3 FEDERATED INFORMATION MANAGEMENT REQUIREMENTS FOR VIRTUAL ENTERPRISES

H. Afsarmanesh♦, C. Garita, Y. Ugur, A. Frenkel,
L.O. Hertzberger
University of Amsterdam, The Netherlands

In VE collaborations, many issues related to information management requirements become crucial, such as the privacy enforcement, information access rights, enterprise visibility levels, data redundancy control, and location transparency aspects. Analysis of the information management requirements in the VE environment lead to a federated information management approach, as the comprehensive support environment for effective information sharing among VE enterprises. In this chapter, the main information management requirements for the VE network are identified and the federated support environmental mechanisms are addressed.

INTRODUCTION

Advanced Industrial Manufacturing Environments require the design and development of a platform to support a number of Small and Medium Size Enterprises (SMEs) to come together and form and function as a Virtual Enterprise (VE). Within this platform, the enterprises involved in a VE need to inter-operate and exchange information in real time. The prominent need here is to be able to work together as a single integrated unit, while individually keeping their independence and autonomy. Although computer networks represent the enabling elements for this platform, there are a large number of challenges and open issues left unresolved, such as the definition of a flexible reference architecture for VEs that support the cooperative information management, and provision of supporting tools for the coordination and interoperability of virtual enterprises. Primarily, the VE environment can be described as a network with a very high degree of

♦ Corresponding author address: University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands, tel.: +31-20-5257512, fax: +31-20-5257490, e-mail: hamideh@wins.uva.nl

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35577-1_37

L. M. Camarinha-Matos et al. (eds.), *Infrastructures for Virtual Enterprises*

information sharing and exchange among its nodes (enterprises). Furthermore, every node within a VE network performs certain task(s) towards VE's common goal. As such, the nodes behavior is similar to agents behavior within a multi-agent system, where enterprises perform distributed tasks. Consequently, a virtual enterprise requires a coordinator to homogenize and control the performance of tasks by its members. The VE Coordinator requires to have certain privileged access to local data of each enterprise, in order to monitor the status of its tasks execution.

Among other characteristics that describe the VE paradigm, the following are of special interest and need proper support by the Information Management (IM) framework: (1) the heterogeneity and autonomy of pre-existing nodes, (2) possibility of both loose- and tight-coupling among the nodes, (3) proprietary vs. sharable node information, (4) negotiation and task status monitoring among the nodes, (5) virtual organization coordination, (6) specific file transfers (e.g. STEP files for product data) and EDI messages exchange for business data definition, and (7) networking aspects.

Advanced mechanisms must be designed and implemented in order to support VE's proper information management (Silberschatz, 1997), (Afsarmanesh, 1997). The general information management characteristics of this network must support: the wide variety of kinds of shared information distributed over the network, different information access and visibility levels for the cooperating nodes, incorporation of pre-existing and proprietary information (legacy information), and management of distributed information associated with the functionality of the "virtual enterprise entity" and its coordination.

Here, for the sake of performing a comprehensive analysis, we consider that the reference architecture of a VE platform constitutes a "VE Interoperation Layer" connected to each enterprise's internal module (for instance the Production Planning and Control system), where the interoperation layer supports, among others, the basic inter-linking among the VE members. In such architecture, the functionalities required for proper management of information in VEs are supported by a "component" within the interoperation layer. Similarly, other functionalities necessary for the VE (e.g. handling the communications, product data processing, EDI, etc.) will be supported through the other modules of the interoperation layer.

In (Afsarmanesh 99) this reference architecture, that is also adopted for the ESPRIT PRODNET II project, is described in details.

The focus of this chapter is the identification of the VE information management requirements and the analysis and description of the federated information management architecture as the proposed support environment for effective information sharing among its enterprises. The content of this chapter can be divided into three main parts. First, the information management requirements of the virtual enterprises are identified. For this task, an analysis study of the VE application domain was carried out and an initial classification of the information needed to be modeled in the VE application domain was defined. Second, an analysis of the general functionality requirements and the information management operations that are required for the VE interoperation layer are described. The third part of this chapter describes how the required information management models and operations, necessary for the VE interoperation layer, can be supported by means of a distributed/federated information management component. Some insight into the related research and projects addressing federated database aspects are also provided.

INFORMATION MODELING REQUIREMENTS FOR VEs

Due to the complexity of the virtual enterprise domain and the various components needed to support its required functionalities, both the analysis of the shared and exchanged information, and achieving a comprehensive classification of this information are difficult. Thus, in order to facilitate this analysis, we followed a step-wise approach: (1) As a first step we divided the study domain into three focus areas that represented the main kinds of interaction and exchange of information between different elements of the VE environment. This analysis identified a wide variety of types of information that needs to be handled at every node, and a large set of requirements for information exchange among the nodes; (2) As a second step towards modeling this diverse information for every enterprise, we divided the information into three categories of Self-, Acquaintance- and VE-related information with corresponding intuitive meanings. Below, we describe these steps in more details.

Definition and study of information focus areas

In this section, we first introduce and briefly describe the three focus areas. Then in the following paragraphs we provide more detailed descriptions for each focus area (see Figure 1):

- Focus Area 1: Addressing the management of information exchanged among different components inside the VE interoperation layer. Here, we consider all the main components that together provide the coherent cooperation layer for every node in the virtual enterprise.
- Focus Area 2: Addressing the exchange of information between a PPC (or enterprise's internal modules) and its VE interoperation layer. For instance, in order to make the PPC information available inside the VE interoperation layer, so that it can be accessed by other components of the layer or by other enterprises in the VE network.
- Focus Area 3: Addressing the information sharing and interoperation between every two enterprises in the VE; namely the information exchange and data access/retrieval queries between the two interoperation layers of the two enterprises. Clearly, within each VE the VE members have specific privileges and there is a wide variety of requests that need to be handled. Also, the fact that an enterprise can be simultaneously involved in several VEs must be carefully considered.

The proper support for these focus areas in turn constitutes the main goal of the design of the information management component of the interoperation layer. A more detailed description with general data examples for each focus area follows next.

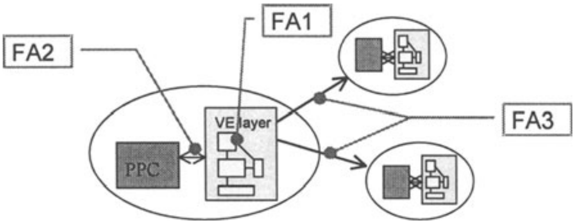

Figure 1 - Focus areas for information management.

***Focus Area 1:** Exchange of information among components of the VE interoperation layer*

Considering that EDI (Electronic Data Interchange) is a standard tool for emulation (coding/decoding) of business related information, here we have chosen the EDI component of the VE interoperation layer as the example environment. To support the EDI module with its management of information, we have identified a given data element set, part of which follows next as an example:

- Partner enterprise identification information
- Partner application-dependent information (interchange contract reference, interchange reference number, message reference number, etc.)
- EDI subset group information
- Order information (order key, date, buyer identifier, currency rate, transport information).
- Order line information (line identifier, article description, quantity, measure unit, reference price, discount, retail price).

Similar to the EDI module, the information of other components of the interoperation layer need to be studied and identified in order to be properly handled by the information management component.

***Focus Area 2:** Exchange of information between the PPC and the VE interoperation layer*

Some of the information that needs to be managed within the interoperation layer involves a part of the information stored in the PPC system of the enterprise. The structure of this information must be carefully defined (if it is not defined according to existing standards), through analyzing the typical PPC information that needs to be used by interoperation layer, or needs to be exchanged with other enterprises.

The sample information that was collected and analyzed for this particular PPC included: order production planning and scheduling, contract information, supervision clauses for specific orders, machine utilization, bill of materials, resource assignments, product description, and supplier information.

***Focus Area 3:** Exchange of information between two VE interoperation layers*

Following is a list of requirements of information sharing among the VE nodes that are foreseen at the moment. However, in the case of focus area 3, since the VE cooperations are new phenomena, the information sharing/exchange *needs* among the enterprises, must also be dealt with as a new phenomenon. Many information management requirements are already identified, however, similar to the introduction of any new tool to the market, in fact more needs will be created only after the benefits gained from the PRODNET tool are fully defined or demonstrated. Currently identified information that needs to be exchanged include: enterprises profile information, product information, contract information (payments condition), information related to VE cooperation and coordination, client orders, orders status reports, order execution diagrams, and unformatted information (blobs)

Initial VE Information Classification

The preliminary analysis step provided a rich set of sample information to be managed by the IM component as identified previously in this chapter. Before any approach to model the collected data, first that data must be properly classified and structured. Complex application domains in general, such as the VE environment, need to handle vast amounts of information of large variety. In such domains, the proper approach for modeling of information is to follow a *stepwise classification approach*. In this section, we first define six categories that primarily represent all the collected information described in previous section. This classification is further used as the base for the information modeling approach described in the next section. Following there is a list of these categories:

1. Enterprise information for public access. A part of the information describing the enterprise and the role it would like to take in potential virtual enterprises, shall be made accessible to all the nodes in the network. Considering the fact that this information may be updated progressively at the PPC, access to the current up-to-date data through the IM component is desired.
2. Enterprise information for members of a virtual enterprise. For every virtual enterprise that a node is involved in, a subset of the data stored in the PPC is needed to be accessed by other members of that VE. Therefore, access to the up-to-date version of such data must be possible. Clearly, the authorization of the VE members accessing such data must be checked by the IM module.
3. Enterprise information for a sister company. Similar to the situation described for VE members, every sister company of an enterprise also needs authorized access to a part of (or most of) the data available at the enterprise.
4. Directory of other relevant enterprises. For every enterprise, it is desirable to keep a directory of some information describing certain other relevant enterprises, being a competitor or a complementary enterprise. This information is usually gathered from other nodes through their "Enterprise information for public access" as described above, and is used by the node's PPC to make new orders, establish VEs, etc.
5. Data from other VE members. Typically, when the enterprise is involved in a VE, it needs to access some information from other members in the VE. This information is gathered from the other nodes through their "PPC Information for members of a virtual enterprise" as described above and is used by the node's PPC, to for instance check the status of the processing of certain order in the VE.
6. Data from another sister company. Similar to the situation described for the data from other VE members, there is a need to access a part of (or most of) the data available at other sister companies.

Once the tailored exchange of information, as defined in items 1 to 6 above, is supported by the IM component at the interoperation layer of every node, then the users at a node can request information from other nodes in the network. To support all these information exchange possibilities, a set of database query/update commands and mechanisms need to be provided through which the involved elements can communicate.

VE Information Modeling Approach

Based on the initial classification of information above (which is in turn based on the Focus Areas analysis), we can define a *base* approach to model the information, in order to support all the VE life cycle activities. This modeling approach is important to be carefully defined, since with a proper strong base for information modeling, both the current and the potential future requirements can be structured and supported. In these general terms, we can define three information categories to cope with the VE information management needs: 1. Self information, 2. Acquaintance information, and 3. Virtual Enterprise information.

Self Information. The Self information represents a part of the enterprise information which is managed and controlled at every node in the network in an independent way, and is not directly related to the concept of VE. For example, we can mention the information that is needed to be managed at the interoperation layer components for their proper operation, the node's self-description, and information necessary for local coordination. A part of the information comes from the internal PPC, in which case the PPC system is responsible for keeping that information properly updated. Different access level mechanisms must be defined here, to properly ensure the information security at every node. With this mechanism, this information can be shared with other nodes. We define three further divisions on the Self information for every node:

1. Self-public information (S-Public). This information includes some general description of the enterprise that in a way advertises the company, and is made accessible to public, e.g. the company profile, enterprise product/services, etc. Some information about the general configuration of the enterprise node also may be included (e.g. availability or not of STEP/EDI services).

2. Self-Associated information (S-Associated). This is a part of the enterprise information that will be made available either to some particular company (e.g. a sister company or a partner), or to a group of enterprises to which this enterprise is related in certain way. For instance, the information regarding the customers and general orders at this enterprise are included here. Access to every piece of this information is only possible by authorized users. Additionally, some access protocol configuration information may be available here (e.g. to access STEP and EDI files), since it may be required for some particular communication or information transfer protocols.

3. Self-private information (S-Private). This type of information stores the data, which is needed to support the regular operation of the interoperation layer components, and is not shared with other nodes in the network. It is intended to be accessed only for local processing. This information includes: components' local configuration information, log files, network communication parameters, passwords for local users, local coordination information, and so on.

Furthermore, in order to satisfy the several information management requirements of the different nodes in the VE, the Information Management System must be a flexible and configurable environment. It must support the definition and update of the enterprises involvement in a VE, due the evolutionary and exploratory interaction among nodes.

In relation to the management of information to support the VE member evolution, two specific stages can be identified: initialization and configuration. The initialization stage is the definition of the enterprise, and its preparation for getting involved in a particular VE. The configuration stage, on the other hand is the proper setting of parameters, so that the enterprise can be involved within every VE, and also it must support the changes of these parameters, during the VE life cycle, whenever it is required. The initialization stage encompasses for instance, the installation of hardware and general software, installation of specific software, and partners search. The configuration embodies the establishment and leaving a VE alliance, the VE contract agreements, and change in VE parameters, among others.

Acquaintance Information. The acquaintance information is obtained from other remote enterprises to serve internal purposes at this node. This information provides a general description of other enterprises, necessary for making a local directory of products and services available in the network. It also provides a facility to share some remote information. Here again, the corresponding access level mechanisms of other nodes ensure their information security. We define the following divisions for the acquaintance information:

1. Acquaintance-public information (A-Public): information which is acquired from the *Self-public information* of other nodes in the network. For example, an enterprise can acquire the general information that other enterprises in the network have made available as a profile. The profile information encompasses: enterprise name, fields of expertise, developed projects, offered services/products, etc.

2. Acquaintance-Associated information (A-Associated): information which is acquired through access to *Self-restricted information* in other nodes. For instance, similar to the *Self-restricted information* described above, some companies have certain level of confidence established with some other partner enterprises that makes them collaborate in a closer way, without necessarily forming a VE. In this case, the companies are willing to exchange particular information under this mutual trust situation. This particular information can be very diverse: text documents, product descriptions, market analysis, and commercialization strategies, inventory information, etc.

Virtual Enterprise Information. This category refers to all that information which is associated with a node regarding its actual involvement in a VE. Here, we systematically define the information into the self and acquaintance sub-categories. Please notice that for every VE in which this node is involved, this framework needs to be repeated. This in turn guarantees the separation and security of every VE's information. Then, the VE information is composed of:

1. VE-Self information: enterprise information related to its participation in a VE, that is managed and controlled entirely by this enterprise. A part of this information will be shared with other VE members. The VE-Self information can be further classified as:

- VE-Self-public information (VES-Public): information which this enterprise makes available to all the members in the VE. This information can include product information, order descriptions, VE costumers descriptions, the schema-description of the information that is available through the *VE Self-restricted information*, for other VE members to investigate and request this node for access privileges if interested, etc.
- VE-Self-restricted information (VES-Restricted): information that this enterprise is willing to make available to certain other authorized VE members. In one given case, the other VE member can be the VE Coordinator; then, examples of this information class include: orders planning and scheduling, logistics, inventory, resource management, billing, order descriptions, enterprise order status, workflows, etc.
- VE-Self-private information (VES-Private): information regarding the participation of this enterprise in a particular VE, that is stored for internal purposes, and that is not shared with any other node. This information includes the configuration of the interoperation layer components for a particular VE, internal order information (e.g. status and progress), internal planning and scheduling, local coordination support information, some VE coordination support information, etc.

2. VE-Acquaintance information: information related to other enterprises regarding their participation in the VE. This information can be remotely accessed from other VE nodes. The acquired VE information can be categorized as:

- VE-Acquaintance-public (VEA-Public): information acquired from other nodes that have defined this information as their VE-Self-public.
- VE-Acquaintance-restricted (VEA-Restricted): information acquired from other nodes in the VE through their VE-Self-restricted information access.

3. VE Messages (VE-Messages): a basic interaction between the members of a VE can be supported through the sending and reception of simple text messages. These messages could be:

- VE-Restricted Messages (VEM-Restricted): messages that are sent to a specific set of partners among a VE.
- VE-Public Messages (VEM-Public): messages for all the partners in a VE.

Furthermore, due to the future information management requirements of the VE domain, we foresee the need to also model the following information structures:

4. VE n-lateral workspace (VE-Workspace). We foresee the need to also model a generic workspace mechanism among the VE members. This facility can be used at the level of two or more particular members, in order to co-work, reach an agreement, or "negotiate" on a particular issue. This workspace is intended to be a communication/interaction area for all the members in the VE. This space can be used for global VE announcements and dissemination of common VE information.

FUNCTIONAL REQUIREMENTS INFORMATION MANAGEMENT

When analyzing the VE information management requirements, besides the identification of required data elements and structures in the VE environment, the identification of the operations that are needed to be performed on the data, is the next important issue. For example, the following operations are required to support the VE functionalities:

- For every component inside the Interoperation Layer, for instance for EDI: the EDI order messages which arrive at the VE member enterprise should be stored in the information management component of the interoperation layer with a common structured form so that enterprises' legacy system can access it.
- For inter-enterprise communication support: any communication service for the interoperation layer needs to store/read inter-VE member messages; furthermore, update/read operations on the information to encrypt and deliver the VE messages are required.
- For the VE coordinator: the VE coordinator needs to gather distributed VE order-related information which can be a detailed commercial specification of the product orders requested from a VE member, production information, and intra-organizational analysis parameters.
- For the legacy system: store/read the VE-related order information that is requested by the VE coordinator who needs to monitor it, and store/read the order information for purchasing products from other enterprises.

As a general strategy to achieve the design of a set of IM query/update operations to support all the requirements, the following sub-tasks were outlined:

1. Identification of the general IM operation design aspects and necessary considerations
2. Analysis of the general information management query/update functionalities required by the interoperation layer modules
3. General design and specification of the set of IM access (retrieve/update) commands and/or service functions
4. Specification of IM query/update functionality

The first point addresses the identification of certain aspects or considerations, which are either desirable or mandatory for the design of the IM operations. The second point involves a careful study of the information management functionality needs for every module of the interoperation layer. At the third point, after the identification of the general service functions for the interoperation layer modules, a more formal specification of the operations to support these functionalities must be outlined.

General considerations for IM query/update operations

There are several important points to be considered when analyzing the required functionality and operations for information management. Following is a list addressing these considerations:

- Compatibility with the general interoperation layer architecture. The interface between IM and the other interoperation layer modules must be in strict compliance with the general architecture design. Some architectural aspects and division of roles among modules have a strong influence on the IM operations design and its latter implementation.
- Loose coupling among interoperation layer modules. The general interaction mechanism between the IM and the other modules must be very clearly defined and kept as simple as possible.

- Operations completeness and simplicity. The set of operations must provide the other modules with all their required information management features, but at the same time must be kept as simple and easy to use as possible.
- Support for visibility access levels on specific information. A mechanism to define and preserve information visibility levels for secure access and retrieval of data among the VE nodes is one important aspect to consider in the design of the IM support operations.
- Lower-level facilities exploitation. The possibility that the IM component may be built on top of an already existing DBMS with its own query/update language support (i.e. the SQL) must be fully exploited, trying to reduce the implementation efforts and reinvention of the wheel.

Analysis of required IM query/update functionalities

As mentioned previously in this chapter, the first step towards the modeling of the information management requirements was the definition of three focus areas. The proper support for these focus areas constitutes one of the main goals of the design of the IM module inside the interoperation layer. Therefore, the analysis of the required IM query/update functionalities can start with an in-depth look to these areas, but this time focusing on the required information access/update mechanisms rather than on the structure of the exchanged information. With this in mind, the access/update requirements that need support within the three areas can be defined as follows:

- For Focus Area 1: information management queries needed by different components inside every interoperation layer.
- For Focus Area 2: information management queries needed to be supported between the company internal module and its interoperation layer.
- For Focus Area 3: information management queries to share information and reinforce interoperation between every two nodes in the virtual enterprise; namely the information exchange and data access/retrieval queries between the interoperation layers of two nodes.

After this general focus areas identification, the next step is the analysis of the requirements for every interoperation layer module to clearly identify their specific query/update operations needs. Two main basic (extreme) approaches can be taken for the identification of requirements for the "format" of the DIMS operations:

- Approach 1: A general, de facto standard database language, e.g. pure SQL. In this case, the IM language would be the SQL language, and all the interoperation layer modules would send their messages to the IM component with embedded SQL commands. The commands would be analyzed and parsed by the query processor of the IM component, where it will be determined if the command merely contains a local query or if it contains a query retrieving information from other enterprises (distributed query). In the latter case, extra actions such as further decomposition and sending of sub-queries to other nodes have to be carried out. Then the SQL language may need to be extended to support the required distributed query processing in the VE environment.
- Approach 2: A comprehensive library of service functions. On the other hand, the IM functionality can be completely "hardwired" in the sense that no general

query language (such as SQL) is applied. In this approach, the specific needs for every interoperation layer module are analyzed and identified, giving the possibility of designing one or more IM libraries containing service functions, which the corresponding modules can invoke. This option is not very flexible for the PCL modules, but it is quite convenient because from these modules' point of view the functions provide a higher level of abstractions than the plain low-level query language commands. For example, instead of having to issue sequences of commands and processing several query results, the modules can "ask" for the definition of certain specific functions to be included in the library, and these functions can encapsulate the command sequences and even do some low-level processing on some query results.

Clearly, the two cases represent the two extreme possibilities and the approach taken for the IM could be a hybrid approach to combine the benefits of both cases.

Specification of IM query/update operations

From the study of the three focus areas described in the previous section, we have determined that IM query/update operations can be provided through the following mechanisms:

- Local queries. The most evident mechanism to support the persistent information management requirements of the interoperation layer modules, is to make a function available which contains a generic query as a parameter. If it happens that the sender of the query, being the PPC or a layer module, knows and specifies that the query is local to this node, in that case, the query is evaluated locally at the IM of the node. This would represent the minimal IM access/query support for the interoperation layer components.
- Generic data management functions. It is also possible to provide access to local data through the generic functions that in turn encapsulate and issue query commands. In this way, the applications do not need to deal with language specificities, and may just provide the necessary information through the access functions. These functions do not necessarily imply a one-to-one mapping in relation to query commands; they can even encapsulate a sequence of language commands.
- Specific functions for modules of the interoperation layer. The support for specific database-related functions can be extremely convenient to facilitate the work of the other interoperation layer modules. Through the specified functions for every module of the interoperation layer, even sequences of query commands and other functions can be encapsulated to provide a higher level of abstraction for the module. The functions can be designed to retrieve local and distributed data as well.
- General (distributed) queries. The distributed VE information from other remote partners must be seamlessly *integrated* with local information at every node. The general distributed IM queries represent queries that can be applied to this *integrated* schema. Please notice that the distributed IM queries imply the need for the development of an advanced distributed query processing.

As mentioned before, nodes of a VE network can be considered as agents within a multi-agent system, where the independent tasks of different agents will be monitored by the coordinator. As such the general goal of the VE will be accomplished. Several VE functionalities are supported through the VE coordination and some associated services, such as the VE partner search and logistics. These advanced functions generate distributed/federated queries that involve accessing several remote SMEs within the VE.

For instance, the VE coordinator, in order to direct the enterprises towards cooperative behavior, have privileged access to certain proprietary local data at different nodes (the local data related to subtasks of each enterprise) in order to monitor the status of their tasks execution. Considering the autonomy of SMEs involved in a VE, gathering the information to monitor the entire client order can only be supported through advanced multi-database architectures and query processing. Here, the coordinator should be able to issue a complex query with consequences on several enterprises. In principle, the complex compound query processing may involve several steps, including the query decomposition, issuance of several remote subqueries to involved SMEs, collection of partial results from SMEs, and generation of the final result to the coordinator. From the coordinator perspective, data access, which is in fact access to integrated data, is not concerned about the data distribution over the VE network.

The integrated data definition of an enterprise over which a federated query is issued is actually the union of all export data definitions that each VE member is willing to share and exchange with this enterprise. However, all these details should be hidden from the query issuer by the federated query processing functionality.

FEDERATED INFORMATION EXCHANGE AMONG ENTERPRISES

As mentioned earlier in this chapter every enterprise must share and exchange a part of its information available at its interoperation layer, with other enterprises. Similarly and at the same time, some information from other nodes must be accessible and acquired by this enterprise. Furthermore, proper sharing and exchange of information must be supported when an enterprise is involved in more than one VE. However, the visibility levels of the enterprise information from external nodes, must be carefully determined by the node, in order to ensure both its autonomy and information privacy.

In order to support these requirements, and proper interoperation among VE interoperation layers in different nodes, a federated database architecture can be designed and developed for the IM module (Afsarmanesh, 1997a), (Afsarmanesh, 1997b), (Camarinha-Matos, 1997), (Camarinha-Matos, 1998). Here as an example and for the sake of analyzing different features and facilities of federation and how it matches the requirements of the management of information in VEs, we address the PEER federated database system, developed at the University of Amsterdam (Wiedijk, 1996), (Afsarmanesh, 1995), (Afsarmanesh, 1994), (Afsarmanesh, 1993), (Tuijnman, 1993). PEER is a fully federated, object-oriented information management system with special emphasis on the node's autonomy and the complete distribution of both data and control within the cooperative network. Namely, it supports the sharing and exchange of information among cooperating

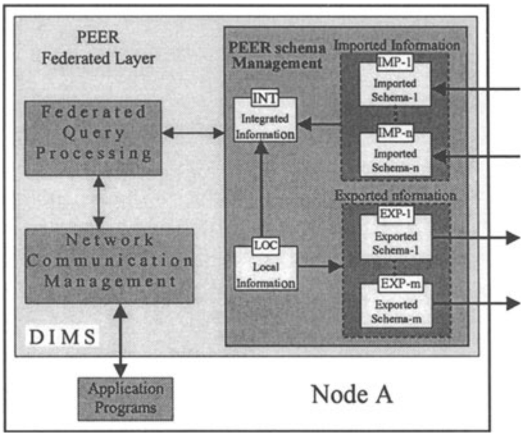

Figure 2 - PEER federated information management layer

autonomous and heterogeneous nodes without the need for any centralization and data-redundancy.

In PEER, interdependencies between the information of two nodes are established through the schemas defined on their information; thus there is no need to store the data redundantly in different nodes. Every node's information is represented by several schemas (see Figure 2): a local schema (LOC), several import schemas (IMP), several export schemas (EXP) and an integrated schema (INT) (Tuijnman, 1993). The local schema is the schema that models the data stored locally. An export schema models some information that a database wishes to make accessible to other nodes (usually, a node defines several export schemas). The various import schemas model the information that is accessible from other databases. The integrated schema presents a coherent view on all accessible local and imported (remote) information. The integrated schema can define a particular classification of objects, which are usually classified differently by the schemas in other nodes. A prototype implementation of the PEER system is developed in the C language in UNIX environment, and includes two user interface tools (Afsarmanesh, 1994), a Schema Manipulation Tool (SMT) and a Database Browsing Tool (DBT).

In general, the "distributed information management systems" can be found within a wide variety of architectural approaches; some taxonomies are described in (Bright, 1992) and (Sheth, 1990). These systems may run on different hardware architectures ranging from several tightly coupled computers with a centralized control to geographically distributed (both physically and logically) machines. The data itself may also be represented and modeled in different ways. But the fundamental features that make these systems differ from a traditional centralized database system is that the data is partitioned and physically stored on different computers (Garcia-Molina, 1995).

Some other related systems or conceptual models in this area are, for example:

- The research presented in (Abiteboul, 1991) describes the definition of object-oriented views and the "imaginary objects" in views, where these objects do not correspond to objects defined in the underlying database.

- The schema architecture of UniSQL/M (Kelley, 1995) supports the integration of relational and object-oriented schemas. Object identifiers can be generated for virtual instances of an imaginary type based on relational tables in the schema. The related research presented in (Kim, 1995a) introduces an elegant classification of the structural conflicts involved in multidatabase integration, and how they are resolved within the UniSQL/M.
- Pegasus (Shan, 1995) supports the definition of virtual types and functions, where functions represent either properties, relationships, or computations. Objects in different databases can be defined as equivalent using the type combine primitive.
- The research presented in (Chomicki, 1994) extends the OSQL language with the support for definition of declarative implicit and explicit multidatabase mappings among types and functions (either stored or derived where the value is defined using an OSQL query), of the component databases.
- The COCOON data model supports an object algebra and views for multi-object-bases (Scholl, 1994), (Radeke, 1994).
- The IRO-DB (Essmayr, 1996) supports homogeneous access to heterogeneous and distributed databases. This system is a federated database system which architecture consists of three layers: the local layer, which supports access to heterogeneous component database systems; the communications layer, to provide services for remote database and object access; and the interoperable layer that integrates the various local schemas into an interoperable schema which combines data from the local databases and handles inconsistencies in structure, naming, scaling behavior and semantics.
- The architecture for the system VHDBS presented in (Wu, 1996) aims at the support for cooperative access to distributed heterogeneous databases. It uses an object-oriented data model as a common data model to integrate heterogeneous data.

Federated database systems are multi-database systems, in which every node in the federation maintains its local autonomy on the data and defines a set of export schemas through which the data is made available to other specific nodes. Also, every node will be able to import data from other nodes through their import schemas, and access their data according to the pre-defined “access permissions”. As a consequence of this general interaction facility, the approach allows the cooperation between the federated nodes in order to accomplish a common or global task, while the local autonomy and independence of every node is preserved and reinforced. Thus, there is no need for a centralized repository of data or control and no need for data redundancy in the network.

Federated Databases and VE Information Management Requirements

In order to fully support the important information management requirements of the VEs, we can use as the base a federated architecture that provides for both the data exchange and the cooperation on joint work.

The Information Management System must be able to allow a high level of interaction between nodes, where both the information and enterprise configurations are constantly changing due to the different levels of dependencies among the production chains of the enterprises in VE. To accomplish these tasks, a federated architecture such as the PEER system described in the previous section) provides the base for an effective information sharing and cooperation among different enterprises, to support many features, among which the following can be emphasized:

1. Objects stored in a node can be shared with other nodes.
2. It is possible to access up-to-date the remote objects shared by other nodes.
3. Different levels of access privileges and information visibility for other nodes.
4. The physical and logical information distribution among the nodes becomes transparent to the users.

The PEER federated system can support many of the required features to model the dynamic relationships between the members of a Virtual Enterprise:

- Different types of schemas are suitable to share and exchange the information maintaining the basic necessary level of security access. Thus, for every node in the domain of the enterprises network, the self-information would be stored in the export schema to allow the rest of the network to access it, the private information would be located in the local schema to guarantee its confidentiality, the acquaintance information is composed of a set of imported schemas, and the VE information is modeled as an integrated schema.
- Through its export schemas an enterprise can determine which other enterprises are allowed to access and/or update a specific subset of its local information.
- Through the importation of export schemas, the shared information are read on-line and always reflect the current state.
- No redundant information is needed to be maintained in the network.
- Through the schema integration and remote referencing it is possible to provide some support for enterprises to negotiate the products and/or services contracts, monitor the progress of an order along the network, modify and re-adjust the incomplete order information (e.g. the delivery date), and coordinate tasks with other enterprises in a value-added chain.
- Through the remote referencing and extended local schemas, the local schedule of an enterprise can establish references to an offer submitted by one or more different enterprises, therefore representing joint work.

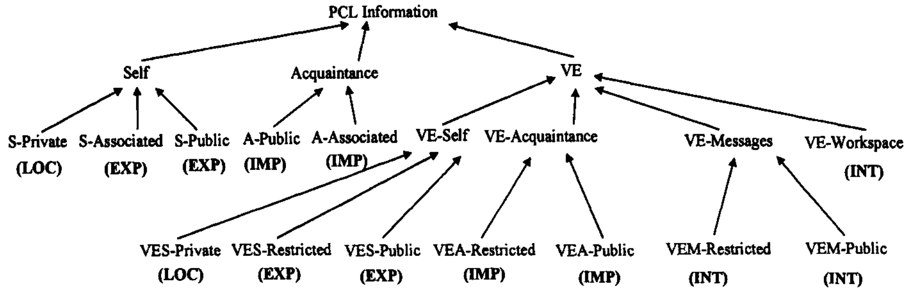

Figure 3 – Classification of VE information supported by different schema types

In Figure 3, different information categories in a enterprise, as identified earlier in this chapter, are illustrated in a tree structure. Furthermore, the support of different leaf nodes in this information categories tree are represented through associating them with elements of the federated schema, namely the LOC, IMP, EXP, and INT schemas is represented. As can be seen in this figure, whenever support for private information is needed (i.e. S-Private, VES-Private), the information is defined within the LOC schema. All the acquaintance information (i.e. A-Public, A-Associated, VEA-Restricted, VEA-Public) can be made available through the IMP schemas. The categories which concern restricted or public data to make available to other interoperation layers (S-Associated, S-Public, VES-Restricted, VES-Public) are modeled within the EXP schemas. Finally, the support to workspaces and messages within VEs is represented through the INT schema of an enterprise.

Clearly, the federated architecture is a strong base approach for VE paradigm. However, in order to effectively support the cooperation of enterprises and the dynamic generation and cease of the virtual enterprises, it is necessary to even further extend the federated architecture with many other features, such as an on-line message passing mechanism between the nodes, to facilitate the dynamic and asynchronic interactions. For example, for follow-up advanced features this is needed to support the dynamic importation / integration of schemas with the updates notification and automatic information submission.

The Federated Query Processing element of the VE Information Management module is responsible for provision of access to the privileged proprietary VE information for which an enterprise is authorized , while hiding the data location details from the end user. The data access is performed against the data defined by export schemas, over the VE members. Through its export schemas an enterprise can allow the other enterprises, depending on their roles, to access a specific subset of its local information. Since the local data related to VE subtasks is independently updated by the legacy system, the VE enterprise coordinator for instance, needs to frequently access the most recently updated data in the legacy system. The support of this demand is also under the responsibility of the VE Information Management federated query processing.

The detailed design of a federated information management for PRODNET can be found in (Afsarmanesh 99).

CONCLUSIONS

Based on the described VE information modeling analysis, the following requirements must be considered and supported when designing VE information management component:

- Provision for the basic DBMS capabilities, such as transaction management, persistence, multi-user capabilities, security access levels. Clearly, on top of these basic functions, the more advanced set of functionalities will be developed.
- Consider the expanding nature of the network of enterprises. The IM component must be open and flexible to support the easy expansion of the network as it grows up.
- Proper support for handling the EDI and STEP data must be provided. The specific design considerations to model represent and manage EDI and STEP information must be carefully taken into account in the development phases.

- Storage and management of a wide variety of kinds of information depending on the business process of the involved enterprises need to be supported.
- Storage of objects of "any" size must be supported, as an outcome of the potential diversity of data.
- Often, for a large amount of information the structure can be common for all the VE members, but also the possibility of storing objects as "blobs" must be provided in order to support the interaction of enterprises.
- Management of information for local coordination that the enterprise for its paper functionality.
- Definition of enterprise information which is shared with other enterprises with different security levels (public, restricted, private) must be supported. The concept of enterprise autonomy must be reinforced, where the enterprise has its own information, and decides by itself which subset of this information is made available to other determined members.
- Importation of the available information from one node into another one needs to be supported. This will allow the sharing and exchange of information among different enterprises.
- Storage and management of information handled by component modules of the interoperation layer must be supported.

In terms of IM operational requirements, the main conclusions as the outcome of the performed analysis are summarized below:

- An information access mechanism among the VE nodes is required, where secured fine-grained visibility levels are defined locally at every node to determine which other VE partners are allowed to access which part of the local information.
- The distributed query processing and the export/import mechanisms described in this chapter entirely support and reinforce the distributed/federated IM selected approach for managing the distributed information in VEs.

Furthermore, depending on the functionalities provided by other modules of the interoperation layer, the support of the following features within the IM may prove beneficial:

- A "notification mechanism" associated with the operations of insertion, deletion, and modification on a particular class of information may be needed. This feature in turn can support the update notifications on shared information.
- Support for the "joint workspace area" between two or more enterprises may be useful. This point aims at the collaboration support between two or more enterprises. Additionally, this feature will support the information management and exchange requirements for any action that requires reaching an agreement by two or more parties, e.g. in order to negotiate a contract.

Acknowledgements

This work was funded in part by the European Commission, Esprit programme. The authors also thank the valuable contributions from their partners of the PRODNET II consortium: CSIN (P), HERTEN (BR), MIRALAGO (P), University of Amsterdam (NL), Universidade Federal de Santa Catarina (BR), Universidade Nova de Lisboa (P), Lichen Informatique (F), ProSTEP (Germany), Uninova (P), ESTEC (P).

REFERENCES

1. Abiteboul, S; Bonner, A. - Objects and Views, in Proceedings ACM SIGMOD91, May 1991.
2. Afsarmanesh, H; Tuijnman, F.; Wiedijk, M.; Hertzberger, L.O.- Distributed Schema Management in a Cooperation Network of Autonomous Agents. Proc. of the 4th Int. Conf. on Database and Expert Systems Applications (DEXA'93), Lecture Notes in Computer Science 720, pages 565-576, Springer-Verlag, September 1993.
3. Afsarmanesh, H. et al. Flexible and Dynamic Integration of Multiple Information Bases, Proceedings of the 5th IEEE Int. Conf. on "Database and Expert Systems Applications DEXA'94", Athens, Greece, Lecture Notes in Computer Science (LNCS) 856, Springer Verlag, Sept. 1994.
4. Afsarmanesh, H.; Wiedijk, M.; Hertzeberger, L.O.; Negreiros Gomes, F.; Provedel, A.; Martins, R.C.; Salles, E.O.T. (1995) - A Federated Cooperation Architecture for Expert Systems Involved in Layout Optimization. in Balanced Automation Systems, L.M. Camarinha-Matos and H. Afsarmanesh (Eds.), Chapman & Hall, July 1995.
5. Afsarmanesh, H; Camarinha, L. - Federated Information Management for Cooperative Information - in proc. of 8th Int. Conf. on Database and Expert Systems Applications (DEXA'97), Sept. 1997.
6. Afsarmanesh, H; Garita, C; Hertzberger, L.O.; Santos, V. - Management of Distributed Information in Virtual Enterprises : The PRODNET Approach – in proceedings of the Int. Conf. on Concurrent Enterprising (ICE'97), October 1997.
7. Afsarmanesh H., Cesar Garita, Yasemin Ugur O., Anne Frenkel, and L. O. Hertzberger. "Design of the Federated Information Management for PRODNET". In the Pro-VE'99 Book: "Infrastructures for Virtual Enterprises" (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, 1999.
8. Bright, M. et al; A Taxonomy and Current Issues in Multidatabase Systems, IEEE Computer, v 25, n 3, pages50-60, March 1992.
9. Camarinha-Matos, L.M.; Afsarmanesh, H; Garita, C; Lima, C - Towards an Architecture for Virtual Enterprises; The 2nd World Congress On Intelligent Manufacturing Processes & Systems, Budapest, Hungary, June 10-13, 1997.
10. Camarinha-Matos, L.M.; Afsarmanesh, H; Garita, C.; Lima, C - Towards an Architecture for Virtual Enterprises. Special issue of the journal of Intelligent Manufacturing with the focus on agent based Manufacturing, Volume 9, Number 2, Pages 189-199, Chapman and Hall publications, March 1998.
11. Camarinha-Matos, L; Afsarmanesh, H. "The PRODNET Infrastructure". In the Pro-VE'99 Book: "The PRODNET Architecture" (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, 1999.
12. Chomicki, J. et al - Declarative Definition of Object-Oriented Multidatabase Mappings, in [Oszu94]
13. Essmayr W. et al. - Authorization and Access Control in IRO-DB, in Proceedings of the 12th International Conference on Data Engineering, IEEE Compurter Society Press, February 1996.
14. Garcia-Molina H, Hsu M. - Distributed Databases, in [Kim95].
15. Kelley W. et al. - Schema Architecture of the UniSQL/M Multidatabase System, in [Kim95].
16. Kim, W.; editor, - Modern Database Systems: The Object Model, Interoperability, and Beyond; ACM Press, 1995.
17. Kim, W. et al- On Resolving Schematic Heterogeneity in Multidatabase Systems, in (Kim, 1995a).
18. Oszu M., et al.; editors - Distributed Object Management, Morgan Kaufman Press, 1994.
19. Radeke E., Scholl M. - Federation and Stepwise Reduction of Database Systems, in Proc. of the First Int. Conf. "Applications of Databases (ADB94)", Vadstena, Sweden, 381--399, June 1994.
20. Scholl M. et al - Object Algebra and Views for Multi-Objectbases, in [Oszu94]
21. Shan, M. at al - Pegasus: A Heterogeneous Information Management System, in [Kim95]
22. Sheth, A; Larson J. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, ACMCS, v. 22, n. 3, September 1990.
23. Silberschatz, A.; Zdonik, S. – Database Systems: Breaking out the box, SIGMOD Record, v. 26, n. 3, September 1997.
24. Tuijnman, F.; Afsarmanesh, H. - Management of shared data in federated cooperative PEER environment. Int. Journal of Intelligent and Cooperative Information Systems (IJICIS), 2(4): 451-473, December 1993.
25. Wiedijk, M.; Afsarmanesh, H.; Hertzberger, L.O. - Co-working and Management of Federated Information-Clusters. Proceedings of the 7th Int. Conf. on Database and Expert Systems (DEXA'96), Lecture Notes in Computer Science 1134, pp 446-455. Springer Verlag, Sept. 1996.
26. Wu X - An Architecture for Interoperation of Distributed Heterogeneous Database Systems, in Proc. of the 7th Inter. Confer. on "Database and Expert Systems Applications DEXA'96", Zurich, Switzerland, Lecture Notes in Computer Science (LNCS 1134), Springer Verlag, Sept. 1996.