01 SUPPORTING INFORMATION ACCESS RIGHTS AND VISIBILITY LEVELS IN VIRTUAL ENTERPRISES

A. Frenkel, H. Afsarmanesh, C. Garita, L.O. Hertzberger
Computer Science Department, University of Amsterdam
{annef, hamideh, cesar, bob}@wins.uva.nl

In the Virtual Enterprise (VE) paradigm, enterprises must share and exchange a part of their internal local information in order to interact and cooperate with each other towards the achievement of the global VE goals. However, the information access level for every VE partner depends on several "determining features" such as the cooperation relationships, degree of trust, the function of the enterprise in the VE, and contractual/legal agreements, among other issues. Considering these aspects, every enterprise needs to precisely define the appropriate access rights and visibility levels on its information individually for every other VE partner. This paper focuses on the VE access rights definition mechanism designed and implemented within the Distributed Information Management System in the context of the ESPRIT project PRODNET II.

1. INTRODUCTION

Virtual Enterprises (VE) represent collaborations of groups of pre-existing autonomous enterprises, which selectively share their expertise, skills and resources towards the accomplishment of a common product or service. Enterprises involved in a VE must share and exchange a part of their information in order to achieve the common VE goals and to support each other's functionalities. However, it is clear that among competitive enterprises in a VE, there are some features that will determine the access level to the local information of every enterprise. For instance the amount of trust between them, the function of the enterprises in the VE, the contractual/legal agreements within enterprises are examples of these "determining features". For this purpose, in every VE instead of individually considering enterprises, the determining features are merged into one characteristic called "role" to decide on their access rights. As expected, not all the members of a VE play the same role and therefore they do not need to have the same access level to the local information of other VE members. For this reason, every enterprise needs to precisely define the specific access rights and visibility levels on its own information for every other VE partner.

As a result, support for the security of shared data and provision of different access rights to shared data -based on other enterprise's role in the VE- are required to be provided and reinforced within each VE. As an example of this role-based

access rights definitions, it is clear that the regular partners involved in the supply chain may require to exchange certain data with other enterprises, while the VE coordinator enterprise needs to monitor the progress of the VE goals and activities at the partner sites. Therefore, the fact that typically the VE coordinator role implies a higher (broader) level of visibility on other members' information than a regular partner role, must be properly supported by the VE information management system. As a result, it is not realistic for the information management approach within a VE network to assume that a single global schema defines all the information that is visible and exchanged by all partners.

Considering the agility in this paradigm, namely the fact that a VE can be created dynamically, may consist of many enterprises and at the same time can be of temporary nature, definition of individual access rights is a cumbersome and error-prone task. To accomplish this objective, in the Esprit project PRODNET II, an innovative mechanism is designed and implemented to fully support the access rights definition in a systematic and configurable way. In (Afsarmanesh, 1999a) and (Afsarmanesh, 1999b) the basic idea of access rights definition mechanism in within the DIMS component of PRODNET was briefly addressed. The focus of this paper is the detailed description of this mechanism within the DIMS. This mechanism is based on a federated database architecture specifically tailored for this purpose.

The structure of this paper is organized as follows. First, in the introduction section, the general PRODNET architecture and the components of the DIMS module are described. Section 2 presents some considerations for managing and accessing distributed information in VEs. Section 3 analyses the details of the DIMS functionality that supports the definition of access rights and visibility levels in VEs. In order to illustrate further the application of the proposed concepts, an example based on the PRODNET demonstration scenario is provided in Section 4. Section 5 addresses some extensions and future directions that this research work can follow. Finally, Section 6 draws the main conclusion of this paper.

1.1 General PRODNET architecture

In order to illustrate the role of the DIMS in the PRODNET architecture, it is necessary to first introduce the general PRODNET node architecture (see Figure 1). This architecture has been extensively reported in other papers (Camarinha-Matos, 1998), (Afsarmanesh, 1999b), and here only the basic elements are described.

Every enterprise in the PRODNET network of potential VE-members is considered as a *node* consisting of three major components: an Internal Module, a PRODNET Cooperation Layer (PCL), and an Advanced VE Coordination Functionalities (ACFs) module. The Internal Module of a node basically consists of the internal information management systems of the company, such as its Production Planning and Control systems (PPC). The ACFs module provides some additional functionalities to extend the scope of the PCL, including the coordination and monitoring of VE-related activities and logistics. The Distributed Business Process Manager (DBPMS) module represents one of these ACFs. Finally, the PCL is the fundamental component that allows the enterprise to interoperate with others in the context of the PRODNET VE network. The PCL itself consists of several internal *components*, briefly described next.

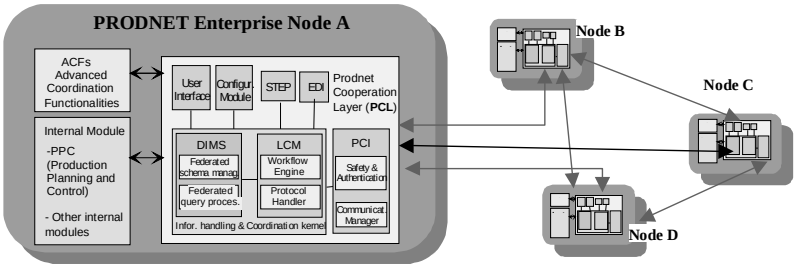

Figure 1 - Description of the PRODNET general architecture

- The Configuration component: allows the set up of certain elements and functionalities of the PCL, for every VE in which a given enterprise is involved.
- The STEP and EDI modules: support the exchange of technical product data and the commercial order-related data respectively.
- The Human Interface module: supports the end-users interactions with the PCL.
- The Local Coordination Module (LCM): executes and controls the internal PCL workflow, which specifies the desired cooperation behavior of each PCL (Camarinha-Matos, 1999c).
- The PRODNET Communication Interface (PCI): is responsible for the actual communication channel among nodes in the VE network (Osorio, 1999).
- The Distributed Information Management System (DIMS): supports all the distributed information management requirements for the PCL operation.

The DIMS reference architecture and its internal components are described briefly in the following section.

1.2 DIMS internal architecture components

In order to cope with the complexity of the VE application domain, it is clear that advanced mechanisms must be designed and implemented in order to support the intricate VE distributed information management requirements. After an in-depth study of the VE application domain, and after modeling and classifying the information involved in the PRODNET application, it was possible to identify which part of the information needed to be kept local, imported, exported, and accessed in an integrated way among the different enterprise nodes. In order to support all the information management requirements recognized during the analysis phase, a federated database architecture was conceived during the design phase of the DIMS module. Following this approach, the general design of DIMS in PRODNET is based on the definition of the *PCL integrated schema*, which is represented and handled in all nodes. Data can be imported/exported and seamlessly accessed through this PCL schema, but the proper access rights are defined locally at every enterprise to specify the rights for external nodes. Therefore, although the sharable data at every node can be accessed through the same schema structure representing the PCL information, DIMS properly preserves the federated information access and visibility constraints by means of well-determined export schema definitions. For a detailed description of the DIMS federated approach and integrated schema please

see (Garita, 2000b), (Afsarmanesh, 1999a). In this section, the main components of the DIMS applications server are described.

The general reference architecture of the DIMS embodies the following components: the DIMS Server Agent, the Federated Query Processor, the Export Schema Manager and Tool, the Internal DIMS Database Manager, and the DIMS Kernel Configurator. A brief description for every component is given in the paragraphs below (for more details see (Garita, 2000b):

- DIMS Server Agent: it is responsible for receiving and dispatching all the DIMS service requests issued by the other PCL modules.
- DIMS Federated Query Processor (FQP): its main objective is to transparently support the access to data distributed over the nodes of the VE network, taking into account the specific visibility access rights (represented by federated export schemas) defined for every node (Garita, 2000a).
- Export Schema Manager (ESM) and Tool (ESMT): enclose the functionality to create and maintain the access rights and the hierarchy of visibility levels that are specified for a given node.
- Internal DIMS Database Manager: it represents the server tier of the DIMS which provides all the basic functionalities that are expected from a database management system including: transaction management, data storage and retrieval, stored procedures management, etc. The DBMS that was used as "construction ground" for the DIMS is the Oracle DBMS.
- DIMS Kernel Configurator: allows the user to specify the values of certain DIMS operation parameters, e.g. communication port numbers and timeout duration.

The work presented in this paper focuses on the functionality of the ESM and ESMT component of the DIMS. Namely, these components support the required access rights and visibility levels among VE partners. These access rights definitions are actually used by the FQP component when evaluating a distributed DIMS query coming from the end-user or other PCL components such as the DBPMS. Therefore, in order to fully understand the approach for information access presented in this work, it is necessary to briefly describe the FQP component functionality and its relationship with ESM.

1.3 The DIMS Federated Query Processor

The main objective of the DIMS Federated Query Processor module is to support transparent access to data (which can be distributed over the nodes of the VE network), taking into account the specific visibility access rights (represented by export schemas) defined for every node. The DIMS FQP functionality enables end users to query authorized VE-related information, while hiding the data location details. The FQP mechanism that provides access to distributed VE information, considers the access rights that have been defined for each enterprise using the ESM/ESMT functionality (see (Afsarmanesh, 1999b) for more details). Thus, in order to identify if the requested data is privileged for the query issuer, the FQP component must work together with the ESMT. The FQP component controls the access to the data using these export schema definitions generated by using ESMT. In general, the processing of general federated queries can be summarized as

follows: when the query arrives at the DIMS, it is analyzed and decomposed into a set of single-site sub-queries, each of which needs to be sent to only one site (VE node) to be processed. After that, the results of the sub-queries are gathered and merged into the final result. If necessary, the FQP module interacts with the corresponding PPC to retrieve up-to-date local production data, during this process. To understand the internal operation of the FQP module in DIMS and its relationship with the ESMT, please see (Garita, 2000a) and (Afsarmanesh, 1999a), where a more detailed description of the FQP application in PRODNET is provided.

2. CONSIDERATIONS FOR INFORMATION ACCESS IN VE

In this section, some general considerations for information access in virtual enterprises are provided and in specific the concept of 'information visibility levels' and 'roles' in virtual enterprises are introduced.

2.1 The Concept of Visibility Levels in VEs

Since different enterprises are entitled to different visibility levels and access rights on other nodes' information, every node in the network of potential VE members must decide which part of its local information to make available to every other partner in every particular VE in which it is involved. To accomplish this objective, every node can protect its autonomy and privacy by defining one detailed *individual export schema* based on its PCL local schema, for every other node with which it shares information (see (Afsarmanesh, 1999b), (Afsarmanesh, 1999a)). This approach is based on the federated database architecture described in Section 1, and it has proved within the PRODNET project to adequately support the indicated VE information access requirements (Afsarmanesh, 1999a).

Although the suggested approach basically defines an individual export schema on top of the local schema for every external "user", we have further generalized this basic idea to the definition of a complete *hierarchy of export schemas*. This hierarchy allows the grouping and classification of common export schema characteristics, facilitating the task of individual export schema definitions.

Another approach to integrated schema management in the context of federated databases can be found in (Fankhauser, 1998), where the emphasis is on a federated database approach with one integrating schema that uses an ODMG interfaces to different DBMSs. Also, in (Rosenthal, 1999) and (Abiteboul, 1991), other approaches are presented to define user-specific 'views' for supporting authorized database access based on the underlying database schemas. However, in contrast to the DIMS architecture, these approaches do not aim at the specific support for the VEs and their requirements. Namely, there is no notion of enterprise role and role hierarchy for export schemas, as explained in more details in the following section.

2.2 The Concept of Roles

In a virtual enterprise environment, the access level to local information of other VE partners depends on several aspects, such as the cooperation relationships, degree of

trust, contractual/legal agreements, and the function of every enterprise in a given VE. In our proposed mechanism, we suggest to merge these "determining features" into one characteristic called "*Role*". Different concepts of role, in the context of VEs, have been introduced in other related works. For instance, a *role* can be defined as the "access control groups" (Edwards, 1996), as the "named set of privileges" (Osborn, 1997), or as the "task-oriented relationships" (Gladney, 1997). In this paper, a *role* is defined by every enterprise as a *category* or *class* that is associated to partners depending on a combination of the determining characteristics mentioned above in order to facilitate the access rights definition. In this paper, we do not address how roles are defined in every enterprise; that is described in other research as addressed below. Rather we focus on the definition of an innovative mechanism to implement the access rights and information visibility based on roles.

As mentioned in (Lupu, 1999): "the notion of role implies some context, which explains how roles interact, and the relationships between". Thus, based on this definition, the partner role concept can be used to determine the position of each partner in a hierarchy of access rights definitions. Thus, in our approach, the partner role concept can be used to determine the position of each partner in a hierarchy of access rights definitions. Thus, an extended idea of the concept of role is handled, where a complete *hierarchy of roles* is defined, representing the relationships among enterprises such as the coordinator, supervisor, and regular partners. The advantage of defining this hierarchy of roles is that then it is possible to associate to every 'different category of functions' (i.e. 'roles') an individual definition of the access rights. In this way, both the description of the VE in terms of roles and associated privileges and its relation with the export schemas, remain relatively fixed as enterprises join/leave the VE or change the role within the VE as mentioned in [Barkley99]. Also in [Smith98] the importance of a flexible and dynamic support of roles in a shared environment is described, in relation to the shared space applications and programming environments.

The functions/responsibilities assumed by an enterprise are established during the initialization/creation stage of the VE. Thus, according to the VE initialization/creation model (described in (Camarinha-Matos, 1999a)), once the VE partners are identified and selected after some negotiation process, the VE structure (topology) is defined as well as the role that each partner will play in the VE. This VE topology and role information is distributed among the VE members and the generated VE configuration is loaded at every partner site. After this stage, formal contracts need to be signed with individual partners, in order to formalize their agreement and determine their rights and obligations in the context of the new collaboration. Contract terms can then be defined in terms of "supervision clauses" that can for instance enforce the rights of the VE coordinator on monitoring, at individual partner site, the progress of a local production order. The supervision clauses can be defined by the VE coordinator, based on the VE contract, and distributed to every VE partner as described in (Klen, 1999) (Spinosa, 1998). Each individual partner can in turn load this information, validate it, and then define the proper information access rights and visibility levels for the VE coordinator and other VE partners.

3. DIMS FEDERATED EXPORT SCHEMA MANAGER

This section describes the Federated Export Schema Manager: the DIMS component that defines the required access rights and visibility levels among VE partners.

3.1 Design of VE Export Schemas

In Figure 2, the design of the database schema related with the export schema management is presented. Through this schema, the recursive definition of elements of the export schema hierarchy and the role hierarchy are supported.

Before the definition of the export schema hierarchy, it is necessary to define the hierarchy of roles. For every different function that a given partner is going to play in the VE a role (ROLE) is defined. Every ROLE has as attributes: the general type of the role (e.g. coordinator, supervisor, regular), the name that identifies the role and the identification of the parent of the role in the hierarchy.

As mentioned in the previous section, the access rights are expressed in terms of the role that enterprises play in order to carry out the goals of the VE. Thus, for every VE role, an external schema set (Export_Set) is defined, which at the end corresponds to the partner's export schema. Through the Export_Set, the proper visibility levels for the partners on the local schema of the enterprise are specified. An Export_Set can be either a single or a dependent export set depending if they are based on other export sets or not. With this approach, on one hand, the support for the general export schemas definition is provided, where not only the pre-defined export schema definitions at the level of VE, coordinators, supervisors, and partners, are considered, but also other hierarchies can be defined and supported as necessary. On the other hand, an Export_Set consists of a set of schemas, which in turn can be single schema (EXP) or dependent schema (Dependent-EXP) following this definition strategy. More details of the schema definition for the enterprise export schema can be found in (Afsarmanesh, 1999a).

To operate on the described schema, an "Export Schema Manager" (ESM) module has been developed. The ESM is used for instance, to create a basic export schema, and then, to define dependent partner export schemas based on it, is also used to create the hierarchy of roles. The ESM will ensure that the export schema and the role hierarchies remain consistent, and that the schema definitions for every dependent partner export schema are properly created. Of course, simpler information access scenarios, which can also occur, are fully supported. For instance, even if no hierarchy is needed or required and only independent export

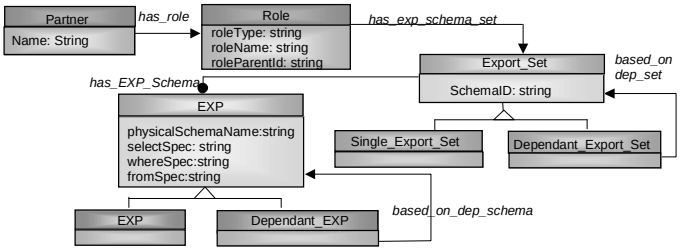

Figure 2 - Schema definitions for partner export schema management

schemas for every partner are required, the ESM can support this functionality operating on the described schema in the same way. In this simplest case, only independent roles associated with Export Set instances will be managed; no further composition is required here.

3.2 Definition of VE Export Schemas using the ESMT

In this section, a specific methodology for the definition of the VE Export Schemas using the Export Schema Manager Tool (ESMT) is presented. As mentioned in the section 3.1, the Export Schema, that will define visibility levels and access rights to other nodes' information, consists internally of a set of export tables (EXP) and dependent export tables (Dependent-EXP). The EXP refers to an export schema defined on one database table, and the Dependent-EXP is a schema defined on an EXP. Then, the Export Schema is assigned to a role, and every partner will be associated with one of the roles that have been defined.

The ESMT was developed for DIMS to support the definition and creation of the Export Schemas as part of the local information that is shared with other enterprises. It helps the user of PCL to define and create the export schemas, during the configuration phase of the VE. Also, it can be used during the execution phase of the VE, as access rights need to be modified due to for instance agreement modifications or when defining access rights to new partners that have just joined the VE.

The main window of the ESMT interface tool contains a menu bar that enables the user to perform different operations, this menu supports the definition of different schemas. There is also a status window where control messages generated during the creation of the export schema are displayed. The VE Export Schemas and its relation with every node in the VE need to be defined in several steps. These steps and the action associated to each of them in the ESM tool are described below:

1. Define the Role Schema hierarchy. A complete hierarchy of the roles that every node can play in a VE is created , based on for instance, the three basic kinds of roles: coordinator, supervisor, and regular partner. The Create Role Schema option allows the user to create this role hierarchy schema, only specifying the type of role, the name that will identify it and its parent on the hierarchy.
2. Create an EXP schema for every database table. The EXPs determine which instances (i.e. horizontal partitioning) will be available, for instance, to the Coordinator node. In this case, it is also necessary to define which database tables will be accessible by another node. In the ESMT, the Create EXP window supports the creation of the EXPs, the user has to enter a unique name for the EXP name or also they could be generated automatically. Then, the attributes that define the export schema has to been specified, i.e. the original table on which the schema is based on, the list of attributes selected from the table, and the conditions on the specified attributes.
3. Define one or more Dependent-EXPs for every EXP schema defined above. The Dependent-EXPs determine which attributes will be available for instance, to the nodes in the Supervisor level. The Create Dependent-EXP window supports the creation of the Dependent-EXPs. This operation is equivalent to the previous one, except that the reference is not a table, but an EXP.

4. Define one or more Dependent-EXPs for every Dependent-EXP defined above. This operation is equivalent to the previous one, except that the reference is not an EXP, but another Dependent-EXP. These schemas determine which attributes will be made available for instance, to the nodes in the Enterprise level.
5. Define the EXP/Dependent-EXP Set. This set will group the EXP and Dependent-EXPs that will specify the proper visibility levels for the different kind of roles. The Create EXP/Dependent-EXP set window supports the creation of the export schema set. A EXP/Dependent-EXP hierarchy which represents the specific instance diagram for the export schema hierarchy is displayed and it shows how EXP schemas are defined for each database table, and how the Dependent-EXPs are based on the EXPs. Also, the EXP/Dependent EXP Set is displayed showing how the set is taking form according to what the user has selected or removed from the hierarchy. Finally, the SQL description of the chosen database-table, EXP schema or Dependent-EXP schema is shown.
6. Create the association Role-Export Schema. Each role in the hierarchy of roles previously defined will be associated with one EXP/Dependent-EXP Set defined above, so that every defined role is able only to access predefined information. The advantage to define the role hierarchy and associate every role with a EXP/Dependent-EXP Set is that then it is easier to define different visibility levels for every actual VE partner.
7. Assign an Export Schema to an Enterprise (Figure 3). This will associate an enterprise with one of the roles of the Role Schema hierarchy defined above, so that every node will be able to access different information, according to the role that will play in the VE. The Create Enterprise EXP Schema window supports the definition of the export schema for an enterprise. It helps to associate a specific enterprise (Enterprise ID) with a specific Role (Enterprise Role ID). The Schema Specifications shows the Schema set, i.e. the EXP schemas and EXP-Dependent schemas that the particular enterprise will be able to access, and the SQL description of the chosen EXP schema or Dependent-EXP schema.

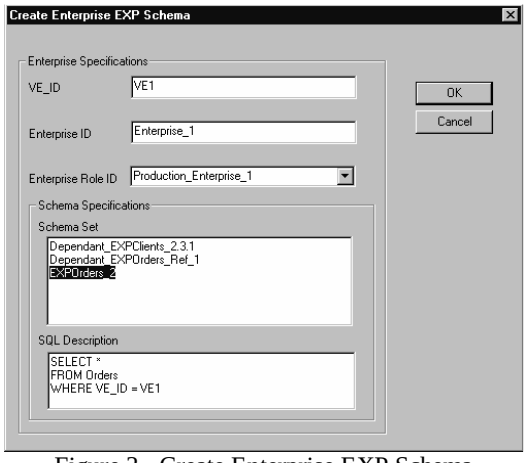

Figure 3 - Create Enterprise EXP Schema

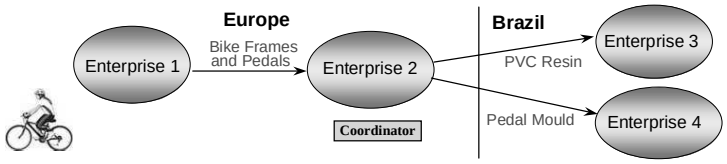

Figure 4- PRODNET Demonstration Scenario

4. EXAMPLE OF EXPORT SCHEMAS IN PRODNET

In this section, a specific case demonstration of the DIMS advanced functionalities, based on the general PRODNET scenario is presented.

4.1 PRODNET scenario

Based on the PRODNET Demonstration Scenario described in (Camarinha-Matos, 1999b), in this section a specific demonstration of the DIMS advanced functionalities is presented. The aim is to demonstrate and validate the Export Schema Manager. For this reason this demonstration focuses only on the export schema definition, thus the federated query processing is not addressed. The Federated Query Processing functionality has been extensively reported ((Garita, 2000a), (Afsarmanesh, 1999a)), here only the basic elements are described.

Figure 4 represents a general overview of the VE scenario to be used as a reference for the ESM demonstration. It presents the VE purchase orders that were requested and received on each node as explained in (Camarinha-Matos, 1999b). In this intercontinental VE, Enterprise 1 is a bicycle producer that requires bicycle pedals and bicycle frames to supply its internal production. And therefore it issues a purchase order to Enterprise 2 for each product. To attend the order of bicycle pedals, Enterprise 2 needs pedal moulds and some raw material (in this case PVC resin), which are ordered to Enterprise 4 and 3 respectively. The bicycle frames are produced and supplied by Enterprise 2 itself. For this demonstration, we use the same data used in general PRODNET demonstration except that the "Bike Frames" and "Pedals" orders are both asked at the same time, in order to have a more interesting demonstration. As shown in Table 1, Enterprise 2 stores locally the production information of bicycle frames, Enterprise 3 the production information of PVC resin, and Enterprise 4 the production information of pedal moulds.

Table 1 - Sample of Local production information at every VE member

a) Enterprise 2

Enterprise Id	*Name*	*Description*	*StartDate OfProduct*	*EndDate OfProduct*	*Delivery Date*	*Requested Quantity*	*Produced Quantity*	*Price PerUnit*
Enterprise 2	Bicycle Frame	Bicycle Frame	18-May-99	1-Nov-99	3-Nov-99	100	50	10
Enterprise 2	Pedal	Bicycle Pedal	22-Aug-99	5-Nov-99	8-Nov-99	200	100	2

b) Enterprise 3

Enterprise Id	*Name*	*Description*	*StartDate OfProduct*	*EndDate OfProduct*	*Delivery Date*	*Requested Quantity*	*Produced Quantity*	*Price PerUnit*
Enterprise 3	PVCResin	Bicycle PVC Resin	1-Sep-99	28-Oct-99	30-Oct-99	70	40	1

c) Enterprise 4

Enterprise Id	*Name*	*Description*	*StartDate OfProduct*	*EndDate OfProduct*	*Delivery Date*	*Requested Quantity*	*Produced Quantity*	*Price PerUnit*
Enterprise 4	PedalMould	Bicycle Medal Mould	10-Sep-99	28-Oct-99	30-Oct-99	1	1	20000

4.2 Demonstration Case

The main goal of this case is to demonstrate the definition and creation of the export schemas. This definition takes place at every enterprise and is part of the Enterprise Configuration/Reconfiguration step as explained in (Camarinha-Matos, 1999a). Furthermore, this case also shows how easily and dynamically the access rights for an enterprise can be changed at any time during the VE Operation phase, due to definition of new bilateral agreements or other reasons.

As mentioned above, it is not desirable that all the enterprises involved in a VE have the same access rights to information of a specific node. Thus, for instance an enterprise with role 1 (e.g. Regular Partner) should not access the same information, that for example the enterprise with role 2 (e.g. Coordinator) can access. In this specific demonstration case, Enterprise 2 will define the information access rights on its local information for the enterprises with a "Regular Partner" role, i.e. Enterprise 3 and Enterprise 4.

The main steps in the general process to define the access rights are illustrated in Figure 5. The process includes the definition of EXPs for the database tables of the production orders (*Dbpms_ProductionOrder*), the requested orders (*Dbpms_RequestedOrder*) and the requested items (*Dbpms_Requested_Item*) as specified respectively in the EXP hierarchy in (5-1b). The other information represented in Figure 5 depicts the steps in definition of one of these export schemas, namely the ExpProdOrder1. The creation of this export schema is shown in (5-1a). This EXP determines that all the attributes (*) of the *Dbpms_ProductionOrder* table in Enterprise 2 can be made available to Enterprise 3 and Enterprise 4, with the conditions that VE must be VE1 and the information must be about Pedals (VE = VE1, Name = Pedals). The next step is the creation of the EXP Set (5-2). This EXP Set groups the EXPs and/or the Dependent-EXPs that will specify the proper visibility level which Enterprise 2 will give to the enterprises with a Regular Partner role. The third step is to update the Role Schema hierarchy. Assuming that the role Coordinator has been created previously, the role Regular Partner can be created (5-3b) as its restriction and associated with the EXP Set that is just created (5-3a). At last, it is necessary to specify which role Enterprise 3 will play in the Virtual Enterprise, in this case, it is the Regular Partner (5-4). The same must be specified for Enterprise 4. In this way, both nodes have access to the same information.

In order to validate the definition of the export schemas for Enterprise 3 and

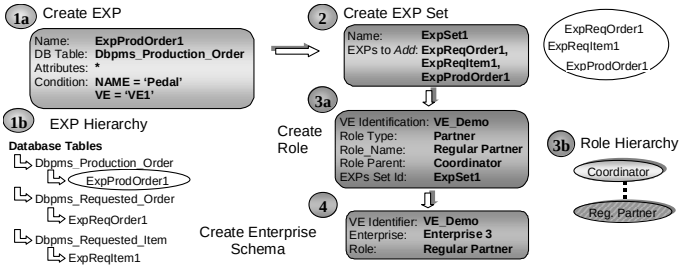

Figure 5 – PRODNET DIMS Demonstration Case

Table 2 - Final Results for Demonstration Case

a) Results at Enterprise 2

Enterprise Id	*Name*	*Description*	*StartDate OfProduct*	*EndDate OfProduct*	*Delivery Date*	*Requested Quantity*	*Produced Quantity*	*Price PerUnit*
Enterprise 2	Bicycle Frame	Bicycle Frame	18-May-99	1-Nov-99	3-Nov-99	100	50	10
Enterprise 2	Pedal	Bicycle Pedal	22-Aug-99	5-Nov-99	8-Nov-99	200	100	2
Enterprise 3	PVCResin	Bicycle PVC Resin A	1-Sep-99	28-Oct-99	30-Oct-99	70	40	1
Enterprise 4	PedalMould	Bicycle Medal Mould	10-Sep-99	28-Oct-99	30-Oct-99	1	1	20000

b) Results at Enterprise 3 and Enterprise 4

Enterprise Id	*Name*	*Description*	*StartDate OfProduct*	*EndDate OfProduct*	*Delivery Date*	*Requested Quantity*	*Produced Quantity*	*Price PerUnit*
Enterprise 2	Pedal	Bicycle Pedal	22-Aug-99	5-Nov-99	8-Nov-99	200	100	2
Enterprise 3	PVCResin	Bicycle PVC Resin	1-Sep-99	28-Oct-99	30-Oct-99	70	40	1
Enterprise 4	PedalMould	Bicycle Medal Mould	10-Sep-99	28-Oct-99	30-Oct-99	1	1	20000

Enterprise 4, the DIMS Browsing Interface of the Federated Query Processor, described in more detail in (Garita, 2000a), can be used to execute certain specific queries on the PCL schema. For example, any of the Regular Partner (i.e. Enterprise 3 and Enterprise 4) can ask the query *GetProductionOrder* to gather the production information about the orders in the VE. Thus, the DIMS at Enterprise 3 (or Enterprise 4), after sending the query to the other enterprises, collect, process and merge the results, will present the results through the Browsing Interface. In this case, Enterprise 3 (or Enterprise 4), as a Regular Partner node, will not get all the information from Enterprise 2, since this enterprise has defined the access rights such that only the information of Pedals will be presented, as it is shown in Table 2b. But if the Coordinator node (i.e. Enterprise 2) asks the same query, then this node will be able to see all the information, since the Coordinator does not have this restriction, as it is shown in Table 2a.

In a similar way, later Enterprise 2 can for instance limit the access rights for Enterprise 4. For this purpose, Enterprise 2 will change the information visibility rights defined for Enterprise 3 by modifying the Export Schema associated to Enterprise 3. Please notice that the changes on the visibility levels are only for Enterprise 3 but not for the other enterprises (e.g. Enterprise 4) with a Regular Partner role.

The process to change the access rights is very similar to the previous case, except that at the first step instead of create “EXP” the “Dependent-EXPs” need to be defined for the EXPs of the production orders (*ExpProdOrder1*), the requested orders (*ExpReqOrder1*) and the requested items (*ExpReqItem1*). For example, the Dependent-EXP for the EXP *ExpProdOrder1* in Enterprise 2 determines that only some attributes (e.g. EnterpriseId, Name, Description, Delivery Date and Produced Quantity) can be made available to Enterprise 3. As such, when Enterprise 3 asks through the DIMS Browsing Interface for the GetProductionOrder query, to gather the production information about the orders in the VE, only the specified attributes related with ‘Pedals’ will be received from Enterprise 2, i.e. the information for the StartDateOfProduct, EndDateOfProduct, RequestedQuantity, PricePerUnit attributes in Table 2 for Enterprise 3 will be not accesible. At the same time, Enterprise 4 can still access the entire set of information.

These scenario cases have demonstrated how the ESM tool of the DIMS can be used at the VE creation and set up phase (initialization) to define specific information visibility rights to other partners, and also gives an idea of how the

information access rights for other partners can be changed during the VE operation time.

5. EXTENSIONS AND FUTURE WORK

The research work presented in this paper can be extended in many directions, some of which are addressed in this section. In particular, the concept of export schema templates and a mechanism to automatically create export schemas are introduced.

5.1 Export Schema Templates

Besides the concept of *role* defined in Section 2.2, there are other mechanisms that could simplify the task of export schema definition for a specific VE partner. For instance, the concept of "export-schema-set *template*" can be introduced. Such templates represent a predetermined PCL export schema set, which will be used to create new export schema sets with similar characteristics. The idea of the template is to capture the general descriptions of certain VE member roles; for example, a default regular VE partner, or a default VE coordinator. In this way, a set of templates (built-in default descriptions of general export schema sets) are pre-defined and stored in DIMS, and with them the creation of new export schema sets based on these profiles can be carried out. A constructor function for an export schema set template would include a list of parameters that would be used to create the particular export schema set instance. The concept of template suggested here is similar to the use of class template handled in object-oriented programming languages such as C++ (see (Pohl, 1997)).

As a simple example of template definition and instantiation, please see Figure 6. In this figure, a template for export schema sets is defined, where it is possible to specify parameters for the VE identifier and the client (VE member) identifier. If many export schema sets present a given "pattern" such as the characteristics defined for this template, then the template can be used many times to create any number of export schema instances. Please notice that a template could be based on other template definitions, giving place to the concept of *dependent templates*.

Other applications of the concept of "templates" and "roles" applied to VE modeling can be found in (Oyvind, 1999). In that approach the template refers for instance to the specification of the common features of a collection of objects of a particular type e.g. enterprises, so that other object collections can be instantiated using it. These other definitions of roles and templates are related to our suggested approach; however the emphasis in that work is on formally modeling the general characteristics, behavior and interactions among abstract entities involved in a VE, while our emphasis is on the specific application of these concepts to the definition and support of the visibility levels for information exchange among the VE partners.

5.2 Automatic creation of export schemas

At the VE creation time, the task of definition of initial export schemas for every VE partner can be automated to a great extent in some specific cases. Based on the functionality offered by the ESM of DIMS and on VE contract-related information, a specific procedure can be developed that takes advantage of this information and that automatically generates the export schemas for every VE partner. To see how this could be supported, it is necessary to reference some of the steps involved in the VE creation phase, as described in (Camarinha-Matos, 1999a).

Please notice that the definition of access rights based on the concept of roles given in Section 2.2, corresponds in reality to the definition of the proper export schema set for every other partner. Also, notice that the supervision clauses can be formally and precisely modeled by data structures, and they can be distributed for example, as a text file that can be parsed locally at every node. For instance, the data associated with a supervision clause for a specific production order could indicate the information that needs to be made available to the VE coordinator (see also (Klen, 1999)). This information can include the identifier of the production order, the real and planned dates for starting and ending the production, the delivery date, the status of the order production (e.g. in progress, delayed, completed), etc. Therefore, the supervision clause for a production order can be modeled by a C-like structure as follows:

```
typedef struct {
        Identifier  VEid;
        Identifier  VEPartnerId;
        Identifier  identifierRequestedOrder;
        Boolean     requiresDeliveryDate;
        Boolean     requiresPlannedStartDate;
        Boolean     requiresRealStartDate;
        …
} RequestedOrderSupervisionClause;
```

In general, if the data structures describing the supervision clauses are commonly defined in advance and are well-known to all partners of a given VE, it is possible to create a set of predefined export schema templates (see previous section) for specific types of supervision clauses. These templates can be instantiated with parameters which values are extracted from the supervision clauses' information. For instance, if the supervision clause specifications are distributed as a text files, then the file can

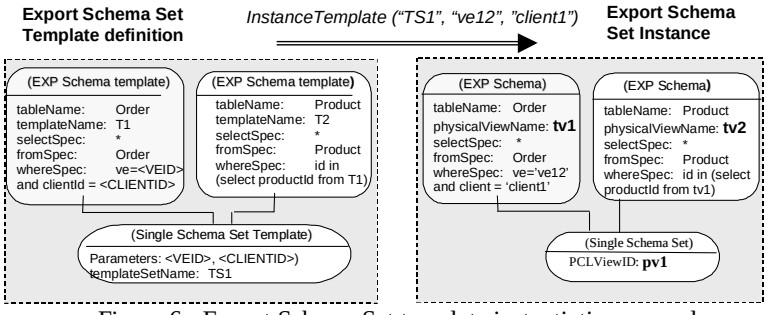

Figure 6 - Export Schema Set template instantiation example

be parsed and processed to determine the parameters for the construction functions associated to specific 'export schema templates'. Subsequently, when the templates are instantiated, the actual export schema sets for every VE partner are created automatically. Once the export schema sets are defined, the operation phase of the VE can start since the proper visibility levels for information are already properly defined according to the contract and supervision clauses. Clearly, if it becomes necessary, a human operator can modify the generated export schemas at any time during the VE life cycle (this feature is mandatory to be provided for VE evolution).

The automation of the export schema creation is based on the idea of VE contracts and VE supervision clauses. The examples provided in this paper are based on the PRODNET approach. Regarding other approaches, in (Hoffner, 1999) an approach towards automation of contract match making is described where standard form contracts and standard contract clauses are analyzed. In (Oyvind, 1999) contracts are used to specify the roles and relationships among "objects", such as enterprise objects. However, our presented approach exhibits unique features towards the automatic reinforcement of the contract and supervision clauses via the definition of fine-grained access rights based on export schema definitions.

6. CONCLUSIONS

In the VE environment, a crucial requirement is an information access mechanism among the VE nodes, where secure fine-grained visibility levels are defined locally at every node to determine which other VE partners are allowed to access which part of the local information. The Export Schema Manager component developed for the DIMS module in the PRODNET II project properly supports the definition of visibility levels and access rights for the information accesses from other VE nodes.

The described export schema hierarchy management features are used to adequately support the definition of information access rights based on the partner relationships established in the VE existing legal contracts or bilateral agreements among VE partner enterprises.

The Export Schema Management tool incorporates advanced user interface graphic elements and provides a comprehensive and friendly environment for the end users. Using the ESMT, it is possible to assign enterprises the proper access rights to the information they need by assigning to enterprises the role associated with their functions, and then associating to roles the adequately information access rights, i.e. the well-defined export schemas. Also, future work is foreseen regarding the extension of the ESM functions towards the support of the automatic creation of export schemas and export schema templates.

7. REFERENCES

1. Abiteboul, S., and Bonner, A. (1991). "Objects and Views." *ACM SIGMOD International Conference on Management of Data*, Denver, Colorado, 238-247.
2. Afsarmanesh, H., Garita, C., Ugur, Y., Frenkel, A., and Hertzberger, L. O. (1999a). "Design of the DIMS Architecture in PRODNET." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic, 127-146.

3. Afsarmanesh, H., Garita, C., Ugur, Y., Frenkel, A., and Hertzberger, L. O. (1999b). "Federated Information Management Requirements for Virtual Enterprises." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic, 36-48.
4. Camarinha-Matos, L. M., and Afsarmanesh, H. (1999a). "The PRODNET Architecture." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic, 109-126.
5. Camarinha-Matos, L. M., Afsarmanesh, H., Antunes, C., Clavier, J.-F., Garita, C., Gibon, P., Klen, A., Lenz, H., Lima, C., Mota, J., Osorio, L., Rabelo, R., Ribeiro, H., Schreiber, A., Spinosa, M., and Ugur, Y. (1999b). "The PRODNET Demonstrator." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic, 279-290.
6. Camarinha-Matos, L. M., Afsarmanesh, H., Garita, C., and Lima, C. (1998). "Towards an Architecture for Virtual Enterprises." *Special issue of the Journal of Intelligent Manufacturing with the focus on agent based manufacturing*, 9(2), 189-199.
7. Camarinha-Matos, L. M., and Lima, C. (1999c). "PRODNET Coordination Module." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic.
8. Edwards, K. (1996). "Policies and Roles in Collaborative Applications." *ACM Conference in Computer-supported Collaborative Applications*, Cambridge, MA.
9. Fankhauser, P., Gardarin, G., Lopez, M., Munoz, J., and Tomasic, A. (1998). "Experiences in Federated Databases: from IRO-DB to MIRO-Web." *24th International Conference on Very Large Data Bases - VLDB'98*, New York City, New York, 655-658.
10. Garita, C., Afsarmanesh, H., Ugur, Y., and Hertzberger, L. O. (2000a). "Federated Query Processing for Distributed Process Coordination in Virtual Enterprises." *4th IEEE/IFIP International Conference on Information Technology for Balance Automation Systems in Production and Transportation - BASYS'2000*, Berlin, Germany.
11. Garita, C., Ugur, Y., Frenkel, A., Afsarmanesh, H., and Hertzberger, L. O. (2000b). "DIMS: Implementation of a Federated Information Management System for PRODNET II." *11th International Conference and Workshop on Database and Expert Systems Applications - DEXA '2000*, London, England.
12. Gladney, H. M. (1997). "Access Control for Large Collections." *ACM Transactions on Information Systems*, 15(2).
13. Hoffner, Y. (1999). "Supporting Contract Matching." *9th International Workshop on Research Issues in Data Engineering - Information Technology for Virtual Enterprises - RIDE-VE'99*, Sydney, Australia, 64-24.
14. Klen, A., Rabelo, R., Spinosa, M., and Ferreira, A. (1999). "Distributed Business Process Management." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic, 241-258.
15. Osborn, S. (1997). "Mandatory Access Control and Role-based Access Control Revisted." *Second ACM Workshop on Role-based Access Control*, Fairfax, VA, USA.
16. Osorio, L., Antunes, C., and Barata, M. (1999). "Communication Infrastructure." Infrastructures for Virtual Enterprises - Networking Industrial Enterprises, L. M. Camarinha-Matos and H. Afsarmanesh, eds., Kluwer Academic.
17. Oyvind, J., Milosevic, Z., and Wood, A. (1999). "Modeling Virtual Enterprises and the Character of their Interactions." *9th International Workshop on Research Issues in Data Engineering - Information Technology for Virtual Enterprises - RIDE-VE'99*, Sydney, Australia, 19-26.
18. Pohl, I. (1997). *Object-Oriented Programming using C++*, Addison Wesley.
19. Rosenthal, A., and Sciore, E. (1999). "First-Class Views: a Key to User-centered Computing." *ACM SIGMOD Record*, 28(3).
20. Spinosa, M., Rabelo, R., and Klen, A. (1998). "High-Level Coordination of Business Processes in a Virtual Enterprise." *10th International IFIP Conference PROLAMAT'98*, Trento, Italy.