Engineering: Research and Applications—an International Journal, Volume 5, No. 3 (September), pp. 233–252.
21. Presley, Adrien and Rogers, Richard L., Process Modeling to Support Integration of Business Practices and Processes in Virtual Enterprises. Proceedings of IEEE International Engineering Management Conference, August 18–20, 1996, Vancouver, Canada.
22. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy F., and Lorensen, W., 1992, *Object-Oriented Modeling and Design*, New Jersey: Prentice Hall.
23. Srinivas, K., Reddy, R., Babadi, A., Kamana, S., Kumar, V., and Zhao, D. Z., 1992, MONET: A Multi-media System for Conferencing and Application Sharing in Distributed Systems. *CERC Technical Report*, CERC-TR-RN-91-009. Concurrent Engineering Research Center, West Virginia University.
24. Stover, R. N., 1993, EDM as an Enabling Technology. Special Report/Engineering Document Management, *Computer Aided Engineering*. August, EDM1-EDM16.
25. Trapp, G. 1991. Sharing Information: a CALS/CITIS, Concurrent Engineering and PDES/STEP Synergy. *CERC Technical Report*, CERC-TR-TM-91-011. Concurrent Engineering Research Center, West Virginia University.
26. Vernadat, Francois B., 1996, *Enterprise Modeling and Integration*. Chapman & Hall, London.
27. Wong, A., and Sriram, D., 1993, SHARED: An Information Model for Cooperative Product Development. *Research in Engineering Design*, Volume 5, pp. 21–39.