Chapter 3

Analysis of VE Information Management Requirements and the Proposed Federated Approach

3.1 Introduction

The wide range and variety of characteristics describing the VE paradigm introduce particular information management requirements that need to be addressed by any IT platform aimed at the support of this form of enterprise collaborations. For example, a set of general challenges faced in the design and development of the information management system of a generic VE support platform, were already introduced in Chapter 2. Many of these requirements stem from the high degree of heterogeneity and autonomy of the pre-existing nodes that simultaneously need to act together as a single unit. However, some of these general requirements may become more relevant or acquire different significance depending on the VE application domain and the target objectives of the particular VE support platform under consideration. For instance, the information management requirements to support VEs in sectors such as industrial manufacturing, tourism, and concurrent engineering, may share some common characteristics, but others may be substantially different depending on the aims of the VE support platform.

Therefore, the objective of this chapter is threefold. First, it introduces several examples of VE collaborative scenarios in the industrial manufacturing sector. Secondly, taking as a reference the described VE scenarios, this chapter describes the analysis of the specific VE information management requirements for the Distributed Information Management System (DIMS) of a given VE support platform focusing on the industrial manufacturing sector. These requirements include both the information
modeling and the functional requirements for the DIMS. Finally, based on the results of the performed analysis, this chapter describes the federated information management architecture that is proposed as the support framework for effective information sharing among the VE member enterprises. As will be shown in forthcoming sections of this chapter, the proposed federated architecture introduces particular advantages and novel features to adequately support the definition of information access rights based on the existing legal contracts among the VE partner enterprises.

Furthermore, in this chapter it is assumed that the reference architecture of a VE support platform is represented by a VE Cooperation Layer (VCL) connected to each enterprise's internal modules (e.g. the ERP/PPC -Production Planning and Control-system), where the VCL supports among others, the basic inter-linking among the VE members (see also Chapter 1). In such architecture, the functionalities required for proper management of information in VEs are supported by the Distributed Information Management (DIMS) component within the VCL. Similarly, other functionalities necessary for the VE platform (e.g. communication handling, technical product data management, EDI processing, etc.) are supported through the other modules of the VCL (see for instance, the PRODNET reference architecture described in Chapter 1).

The content of this chapter is structured as follows. First, several examples of industrial manufacturing VE scenarios are introduced in Section 3.2. Secondly, the information management requirements identified for the reference VE scenarios are classified in Section 3.3. For this point, an analytic study of the VE application domain is presented and an initial classification of the information that needs to be modeled in the VE application domain is defined. In Section 3.4, an analysis of the general functionality requirements and the information management operations that are needed to support the VE Cooperation Layer are described. Section 3.5 of this chapter illustrates how the required information management models and operations, necessary for the VCL, can be supported by means of a distributed/federated information management component. For this purpose, the features offered by the PEER federated system are presented, and its eventual application to the VE domain is demonstrated as a reference. Furthermore, an example regarding the DIMS federated approach for data integration to support certain VE collaborative scenarios is provided. Finally, a summary of the main conclusions achieved after the data modeling and functional requirement analysis of DIMS, is presented in Section 3.6.

### 3.2 Industrial Manufacturing VE Scenarios

In this section, some examples of industrial manufacturing VE scenarios are described, in order to define the scope of the information management requirement analysis addressed in further sections of this chapter. The scenario cases included in this section are based on the VE demonstration scenarios that were elaborated for the PRODNET project and presented at the Pro-VE'99 international conference in October 1999 in Porto, Portugal (see also [43]).

Before describing the specific VE collaboration cases, it is necessary to first introduce the assumed reference VEs and their corresponding VE partners. Thus, Fig-
3.2. Industrial Manufacturing VE Scenarios

Figure 3.1 represents a general overview of an intercontinental VE scenario to be used as a reference for the DIMS requirement analysis. In this scenario, the roles played by each VE member are described next:

- Enterprise 1 (E1) is a European bicycle producer that requires bicycle frames and pedals to support its internal production.

- Enterprise 2 (E2) is a European manufacturer company, which produces and supplies bicycle components, including frames and pedals.

- Enterprise 3 (E3) is a PVC (polyvinyl chloride) resin supplier located in Brazil. Here, it can be mentioned that the PVC resin (after being mixed with plasticizers and other components) can be used for example, in the injection molding process for the bicycle pedals required in this VE scenario.

- Enterprise 4 (E4) designs and creates product moulds. For instance, in this VE scenario, a new mould will be created and used in the construction of a new type of bicycle pedals. This enterprise is also located in Brazil.

Having defined these general enterprise roles, let us consider two specific VEs established among these companies as follows:

- Virtual Enterprise 1 (VE1): in this VE, Enterprise 2 produces and supplies bicycle frames for Enterprise 1. Enterprise 1 is the coordinator in this VE.

- Virtual Enterprise 2 (VE2): this VE involves all four enterprises. Namely, in this VE, Enterprise 2 must deliver a new kind of bicycle pedal to Enterprise 1, for which it needs PVC resin material from Enterprise 3, and a specific pedal mould from Enterprise 4. In this case, Enterprise 2 is the VE2 coordinator.

![Figure 3.1: General VE Reference Scenario.](image-url)
These VEs can operate in a simultaneous and independent manner for an indefinite period of time, and a given company may play different roles in different VEs (e.g. Enterprise 2 is a regular partner in VE1 and the coordinator in VE2).

Based on these enterprise roles and VE scenarios, some VE operation cases are described in the next sections. In order to facilitate the presentation of the involved steps and functionalities, most of the cases focus on the main interactions that take place between two given companies at a time. These cases will contribute to the identification of the information modeling and operational requirements for the DIMS.

### 3.2.1 Scenario Case 1 – Interactions between VE Members

This scenario case shows some of the possible interactions among enterprises E1 and E2 within the context of VE1, with focus on commercial order functionalities. This scenario represents a situation in which E1 searches for suppliers of bicycle pieces (e.g. frames), and selects E2 with whom a contract is signed. After this, E1 can generate purchase orders for E2. Once an order is accepted, E2 starts the internal production, which can be monitored by E1 (VE1 coordinator) according to the VE contract. When the ordered frames are ready, they are dispatched to E1. An example interaction scenario for a VE established among these two companies is described in Table 3.1 below (see also Figure 3.2), where the capital letters inside parenthesis also indicate and correspond to the arrows in the diagram.

<table>
<thead>
<tr>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1 searches for a bicycle frame supplier and selects E2;</td>
</tr>
<tr>
<td>A contract is signed and VE1 is created; (A)</td>
</tr>
<tr>
<td>E1 generates a purchase order and sends the corresponding EDIFACT message to E2; (B)</td>
</tr>
<tr>
<td>Order acceptance is generated by E2 (e.g. via PPC based on contract rules) and sent to E1; (C)</td>
</tr>
<tr>
<td>PPC at E2 generates a production order according to E1 order; (D)</td>
</tr>
<tr>
<td>E1 monitors and supervises the VE orders (J);</td>
</tr>
<tr>
<td>E1 can request “production follow-up” bulletins from E2 (K);</td>
</tr>
<tr>
<td>E2 delivers the product and delivery record to E1; (E, F)</td>
</tr>
<tr>
<td>E1 receives and inspects the frames;</td>
</tr>
<tr>
<td>E1 sends a “reception report” to E2 if any manufacturing problem is found in the frames; (H)</td>
</tr>
<tr>
<td>E2 (PPC system) generates an invoice and sends it to E1; (G)</td>
</tr>
</tbody>
</table>

Table 3.1: Main enterprise interactions to support Scenario Case 1.

### 3.2.2 Scenario Case 2 – VE Order Related Functionalities

This scenario case illustrates some of the interactions between Enterprises 2 and 3 within the context of VE2, also focusing on order-related functionalities. In this case, E2 is looking for a supplier of PVC resin and signs a contract with E3. Based on this contract, a long-range collaboration is established in which the supply of this material will be delivered on request. For instance, E2 can dynamically issue PVC resin orders for E3. After E3 has accepted and confirmed a given requested order from E2, it can initiate the internal production of the resin, and finally deliver the material to E2.
3.2. Industrial Manufacturing VE Scenarios

An example of the sequence of interactions among these companies is described in Table 3.2 below (see also Figure 3.3).

<table>
<thead>
<tr>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2 searches a PVC resin supplier;</td>
</tr>
<tr>
<td>E2 identifies E3 as a resin supplier;</td>
</tr>
<tr>
<td>Contract is signed and a long-range supply is planned; (A)</td>
</tr>
<tr>
<td>Suppliers data is configured in E2;</td>
</tr>
<tr>
<td>E2 generates a programmed order and sends it to E3; (B)</td>
</tr>
<tr>
<td>E3 sends notification of order acceptance to E2 (according to the contract rules); (C)</td>
</tr>
<tr>
<td>E2 monitors and coordinates the VE orders (J);</td>
</tr>
<tr>
<td>Production is started by E3;</td>
</tr>
<tr>
<td>E2 can request “production follow-up” bulletins; (E)</td>
</tr>
<tr>
<td>E3 delivers both resin and invoice to E2; (F, G)</td>
</tr>
<tr>
<td>E3 (PPC system) generates an invoice and sends it to E2;</td>
</tr>
<tr>
<td>E2 receives and inspects the product;</td>
</tr>
<tr>
<td>E2 sends a “reception report” to E3 if any manufacturing problem is found in the product; (H)</td>
</tr>
<tr>
<td>E2 sends the invoice for payment; (I)</td>
</tr>
</tbody>
</table>

Table 3.2: Main enterprise interactions for commercial order support within Scenario Case 2.

3.2.3 Scenario Case 3 – Product Design Negotiation

This third scenario case focuses on the interactions that take place between Enterprises 2 and 4 in order to agree on the design of the pedal mould that E2 needs to order from E4. In this case, E2 designs a proposed pedal mould using specific CAD tools (e.g. AutoCAD), a Product Data Management (PDM) editor, and local visualization tools with markup facilities (e.g. IntraVISION). The related technical product information is sent to E4 using an EDI/CONDRA message to exchange the STEP information associated with the pedal mould design. The EDI/CONDRA messages allow the referencing of product data files, containing CAD descriptions, together with pertaining trade data like price, shipping address, part number, etc. When the message is received and interpreted at E4, the pedal design is studied and revised
using similar tools. For instance, new versions of the PDM files can be generated, and the graphical design can be red-lined or marked-up to reflect suggested changes. Then, another CONDRA message is sent from E4 to E2 with the proposed changes, and the design is finally agreed among both companies. After this, the new pedal mould can be ordered from E4 as any other product following a sequence of steps similar to scenario cases 1 and 2. The described scenario is detailed out in table Table 3.3 (see Figure 3.4).

<table>
<thead>
<tr>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2 searches a mould maker supplier;</td>
</tr>
<tr>
<td>E2 identifies E4 as a pedal mould supplier;</td>
</tr>
<tr>
<td>E2 sends proposed pedal design to E4 using an EDI/CONDRA message; (C)</td>
</tr>
<tr>
<td>E4 and E2 interactively analyze the product design, exchanging STEP models; (E)</td>
</tr>
<tr>
<td>E2 agrees with the design and sends a “design acceptance” to E4; (F)</td>
</tr>
<tr>
<td>E4 sends proposal for the mould production; (G)</td>
</tr>
<tr>
<td>E2 evaluates the proposal and sends a “confirmation order” to E4; (H)</td>
</tr>
<tr>
<td>Contract is signed and sporadic supply is planned; (A)</td>
</tr>
<tr>
<td>Suppliers and product data are configured in PPC;</td>
</tr>
<tr>
<td>E2 generates and order entry and sends it to E4; (B)</td>
</tr>
<tr>
<td>E4 sends the order and the project confirmation; (D)</td>
</tr>
<tr>
<td>E2 manages the VE orders; (N)</td>
</tr>
<tr>
<td>E2 can request “production follow up bulletins” from E4; (I)</td>
</tr>
<tr>
<td>E4 sends all exportation (process) and releases documentation with product; (J) (K)</td>
</tr>
<tr>
<td>E2 receives and inspects the product;</td>
</tr>
<tr>
<td>E2 sends a “reception report” to E4 if any manufacturing problem is found in the product; (L)</td>
</tr>
<tr>
<td>E2 sends the invoice for payment; (M)</td>
</tr>
</tbody>
</table>

Table 3.3: Main interactions among VE members to support product design negotiation.

3.2.4 Scenario Case 4 – VE Monitoring and Coordination

In this scenario case, the emphasis lies on the VE monitoring and coordination functionalities, which are crucial for the proper operation of the VE as a whole entity.
As mentioned in Chapter 1, "VE Coordinator" is one of the roles that a VE member enterprise can assume within a VE. The VE Coordinator is responsible, among other duties, for the periodic control and monitoring of the tasks assigned to VE Member Enterprises. For instance, as described in the previous VE scenario cases, the VE coordinator enterprises (e.g., E1 in VE1 and E2 in VE2) "monitor and coordinate" the VE orders that are being fulfilled by themselves and other partners. This interaction was indicated as a single step in the scenario cases, but it actually involves the execution of several other tasks. Namely, based on the specific production information gathered from the VE partners, the VE Coordinator can be able to detect potential conflicts in the execution of the tasks assigned to different VE members, and even to analyze and propose solution approaches.

For example, in this scenario case, the coordinator of VE2 (E2) needs to monitor and coordinate the tasks associated with the VE orders that have been sent to Enterprises E3 (PVC resin) and E4 (pedal mould), as well as the tasks related to its own local production (bicycle pedal production for E1). Please notice that E2 needs to retrieve up-to-date information from E3 and E4, but that other VE partners, such as E1, should not be able to have access to this internal information from E3 and E4. Since this internal information is not required for E1, these enterprises have decided to protect it and to avoid the exposure of protected information to eventual competitors.

As mentioned in Chapter 1, the access rights on local information must be agreed, defined, and distributed among the VE members in the form of VE contract clauses during the VE creation phase (although, they can be modified later). However, in this particular scenario, the focus is put only on the VE operation phase. A sample sequence of interactions among Enterprises E2, E3 and E4 regarding the monitoring and supervision tasks within VE2, is provided in Table 3.4 (see also Figure 3.5).

Please also notice that most of the production information associated to the global VE goals is fully distributed and managed independently at the remote sites of the VE members accomplishing particular tasks. Furthermore, in order the carry out the monitoring of VE tasks (e.g. task status, progress, etc.), the VE coordinator
Steps

<table>
<thead>
<tr>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPCs notify VE coordinator module about availability of new production-related data; (A)(B)(C)</td>
</tr>
<tr>
<td>VE Coordinator receives notifications and requests up-to-date production information about the VE orders from all producer enterprises in VE2 (including local production information); (D)(E)(F)</td>
</tr>
<tr>
<td>VCL receives coordinator request and determines if information must be retrieved online from ERP/PPC (PPC pull approach) or if the information is already available in VCL (PPC push approach); (G)(H)(I)</td>
</tr>
<tr>
<td>If necessary, up-to-date information must be pulled out from ERP/PPC system into VCL; (J)(K)(L)</td>
</tr>
<tr>
<td>Remote VCL must verify access rights on local information being requested from other enterprises; (M)(N)</td>
</tr>
<tr>
<td>Remotely stored information must be sent back to VCL of VE coordinator; (O)(P)</td>
</tr>
<tr>
<td>VCL merges the distributed information gathered from other VE nodes and returns it to VE Coordinator (e.g. an end-user or client application) to support decision making processes; (Q)</td>
</tr>
</tbody>
</table>

Table 3.4: Main VE member interactions to support VE coordination and monitoring.

must rely on a mechanism able to retrieve the most recently updated data available at one or more given VE partner sites, when necessary. Otherwise, it would not be possible for the VE coordinator to detect production delays with enough time to trigger re-scheduled production plans.

Besides the functionality to support VE task monitoring and coordination, this scenario case also introduced the need to support proper access rights and visibility levels among VE members. In fact, this requirement must be thoroughly satisfied during the complete VE life cycle (including the VE functionalities described in the previous scenario cases), but it becomes especially evident when several different VE member roles are defined within a VE. Namely, depending on the VE partner roles, different information items must be made accessible for other nodes. For example, the VE coordinator needs to gather some production information from a given VE

Figure 3.5: VE Scenario Case 4 – VE monitoring and coordination.
member; however, the requested information may need to be withheld from other partners due to security reasons or because it is simply irrelevant to other partners. This situation will be further analyzed in Section 3.5 of this chapter.

3.3 Information Modeling Requirements for VEs

Considering the main functionalities addressed by a VCL focusing on the industrial manufacturing scenarios described in Section 3.2 and the studies performed by the author in the related R&D projects, the following information management requirements are of special interest and need specific support by the corresponding Distributed Information Management System (DIMS) component of the VCL:

- Exchange of technical product data (e.g. STEP files for product data).
- Exchange of commercial orders data (e.g. EDI messages).
- Enforcement of autonomy and privacy of pre-existing industrial SMEs. Namely, VE member enterprises must be able to individually decide which part of their local information must be made available to other enterprises based on VE agreements or contracts.
- Monitoring of distributed production order information among VE nodes.
- Management of the necessary information to support the proper coordination and execution of services offered by different VE platform components. Specific information is needed to be handled for the support of internal VCL coordination mechanisms provided for instance by a workflow management component.
- Proper integration of information coming from the ERP/PPC and other internal company systems (e.g. other systems that may also need to interact with the VE layer, such as for example product design and engineering systems).
- Cooperative information management, regarding the information that a given enterprise decides to make available to other specific nodes in the VE network.
- Management of VE configuration data, concerning the definition of the VEs and the parameters for the operation of the internal VCL components.

It must be mentioned that although this list of requirements is quite varied in nature and involves functionalities that correspond to specialized VE Cooperation Layer modules (e.g. specialized STEP and EDI modules), the VE information management component (i.e. DIMS) must also represent and handle a part of the information associated with these functionalities.

These and other preliminary information management requirements are clearly identifiable based on the described VE reference scenarios and a general description of a VE platform architecture (other challenging enterprise collaborative scenarios are described in [103]). However, in order to achieve a complete identification of
the information modeling and data access requirements for the corresponding information management component, a systematic analysis of the VE cooperation layer functionality and application domain is mandatory.

Due to the potential complexity of the virtual enterprise domain and the diversity of the 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 tasks. Thus, in order to facilitate this analysis, the following step-wise approach is applied:

1. As a first step, the study domain is divided into three focus areas that represent the main kinds of interactions and exchange of information between different elements of the VCL reference architecture.

2. As a second step, a comprehensive classification and structuring of the data needs to be performed. This analysis will allow the identification of a wide variety of categories and types of information that need to be handled at every node, and a large set of requirements for information exchange among the nodes.

3. As a third step towards the modeling of this diverse information for every enterprise, the information is divided into three categories of Self-, Acquaintance- and VE-related information (with corresponding intuitive meanings).

In the following subsections, these steps are described in more details.

3.3.1 Analysis of Information Focus Areas

In this section, the three focus areas for the analysis of the DIMS information management requirements are introduced and briefly described (see Figure 3.6):

**Focus Area 1 - FA1**, focuses on the Internal VCL support. Namely, it encompasses the management of the information that is exchanged among different components inside the VCL. Here, this focus area addresses all the main components that together provide the coherent cooperation layer for every node in the virtual enterprise. For instance, to support the EDI module of the VCL with its management of information, the identified set of data elements includes, among others: partner enterprise identification information, partner application-dependent information, EDI subset group information, order information, and order line information.

**Focus Area 2 - FA2**, focuses on the PPC-VCL interoperation support, addressing the exchange of information between a ERP/PPC (or other enterprise internal modules) and its VCL. Some of the information that needs to be managed within the interoperation layer involves a part of the information stored in the ERP/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 ERP/PPC information that either needs to be used by
interoperability layer, or needs to be exchanged with other enterprises. For example, part of the information that has collected and analyzed for a particular PPC includes: the order production planning and scheduling, contract information, supervision clauses for specific orders, and machine utilization.

**Focus Area 3 - FA3**, focuses on the interoperability support among enterprise nodes. It addresses the information sharing and interoperability between every two enterprises in the VE; namely the information exchange and data access/retrieval queries between the two VCLs of two given 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 identified information that needs to be exchanged may include: enterprises profile information, product information, contract information, information related to VE cooperation and coordination, client orders, orders status reports, order execution diagrams, and unformatted information, among many other possible information classes.

The proper support for these three focus areas in turn constitutes the main goal of the design of the DIMS for the VE Cooperation Layer. A more detailed description, with several data examples for each focus area, follows next.

### 3.3.2 VE Information Categorization

The study of the VE scenarios and the requirement analysis for the DIMS define a rich set of sample information that needs to be managed by this component. Before any approach to model the collected data however, first that data must be properly classified and structured. The proper approach for modeling of information is thus to follow a *stepwise classification approach*. In this section, six categories (classes) of data are defined, which primarily represent all the collected information from the three focus areas. This classification is further used as the base for the information

---

**Figure 3.6: Focus areas for information management requirement analysis.**
modeling approach described in the next section. Following there is a list of these categories (see also [71, 12]):

1. Public enterprise information: information accessible by all the nodes in the network (i.e. the network formed by all the enterprises enabled with a VE Cooperation Layer; see Chapter 1). For example, 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 DIMS component is desired.

2. Enterprise information for members of a virtual enterprise: a subset of the data stored in the internal enterprise modules that needs to be accessed by other members of that VE. Namely, for every virtual enterprise that a node is involved in, a subset of the data stored in the PPC and other internal ERP systems 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 DIMS module.

3. Enterprise information for a "sister" company: part of the data accessible to other enterprises with a higher degree of trust than regular partners of this enterprise. In this case, a sister company represents an enterprise that has a high degree of access to the internal information of a given company, due to for instance, a solid mutual trust that has been built during long-term collaboration relationships. Thus, similar to the situation described for VE members, every sister company of an enterprise also needs authorized access to a "bigger part" of the data available at the enterprise.

4. Directory of enterprise profiles: directory of some information describing enterprises that represent actual or potential VE partners. At every enterprise, it is desirable to have access to a directory with some information describing some 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 ERP/PPC to make new orders, search and select VE partners, etc.

5. Data from other VE members: 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 "Enterprise Information for members of a virtual enterprise" as described above and is used by internal enterprise applications such as the ERP/PPC, to for instance check the status of the processing of certain order in the VE. Shared data from other members to support team work in areas such as concurrent engineering represents another example of this category.

6. Data from another sister company: similar to the situation described for the data availability to other authorized VE members, there is a need to access a
part of the data available at other sister companies through the corresponding authorized rights.

At each of the above points, there is a need to guarantee that: current up-to-date data can be retrieved through the VLC component, and that the proper authorization of the VE members accessing such data is properly checked by the VCL. Please also note that a VE information management approach based on provision of access to duplicated information stored at every node would not be suitable to cope with certain VE scenarios in which sensitive data is frequently updated. For instance, in order to support the periodic monitoring of the internal production status for a given purchase order, an on-line information access mechanism is more convenient than a mechanism based on duplication and actualization of local production information.

Once the tailored exchange of information, as defined in items 1 to 6 above, is supported by the DIMS component at the VCL of every node, then the users at a node can request information from other nodes in the network. In order 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 (this issue will be addressed in Section 3.4).

3.3.3 VE Information Modeling Approach

Taking into account the preliminary classification of information described above (which is actually based on the Focus Areas analysis), the following basic reference model is defined for all the information classes treated along the main VE life cycle activities. It is important to carefully define this reference model, since with a proper strong base for information modeling, both the current and the potential future requirements can be structured and supported. Following these general assumptions, the reference model is composed of three basic categories of information aimed at the classification and modeling of VE information. Namely, specific categories are defined for: Self information, Acquaintance information, and Virtual Enterprise information. These categories (and their subsequent subcategories) are described below and illustrated in Figure 3.7.

Self Information category

This category corresponds to the part of the enterprise information which is managed and controlled at every node in the network in an independent way, and that may not be directly related to the concept of VE (for example, the information that is needed to be managed by the VCL components for their internal operation). Part of this information comes from the internal ERP/PPC system, in which case the PPC system is responsible for keeping that information up-to-date. Also, in order to properly ensure the information security at every node, different access level mechanisms must be defined. Through these mechanisms, this information can then be shared with other nodes. Here, three further sub-categories are defined for the Self information for every node:
i. Self-public information: for instance, 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).

ii. Self-restricted information: 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 a
certain way. This particular information can be very diverse: text documents,
product descriptions, market analysis, commercialization strategies, inventory
information, customers, etc. Access to every piece of this information is only
possible by authorized users.

iii. Self-private information: this type of information includes the data, which is
needed to support the regular operation of the VE cooperation layer itself, 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.

Acquaintance Information category

The acquaintance information is obtained by one enterprise from other remote en-
terprises to serve different internal purposes. For instance, this information provides
a general description of other enterprises, necessary for making a local directory of
products and services available in the network. For this category, the correspond-
ing access level mechanisms of other nodes ensure their information security. The
following divisions are defined for the acquaintance information:

i. Acquaintance-public information: 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.

Figure 3.7: Classification of VE information supported by different schema types.
ii. Acquaintance-restricted information: 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.

Virtual Enterprise Information category

The VE Enterprise Information refers to all that information which is related to the actual involvement of a particular enterprise node in a given VE. Here, this information is systematically defined with the corresponding self and acquaintance sub-categories. Therefore, the VE information is composed of the following categories:

i. 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 is further divided in:

- VE-Self-public information: information which this enterprise makes available to all the members in the VE. This information can include product information, order descriptions, VE customer 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: 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: production orders, internal production plans and schedules, enterprise order status, inventory, resource management, order descriptions, etc.

- VE-Self-private information: information regarding the participation of this enterprise in a particular VE, that is stored only for internal purposes, and is not shared with any other node. This information includes the configuration of the interoperation layer components for a particular VE, internal confidential order information (e.g. internal status and progress), domestic planning and scheduling, local coordination support information, some VE coordination support information, etc.

ii. 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: information acquired from other nodes that have defined this information as their VE-Self-public.
- VE-Acquaintance-restricted: information acquired from other nodes in the VE through their VE-Self-restricted information access.

iii. VE Messages: a basic model for interaction and collaboration between the members of a VE can be supported by the sending and reception of "VE messages". Namely, VE members need to exchange asynchronous messages during negotiation or re-negotiation of VE contract clauses. These messages can be also represented and stored in the VE information management system, including for instance: VE-Restricted Messages (sent to a specific set of partners among a VE) or VE-Public Messages for all the partners in a VE.

iv. VE n-lateral workspaces (VE-Workspaces): this point addresses 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. However, this requirement is more relevant in other sectors such as concurrent engineering, than in general industrial manufacturing VE scenarios.

In the next section, the functionalities (operations) that need to be provided by DIMS to support the global VE operation are identified.

### 3.4 Functional Requirements for VE Information Management

Besides the identification of required data elements and structures for the VE environment, the identification of the operations that need to be performed on the data, is the next important issue. For this purpose, the following sub-tasks are necessary as a general strategy to achieve the design of a set of DIMS query/update operations to support all the possible requirements:

1. Identification of the general DIMS operation requirements and necessary design considerations.

2. Analysis of the general information management query/update functionalities required by the VCL modules.

3. General identification and specification of the set of required DIMS access (retrieve/update) commands and/or service functions.

These three main steps are described in the following sections. The first point addresses the identification of certain aspects or guidelines, which are either desirable or mandatory for the design of the DIMS operations. The second point involves a careful study of the information management functionality that needs to be provided for every module of the interoperation layer. At the third point, after the identification of the general service functions for the VCL modules, a general design and specification of the operations to support these functionalities must be outlined.
3.4.1 General Requirements for DIMS Functionalities

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

- Support for the fundamental functionalities of the VE infrastructure. The DIMS operations must provide information management facilities to support all the required functionalities of the VE support platform, as mentioned previously in Section 3.1.

- Compatibility with the reference VCL architecture. The interface between DIMS and the other VCL modules must be in strict compliance with the general architecture design. Some architectural aspects and division of roles among different modules can have a strong influence on the DIMS operations design and its latter implementation. For instance, the VE cooperation layer architecture may define the specific types and interoperation mechanism that the information management system must follow in order to interact with other components within the VE layer.

- 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.

- Flexibility and configurability. Furthermore, in order to satisfy the wide variety of information management requirements of the different nodes in the VE, the corresponding VE Information Management System must be a highly flexible and configurable environment. Namely, the system should be scalable and new information management functionalities should be easily adapted and made available to other layer components.

- Access to up-to-date distributed information. The DIMS must provide the necessary functionalities to allow seamless access to VE distributed data that is physically located at different VE nodes. The provided mechanisms need to be implemented in such a way that up-to-date information can always be retrieved if required by the query-issuer node.

- Support for visibility access levels on specific information. Based on the VE information modeling task described in Section 3.3.3, a mechanism to define and reinforce 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 DIMS support operations.

In addition to the considerations above, the possibility that the DIMS component may be built on top of an already existing DBMS with its own query/update language support (e.g. the SQL) must be fully exploited, trying to reduce the implementation efforts and the "reinvention of the wheel".
Another list of general information management requirements related to VE support has also been identified in [133]. However, in that work the requirement analysis focuses mostly on the area of engineering sector, and the need to support and reinforce fine-grained access rights among VE members playing different roles is not analyzed.

### 3.4.2 Analysis of required DIMS query/update Functionalities

Having defined the general design considerations described in the previous step, it is possible to proceed to analyze the required DIMS query/update functionalities. For this task, an approach similar to the analysis of the information models is applied as described next.

As mentioned previously in this chapter, the first step towards the modeling of the information management requirements is the definition of three focus areas. Similarly, and in the same way, the analysis of the required DIMS query/update functionalities can therefore start with an in-depth look to these areas, but now concentrating on the required information access/update mechanisms instead of on the structure of the exchanged information. Consequently, the access/update requirements, i.e. information management queries that need to be supported within the three areas can be defined as follows:

- In Focus Area 1: queries needed by different components inside every interoperation layer. For every component inside the VCL, 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 systems can access it.

- In Focus Area 2: queries that need to be supported between the company's internal modules (e.g. ERP/PPC system) and its interoperation layer. As an example, legacy systems need to store/read the VE-related order information that is requested by the VE coordinator who needs to monitor it. Furthermore, they need to store/read the order information for purchasing products from other enterprises.

- In Focus Area 3: queries needed to share information, and support the interoperation between every two nodes in the virtual enterprise. For instance, in this case, any communication service for the interoperation layer needs to store/read inter-VE member messages. Update/read operations on the information to encrypt and deliver the VE messages are also required. Furthermore, in order to enable VE coordination, it is necessary 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, etc.

After the identification of these general focus areas, the next step is the analysis of the specific requirements for every interoperation layer module to clearly identify their particular query/update operations needs. This issue is the subject addressed in the paragraphs below.
3.4.3 General Specification of DIMS Access Functionalities

Based on the study and analysis of the three focus areas, the DIMS query/update operations must be provided through the following mechanisms:

- Local SQL queries.
- Generic data management functions.
- High-level functions for specific modules of the VCL.
- Generic distributed/federated queries.

These categories represent the services that the DIMS makes available to other VCL modules and also to the ERP/PPC (through specific VCL services). Each of these mechanisms is described in the following subsections. In addition, for every mechanism some example DIMS functions are provided (the complete list of the identified functionalities is described in [68]).

Local SQL queries

This mechanism makes some functions available to allow both the ERP/PPC and other VCL modules to issue a generic data management query to be evaluated locally at the DIMS of the node. These functions represent the minimal DIMS access/query support for the interoperation layer components. For example, the queries may be formulated in a language such as SQL. The definition of this DIMS function could be:

- Result DIMS_LocalSQLQuery (query)

Where Result represents some kind of record set data type. This functionality may be directly supported by making available ODBC facilities to client modules.

Generic data management functions

The functions described in this section aim at providing access to local data through some form of generic mechanisms, which encapsulate and internally represent query commands. Using this facility, the applications do not need to deal with database language specificities, and may just provide the necessary information through these access functions. Furthermore, 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. For example, in certain application domains specific information access methods are sometimes necessary to handle definitions complying with a specific industrial standard such as the ISO Express language. In the case of DIMS, the functions that are under this category, include:

- Access protocol functions. The access protocol functions involve connecting and disconnecting functions to the DIMS, as well as local transaction support. As an example of these two functions, the following definitions are given:
- DIMSConnection DIMS_Connect (module_id, password)
- Result DIMS_StartTransaction ()

- General data access functions. The general data access functions basically cover the select, update and delete commands on the table records. For instance, the following function would return all the records for a given table:

- Result DIMS_SelectAll (table_name)

- Meta data access functions. The meta data access functions are used to perform operations on the database schema. These functions can only be used during the “development of the VCL”. Namely, most of the meta-data access functions are used only by the DIMS designers for the DIMS development purposes and are not made available to the other modules. These functions include operations for database creation/deletion and schema definition. For example:

- Result DIMS_Create_Table (table_name, desc)

- Other operations. Besides the functions described so far, other miscellaneous functions may be defined. For example, the following function can be used to get the next integer number in an internally managed number sequence:

- Result DIMS_GetSequenceNumber (sequence_id)

**High-level functions for specific modules of the VCL**

The support for certain specific advanced database-oriented functions can be extremely convenient to facilitate the work of the other VCL modules. Through the specified functions for every module of the VCL, sequences of query statements and generic data management 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. For instance, the VE monitoring and coordination module (e.g. represented by the DBPMS module in PRODNET) needs functions to retrieve specific information related to the requested (purchase) orders, requested order items, and internal production orders, that have been assigned to a given set of the VE partners. These kinds of high-level functions involve the retrieval of distributed information that is physically stored along the VE network.

In order to identify these functions, the specific requirements of each VCL module need to be studied. As an example, the following operations are respectively required for the PCI, STEP, and ACF modules of the VE cooperation layer in PRODNET (see Chapter 1):

- Result DIMS_GetPublicKey (VE_id, VE.Partner, PublicKey): gets the public key of the specified node for the encryption algorithm used by the PCI module, for a given partner in a given VE.
3.4. Functional Requirements for VE Information Management

- Function DIMS.StoreBlob (blob): stores the data as a “blob”, and DIMS provides an identification for this blob (the blob-id).

- Function DIMS.PutAuditingInformation (audit-info): stores the requests and acknowledgements of control actions of a distributed production process plan.

Please notice that besides the functionalities described in this section, many other functions can be identified for each module of the VCL (see [68]).

Generic distributed/federated queries

The VE reference scenario cases introduced in Section 3.2, as well as the VE information categorization described in Section 3.3, introduce some challenging requirements regarding the integrated access to local and distributed information at a given VE node. Namely, the distributed VE information from other remote partners must be transparently integrated with local information at every node and made available for VCL modules operation. Therefore, the general DIMS queries described in this section are queries that are applied to the VCL integrated schema, which provides a comprehensive view of all the data that is available to be access at a given node, including the distributed VE information that is made available by other enterprises. These distributed/federated queries may be stated for instance, in the SQL language. The specific format that has been identified for these queries as a result of the requirement analysis phase is described in the next paragraphs.

In order to request these DIMS operations, the sender of the query can specify two parameters (besides the query itself) if it wishes to determine the specific “context” in which the query must be evaluated. These parameters are: the VE identification, and the VE partner identification.

In a typical federated query, the sender specifies only the VE identification as the context for the query. However, if the sender wishes, it can also direct this query to specific nodes. If one or both of these parameters are specified, they will be analyzed in order to distribute the original SQL query among the relevant set of partner nodes. Notice that in the case that the two parameters are explicitly provided, there is no need to parse and decompose complicated SQL queries into subqueries at the sender DIMS. Namely, the original SQL queries are sent directly to the specified VE partner nodes without parsing. In this case, the detailed interpretation and parsing of the queries will be done locally by the internal database engine at the receiving node. The format for the specification of queries for distributed VE information access is represented as follows:

- Function DIMS.DistributedQuery(VE_id, VE_Partner_id, query)

If both the VE and VE partner are explicitly given, the specified query will be sent to the given partner (VE_Partner_id) and evaluated in the context of the specified VE (VE_id). However, either one or both of the VE_id and the VE_Partner_id parameters may contain a “star value”, which means that this parameter must be interpreted as
a wild card. The possible combinations of these two parameters, when containing the star are analyzed and described below (see also Figure 3.8):

- **Result DIMS.DistributedQuery (‘*’, VE.Partner.id, query).** In this case, the query will be evaluated in the context of all the VE s associated to the given partner.

- **Result DIMS.DistributedQuery (VE.id, ‘*’, query).** The query will be evaluated against all the partners participating in the given VE.

- **Result DIMS.DistributedQuery (‘*’, ‘*’, query).** For this case, the query must be sent to all the partners participating in all the VE s in which this node is involved.

The formulation of federated queries as specified above, supports the required functional specificities of the application domain (in this case, the Virtual Enterprise Paradigm), while reinforcing the preservation of relevant context through the provision of the VE and VE partner identification by the DIMS user. Namely, all queries related to VE information retrieval among VE partners can be stated through the general distributed DIMS queries defined in this section.

Finally, the use of the generic federated queries described in this section, will be further clarified in the next section with some examples regarding VE coordination and monitoring tasks.

Figure 3.8: Examples of processing of different DIMS federated queries.
3.5 Federated Information Management for VEs

Some of the information modeling and data access requirements that have been identified for DIMS in the previous sections point out the need for an advanced information retrieval mechanism, through which VE members are able to issue queries involving access to distributed data located at other VE nodes. The support for the information categories introduced in Section 3.3.2, as well as the generic distributed/federated queries described in Section 3.4.3, are examples of DIMS requirements that involve this distributed data access mechanism.

In the support for this mechanism in DIMS, the main issues to be considered are:

- The VE members must be able to have seamless access to up-to-date distributed VE information through the on-line execution of distributed queries. This means that the VE partners should not be concerned about specific details regarding the physical location of the distributed data in the VE network. The physical address and internal organization of the data being stored in another VE node must remain hidden to end-user and applications at the query issuer node. Furthermore, the evaluation of the distributed queries must return the most up-to-date data from the VE nodes, when required.

- In all the tasks involved in the retrieval of distributed information, the autonomy of each VE node must be maintained and reinforced. Namely, considering the classification of VE information carried out in Section 3.3.3, every VE node must be able to determine which information is made available for access to other specific nodes, and which part of the information must be kept protected from external access. The concept of autonomy must be reinforced through the careful definition of fine-grained access rights and visibility levels on local information for other VE partners nodes. These access rights will be taken into account by the distributed query processing tasks during the evaluation of individual subqueries at each VE node.

In order to illustrate the application of these points, let us consider a specific case of VE coordination and monitoring within the context of a VCL architecture in the industrial manufacturing sector. Please remember that in general, every node in an industrial VE can play different roles, such as supplier, producer, or final consumer of good or services [7]. Furthermore, one node will assume the role of VE coordinator in order to monitor and coordinate the independent tasks of the other nodes (VE partners) towards the accomplishment of the global VE goals.

Please also notice that to achieve this important task, the VE coordinator needs to have privileged access to certain proprietary data that is located at different nodes (the local data related to subtasks of each enterprise) in order to monitor the status of their tasks execution. 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: query decomposition, issuing several remote subqueries to the involved enterprise nodes, collection of partial results from these nodes, and generation of the final result to the coordinator. However, from
the coordinator perspective, the distributed data access should take place without being concerned about the data location over the VE network. This means that the information must be integrated in some way in order to allow this transparent access to distributed data (point 1 above). The distributed information for the VE coordinator can be gathered for instance, through the use of DIMS functions such as the distributed/federated queries described in Section 3.4.3.

Furthermore, the VE integrated information that must be made available for access to the VE coordinator, actually corresponds with the union of all the data definitions that each VE member is willing (or is obliged) to share and exchange with the VE coordinator. However, this information should not be available for access to the other regular VE partners since on one hand, it is probably not relevant to their role as regular partners, and on the other hand, it may contain sensitive data that should not be accessible for other enterprises (see Section 3.2.4). Consequently, each node must carefully decide which part of the information must be made available to which specific VE partner. In other words, it becomes clear that there is a strong need for a mechanism to support and reinforce the autonomy of every individual VE node (point 2 above).

In summary, we can conclude that every enterprise must share and exchange a part of its information available at its VCL, with other enterprises. Similarly and at the same time, some information from other nodes must be made accessible in order to be acquired and integrated with the local information of this enterprise. However, the visibility levels of the enterprise information from external nodes, must be carefully determined by the node data (mostly based on other enterprise’s role in the VE), in order to ensure both its autonomy and information privacy. In addition, proper sharing and exchange of up-to-date information must be supported when an enterprise is involved in more than one VE.

In order to support these particular requirements, and the proper interoperation among VE interoperation layers in different nodes, a federated database architecture has been designed and developed for the DIMS module of VCL [71, 12, 44].

In the next sections, two preliminary examples are provided to illustrate some of the different features and facilities of the federated architecture, and how it matches the requirements of the management of information in VEs. In the first case, the approach presented in Section 3.5.1 is based on the PEER system and its fully federated information management architecture, which has been developed at the CO-IM group of the University of Amsterdam but its development is not a part of this dissertation. This general federated approach covers some of the basic information management requirements for VE support, but it still needs to be substantially extended and adapted to properly satisfy the particular VE paradigm requirements. In fact, in the proposed DIMS design, the general federated architecture features and principles have been properly extended and tailored for this purpose. Therefore, Section 3.5.2 includes an initial example of data integration for VE support based on the DIMS federated approach. More examples and details about the application of the DIMS federated architecture to different VE scenarios will be addressed in Chapter 4.
3.5.1 Application of PEER Architecture for VE Support

In this section, the PEER federated database system is described in details, and its potential application to support some VE information management requirements is demonstrated.

The PEER Federated Database Architecture

Federated database systems are multidatabase 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. Conversely, 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 data integration 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.

In this context, PEER is a fully federated and 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 [169, 16, 156]. Namely, it supports the sharing and exchange of information among cooperating autonomous and heterogeneous nodes without the need for any centralization and data-redundancy.

Every node is represented by several schemas (see Figure 3.9): a local schema (LOC), several import schemas (IMPs), several export schemas (EXPs) and an integrated schema (INT) [14]. The local schema is the schema that models the data stored locally. The various import schemas model the information that is accessible from other databases. An export schema models some information that a node wishes to make accessible to other nodes (usually, a node defines several export schemas). The integrated schema presents a coherent view on all accessible local and remote information. The integrated schema can define a particular classification of objects which are classified differently by the schemas in other nodes.

More specifically, the local schema is the private schema defined on the physical data in a given node. Derived from the local schema there are export schemas, each one defining a particular view on some local objects. Export schemas restructure and represent several related concepts in one schema. For every export schema, the exporter node manages the information on nodes who can access it, by keeping their node-id, their access rights on the exported information, and the agreed schema modification conditions to notify the nodes who use this export schema of its changes. To obtain access to data in another node's export schema, a node has to input it as import schema. An import schema in one node is an export schema in another node.

Originally, a node’s integrated schema is derived from its local schema and various import schemas. In later stages of integration, instead of the local schema, the previous integrated schema will be used as a base. At the level of the integrated schema, the physical distribution of information becomes hidden, and the contributing nodes are no longer directly visible to the end user. Different nodes can establish different
correspondences between their own schema and other nodes' schemas, and thus there is no single global schema for the network.

The functionality of the PEER federated system is supported by the specific architecture of each PEER node and its PEER-kernel, and by the existence of the community dictionary in the network. The community dictionary is the "source of information" within the network and can be consulted at any time by other nodes. Its function is to provide up-to-date information on all the nodes in the network. It contains the network addresses of the active nodes and their current state. For every node, it also stores its export schemas and the specific access rights and schema modification rights that the node supports. The dictionary can also be used as the general store for other static information that concerns the entire community such as the objects' name-tables. Although, in the architectural design of PEER, the community dictionary is represented as a separate node to be accessed when needed, any other node in the community can keep local copies of all or parts of its information.

To this federated architecture, some extensions can be added to support a real co-working environment where the remote object references are handled together with the local objects [169, 16]. To define the interrelations among the local and remote objects, the local schema needs to be extended. An extended local schema extends the local schema by a specific type called extended local type. Extended local types are derived from a type defined in an import schema and support the interrelation of specific local objects through a remote reference to objects accessible from other nodes. The remote reference is a mapping (attribute) defined on a local type (as its domain) and an extended local type, corresponding to a remote type, as its range. Actually, instances of an extended local type are object-identifiers of the remote objects that are stored locally at the node as references to the real remote objects.

Figure 3.9: PEER federated information management layer.
In general, the schemas for a node are defined using the **PEER Schema Definition and Derivation Language** (SDDL) [14]. SDDL includes facilities for defining schemas, types, and maps, and for specifying the derivation among a derived schema and a number of base schemas using a set of type and map derivation primitives. An integrated type is constructed from other types defined in different nodes by the union, restrict, and subtract primitives. Map integration is accomplished by specifying a map as either a union of other maps, or as a threaded map, or some combination of these two primitives. A prototype implementation of the PEER system is developed using the C language in the UNIX environment, and includes two user interface tools [15]: a Schema Manipulation Tool (SMT), and a Database Browsing Tool (DBT). A client library has also been developed for Windows NT.

In terms of other related approaches, some taxonomies and surveys of architectures for federated database systems are described in [33, 146, 28]. Furthermore, some other systems or conceptual models that are closely related to PEER are included here as a reference:

- **UniSQL/M.** This multidatabase system is a commercially available product, which schema architecture supports the integration of relational and object-oriented schemas [97, 21]. Structural conflicts involved in multidatabase integration can be resolved within the UniSQL/M system through SQL language commands.

- **Pegasus.** Pegasus is a multidatabase prototype that supports the definition of virtual types and functions, where functions represent properties, relationships, or computations [144, 28]. Both data definition and data access operations are expressed using the HOSQL language (Heterogeneous Object SQL). This language is also used to define the Pegasus global schema based on imported types and functions.

- **VHDBS.** The architecture of the VHDBS system aims at the support for cooperative access to distributed heterogeneous databases [172, 171]. VHDBS is based on OMG/CORB A 2.0, and uses an object-oriented data model as a common framework to integrate heterogeneous data. The kernel component of the architecture is a federated management system based on a four-level schema architecture constituted by the local schema, server schema, federated schema, and client schema.

- **VODAK.** This is a distributed multimedia object-oriented database management system addressing the dynamic integration of heterogeneous autonomous information sources [35, 28]. The VODAK data model incorporates all the main elements of an object-oriented model including inheritance, class, and object method definition mechanisms.

- **Disco.** The Distributed Information Search COmponent (Disco) system is a search mediator for retrieving information distributed among heterogeneous data sources within an Internet/Intranet network [155]. It is based on extensions of ODMG-93 and OQL for data modeling and access language specifications.
The two main components of the architecture correspond to a mediator, used to manage and homogenize the metadata information about the different data sources, and a set of wrappers that carry out the execution of access queries on the particular information sources.

**PEER-based Information Management Support for VEs**

As mentioned earlier in this chapter, in order to fully support the complex information management requirements for VE support in industrial VEs, a federated architecture can be used as the base to provide for seamless access to distributed data with proper definition of access rights at each VE node.

In particular, the PEER federated modeling approach described in the previous section can support some of the basic functionalities for an effective information sharing and cooperation among different enterprises. Among the generic PEER capabilities that can be used for this purpose, 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 remote objects shared by other nodes.
3. Different levels of access privileges and information visibility for other nodes can be defined.
4. The physical and logical information distribution among the nodes becomes transparent to the users.

Applying these fundamental capabilities, the PEER federated system can support some of the required features to model the dynamic relationships between the members of a Virtual Enterprise, for example (see also [69, 7]):

- **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 is read on-line and always reflects 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 planning of an enterprise can establish references to orders submitted by one or more different enterprises, therefore representing for example, joint work within a supply chain.

In Figure 3.10, the different information categories that were identified earlier in this chapter and illustrated in a tree structure (see Figure 3.7) are represented, where additionally every leaf node is associated with elements of the PEER federated schema, namely the LOC, IMP, EXP, and INT schemas. 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 creation and dissolution of the virtual enterprises, it is necessary to further extend the general federated architecture approach with many other features, such as for instance:

- Definition of access rights (e.g. through export schemas) among VE member enterprises based on for instance, existing trust relationships, legal contracts, and supervision clauses.

- Provision of high-level functions regarding specific functionalities required to support the VE life-cycle. For example, specific functions are required to properly support VE creation phase and VE monitoring tasks.

Figure 3.10: Classification of VE information supported by different schema types.
- Explicit management of information represented and stored according to standards and models related to the VE application domain, such as EDI, STEP, distributed business processes, etc.

- Definition of a standard interoperability approach through which internal enterprise modules and VCL components can make use of distributed information management functionalities.

- The traditional federated query processing tasks need to be extended to support a high level of flexibility and configurability in order to support different interaction scenarios among both internal and external modules of the VE Cooperation Layer. For this purpose (as will be illustrated in Chapter 4), the use of workflow management can be incorporated in the design and implementation of the DIMS federated query processing tasks within the VCL.

- The federated data exchange must support the required level of security and authenticity in the communication channel used for data exchange among enterprises.

In the specific case of PEER, other extensions that would be required to apply this particular federated system to support different VE scenarios can be found in [7]. For instance, an on-line message passing mechanism between the nodes, to facilitate more dynamic and asynchronous interactions can be incorporated. Also, automatic update notifications may be considered to support the on-line status monitoring of production orders within the VE. In addition, changes in the definition of export schemas may need to be reported to the VE coordinator in some cases.

Taking into account some of these extensions and specific requirements for VE support, in Chapter 4 it will be explained how the generic federated database architecture has been used as a conceptual base in order to design and implement the DIMS to support VE collaborations in the industrial manufacturing sector. Namely, the PEER system is not directly used in the federated approach proposed in this thesis, but a similar federated architecture is properly extended and tailored in the DIMS components according the specific requirements of the VE application domain. The next section provides an example of how the DIMS federated architecture can be used to support the integration and exportation of information within the VE environment.

### 3.5.2 DIMS Federated Data Integration for VE Support

As mentioned previously, the DIMS module follows a federated database approach (influenced by the PEER architecture), in order to allow transparent access to the VE distributed information, with proper support for enterprise node autonomy and access rights. For example, taking as a reference the VE demonstration scenario presented in Section 3.2 (Figure 3.1), the DIMS integration approach for distributed VE data is illustrated in Figure 3.11. This figure shows the DIMS components at each VE member enterprise node (i.e. enterprises E1, E2, E3 and E4) in relation to the two VEs that were introduced in Section 3.2 (i.e. VE1 and VE2). Similarly to the PEER
federated approach, each DIMS component handles different database schemas to represent and manage the local enterprise data ("Local Data"), VE integrated data ("VE Int Data") and VE export data ("VE Exp Data"). Please notice that in this case, the importation of export schemas into the integrated schema is done in an implicit and automatic way since all enterprises share the same integrated schema definition; i.e. there is no need to explicitly handle import schema definitions that would be mapped into the integrated schema.

The DIMS local schema manages the local enterprise data that needs to be shared with other enterprises or that is necessary to support the operation of other VCL components. Internal enterprise modules such as ERP/PPC systems will interact with the DIMS in order to store information in the local schema when necessary. Based on the local schema, enterprises are able to define the export schemas for other VE partners. Therefore, the objective of supporting and reinforcing enterprise autonomy and access rights to local information is achieved. Furthermore, the local node information and the distributed information that is made available to a given node (through export schemas) is integrated into a VE integrated schema. The DIMS provides different query access functionalities that allow end users and client applications to issue queries on this integrated schema, while hiding the fact that the information is actually physically distributed in other nodes. Also, during this access to distributed information, the remote export schemas are used in order to guarantee that the access rights that have been defined at each node are properly considered and reinforced.

For instance, let us consider a DIMS federated data integration example in the context of the VE scenario described in Section 3.2 (see Figure 3.11). In this case, every enterprise must be able to define the access rights on local information for each other partner in each VE in which it is involved. In this definition, the role that each enterprise plays in each VE must be carefully considered. For example, the DIMS supports the following data sharing and exchange situations among the enterprises represented in Figure 3.11:

- Integration of distributed information for VE monitoring and coordination. Please remember that E2 is the coordinator of VE2. Therefore, enterprises E1, E3 and E4 will make available internal production information to E2, which is required for VE monitoring and coordination activities as specified in commonly agreed VE contracts or supervision clauses. This information is defined through the "E2 Exp Schema" of the "VE2 Exp Schema" at each enterprise, and integrated into the "VE Int Schema" at E2. In this way, through the VE integrated schema at E2, the VE coordinator module or end-users at E2 are able to monitor the up-to-date progress of the production orders for the PVC resin at E3, and the pedal mould at E4. Furthermore, since the DIMS provides specific access functions on the VE integrated schema, E2 components do not have to know the details about the physical location of the VE data at the other partners’ nodes.

- Protection of sensitive information among VE partners. Please notice that the regular partners of VE2 may not want to share with other partners, the potentially sensitive information that is made available to E2 (VE coordinator)
in VE2. For this purpose, every enterprise has the possibility to specifically configure different access rights to other VE partners in VE2. For instance, E4 may decide (based on VE agreements) that local internal scheduling information related to the pedal mould production must be made available to E2, but not to E1 and E3, because there is no need for it at all. Thus, E4 may define that the data that is shared through the E1 and E3 export schemas, is null data. Of course, the export schemas may be modified on-line in order to reflect changes on the policy of data sharing with other partners. Namely, at some point of the VE2 operation, E4 may decide to make available a part of its internal production information exclusively to E3, so that E3 may count on some mould production details at E4 in order to better determine the composition of the PVC resin that should be delivered to E2. This exchange of information among partners takes place in exactly the same way as for the VE coordinator. In other words, every VE partner can also have seamless access to other partners’ information through the VE integrated schema of the DIMS, and the corresponding access rights are reinforced in the same manner. In this particular example, E4 will make available the desired mould production information to E3 via the “E3 Exp Schema” within “VE2 Exp Schema”. Changes in this export schema will be immediately detected in the accesses to the “VE2 Int Schema” of E3.

Figure 3.11: Example of federated integration/exportation of VE data.
• Support for visibility access levels within the context of specific VEs. In the DIMS approach, the export schemas are defined for specific partners within specific VEs. For instance, since E2 is involved with E1 in two different VEs, E2 has defined specific export schemas for E1 in the context of VE1 and VE2 (and the same applies for E1). Since E1 is the VE coordinator of VE1, this means that E1 will have a higher visibility level on the internal production of E2 in the case of the bicycle frames order within VE1, but a regular visibility level on the production information for the bicycle pedals that are produced within VE2. Thus, the case when a given enterprise plays different roles in different VEs, and therefore, its visibility levels on the local information are different, is properly handled. Furthermore, since E1 and E2 have not defined any export schema for E3 and E4 in the context of VE1, these latter enterprises will not have any access to VE1 information, which is consistent with the fact that they are not members of VE1 (although they may access information in the context of VE2).

In general, many other scenario cases for federated data sharing and exchange among these VE member enterprises can be supported in a similar way. Moreover, the support for this kind of data sharing and exchange based on the different roles played by the VE partners, would be very difficult to achieve without a federated database architecture approach. The DIMS federated database architecture has proven to provide particularly attractive features for handling some open-issues associated with the management of information in VEs. For example, the fine-grained definition of access rights that can be achieved by defining export schemas based on projections and selections of fields on underlying database definitions, is one of the features that is not well supported by other VE infrastructures. In fact, as was analyzed in Chapter 2, the DIMS federated approach represents the major distinguishing characteristic in relation to the other projects. Furthermore, the DIMS properly addresses the extended functionalities that would be necessary to incorporate to the PEER system in order to directly apply it to support real case VE collaboration scenarios (see Section 3.5.1). The details about the internal DIMS components and mechanisms that support these facilities are described in Chapter 4 of this thesis.

### 3.6 Conclusions

After the extensive analysis of the information management requirements for DIMS, a main conclusion is that to handle the complex requirements set by the VE paradigm, an advanced federated information access mechanism among the VE nodes is required.

In order to introduce the motivation for applying federated information management in VEs, we discussed the fact that even though enterprises involved in a VE must share and exchange a part of their information in order to achieve the common VE goal, it is also true that not all members of a VE play the same role and not all of them should have the same access level to the information stored in other enterprises. In general, it is clear that among competitor enterprises in a VE the amount of trust is limited, and that every enterprise needs to precisely define the specific access rights
and visibility levels on its information for every other VE partner. As a result, within
the VE, support for the security of shared data and provision of different rights to
access shared data – mostly based on other enterprise’s role in the VE – are required
to be provided and reinforced.
In this context, the DIMS federated architecture approach proposed in this chap-
ter, offers certain novel features and mechanisms for distributed information manage-
ment in the context of VE collaborations. Among others, we can mention:

- VE enterprise members have transparent access to up-to-date VE distributed
  information through different DIMS query functionalities applied on the inte-
  grated schema.

- The DIMS integrated schema represents and provides access to all the informa-
  tion classes that are necessary to support the operation of the VE Cooperation
  Layer as a whole unit.

- VE partners can rely on a mechanism to define and configure secure fine-grained
  access rights on local information for other VE partners depending on their role
  and on specific VE agreements, contracts or supervision clauses. These access
  rights are applied and reinforced in all DIMS tasks involved in the retrieval of
  VE distributed information.

- There is no need to maintain redundant information along the VE network
  nodes.

- The DIMS federated system incorporates specific high-level information man-
  agement functionalities for VE creation and operation phases (e.g. VE moni-
  toring and coordination tasks).

In this way, the general federated database architecture concepts and principles
have been specifically tailored in the DIMS architecture to handle the complex in-
teroperability and information management functionality requirements for VEs, set
by the target industrial/manufacturing SMEs (these and other advanced federated
features offered by the DIMS architecture will be further analyzed in Chapter 4).
Furthermore, based on the VE scenario cases presented in Section 3.2, and the
VE information modeling analysis described in Section 3.3, the following requirements
must be considered and supported in the design phase of the DIMS:

- 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 design and development
  phases of DIMS.

- Storage and management of a wide variety of kinds of information need to be
  supported, depending on the internal business processes defined at each enter-
  prise.
3.6. Conclusions

- 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 (specially considering the application of standards such as EDIFACT), but also the possibility of storing objects as “blobs” must be provided in order to support the interaction of enterprises.

- Definition of enterprise information that 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.

- Integration 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 each component module of the VCL must be supported.

In terms of DIMS operational requirements, the main conclusions as the outcome of the analysis performed in Section 3.4 are summarized below:

- DIMS functions must provide seamless access to up-to-date information that is distributed among the VE nodes.

- 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.

- DIMS query/update operations must be provided through various mechanisms, including local SQL queries, generic data management functions, high-level functions for specific VCL modules, and generic distributed/federated queries.

- Considering the dynamic and evolving nature of the VE life-cycle, the DIMS component must be flexible and configurable enough to support the easy evolution of the VEs as they follow the different stages of their life cycles.

Furthermore, depending on the functionalities provided by other modules of the VE cooperation layer, the support of the following “advanced” features within DIMS may prove beneficial (although their support is not strictly mandatory):

- 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 e.g. the status of a production order.
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.

Finally, Chapter 4 of this thesis will describe how the general federated database architecture concepts and principles have been specifically tailored in the DIMS design, in order to handle the information management requirements identified in this chapter.