8 THE PRODNET ARCHITECTURE

L. M. Camarinha-Matos[1♦], H. Afsarmanesh[2]
[1]New University of Lisbon, Portugal
[2]University of Amsterdam, The Netehrlands

The PRODNET II consortium designed and implemented an open infrastructure to support industrial virtual enterprises with special focus on the needs of small and medium sized enterprises. The proposed architecture is described summarizing its main components. The inter-modules communication mechanisms are presented and an illustrative example is given. An approach for hierarchical coordination is proposed and examples of advanced coordination functionalities are described. Finally the creation and configuration steps for a virtual enterprise are discussed.

INTRODUCTION

Basic architecture

The development of an open and flexible support infrastructures represents a major requirement for the wide adoption of the Virtual Enterprise (VE) paradigm in the industrial world. Recent advances in communications and computer networks represent enabling factors for virtual enterprises but there are still many unresolved issues and challenges to be dealt with in the development of supporting platforms. It is necessary to design reference architectures, develop and improve information interchange standards for various classes of information, address interoperability and information integration and exchange issues, and design new coordination approaches. The development of appropriate re-engineering methods and the integration process of the existing legacy systems in each enterprise also represent major tasks.

The PRODNET II consortium designed and developed an open infrastructure to support industrial virtual enterprises (Camarinha-Matos et al., 1998), (Camarinha-Matos, Afsarmanesh, 1998). The focus of this work was put on the needs of the small and medium size enterprises (SME), but the achieved results can be as well applied to large companies.

♦ Corresponding author address: Universidade Nova de Lisboa, Quinta da Torre, 2825 Monte Caparica, Portugal, tel. +351-1-2948517, fax +351-1-2941253, e-mail: cam@uninova.pt.

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

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

Figure 1 shows the basic PRODNET architecture that comprises three main blocks: The PRODNET Cooperation Layer (PCL), the Advanced Coordination Functionalities and the Company's Internal Module.

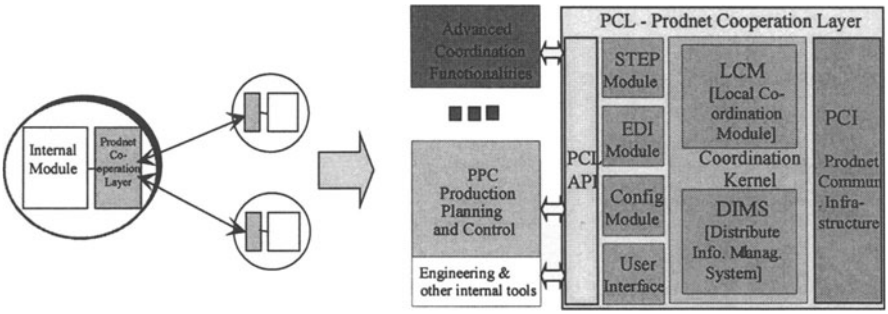

Figure 1 – Basic PRODNET Architecture

INTERNAL MODULE

The so-called Internal Module represents the various legacy systems and applications that run in the enterprise like a Production Planning and Control (PPC) system or Enterprise Resources Planning (ERP) system, a CAD system, a Product Data Management System (PDM), other engineering and internal tools.

PPC - Production Planning and Control System

This is the most important component of the internal module of a VE node. It includes functions such as: (i) Industrial logistics management: Orders flow management, Product data management, Sales forecasts handling, and Actual requirements planning; (ii) Master production scheduling; (iii) Production control; (iv) Quality control / Tracking; and (v) Industrial costing.

Building a system to integrate all manufacturing PPC and the engineering support components has been the focus of attention of many research activities during the last years, and is outside of the scope of PRODNET II. Within this project, the PPCs of different enterprises may collaborate through the Cooperation Layers, where every enterprise's Cooperation Layer only deals with those parts of the PPC, ERP and PDM information the company decides to make available to other VE partners.

Extensions and re-engineering to these legacy systems are also necessary to support the interactions with the cooperation layer.

The electronic exchange of orders is a basic functionality to be provided by a VE supporting platform. Besides the exchange of the order itself, it is necessary to support its follow up in order to cope with, for example, delays in orders processing, temporary incapacity of a supplier, changes in incomplete orders, the need to re-adjust delivery times, and so on. A client company might even want to know details about the manufacturing status of ordered products or components in order to prevent any difficulties for itself. Orders management is therefore, one of the most important functionalities to be supported in a VE environment.

In the PRODNET II architecture it is assumed that efficient orders management is only possible if a company already has a PPC system which provides the company's internal orders management. Otherwise it seems unfeasible, considering the amount and type of information required to be handled and decisions that need to be made, to perform such management by a component like PCL. Thus, a basic requirement in order to accomplish the orders management is the existence of a strong interaction between the PCL and the PPC system. This implied some reengineering of the legacy PPC system in order to make it reactive to requests coming from the network and to make it able to understand the adopted EDI format.

In order to investigate the necessary extensions and re-engineering in a typical PPC system, an existing system (from the CSIN partner) was used as a test-bed (Figure 2).

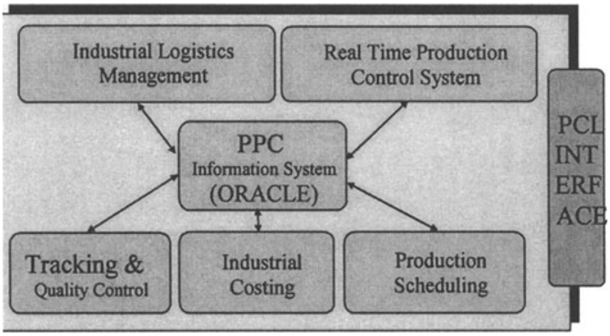

Figure 2 - The architecture of the re-engineered PPC in PRODNET

PRODNET II considers three types of orders: *Regular*, *Incomplete* and *Imprecise* (Camarinha-Matos and Lima, 1998a). There is no vague piece of information (incomplete or imprecise data) in a regular order, which means all information related to that order is known. On the other hand, an incomplete order has at least one piece of data missing, whilst an imprecise order has at least one piece of data specified in an uncertain way. These two types of orders require a special handling process as the data required to fulfill the order will arrive incrementally. In spite of the incompleteness or vagueness of information, it is often the case that a company starts the production based on the available information as well as propagating that vagueness to its suppliers. This functionality was developed in the PPC system, as an extension to the legacy system of CSIN.

Another important aspect when a company gets involved in a VE environment is the proper management of quality related information. In order to allow the definition of liabilities, regarding quality in a scenario of distributed manufacturing, it is necessary that each node is able to track all components involved in its products. This functionality was also included in the developed system.

Engineering tools

Management of product data is a very important and difficult task for any industrial company, which in turn justifies the emergence of a new class of tools known as

PDM (Product Data Management) systems. In fact there are a lot of interdependencies between engineering and business data. For instance, the PPC system needs the Bill of Materials (BOM) for the products designed by the engineering sector. Having a single large system to handle all of these data is not a realistic situation at the industries today. Therefore, the integration of different applications such as PPC, CAD and PDM is a common approach followed by manufacturers.

In PRODNET II some legacy tools from the ProSTEP partner (the PDM Editor and the INTRAVISION CAD viewer) were used in order to investigate the interactions between these components and the PCL. All product data exchanges between members of the VE are based on the STEP standard, namely the AP214, as illustrated in Figure 3.

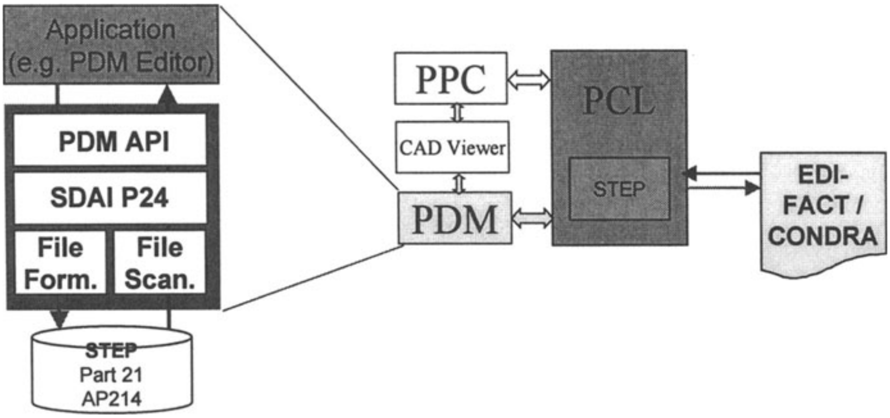

Figure 3 – Integration of STEP data

As the PRODNET architecture supports STEP data exchanges, and there are several legacy CAD systems that have interfaces to STEP, no particular CAD system was selected for PRODNET. However, a STEP-compliant CAD viewer is used in order to support the product negotiation phase in a VE, even when a SME doesn't have a STEP compatible CAD system.

THE PRODNET COOPERATION LAYER

The PRODNET Cooperation Layer (PCL) is the main component of the PRODNET infrastructure. This layer is responsible for supporting all interactions between a company and its VE partners.

DIMS - Distributed Information Management System

The Distributed Information Management Subsystem (DIMS) in the PRODNET Cooperation Layer is responsible to model and manage the exchange of all integrated cooperation-related information, while preserving the autonomy and information privacy of involved enterprises. Different levels of information visibility for other enterprises, defined at every enterprise, guarantee the enterprise's autonomy on sharing its information. Furthermore, the proper privacy of information is preserved when an enterprise is involved in more than one VE.

DIMS is therefore responsible for the modeling, representation, management and access to the data necessary to support both:

(1) intra-enterprise information exchange – a specific set of query-update service functions are developed for interactions with DIMS, namely to be used by the enterprise's internal modules, and the other sub-components of the PCL infrastructure;

(2) inter-enterprise information exchange for necessary interoperation among autonomous enterprises, through supporting the following features:

- Federated cooperative architecture (no data-redundancy for the sake of information sharing, no centralization of data/control)
- Provision of access rights based on the individual *role* of every other enterprise
- Preserving the security/visibility of shared data within one VE and among different VEs.

Figure 4 shows the main components of DIMS developed by the University of Amsterdam (Afsarmanesh et al., 1998).

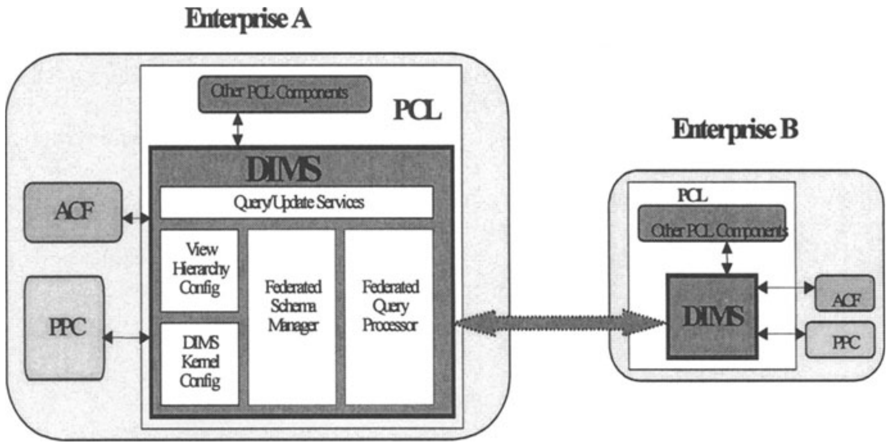

Figure 4 - The internal components of DIMS

The EDI Module

This module is responsible for preparation (formatting) and receiving (decoding) orders and other business-related messages in EDIFACT format. Among its other functionalities, it will check and parse EDIFACT syntax (for various versions of the standard), check for the completeness of messages contents, and generate appropriate formats for sending out the EDI messages. It will also detect and extract EDI messages that may embed some information represented in other formats, such as the order-associated STEP specifications (via the CONDRA message).

Figure 5 shows the main components of this module developed by the Lichen Informatique.

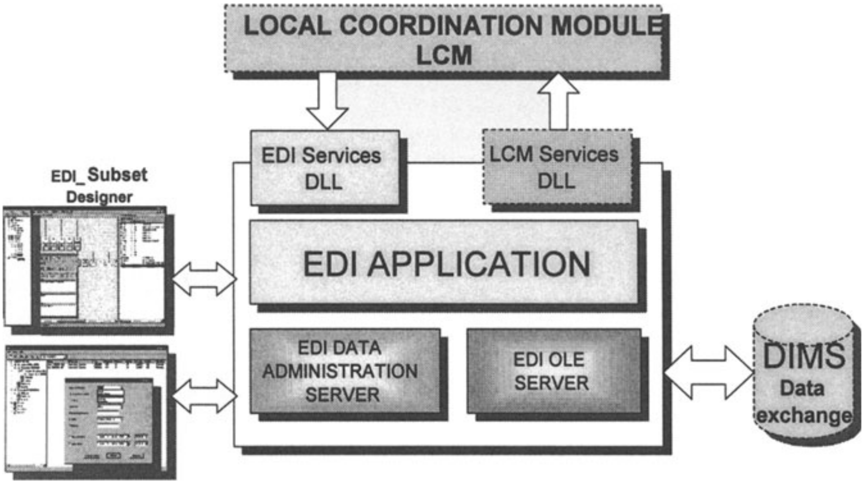

Figure 5 - Integration of EDIFACT in PCL

The STEP Module

The function of the STEP module is to handle the technical product data used within the PRODNET. Ideally all product data should be exchanged in STEP format. The STEP services provided to PRODNET will allow for the transmission and reception of STEP files that have been encoded according to a defined schema, as described in Part 21 of the STEP standard. In particular the AP214 is considered. See Figure 3 representing the components and the architecture of the STEP standard support in PRODNET. STEP data handled within PRODNET can then be accessed through the usage of the Standard Data Access Interface (SDAI), or via a SQL interface.

The ProSTEP company is the responsible for this module.

PCI - PRODNET Communication Infrastructure

The PCI module in PCL is responsible for handling all communications with the other nodes in the network. It includes functionalities such as: Selection of communication protocols and channels; Basic communications management; Privacy mechanisms (cryptography); and Secure message communication channels between nodes.

The main characteristics of the module developed by the ESTEC company are:

- Support for Multiple Communication Resources
- Message Delivery under Quality Constraints
- Message Privacy, Integrity and Authentication
- Intelligent Communication Management.

Figure 6 shows the structure of the communications infrastructure.

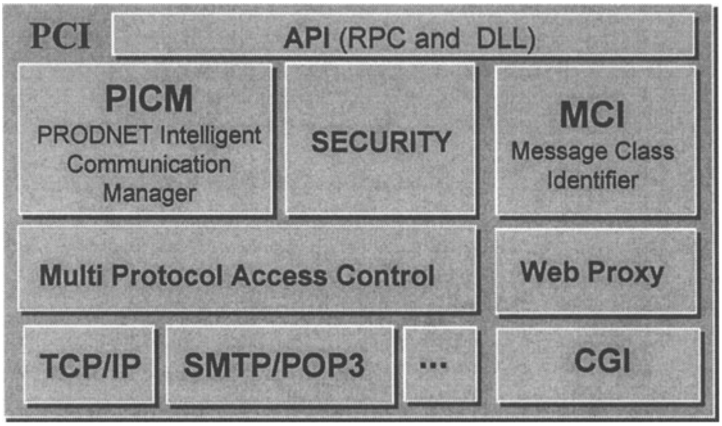

Figure 6 - The PRODNET Communications Infrastructure

Regarding the security aspects, the following features are included in PCI:

- Symmetric and Asymmetric Cryptography
- Authentication based on Signed Certificates
- Adoption of (emerging) Security Standards
- Digital Signature & Certificate Management
- Tunable Security Level.

Complementarily, the following information auditing functions are provided:

- Communication Events Logging
- Registration of Security Exceptions
- Evaluation of Communication Quality
- Legal Conformity Awareness.

Configuration and User Interface modules

The PRODNET platform is intended to support a large diversity in virtual enterprises and their interconnection and inter-operability modes. This means support for a large heterogeneity in terms of the available / installed services and management procedures in every enterprise. Therefore, it is necessary to specify the desired cooperation behavior of every enterprise by an explicit plan. Namely, each enterprise has to define its particular activity flow plans that will be "executed/controlled" by the Local Coordination Module of PCL.

Additionally, the Configuration Component will allow a specification of the structure of the VE, and the involvement characteristics of all its members. The User Interface component of PCL provides interaction facility between the human operator (responsible for the interactions of the enterprise with the VE) and the PCL. The level of human intervention in this process will depend on the internal policy of each company and will be specified at the configuration phase of the VE creation through the configuration specification and the workflow plan definition.

LCM - Local Coordination Module

This component acts as a workflow engine, according to the reference model of the Workflow Management Coalition and is the "executor/controller" of the activity flow plans defined by its Configuration Component (i.e. the graphical workflow editor). Namely, it is responsible for the behavior of the PCL internal modules interactions. The LCM handles all cooperation events with the VE enterprises according to the specified rules for this particular enterprise. These events have an asynchronous nature and are either provoked by other nodes of the VE, by the Internal Module of the enterprise, or by the human operator through the Human Interface.

Figure 7 shows the main components of the LCM system developed by the New University of Lisbon (Camarinha-Matos and Lima, 1998b).

The main characteristics of this module are:

- Compliance with the Workflow Management Coalition reference architecture, namely in terms of accepting WPDL (Workflow Process Definition Language)
- Coordination engine, supporting multiple workflows and sub-workflows, automatic invocation of services to implement workflow activities, management of parameter passing, conditioned transitions, AND/OR splits and joins, temporized and cyclic activities.
- Graphical-oriented coordination monitor to inspect and interact with the processes' execution.
- Interface with a Graphical Workflow Editor.

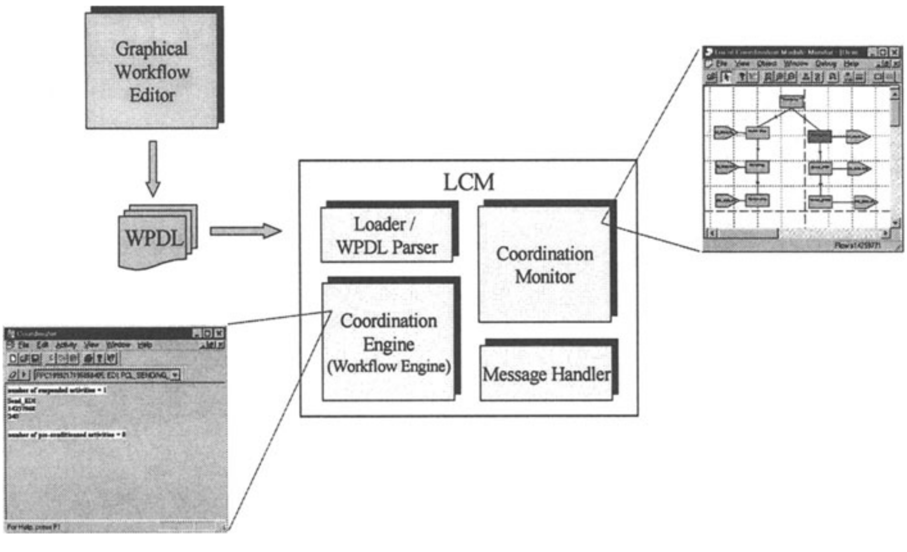

Figure 7 - Local coordination in PCL

INTER-MODULES COMMUNICATION APPROACH

Basic aspects

There are three types of services defined inside PCL to carry on the general functionalities of the enterprise through its modules: *Core*, *Auxiliary* and *Internal*. Core Services are "macro services" offered by PCL and coordinated by LCM to the legacy systems, such as: PCL_SendingOrder, PCL_ReceivingCondraMessage, PCL_GetBpKey, etc. Auxiliary and Internal services are defined by different modules inside PCL, and are classified based on their relation with the coordination task performed by LCM. For instance, DIMS_GetStepDataExchange is an internal service to PCL, which is used by the EDI module to read STEP data from the DIMS. On the other hand, DIMS_AttachFilesReference is an auxiliary service used by LCM when sending out a CONDRA message.

The invocation of services provided by PCL and its modules, as well as the interfaces among the modules, follow a uniform general mechanism.

The coordination of activities is performed by LCM and suggests some particular features for this common interface definition. First, since LCM has to handle and manage many events simultaneously, a basic requirement for many services inside PCL is to support their asynchronous invocation. Second, LCM has to manage the execution of all core services instances, which means that each instance has to be uniquely identified along its duration. Therefore, a control "tag" should be associated to each core service instance. Third, as LCM needs to receive/send data from/to the legacy systems in order to properly support the core services execution, the interface should support a "data flow" between LCM and legacy systems. The requirement is also applied to the modules' interfaces within PCL, since their auxiliary services are to be used by LCM to properly accomplish a core service execution. Therefore, the common interface should also support the "data flow" among PCL modules.

The core services are realized and configured as workflow plans by a human operator. The activities involved in a workflow plan are implemented via the invocation of auxiliary services of different PCL modules.

PCL Common Interface Services

At the base, the common interface of PCL offers two generic services: *ServiceRequest* and *ServiceAnswer* (Figure 8). The ServiceRequest is used to activate both the core services from PCL or to invoke auxiliary/internal services among PCL modules. The ServiceAnswer is used to return the answer to a particular request previously received. The same uniform interface mechanism, with the ServiceRequest and ServiceAnswer is used both within one PCL, and among the PCLs in two different enterprises.

The templates for both services are shown below, using the "C" notation to represent the interchanged arguments.

ServiceRequest (*PclToken* * token,
PclParametersList inputParameters,
PclServiceResult * outputParameters);

ServiceAnswer (*PclToken* * token,
PclServiceResult outputParameters);

In the definition:

- *token* constitutes all the *mandatory* and *common* information for all PCL modules (as described below);
- *inputParameters* are the parameters required by the service being requested;
- *outputParameters* are used for sending back the results produced by the requested service, in two manners: synchronously, as the output parameters of a *ServiceRequest* invocation, and asynchronously, as the output parameters in a *ServiceAnswer*.

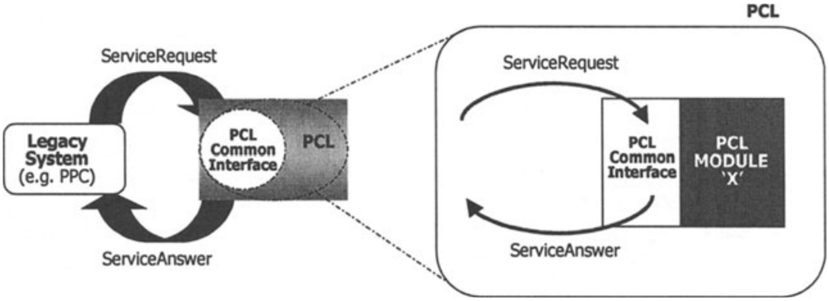

Figure 8 – PCL common services interface

The *PclToken* structure is used mainly for control purposes. It also supports the background context definition of a PCL message that flows among the PCL modules. The PclToken structure is shown below and its attributes are identified.

```
typedef struct{
    Identifier32          tokenId;          // unique identifier
    VENode                sender;           // VE context
    VENode                receiver;         // VE context
    ModuleIdentifier clientModule;          // calling module
    CoreServiceType coreServiceId;          // workflow plan ID
    ModuleServiceType     moduleServiceId;  // auxiliary service requested
    ServiceCallMode serviceCallMode;        // service calling mode
    OpaqueReference    lcmRequestId;        // unique identifier used by LCM
    TimeStamp             creationTime;     // for auditing/log purposes
} PclToken
```

Communication Process and Implementation Paradigm

The PCL common interface has adopted the client-server paradigm for modules' interoperation. Therefore, each module contains its "interface server" supporting the requests that are handled by its "interface client", and running together with the other modules. Figure 9 shows an example of this implementation, considering DIMS and LCM.

It is important to notice that this model is totally independent of the communication process implemented to support the client-server interoperation. Different PCL modules have adopted two different kinds of processes for their interface implementation: RPCs and sockets. The clients are provided as DLLs (Dynamic Link Library) that are loaded by the modules that need to interact with the "DLL provider" (Figure 9).

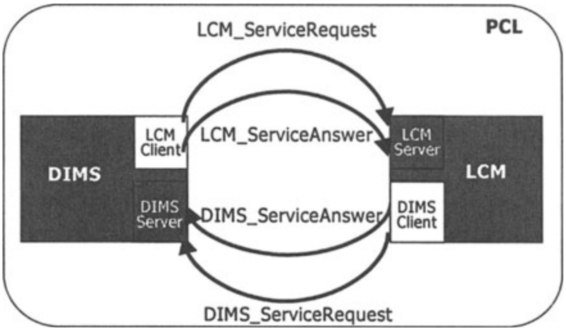

Figure 9 – Communication between modules

A SIMPLE EXAMPLE

In order to illustrate the interactions among the various modules of the PRODNET infrastructure let us consider a simple example involving the sending of an order from a company A to a company B.

Figure 10 shows the main steps involved in this process that must be supported by the infrastructures installed in each company. The various modules involved in the implementation of various steps are also represented on the right hand side of the box representing an activity.

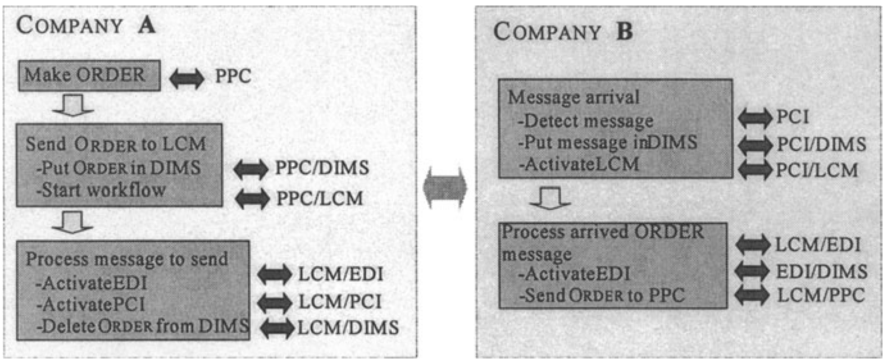

Figure 10 - Sending and receiving an order

The workflow plans implicit in this diagram are configured via the Workflow Graphical Editor and correspond just to one example of a possible enterprise behavior. For instance, in this example, when the order arrives at company B, it is automatically sent to the local PPC, while in another enterprise the procedure might be different. For instance, an alternative workflow could involve a connection to the human interface, in order to get the order acceptance by a human operator.

HIERARCHICAL COORDINATION ARCHITECTURE

Motivation

The modules described in previous sections constitute the so-called PRODNET reference architecture that supports the basic requirements for cooperation among enterprises in a VE environment.

This reference architecture is however, limited if advanced VE-related coordination services are required to be supported. Analyzing potential future scenarios for VE environments, particularly in terms of VE coordination, different levels of requirements can be identified, what suggests a hierarchical coordination architecture. For instance, at least two distinct levels – VE level coordination and VE-member level coordination – should be supported by the infrastructure. Therefore, during the project, the initial PRODNET basic infrastructure evolved into a hierarchical architecture.

Three-layered reference architecture

In order to support the above requirements a 3-layer hierarchical reference architecture is proposed by PRODNET, as represented by the Core Cooperation Layer (CCL), the Enterprise Cooperation Layer (ECL) and the PRODNET Cooperation Layer (PCL), defined hierarchically on top of each other, shown in figure 11. In this architecture, the so called "management / coordination functionalities" are separated from the basic "cooperation functionalities", as described bellow. Namely, every cooperation layer constitutes some management / coordination functionality as well as a lower level of cooperation layer. Please notice that in the new 3-layered architecture the PCL has much more complexity than the PCL defined in the original PRODNET architecture, though using the same name.

Core Cooperation Layer (CCL). The CCL is positioned in the first level (layer 1) and corresponds to the original PCL for an enterprise, as presented in Figure 1.

Layer 2, the *Enterprise Management Functionalities (EMF)* embodies the CCL and EMF, and it is responsible for coordination of the activities at the enterprise level. The EMF deals with the coordination responsibilities of the enterprise towards the accomplishment of its contracts with the VE and other VE-partners. This layer is also based on the workflow management architecture, in order to show the easily configurable behavior of an enterprise, and its structure is quite similar to the structure of the CCL. The services used at this level are related to the execution of business processes at the enterprise level and are supported by the extensions to the PPC / ERP systems. In the PRODNET II project only some simple services to support the ***contracts management*** are implemented at the EMF layer. However, all services related to PPC, ERP and other internal legacy systems can in principle be made available at this coordination level.

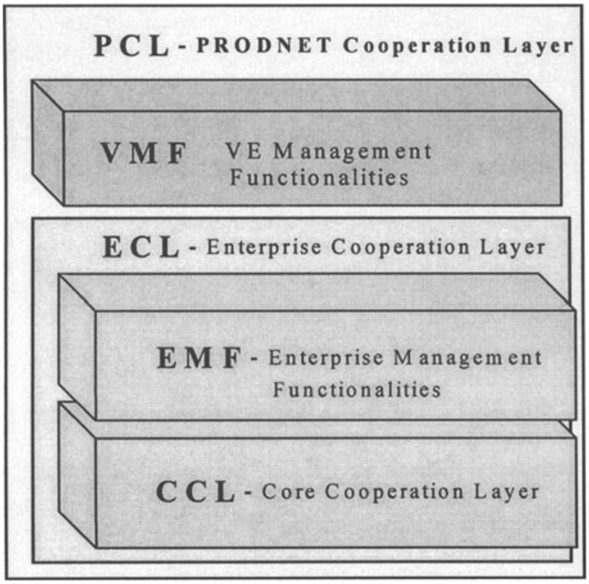

Figure 11 - PRODNET hierarchical coordination architecture

Layer 3, the PRODNET Cooperation Layer (PCL) embodies the ECL and VMF, and is responsible for coordination of activities at the VE level. The *Virtual Enterprise Management Functionalities (VMF)* performs the coordination functionalities at the VE level. Therefore, only the node acting as a VE coordinator will use this component of its PCL to monitor, assist, and modify the necessary activities related to the general VE goal achievement. The VE Management Functionalities (VMF) will resort to the services provided at the ECL of its node to communicate (and cooperate) with the other nodes of the VE. In PRODNET II only two services (called ACF - Advanced Coordination Functionalities) are implemented at this level that consist of:

-Electronic Partners search and selection tool (EPST);

-Distributed business process management system (DBPMS);

but, this infrastructure is developed open, for the future addition of other services.

ADVANCED COORDINATION FUNCTIONALITY

The coordination aspects of a Virtual Enterprise, although very important, still face many open questions, some of them of a non-technological nature. For instance, on one hand the legal barriers and the need for re-organizational changes that imply retraining the people and new roles assignment, even establishing new required power structures among the enterprises, take time to implement and require a very careful approach.

On the other hand, the development of advanced inter-network coordination mechanisms, information sharing, and safety procedures must be accompanied by strong trust building actions, an area that can be technically tackled but whose exact evolutionary shape is hard to anticipate at current stage. These uncertainty factors recommend a step-by-step approach to this problem, instead of an ambitious general infrastructure development approach, as followed by some other projects.

Therefore, PRODNET II followed two complementary approaches:

- A minimal coordination level, mostly based on human decision making, is included in the PCL within its primary reference architecture, as described in the previous sections; and
- A set of experimental components on more advanced coordination support mechanisms, that were investigated and developed within the 3-layered reference architecture.

EPST – Electronic partners search and selection tool

Search and selection of business partners is a very important and critical activity in the operation of a company.

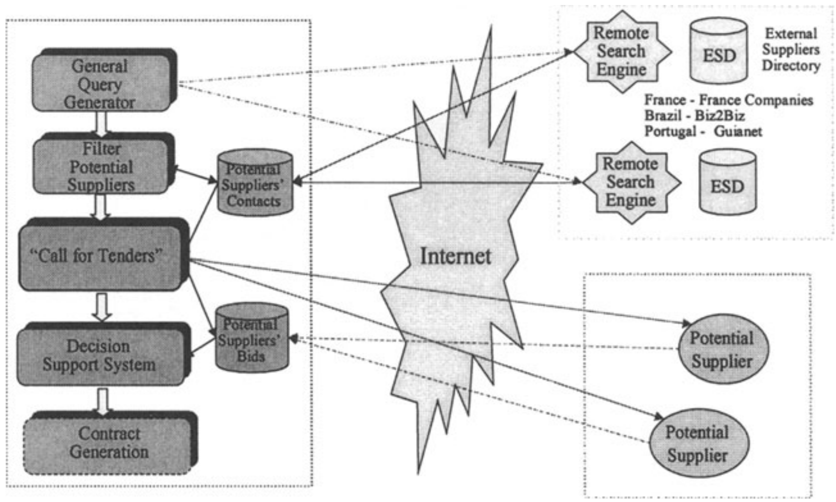

Figure 12 - Partners search and selection

The EPST tool (Figure 12), developed by the New University of Lisbon, Portugal, is a prototype electronic partners procurement tool supports the partners search and selection through the public directories of suppliers available on the Internet. The main characteristics of this prototype include:

- Searches based on access to three directory services: Biz2Biz, France Companies, and GuiaNet.
- Filtering of results according to some user defined criteria
- Generation of Web-based Call for Tenders and e-mail notification for the potential partners
- Bids management
- Simple multi-criteria classification / decision support system for received bids.

DBPMS - Distributed business process management system

The DBPMS prototype developed by the Federal University of Santa Catarina, Brazil (Klen et al., 1998), aims at providing the VE coordinator with the "real-time" information about the current status of the business process execution at the VE-member enterprises. The main components of this module are shown in figure 13.

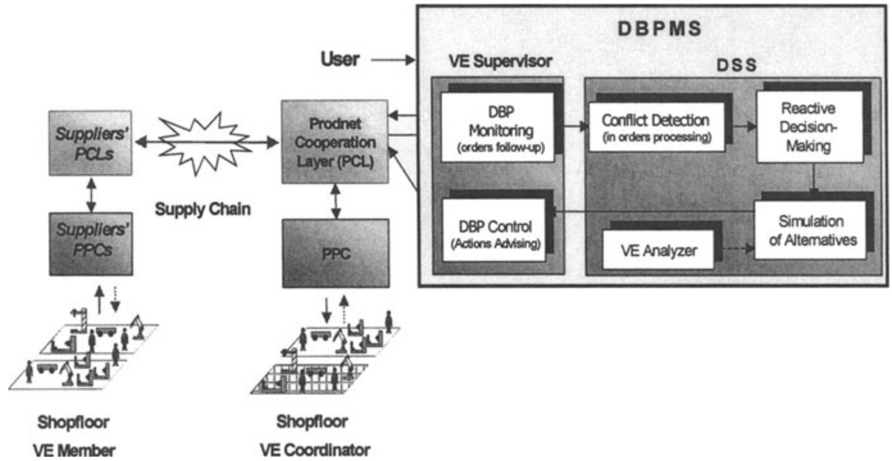

Figure 13 – Distributed business process management functionality

The trust building process among enterprises and the decision to move forward in the integration process at the SME level can be facilitated by the demonstration of such functionalities and their potential benefits.

CREATION AND CONFIGURATION ISSUES

In the design of the PRODNET architecture, a large emphasis was given to developing it as a flexible and configurable environment. Namely, an effective PRODNET infrastructure must satisfy the requirements of the enterprise through different stages of: installation of PRODNET-PCL at the enterprise, enterprise taking part in Ves, and its evolution within the VEs. Therefore, the two main stages include: (1) the *initialization(/reinitialization)* of the PRODNET-PCL to the specificities of the enterprise environment and its business process behavior; and (2) the necessary *configuration(/reconfiguration)* of the PRODNET-PCL, to the VE specifications and regulations, for the enterprise's involvement within different virtual enterprises.

The main outcome of the initialization stage is the proper positioning and manifestation of the enterprise within the "PRODNET network of enterprises", and preparing it for getting involved in VEs. The adjustment of PCL to fit both the internal modules of the enterprise, and the specific behavior and procedures inside the enterprise, is performed at this stage.

The outcome of the configuration stage however, is the proper setting of parameters for involvement of the enterprise within every VE, and support for the readjustment of these parameters, when necessary, during the VE's life cycle. For instance the reconfiguration necessary due to the "feedback" from the VE operation phase, when there is a need to select a new partner.

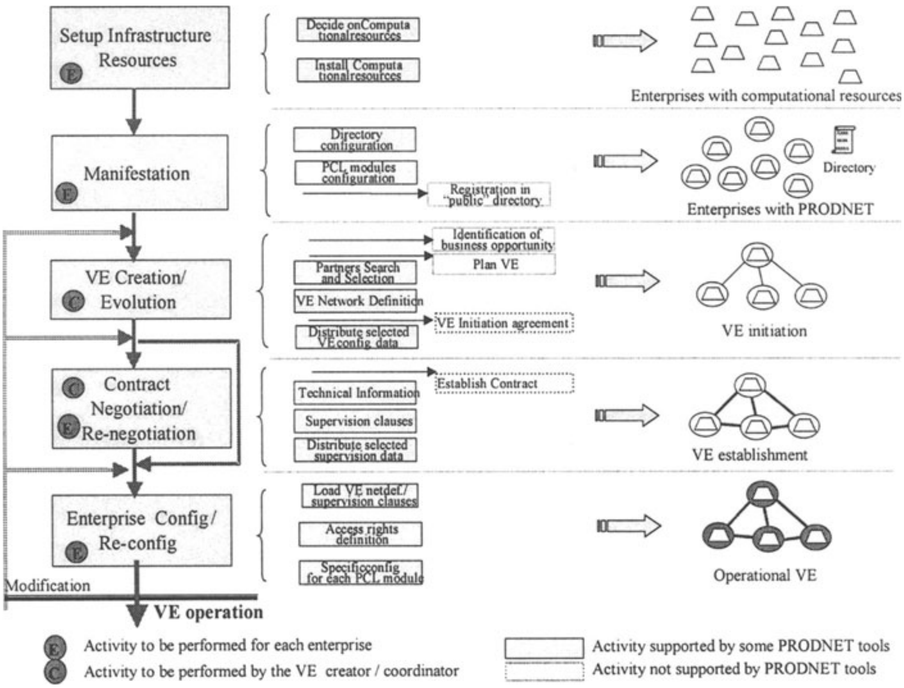

Figure 14 – Main steps in VE Initialization and configuration

In PRODNET we have identified the following main steps for the initialization and configuration stages, in order to tune the PRODNET-PCL to the specificities required for both the environment and functionalities of an enterprise. The main steps in the general process are illustrated in Figure 14. A brief description of every step follows:

- **Setup of Infrastructure Resources**: this step identifies and installs the computational resources for the enterprise. This includes the setting up of: the operating system, the database management system, the connection to Internet, etc., that is needed for the proper functioning of the PRODNET-PCL components.
- **PRODNET Enterprise Manifestation**: once the enterprise has properly installed its resources, as specified in the previous step, it needs to set up the PRODNET-PCL components themselves, what is required to prepare the node for its future involvement in VEs. At this step the enterprise can "proclaim" itself as an enterprise that is enabled with the PRODNET Cooperation Layer. The enterprise announces itself to a public directory. The directory includes

general enterprise information, such as the interest areas/services and general company profile, as well as its "default" PCL configuration.

- **VE Creation/Evolution**: when a business opportunity is identified, the proper VE partners are searched, identified and selected. Some preliminary agreements ("initiation agreement") will be reached, and then the VE topology and its partners' roles are defined and distributed among the VE-members. The same step needs to be followed, also in the case of a VE evolution. For instance, when there is disturbance in the production plan of the VE, the VE coordinator needs to select one or more new partners and modify the topology and agreements within the VE.
- **VE Contract Negotiation/Re-negotiation**: once the topology of the VE and the role of its partners are defined, formal contracts need to be signed between the VE coordinator and individual partners. The contracts contain the rights and liabilities of enterprises towards the VE, and their agreement to the global VE rules and regulations. In specific, both the technical production plans for the enterprises and the "Supervision Clauses" that define the rights of the VE coordinator to observe and monitor the progress of activities at the VE-member sites. Selected parts of the supervision clauses set will then be distributed among the VE-members. Due to the VE evolution, clearly new contracts may be negotiated among VE partners.
- **Enterprise Configuration/Re-configuration**: At this step, every VE partner loads the VE network definition, regulations, and supervision clauses, received from the VE coordinator. Based on these definitions, the enterprise will set the precise information access and visibility rights for every other VE-member. Furthermore, at this step, the specific configuration and tuning of each PCL component will be carried out, in order to support the specific requirements and necessary platform for the VE.

Please notice that some of these steps will be taken only by the creator/coordinator of the VE, while the others will take place at every enterprise, as indicated in Figure 14. This figure also specifies those activities for which some support software is developed, within the PRODNET project.

CONCLUSIONS

The PRODNET II project has developed an open infrastructure that addresses a wide range of requirements set by the VE paradigm with particular focus on the characteristics of SMEs.

Configurability and flexibility are major requirements for any VE-support infrastructure due to many contradictory criteria under which the enterprise must function, what represented some of the main design criteria for the PRODNET architecture. In particular, the workflow-based hierarchical coordination approach developed by PRODNET II seems particularly adapted to these requirements.

Other important features are the support for safe communications, guarantee of privacy and configurability of levels of information visibility necessary for information exchange, interoperability between the EDIFACT and STEP standards, identification and development of necessary extensions to legacy systems such as a PPC, in order to operate within a VE environment.

In terms of advanced coordination mechanisms, PRODNET II contributed to the areas of electronic procurement of partners and distributed business process management supported by an innovative federated information management approach among VE-member enterprises.

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

REFERENCES

1. Afsarmanesh, H., Garita, C., Hertzberger, L.O. - Virtual Enterprises and Federated Information Sharing, Proceedings of DEXA'98, Vienna, Austria, August 1998, Lecture Notes in Computer Science, Springer-Verlag.
2. Camarinha-Matos, L.M.; Afsarmanesh, H. - Virtual Enterprises: Life cycle supporting tools and technologies, in *Handbook of Life Cycle Engineering: Concepts, Tools and Techniques, A. Molina, J. Sanchez, A. Kusiak (Eds.), Chapman and Hall,* June 1998, ISBN 0-412-81250-9.
3. Camarinha-Matos, L.M; Afsarmanesh, H.; Garita, C.; Lima, C. - Towards an architecture for virtual enterprises, *Journal of Intelligent Manufacturing,* Vol. 9, Issue 2, April 1998, pp 189-199.
4. Camarinha-Matos, L.M.; Lima, C. - A framework for cooperation in virtual enterprises, *Proc. of DIISM'98 – IFIP Int. Conf. On Distributed Information Infrastructures for Manufacturing,* J. J. Mills, F. Kimura (Eds.), Kluwer Academic Publishers, USA, 1998, ISBN 0-412-84450-8.
5. Camarinha-Matos, L.M.; Lima, C. - Configuration and coordination issues in a virtual enterprise environment, *Proc. Of PROLAMAT' 98,* G. Jacucci, G. J. Oling, K. Preiss, M. Wozny (Eds.), Kluwer Academic Publishers, Italy, Sept 98, ISBN 0-412-83540-1.
6. Klen, A.; Rabelo., R.; Spinosa, L.M.; Ferreira, A.C. - Integrated Logistics in the Virtual Enterprise: the PRODNET-II Approach, Proceedings of IMS'98 - 5th IFAC Workshop on Intelligent Manufacturing Systems, Gramado, Brazil, 9-11 Nov 1998.