FEDERATED QUERY PROCESSING FOR DISTRIBUTED PROCESS COORDINATION IN VIRTUAL ENTERPRISES

Cesar Garita, Hamideh Afsarmanesh, Yasemin Ugur, L.O. Hertzberger
Computer Science Department, University of Amsterdam, The Netherlands
{cesar, hamideh, yasemin, bob}@wins.uva.nl

In Virtual Enterprise (VE) collaborations, enterprises operate as a single unit sharing their skills, technologies, and costs towards the achievement of a common goal. In order to support the proper cooperation among VE members, an advanced distributed information management approach would be highly beneficial to tackle the problems of physical distribution and access rights, present in many VE scenarios. One such scenario involves the monitoring and coordination of distributed business processes (DBPs) in VE, in order to ensure the fulfillment of the global VE goals. This paper shows how advanced federated database query processing mechanisms were designed and implemented in the framework of the ESPRIT project PRODNET II, in order to support DBPs monitoring and coordination in VEs.

1. INTRODUCTION

In a Virtual Enterprise (VE) environment, enterprises share their skills and technologies in order to deliver a competitive product (or service) that is usually very difficult or impossible to achieve by a given enterprise on its own. Given the natural complexity and diversity of applications that need the VE paradigm, it represents an extremely challenging and intricate field for any IT architecture that aims at supporting the interoperability of Virtual Enterprises and coordination of their joint activities. From the information management point of view, there are many VE characteristics that inherently demand the application of advanced distributed information management approaches. For example, in one specific case of information exchange among VE partners, information needs to be shared in order to allow the proper on-line monitoring and coordination of VE tasks and processes that have been assigned or delegated to enterprises in VE. This means that there must be an enterprise in the VE, representing the VE coordinator, that must have privileged access to specific proprietary local data residing at different nodes. Only by means of on-line access to up-to-date distributed information, it is possible for the VE coordinator to monitor the status of their tasks execution and to direct the enterprises towards a global cooperative behavior. For this purpose, the VE coordinator should be able to issue complex information queries involving diverse

information items, which are physically spread over several enterprises. In this general scenario, the VE information management system must support a high degree of autonomy and information privacy as well as a proper information sharing/exchange mechanism between the VE nodes (Afsarmanesh, 1999a), (Garita, 2000). Considering these requirements, the flexible gathering and integration of information that is necessary for VE, can be naturally supported through a federated database architecture, in which one of the main components is the Federated Query Processor (FQP).

In this paper, a federated database architecture is applied to the field of advanced coordination and monitoring of VE distributed business processes. The presented work has been fully implemented within the ESPRIT project PRODNET II¹. The structure of this paper is as follows. In the introductory section, a general description of the PRODNET reference architecture is provided. Section 2 addresses the main concepts and models associated with VE distributed business process management. Section 3 contains the detailed description of the DIMS federated query processing mechanism that fully supports the access to the distributed information model required for VE monitoring and coordination. Section 4 provides some examples regarding the particular application of FQP mechanism in the context of PRODNET. Finally, Section 5 includes the main conclusions of this work.

1.1 The PRODNET reference architecture

A summarized description of the PRODNET reference architecture is provided in this section (see Figure 1). For a more detailed specification, please see (Camarinha, 1999a). Every enterprise in the PRODNET network of potential VE-members is considered as a *node* composed 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 systems of the company –such as its Production Planning and Control systems (PPC)-, and other engineering systems necessary to accomplish its regular operations.

The PCL component is responsible for the actual inter-operation between a given node and other nodes in the network. PCL is the fundamental component that allows the enterprise to interoperate with others in the context of the VE. The PCL itself consists of several internal *components*:

- The Human Interface module: supports the end-users interactions with the PCL.
- The Configuration component: allows the set up of the elements and functionality of the PCL, for every VE in which a given enterprise is involved.
- The STEP and EDI modules: primarily support the exchange of technical product data and the commercial order-related data respectively.
- The Local Coordination Module (LCM): executes and controls the internal PCL workflow that specifies the desired cooperation behavior of each enterprise PCL.
- The Distributed Information Management System (DIMS): supports all the local and distributed information management requirements for the PCL, and

¹ The ESPRIT project-22647 PRODNET II (1996-1999) involved the following partners: New University of Lisbon, University of Amsterdam, ESTEC, Uninova, Lichen Informatique, ProSTEP, Federal University of Santa Catarina, Herten, Miralago and CSIN.

provides an interoperability platform to exchange and cooperatively manage large amounts of data between VE-members.

- The PRODNET Communication Interface (PCI): represents the entity responsible for the actual communication among nodes in the VE network.

Finally, the Advanced Coordination Functionalities (ACFs) module provides some additional functionalities to extend the scope of the PCL, including the advanced monitoring and coordination of VE-related activities. The Distributed Business Process Manager (DBPMS) module represents one example ACF.

2. DISTRIBUTED BUSINESS PROCESSES IN VES

The monitoring and coordination of the VE tasks and activities is a crucial issue during the operation phase of the VE life cycle. Given the fact that the VE sets forward one or more global goals involving several partners, which in principle can be completely autonomous and independent, there must be a well-defined mechanism to coordinate all their activities towards the global goals. In PRODNET, the DBPMS module, strongly supported by the DIMS and the LCM modules, accomplishes this task. The DBPMS module provides a set of advanced functionalities that have been identified and developed to tackle some specific problems involved in the coordination of VE distributed activities (Klen, 1999).

In PRODNET, the VE monitoring and coordination approach is based on a reference model for Distributed Business Process representation. This model has been used in order to properly represent the abstract concepts of VE, VE member, and its associated complex production chain, in such a way that the coordination and monitoring of VE tasks is adequately facilitated and supported (Klen, 1999). The model is designed and developed at the Federal University of Santa Catarina, based on some concepts and definitions from (AMICE, 1993), among other sources. The complete description of the DBPMS operations and DBP data models are outside the scope of this paper. However, in the next paragraphs, those aspects of the DBPMS data model that are related to DIMS functionality, are analyzed. In particular, the following concepts are defined for the DBPMS:

- Business Process (BP): structure that models the set or pool of VEs within a particular enterprise node. Several VEs can be managed inside a given enterprise node (one enterprise can be involved in many VEs simultaneously).

Figure 1 - Description of the PRODNET node architecture

- Distributed Business Process (DBP): represents a specific VE, irrespectively of the particular VE goals, service or products being addressed. A DBP is associated with the corresponding VE partners (Domain Processes).
- Domain Process (DP): represents a specific VE member enterprise (VE partner) and can have a set of zero or more requested orders e.g. product purchase orders.
- Requested Order (RO): represents a specific Requested Order from another VE member. A RO is composed of (a set of) Requested Items. The RO contains detailed commercial specification related to the purchase order.
- Requested Item (RI): represents a specific Requested Item (or line) associated with one RO. A RI is in turn composed of a set of Production Orders.
- Production Order (PO): contains the detailed description of an internal production order to be supervised. It contains specific internal production chain information that is specific to the PPC of every company.
- VE-Intra: contains extra intra-organizational parameters used for an analysis of the progress evolution of each production order in a given VE.

In reality, the coherent support and management of the above concepts results in a flexible approach to cover the complicated distributed production chain interactions involved in a given VE scenario. Every VE coordinator node has one or more instances of the described data model. In this way, the VE coordinator, e.g. the DBPMS system, can actually ask questions about the orders and monitor their status, progress, and the other information related to other partners in a given VE.

2.1 DBP distributed data management requirements

In order to properly support the *distributed* business process model described in the previous section, a set of *distributed* data structures and data access functions must be developed. It must be noticed that the DBP information needed to be handled by the VE coordinator, is actually spread along the network of the VE member enterprises. In other words, in fact the requested order information with its associated requested items, production orders and VE-intra parameters, represents a distributed “*requested order tree*”, that is constantly updated at the local site of each VE partner fulfilling the requested order. When a given VE coordinator needs to monitor some of the information contained in this tree, it must be provided with the most up-to-date information. This means that the DIMS must implement an access mechanism in such a way that the data stored in the internal module of a remote node, can be seamlessly queried, and accessed by the coordinator node.

For instance, the VE coordinator must be able to ask the following kind of queries on the DBP model involving several VE members:

- Get all the DBP identifiers, i.e. which VEs are being coordinated at this node.
- Get all the DP identifiers associated with a given DBP.
- Get all the RO identifiers for a given DBP.
- Get the RO (including status and progress) for a given RO in a DBP.
- Get all the RO tree information for a given RO in a DBP.
- Get the most up-to-date RI information in a DBP regularly.
- Get the most up-to-date PO information in a DBP regularly.

At the same time, while the distributed queries are issued, the concept of data access rights in this kind of data retrieval must be always supported by the information management component. For instance, the VE coordinator will obviously have more access visibility to the data stored in a given enterprise, than a simple regular VE partner, due to the control, monitoring, and possibly auditing responsibilities (and *rights*) of the VE coordinator. We need to consider that the visibility levels on the local enterprise information must be individually defined and configured for every other partner in every VE. It is a fact that the proper definition of the described visibility levels among VE partners is mandatory in order to support and reinforce the necessary level of trust among cooperative enterprises in this kind of open environments (Afsarmanesh, 1999a).

The DIMS in PRODNET not only supports the transparent access to the complete DBP distributed information model, but at the same time, it supports the autonomy and independence of every enterprise. The federated database mechanism applied for this purpose by the DIMS is described in the following section.

3. FEDERATED QUERY PROCESSING IN PRODNET

In this section, the DIMS federated approach is introduced, focusing on the integrated schema definitions for DBPMS and the Federated Query Processing.

3.1 The PRODNET DIMS Federated Architecture

Analyzing the wide variety of shared and exchanged information among VE member nodes, and achieving a comprehensive description and classification of this information, were a part of the challenging tasks tackled within PRODNET (Afsarmanesh, 1999a). After the extensive analysis phase and in order to support the information management requirements, and proper interoperation among PCLs in different nodes, a federated database architecture is designed for the DIMS module in PRODNET (Afsarmanesh, 1998), (Afsarmanesh, 1999b). The federated architecture approach proved to adequately facilitate and support the sharing of information between enterprises, while providing the necessary visibility levels to ensure their own autonomy and information privacy. The designed federated architecture has its roots in the PEER federated database system, developed at the University of Amsterdam (Wiedijk, 1996). The implementation of the DIMS component was developed in C++ using Oracle, ODBC facilities and RPC (Remote Procedure Call) mechanisms among other tools. Other related approaches to distributed information management in the context of VE support platforms can be found in other related projects such as VEGA (Zarli, 1999), GENIAL (Radeke, 1999) and EisNet (Lee, 1999). However, in comparison to the federated approach chosen for PRODNET, several concepts such as the export schemas and the flexible support for a hierarchy of information access levels, are not directly addressed in any of these architectures.

In the next section, the specific approach taken for the design of the DIMS in PRODNET is addressed in details.

3.1.1 PCL Federated Integrated Schema

After an extensive analysis and modeling of the information management requirements in PRODNET, the final results led to determine specific information models categorized as local, import, export, and integrated information (Afsarmanesh, 1999b). The categorization is based on the fact that: a part of the information generated and stored in an enterprise must be kept **local**; a part of the information that needs to be accessed from the other nodes will constitute the **imported** information from other enterprises; a part of the local information that needs to be shared with other enterprises constitutes the **exported** information to other enterprises; and part of the information that needs to be accessed at an enterprise must be **integrated** into one coherent schema. Based on this and further refined analysis, a general *integrated object-oriented database schema* for enterprise information is designed and developed (Afsarmanesh, 1998), (Garita, 2000). In this paper, we detail out the DIMS integrated schema design for the DBPMS information.

The DIMS integrated schema allows the *integration* of the distributed information of the DBP and provides the applications such as the DBPMS, a unified and coherent schema that hides all the details of the physical distribution of data, when issuing the database queries. A partial and simplified description of this schema is illustrated in Figure 2, basically representing only the information related to the data structures described for the DBP model in Section 2. Namely, the diagram clearly illustrates the relationship between DP and DBP, and the composition of RO, RI, PO, and intra-VE structures as a requested order tree.

By allowing the DBPMS module (or other end-users) to execute queries and other specific high-level data access functions on this integrated schema, the ultimate goal of monitoring and coordination of distributed requested order trees in a VE is accomplished.

3.2 Federated Query Processing Mechanism

As mentioned before, the PCL applications such as advanced coordination modules, and end users need to access VE-related data without worrying about the physical data distribution. At the same, the owner of the data at every enterprise wishes to share parts of its local VE-related data differently, with different classes of users, and keep other parts confidential. The main goal of the federated query processor is to provide access to an integrated view of the restrictive data over the VE network to authorized users, depending on their visibility levels defined at the remote sites. The visibility levels are implemented in terms of export schema definitions, and are managed as described in the next section.

3.2.1 FQP and Export Schema Definitions

Following the approach for the federated database architecture described in Section 3.1, every node in a federated database must define which part of its local information to make available for each other partner in every particular VE that it is involved to control the shared data. For this reason, each enterprise needs to create

specific export schemas on its logical data structures. An export schema definition is a collection of views representing the sharable part of the information (Meng, 1998).

In principle, it is usually an exhausting task to define one individual export schema for each VE partner, to restrict and control the access to enterprise's local data. However, this task is reduced and simplified in PRODNET. To achieve this simplification, we consider that each enterprise plays a specific role (e.g., coordinator, supervisor, regular partner) in each VE. Roles indicate the capability of an enterprise to perform a set of tasks in the VE. As a result, the sharable data can be primarily determined for enterprises based on their roles in the VE. In PRODNET, an Export Schema Manager Tool (ESMT) is developed for the DIMS module to support the creation and update of the export schemas for other enterprises, and to associate each set of export schema definitions to the role played by an enterprise in the particular VE, during the configuration and operation phases of the VE. The export schema elements are at the lowest level implemented as compositions of different table views. In order to identify if the requested data is privileged for the query issuer, the FQP component must work together with the ESMT. FQP controls access to the data using these export schema definitions generated by using ESMT.

To understand the internal operation of the FQP module in DIMS and its relationship with the ESMT, Figure 3 describes the main subtasks of the federated query processing. We can summarize the processing of general federated queries as follows: when the query arrives at the DIMS, it is analyzed and decomposed into a set of single-site subqueries, 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.

Figure 2 - Partial high-level DIMS integrated schema for DBPMS

Figure 3 - Federated Query Processing subtasks and interactions

These subtasks will be explained in details in the following sections with focus on the Distributed Business Process data management requirements in VE.

Other approaches to the federated query processor design and development can be found in (Meng, 1998) and (Kapsammer, 1997). In these works, the query processing mechanism is designed and developed to support a general multi-database architecture where there is a central interoperable layer to handle global queries, and a component layer at each database system participating in the federation. However, this general approach does not fit VE environment peculiarities since each VE is a temporary coalition of enterprises and therefore there cannot be one single enterprise to manage the central layer.

3.2.2 Query Reformulation and Decomposition

The FQP module provides a high-level function library that contains both general and application-dependent functions for the DIMS client applications. Besides those functions, a friendly user interface has been developed, which enables the end-user to issue certain queries by selecting menu items. In general, the query requesters do not need to worry about the low-level database access language and details, when using the DIMS high-level functions. When one of these library functions is called, the DIMS reformulates it into an internal query format, depending on the parameters specified for every function. This reformulated query is then analyzed to determine the specific VE partners involved in the original query. Further, the query is decomposed into a set of simpler subqueries, so that each subquery involves the retrieval of data from only one VE node. Namely, each subquery needs to be sent to only one corresponding partner to be processed locally at that side. At this point, the subquery that needs to be sent to a remote node is called an *external subquery*. It is also important to point out that the FQP mechanism described in this paper operates on the integrated schema described in Section 3.1.1. In PRODNET, all the nodes

have agreed on this common integrated schema, and therefore, it is not necessary to keep extra information about different schema structures at each node.

3.2.3 Query rewriting against Export Schema Definitions

When the external subquery arrives at the corresponding VE node (e.g. node B in Figure 3), it should be evaluated against the export schema defined at this node for the subquery sender (node A) in this particular VE. The first task in the query rewriting process is to find the names of the export schema elements corresponding to the class names in the subquery. The search space embodies the export schema set (set of export schema elements) associated with the role of requester site. After that, the subquery is transformed into queries using only names in the export schemas. The rewriting process and algorithm can be explained better with a simple example. Let us assume that the following subquery (q) is issued by enterprise A to check enterprise B's VE-related orders for those orders whose feedback date has passed; namely the orders whose feedback date is earlier than today's date.

q: *select * from Order, OrderLines*
where Order.feedback <= sysdate AND Order.key = OrderLines.orderkey ;

When the subquery arrives at Enterprise B, the export schema for Enterprise A will be used for its interpretation. First, all the referenced class names in q are replaced by the corresponding export schema elements inside the export schema set for enterprise A. Then for instance, the modified query will be as follows:

q1: *select * from Order_Exp, OrderLines_Exp*
where Order_Exp.feedback <= sysdate AND Order_Exp.key = OrderLines_Exp.orderkey ;

In this example, the Order_Exp and OrderLines_Exp are the elements defined in the export schema of Enterprise A and derived from the Order and OrderLines classes respectively. These elements are associated with the export schema sets defined for the role of Enterprise A in the VE. The next step is to consider the operations that are used in the derivation of export schema with the modified subquery (q1). This is however accomplished automatically by the view definition mechanism used in the DIMS:

q2: *select Order.officialname, Order.definition, OrderLines.name,*
OrderLines.orderstatus, OrderLines.orderprogress
from Order, OrderLines where Order.feedback <= sysdate and Order.orderlevel='Generated'
AND Order.key = OrderLines.orderkey;

Since Enterprise B defined the Order_Exp and OrderLines_Exp schema elements by selecting only generated orders and projecting these orders on certain fields (Order.officialname, Order.definition, OrderLines.name, OrderLines.orderstatus, OrderLines.orderprogress), the final query (q2) reflects this derivation process according to the export schema.

3.2.4 Execution of Federated Query – Interacting with PPC

DIMS communicates with the local data sources of the enterprise, through a specific application programming interface (API), developed for the enterprise's local production and planning system (PPC). The functions in the developed API allow the retrieval of data from the internal database system, and convert the results into

the common data format defined by the PCL interoperability approach. As such, all internal systems of an enterprise can interoperate with its PCL component. Consequently, the DIMS gets the up-to-date data from enterprise's local production system, through the specific API and stores it in its internal database temporarily during the processing of the query. After this step, the modified external subquery or local subquery can be executed on the data stored temporarily in the DIMS database.

To support the DBPMS requirements, specific data retrieval functions for its requested order tree (Get_RequestedOrderTree), requested order (Get_RequestedOrder), requested item (Get_RequestedItem) and production order (Get_ProductionOrder) are provided by the API of the extended PPC. One approach to support the DBPMS, is to trigger the monitoring task of DBPMS when the local production is scheduled in an enterprise and part of its data is generated inside the PPC. This approach (the push mechanism) avoids the repeated pulling of PPC's information by the coordinator, via the DIMS; what constitutes the second approach (pulling approach). Therefore, in the first approach, the PPC informs the coordinator when a part of the requested order tree information is generated in the local DIMS database of the enterprise with the current order status and progress information of the order, so that DBPMS can proceed with its monitoring operation.

After the monitoring operation is triggered, the DBPMS starts gathering information from the involved nodes over the VE network through the DIMS federated query processing mechanism. For instance, DBPMS can first ask the Requested Order Tree structure of an enterprise in a VE. DIMS processes the query and decomposes it using the VE partner identification. In this case, it is clear that there is only one subquery per partner, that needs to be sent out or processed locally. When the subquery arrives to the corresponding site, DIMS rewrites it against the export schema definition for the sender at this site. Then it asks Get_RequestedOrderTree to PPC through the API. DIMS obtains the most up-to-date information in this way, and evaluates the restricted/rewritten query on this data. The obtained result is returned to query issuer and then to its DBPMS.

Figure 4 - Interactions of PCL internal modules for FQP support.

3.3 Workflow Support for FQP

In the PRODNET approach, the DIMS, similar to the other modules of the PCL, does not use a “proprietary” mechanism to communicate either with the legacy systems or the remote DIMS; instead it follows the common PCL approach, that exploits the facilities of the PCL communicator, i.e. PCI (Osorio, 1999), and the PCL workflow manager components, i.e. LCM (Camarinha, 1999b). The workflow management system, which captures the activity semantics as related to the dependencies among operations performed by multiple components, is used to support the federated query execution plan within the VE environment.

There are several specific workflow plans designed to support the FQP mechanism. For instance, Figure 4 represents the basic set of events and the flow of activities involved in the processing of a DIMS federated query. The query in this figure depicts a request initiated by the DBPMS that involves an inquiry about the status of the requested orders, from the involved partners in a VE. The diagram shows the major steps in the activity flows for the part of the query processing where the coordinator node (Enterprise A) sends a subquery to one of the remote nodes (Enterprise B), and requests the most recently updated data (order status) from its PPC.

4. FQP DEMONSTRATION CASES FOR VE COORDINATION

A big demonstration scenario was prepared for the PRODNET project and presented at the Pro-VE '99 conference in October 1999 in Porto, Portugal (Camarinha, 1999c). This scenario involved the functionality of all the PCL modules. In the case of the DIMS, several specific demonstration cases have been developed to show the main functionalities of its federated query processor and ESMT in the PRODNET context. Sections 4.1 and 4.2 represent two such scenario cases. In order to describe these cases, let us first consider a VE scenario in which Enterprise A is a bicycle producer and needs to order a certain number of bicycle pedals and bicycle frames to other enterprises in order to use them in the assembly line of its production (see Figure 5). Now let us assume that Enterprise A issues a purchase order to Enterprise B for both of these items. In order to fulfill the order of the bicycle pedal, Enterprise B needs a pedal mould, and some raw material (PVC resin), which are ordered from

Figure 5 - PRODNET demonstration Scenario

Figure 6 - A Part of Export Schema Definitions at Enterprise B

Enterprise D and Enterprise C respectively. Enterprise B produces and supplies bike frames by itself. As a result, Enterprise B generates the local production information on bicycle frame, Enterprise C generates the local production information on PVC resin, and Enterprise D generates the local production information on the pedal mould. Further, assume that Enterprise B is the coordinator in this scenario.

Also, let us assume that the export schema is defined for order data structures at Enterprise B, as it is partly shown in Figure 6, and that the export schema definition for the coordinator is the same in the other nodes. The specific details on how to define the export schemas for DIMS is outside this scope of this paper.

In the next two sections, some specific demonstration cases are presented as examples to show a part of the FQP functionality.

4.1 Example of the Coordination Functionalities Supported by FQP

The DBPMS may need to gather production order information from all the VE partners, which are involved in producing a VE order, in order to check if there is any delay in the delivery date. In this specific example, the VE Coordinator (Enterprise B) calls a high-level function from the DIMS to retrieve the “production orders” information, while specifying the VE and order identifier parameters. The DIMS federated query processor transforms this request into a set of external subqueries, each which will be sent to one of partners involved in this VE. Then the FQP collects the results of the sub-queries (sent to all partners), and merges and returns the final result to the coordinator as described in Section 3.2. (see Figure 7).

4.2 An FQP application example for regular partners

As a second example of FQP applications, the regular partners in the VE may need to ask information that is distributed over the VE network. Now, let us assume that Enterprise C is expecting a delay in its local production plan, and therefore it would

Ve_Id	EnterpriseId	Name	Description	StartDateOfProduct	EndDateOfProduct	DeliveryDate	Status	Progress	
VE_Demo	EnterpriseB	Bicycle Frame	Bicycle Frame	18-May-99	1-Nov-99	3-Nov-99	3	0	
VE_Demo	EnterpriseB	Pedal	Bicycle pedal	22-Aug-99	5-Nov-99	8-Nov-99	3	0	
VE_Demo	EnterpriseC	PVCResin	Bicycle PVC Resin AK 123	1-Sep-99	28-Oct-99	30-Oct-99	3	0	
VE_Demo	EnterpriseD	PedalMould	Bicycle Medal Mould X-34	10-Sep-99	28-Oct-99	30-Oct-99	3	0	
RequestedQuantity		ProducedQuantity	QuantityIn	RequestedLotSize	ProducedLotSize	LotValue	PricePerUnit	LeadTime	LeadTimeIn
100		50	Each	50	50	10	10	5	day(s)
200		100	Each	100	110	15	2	4	month(s)
70		40	Kg	35	25	15	1	20	day(s)
1	1	Each	1	0	20000	12			

Figure 7 - The final result at the Coordinator Node

like to postpone its delivery date of the PVC resin. For this reason, it needs to ask Enterprise B for some production information about the pedal, such as the delivery date in Enterprise B, and also needs to check it against its own production information, so that it can compare the two delivery dates. Then, it could propose a new delivery date for its local production considering the delivery date of the pedal in Enterprise B. This example clearly shows the fact that using DIMS any partner of a VE can also ask certain questions from other partners in the same VE. This scenario case also shows the fact that a regular partner cannot access all the information that can be accessed by the coordinator of the VE, due to the fact that its information visibility level is properly defined through the export schemas according to its role in the VE. In principle, it is not desirable that regular partners can see all the production information that the Coordinator can see in the competitive environment of the VE. For instance, in this example Enterprise B is willing to show only a part of its production order information, which comprises only of some fields (veid, enterpriseld, deliverdate, description and name) of the pedal order to other regular partners. The final result that is returned from Enterprise B to Enterprise C is represented in Figure 8.

5. CONCLUSIONS

In the VE environment, the coordination of the partner enterprises and their business processes associated with the VE orders, is a crucial issue to achieve the common goals of the VE. In general, VE coordination is a complex process that requires advanced distributed information access mechanisms involving several VE member enterprises. Considering that the business process data is distributed over the nodes of the VE, the needs of the VE coordinator can only be supported through an advanced federated database architecture and specifically, through a federated query processing such as the approach described in this paper. The federated query processing mechanism developed for the DIMS module in the PRODNET project provides controlled access to data dispersed over the nodes of the VE network, with

Ve_Id	Enterpriseld	Name	Description	StartDateOfProduct	EndDateOfProduct	DeliveryDate	Status	Progress	
VE_Demo	Enterprise C	PVCResin	Bicycle PVC Resin AK 123	1-Sep-99	28-Oct-99	30-Oct-99	3	0	
RequestedQuantity		ProducedQuantity	QuantityIn	RequestedLotSize	ProducedLotSize	LotValue	PricePerUnit	LeadTime	LeadTimeIn
70	40	Kg	35	25	15	1	20	day(s)	
Ve_Id	Enterpriseld	DeliveryDate	Description	Name					
VE_Demo	Enterprise B	8-Nov-99	Bicycle pedal	Pedal					

Figure 8 - The final result at Enterprise C

respect to the exported data definitions (export schemas) and the hierarchy of roles played by each VE node. In the future, extensions of the FQP functionalities developed for DBPMS will be extended to support a wider range of applications.

6. REFERENCES

1. Afsarmanesh, H; Garita, C.; Hertzberger, L.O.- Virtual Enterprises and Federated Information Sharing. In Proceedings of the 9th International Conference on "Database and Expert Systems Applications", DEXA'98, pages 374-383, Vienna, Austria, August 1998.
2. Afsarmanesh, H; Garita, C.; Ugur, Y.; Frenkel, A.; Hertzberger, L.O. - "Federated Information Management Requirements for Virtual Enterprises". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, pages 36-48, 1999 (a).
3. Afsarmanesh, H; Garita, C.; Ugur, Y.; Frenkel, A.; Hertzberger, L.O. - "Design of the Federated Information Management Architecture for Prodnnet". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, pages 127-146, 1999 (b).
4. AMICE, *CIMOSA: Open Systems Architecture for CIM*, Springer-Verlag, Berlin, 1993.
5. Camarinha-Matos, L.M.; Afsarmanesh, H. "The PRODNET Infrastructure". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, 1999 (a).
6. Camarinha-Matos, L; Lima, C.P. "PRODNET Coordination Module". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, 1999 (b).
7. Camarinha-Matos, L.M. et al. "The PRODNET Demonstrator". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, 1999 (c).
8. Garita, C.; Afsarmanesh, H.; Hertzberger, L.O. – "The PRODNET Cooperative Information Management for Industrial Virtual Enterprises", accepted to the International Journal of Intelligent Manufacturing on April 2000.
9. Kapsammer, E; Wagner, R.R. "The IRO-DB Approach Processing Queries in Federated Database Systems", in Proc. of Eight International Workshop on Database and Expert Systems Applications DEXA'97, IEEE Computer Society Press, Toulouse, France, 1997.
10. Klen, A.P.;Rabelo, R.J.;Spinosa, L.M.; Ferreira, A. - "Distributed Business Process Management", in Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, 1999.
11. Lee, E.; Noah, W. – An Enterprise Intelligence System Integrating WWW and Intranet Resources. in Proceedings of the Ninth International Workshop on Research Issues in Data Engineering – RIDE-VE'99, IEEE Computer Society, Sydney, Australia, March 1999.
12. Meng, W.; Yu C. T. "Principles of Database Query Processing for Advanced Applications", Morgan Kaufmann Publishers, 1998.
13. Osorio, L.; Antunes, C.; Barata, M. "Communication Infrastructure". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, 1999.
14. Radeke, E. et al – Distributed Information Management in Virtual Engineering Enterprises by GEN, in Proceedings of the Ninth International Workshop on Research Issues in Data Engineering – RIDE-VE'99, IEEE Computer Society, Sydney, Australia, March 1999.
15. Wiedijk, M.; Afsarmanesh, H.; Hertzberger, L.O. - Co-working and Management of Federated Information-Clusters. 7th Int. Conf. on Database and Expert Systems (DEXA'96), Lecture Notes in Computer Science 1134, pp 446-455. Springer Verlag, Zurich, Switzerland, September 1996.
16. Zarli, A. et al – A Framework for Distributed Information Management in the Virtual Enterprise: the VEGA project". In Infrastructures for Virtual Enterprises- Networking Industrial Enterprises (L.M. Camarinha-Matos and H. Afsarmanesh, Editors), Kluwer Academic Publishers, ISBN 0-7923-8639-6, 1999.