A Framework for Virtual Enterprise Support Services

Vaggelis Ouzounis, Volker Tschammer
ECCO – Electronic Commerce Center of Competence, GMD-Fokus,
Kaiserin-Augusta-Allee 31, D-10589, Berlin, Germany
Tel: +49 30 3463 7108; Fax: +49 30 3463 8200;
Email: (ouzounis, tschammer)@fokus.gmd.de

Abstract.
The main scope of this paper is to present a set of enabling services that support the establishment and management of Virtual Enterprises and the integration and interoperation of business processes. Based on a virtual enterprise life-cycle model that we propose, different business domains can integrate their business processes in order to provide transparently services to the end-users and customers of the Virtual Enterprise (VE). The appropriate enabling services are identified, analyzed and certain details regarding their design is given. These enabling services are the VE contract manager, the BP administrator, and the BP controller. Underlying services needed for the integration and execution of shared business processes are also proposed. Our proposed enabling services reduce the cost of development, maintenance and integration of shared business processes in VEs by applying interoperable, distributed object oriented technologies.
Keywords: Virtual Enterprises, Business Processes, Contracts, and Business Objects.

1. Introduction

Rapid advances in communications, open networks (internet, WWW), service provision frameworks, interoperable distributed object-oriented technologies and interactivity have been opening and enabling new opportunities for businesses and new effective ways of doing business electronically. Doing business electronically means to shift critical business processes (BP) [1,2] to the open networks (e.g. internet) and enhance customer and supplier relationships through automated IT systems. The major drivers and motivation behind this transformation are low transaction costs, better customer satisfaction, global operation, lower time to market, and rapid adaptation to market changes. Additionally, companies are realizing that in order to seize the opportunities offered by electronic commerce (EC) they should co-operate in product/service development phase, in marketing and sales activities and share their business processes, resources, core competencies, skills and know how.

A virtual corporation is a kind of horizontal and/or vertical cooperation of independent firms with the possible participation of institutions and/or persons to achieve a service or product on the basis of a common understanding of business [2,3,4,5]. Each member of the virtual enterprise will contribute primarily what it regards as its core competencies. There is a time limit on the existence of the virtual corporation caused by fulfillment of its business purpose. From the viewpoint of an external observer, the virtual corporation appears as a unitary enterprise.

Companies participating in a virtual enterprise to share business processes and/or to outsource costly business processes to other members. A business process can be defined as a set of one or more linked procedures or activities, which collectively realize a business objective or policy goal, normally within the context of an organizational structure [3,4,5,6]. In general, every business process solves a specific business problem. In a single business domain environment, business processes managed and controlled by BP administrators that have full responsibility of the overall business processes running across the domain. BP administrators can create, install, configure, integrate and control BPs without any restriction. On the contrary, in a VE environment BPs should be managed and controlled autonomously and in a distributed manner. The sharing of business processes should be regulated through contractual agreements that specify the terms and conditions of the sharing [1,7].

Virtual enterprises are not a new concept, especially in management studies. A lot of big manufacturing companies, and especially car manufactures, has “links” with their suppliers and customers. These links enable the sharing of business processes and activities between them. Although, the level of integration, the flexibility required and the ITC used for enabling these links is not the appropriate one. Most of the activities are still

performed manually, in a proprietary and complex way. The new advances in internet, web and the distributed component-based technology world, enables the establishment and management of VEs in a cost and productive way.

In this paper, we propose a set of general purpose enabling services that allow the establishment and management of VEs. In particular, in section 2 a state of the art is given and certain information concerning emerging interoperable distributed-component based technologies is provided. In section 3, key technical requirements that the VE enabling services should captured are discussed, while in section 4 a business domain architecture is given. The business domain architecture outlines the basic layers needed for the development of open, secure, interoperable distributed business systems. In order to identify the necessary building blocks required by VEs, in section 5 we propose a life cycle model that specifies all the appropriate steps needed during the establishment, management and execution of a virtual enterprise. Finally, in section 6 we propose a set of basic VE enabling services needed during the whole life cycle model. These components are being developed and integrated into our business domain architecture. Finally, in section 7 and 8 we present our VE enabling services and give information regarding our experiences during the design and development phase.

2. State of the Art

Enterprise systems have been evolved according to the technology changes and evolution. The first approach used in developing enterprise systems was custom-engineered solutions. The provided solutions were in general closed and tailored to company's needs and requirements. The development time and the cost were rather high, while the maintenance and re-engineering was also difficult and ineffective. These solutions adopted mostly by big companies due to the high cost and are characterized by inflexibility, low degree of interoperability, limited security and low level of scalability. The main reason of failure of these systems was the high re-engineering time, i.e. the time needed to change existing business processes in order to respond to the market needs [1,6].

As solution to the above problems, the development of enterprise systems based on proprietary, modular, software modules was proposed. Solutions based on this method were more flexible and rather more maintainable. The cost was medium, but still the problem of interoperability, scalability and evolution according to the business needs still remained.

Current enterprise systems are composed of different products from different vendors and require increased level of interoperability and automation. Current organizations are more distributed and require distributed solutions with increased level of scalability. Current business systems should respond to rapidly changing market needs and conditions. In order to achieve this, the development time of enterprise systems should be minimized and this means that special reusable concepts must be introduced. As solution to the above problems, the distributed component-based systems are being proposed. These systems are composed of basic building blocks, mostly based on object-oriented technologies, that can be bought off the self and reused, customized, configured and integrated into the overall business information systems. This approach provides solutions with low cost, minimum development time, easy maintenance, accepted level of interoperability, scalability and distribution. These technologies are actually integrated development and run time environments that isolate much of the conceptual and technical complexity involved in building business applications. It seams that solutions based on these concepts are perfectly suited to current business needs and requirements.

Enterprise systems based on these concepts are gaining momentum and more commercial products are on the market. Two are the market leaders in this area, the DCOM [14] from Microsoft Corp. and the JavaBeans from Sun Micr. DCOM is a programming language and implementation independent and platform dependent, mostly based on Windows NT systems, while JavaBeans [13] is a programming language dependent and platform independent, due to the Java Virtual Machine concept. For both products there is an integrated development environment for component development and reuse. Unfortunately, both products are not fully complied with CORBA [8,9] specifications in the area of business objects and business component architecture. The Object Management Group (OMG) has played a leading role in establishing open systems specifications for distributed object computing. Until recently, OMG focused on object-level standards. Responding to ease-of-use concerns from members aligned with the JavaBeans standard, the OMG issued a request for proposals for common business component-level-specification last year. Similar market pressures will likely lead to analogous OMG specifications for the interoperability of JavaBeans and DCOM with Corba architecture.

Although there are significant developments in the area of enterprise applications, services and platforms, it is still problematic how different business domains can share services and business processes in a dynamic and

flexible way. The main reason for this failure was the existence of different communication protocols in the past that made this inter-domain cooperation almost infeasible. The advent of internet and the acceptance of TCP/IP and HTTP provide the technological basis for the establishment and management of virtual enterprises.

3. Technical Requirements

Virtual enterprises are becoming feasible on account of technology developments including CORBA, the WWW, and distributed component-based software. From technical point of view, the development of a virtual enterprise involves:

- the establishment of the VE enterprise,
- the management of electronic contracts that specify terms and conditions concerning the services provided,
- the assignment of services and events to contracts,
- the re-engineering of business processes,
- the control and the evaluation of the performance of the shared business processes,
- the invocation of remote business services,
- the handling of business events, and
- the access control and security management.

Within the object oriented and component-based world, solutions to some of the above problems exist. But their integration and interoperation is still elusive.

In order to materialize the concept of VE, current and near future information systems should provide a flexible VE management platform that enables the specification, maintenance, execution and management of interactions occurred across business domain boundaries [11]. These interactions can be simple ones, like the invocation of a service located in a different domain, or more complex ones, like the retrieval of production and marketing data based on specific constraints and the payment for the usage of them. Thus, different sequences of interactions might be occurred with different management needs.

In order to facilitate such kind of scenarios the following key characteristics should be captured:

- specification of the domains that participate in the VE,
- specification of the services that will be provided and invoked,
- specification of the rules and constraints that regulate their co-operation,
- provision of the services and management of the resources needed,
- management of access based on the above specifications,
- management of business events generated during the service provisioning,
- integration of services and business processes in the VE enabling platform, as well as, new ones in an easy and dynamic way.

The main scope of this paper is to propose a set of enabling services that satisfy the above requirements in a consistent and efficient way.

4. The Business Domain Architecture

In order to support the appropriate level of integration, flexibility, scalability and interoperability, a new corporate architecture should be developed. In the following, a layered architecture is proposed based on state of the art technologies. This architecture is providing a platform on which applications, business processes, and business process components can be integrated and managed, in terms of service components. This architecture is our key reference model for current as well as for future implementations. The architecture is composed of the following layers:

- **Secure interoperable infrastructure**. Provides the networking environment for access to services in an interoperable and secure way, (e.g. Corba Platform),
- **Enterprise wide infrastructure.** Provides the interoperable environment and framework required in supporting integration and interoperation of business processes. It also provides the interfaces and protocols for the application components to collaborate and the support mechanism for “Plug & Play” application component integration, (e.g. JavaBeans),
- **Common business objects.** These are objects that provide basic functions used by different upper layer business processes and components (e.g. business objects from OMG),
- **Specialised business processes.** Specialised, application and domain dependent BPs that solve critical business problems inside on domain, like ordering, payment, order tracking. These processes can be seen as service components used by different administrative domains and departments, like sales, production, financial, public administration, etc. (e.g. legacy systems, EDI systems, workflow processes, workflow objects, etc.)
- **VE enabling services**. These are front-end enterprise-wide services that provide the ability to various companies to form VEs.
- **BP management services.** Services that manage the secure access to services, dynamic service execution, accounting, trouble ticketing, performance control, billing, etc. (e.g. TINA, NMF, etc.)

As stated previously, there are significant developments in most of the above layers except in the area of VE enabling services. Most of the developments in the area of VEs are proprietary solutions integrated into the applications. On the contrary, separate, application independent services are required. These services are vital for the establishment and management of virtual enterprises. Additionally, the integration and interoperation of them within the business domain architecture is of equal importance and should be supported in efficient way.
In the following Figure 1 the business domain architecture is depicted.

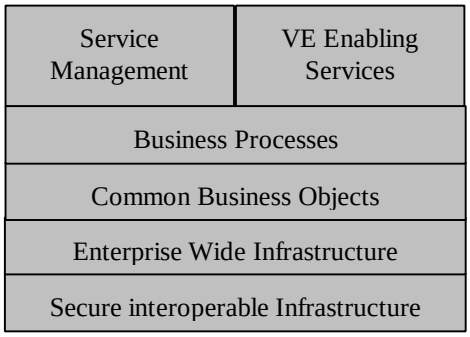

Figure 1: Business Domain Architecture

5. A Virtual Enterprise Life-Cycle Model

In order to identify the necessary VE enabling services, we analyzed the steps and activities required for the cooperation of two or more business domains. Most of these activities are currently performed manually, require significant time, there are no tools and methodologies that can be applied easily, while the integration with existing services and BPs is very difficult. Main focus of our work is to automate, whenever possible, some of these activities and propose services that can be used for the management of them.
In general, the interoperation of shared services and business processes across different business domain boundaries involves a set of actions that can be categorized into three respective phases.
Establishment Phase: during which members of VE are establishing, and configuring the linkage between them. This phase includes the:

- initial negotiations between organisations to agree on specific set of business process interfaces that will be provided by one member to another under certain terms and conditions including security, authentication, payment, reliability and fault tolerance,
- specification of contracts that regulate all the terms and conditions of the service provisioning process between domains
- modification of access rights to allow restricted and secure access to services and business processes based on the contracts,
- configuration and customisation of access control, authentication and authorisation systems for the secure provisioning of business processes, and services to the VE members according to contracts,
- business process re-engineering of current business processes so as to interoperate with processes located in other business domains.
- integration of business processes with external business processes provided by members so as to ensure autonomy, transparency, scalability and distribution,
- testing and performance analysis of the newly developed shared processes and removal of bottlenecks and de-efficiencies,

Provision Phase: during which the provided services can be accessed and invoked in a secure and modest way. This phase includes the:

- invocation of business processes transparently by each member of the VE according to contracts established in the phase one,
- management and control of invoked business processes and resources during real time operation by each domain indepedently,
- management and handling of business events,
- on-line performance evaluation and control of business processes,

Termination Phase: during which the VE members can alter the access rights, interfaces and implementations of the provided services and business processes. In this phase each member can change the contracts, the access control rights, business process interfaces, implementations and in general can re-engineer business processes so as to reflect the new needs of the organization.

6. Virtual Enterprise and Business Process Semantics

A virtual enterprise can be considered as a collection of different business domains that cooperate together by sharing business processes. The sharing process between two or more domains is regulated by peer level contractual agreements between the members of a VE. A business domain participating in a VE is called member of the VE. A business domain may participate in more than one VEs at the same time. VEs can be established in a centralized or on an autonomous, decentralized way. A

VE is established in a centralized way when one domain has total control over the specification, management and maintenance of contractual agreements. Such example may occur in the case when a big company set-ups a VE with a set of suppliers which are small companies. In that case the big company has full control over the contracts. A VE is established in an autonomous and decentralized way, when every company has full control over the contracts and selects at least one company to cooperate with. Such example occurs in a network of small companies that cooperate to produce a tangible product. In this case, a web of companies is created in a incremental way. The following concepts can be easily adapted so as to be used for both centralized, as well as for decentralized VE.

Therefore, in our concept a contractual agreement is called **Virtual Enterprise Contract (VEC)** and consists of:

- a set of generic information about the contract, like duration, start time, etc.
- a set of business domains that contain infos like the address, telephone, director name, etc. and technical characteristics of these domains, like the server ids, the www server used, security data, etc.
- a set of service interface specification offered and deployed by the domains (e.g. Corba IDL interfaces)
- a set of events that can be posted from one domain to another,
- a set of terms regulating the access and provisioning phase (e.g. payment issues, authentication, etc.)
- a set of If-then rules which are the constraints of the contractual agreement.

In figure 2, a class diagram representing the key subclasses of a VE contract is represented. The definition of the basic subclasses of a VE contract are:

- **Generic**: contains info's about duration, start time, description of legal terms and conditions, etc.
- **Domain**: contains administrative info's about the organisations, like address, tel., etc. and as well as technical info's, like server ids, web servers, etc.
- **Service**: contains info's about the provided service within the context of this contract, as well as, the interfaces of these service, performance constraints, etc.
- **Event:** contains infos about the events that can be posted and consumed to/by other domains
- **Term**: contains infos concerning the payment, access rights, authentication, authorisation, billing, etc.
- **Rule**: contains infos regarding the constraints of the contract, like payment discount, exception handling, etc.

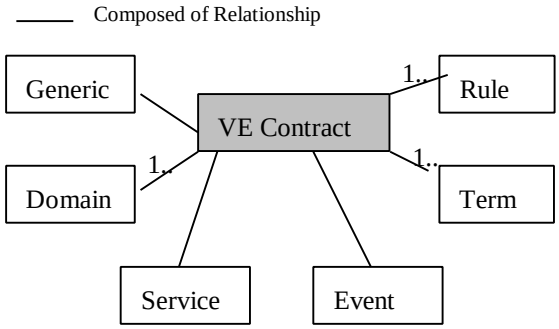

Figure 2: VE Contract Class Diagram

Moreover, a BP is set of one or more linked procedures or activities that collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships. A business process is typically associated with operational objectives and business relationships, for example an Insurance Claims Process, or Engineering Development Process. A process may be wholly contained within a single organizational unit or may span several different organizations (VEs), such as in a customer-supplier relationship. A business process has defined conditions triggering its initiation in each new instance (e.g. the arrival of a claim) and defined outputs at its completion. A business process may involve formal or relatively informal interactions between participants; its duration may also vary widely. A business process may consist of automated activities and/or manual activities.

A business process consists of a network of activities and their relationships, criteria to indicate the start and termination of the process and the activities, and information about the individual activities, such as participants, associated IT applications, business objects and data, etc. In addition to that, a BP's activity issues events and registers for event occurrences specifying the appropriate event handlers. In a VE environment, BP and events should be shared. In a VE environment one BP or activity of this BP must be in position to invoke a BP belonging to another member or issue events that can be consumed by BPs of another VE member. The invocation and the posting of events can be done only according to the contract agreements, i.e. the VE contracts.

In figure 3, a class diagram modeling the above BP concepts is introduced. In our model, a BP is represented by a BP object that is a superclass or a package. The definition of the basic classes of a BP object are:

- **BP General Data**: includes the name of the BP, description, business domain, the responsible participant, etc.

- **BP Life Cycle Operations**: includes operations for executing, suspending, terminating BPs, etc.,
- **BP Public Data:** includes the data structures that can be shared by other BP objects, belonging either on the same business domain or on different one. Additionally, the events that this BP object is raising during execution are specified.
- **BP Public Activities:** includes the activities that can be invoked by other BP objects belonging either on the same business domain or on different one.
- **BP Context**: includes all the internal activities-objects that interact together for providing the BP functionality.
- **BP History**: includes information about the status of the BP object during real time operation invocations occurred.

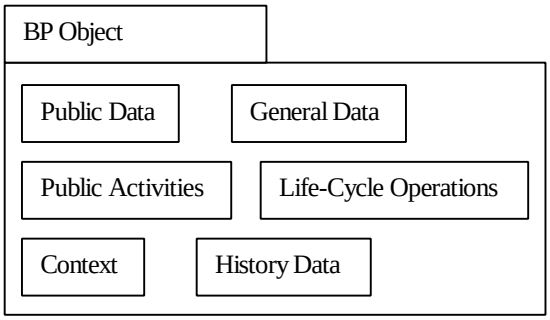

Figure 3: Business Process Object Diagram

Every BP object in one business domain is registered in the **BP directory**. The BP directory maintains all the appropriate information about BP objects located in one domain. This service is used for locating business process objects during run-time execution. During real execution time, BP objects can invoke other BP objects or activities-operations of BP objects across organizational boundaries. The **Interaction Manager** manages all these invocations of BP objects or activities-operations that are public. Additionally, a BP object may issue an event that can be consumed by an operation belonging to another BP object across the VE boundaries. The entity that enables this interaction is called **Event manager.**

7. VE Enabling Services

Based on the above proposed VE life cycle model, we propose a set of enabling services that allow the establishment, configuration, integration and management of VEs and the business processes within the business domain architecture. These components aiming in resolving major problems like interoperability, integration, scalability and reliability. These components effectively support most of the necessary services needed during the three phase of the VE life cycle. Additionally, they provide the appropriate mechanisms to interoperate with business components and services of the underlying layers of the business domain architecture. We provide four classes of enabling services needed for the establishment and management of VEs. There are:

1. VE Enabling Services support the establishment and management of a virtual enterprise and the management of VE contracts, business processes, activities, and business events.

- **VE Contract Manager:** is a component that manages the VE contracts. This component enables the creation of a contract, the deletion and the modification of it. The modification of a VE contract includes the assignment of services and events to a contract and the opposite, and the modification of terms and rules applied in the contract.
- **BP Administrator**: is a component that provides the ability to specify business process objects, modify existing specifications and integrate BP objects with BP objects provided by VE members. It also enables the creation, deletion and modification of public activities and the specification of events and event handlers inside BP objects.
- **BP Controller**: enables the control and monitoring of business processes during real-time operation. It enables the insertion, deletion and control of traces and events used for the capturing of the business process status.

2. Enterprise Support Services are a set of underlying enabling services that supports the execution and management of business processes.

- **BP Directory service**: manages and maintains information about business processes provided either within the business domain or outside the business domain through the VE contract. In general, every business process that can be invoked, either inside or outside the business domain, is registered in the BP directory.
- **BP Event service:** manages the registration and notification of BP events issued either inside or outside the business domain. A BP event can be issued by a BP belonging to Business Domain A and consumed by a BP belonging to Business Domain B.
- **BP Invocation Service:** manages the invocations of BPs and event notifications across business domains boundaries based on the VE contracts.
- **Security service**: specifies and manages security policies enforced in different business

components and business processes.

3. Business Processes provide all the necessary business logic and services to solve specific business problems. All business processes are formed in terms of Business Process Objects and can be managed and administrated by the above mentioned VE enabling services. Due to the flexible association between BPs, categorization and decomposition of BPs can be performed.

- **Business Processes**: provide all the necessary business logic to employees, VE members, customers, suppliers, etc. These business process components have been developed by re-using common, primary business processes and objects and have been installed, configured and integrated by the BP administrator. These business processes are controlled and managed during real time operation by the BP controller that detects performance bottlenecks and de-efficiencies. The business processes can be analysed in more primitive and common business objects so as to increase the degree of re-usability and to minimise the development cost. Common Business objects are primitive, general purpose, business sector independent objects used in the development of value added, application specific BPs.

4. Distributed Support Services provide all the necessary secure distributed network services for the interoperation of different business domains. These services basically provided by Corba Service Architecture and used for the development of the above mentioned building blocks and components.

In the above figure, mapping between these proposed enabling services and the layered business domain architecture proposed previously is done.

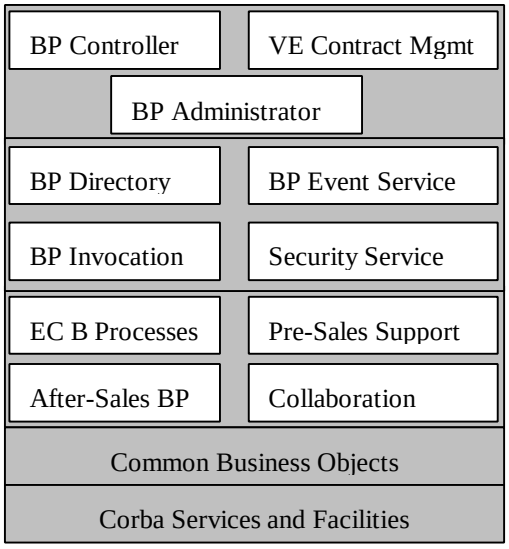

Figure 4: Building Blocks of a VE Infrastructure

In the following paragraphs, every VE enabling service is further analyzed and certain design and implementation details are given.

7.1. VE Contract Manager

As stated before, the VE Contract Manager is a component that manages VE contracts. The main operations provided by this component are:

- contract management, like creation, deletion and modification of a VE contract,
- service assignment operations like assign deassign service to a contract,
- event assignment operations like assign and deassign event to a contract,
- terms and rule administration operations like modification of payment terms,
- publish operations, like send and receive contract to VE members,
- contract storage operations, like store, retrieve and delete contract from the database

The main functional components of the VE contract manager and the relationships with other classes are depicted in the following figure.

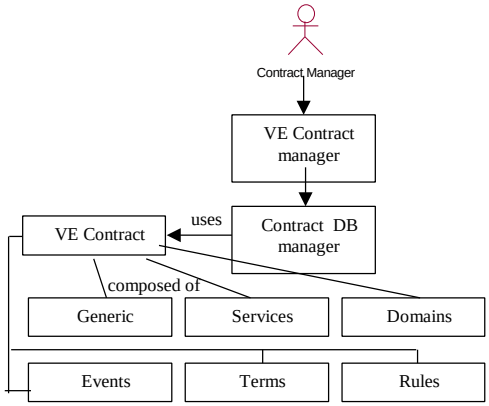

Figure 5: VE Contract Manager Model

7.2. Business Process Administrator

The BP Administrator is a component that manages all the necessary actions for the creation, configuration and integration of business process objects inside the business domain and inside the VE. The main operations provided by this component are:

- BP object management operations, like creation, specification, deletion and modification of BP objects
- BP object public operation management, like creation, deletion and modification of public activities, specification of events and event handlers

inside BP objects,

- BP object integration operations, like linkage of BP objects inside the business domain and inside the VE domain
- BP object registration operations, like registration and unregistration of BP objects to BP directory

The main functional components of the BP administrator manager and the relationships with other classes are depicted in the following Figure 6.

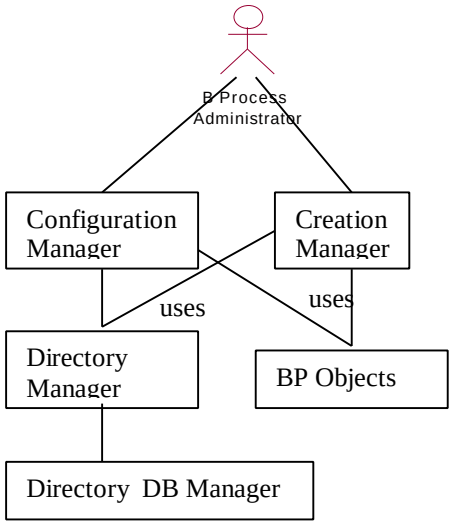

Figure 6: BP Administration Model

7.3. Business Process Controller

The BP Controller enables the control and monitoring of business processes during real-time operation. Actually, it provides operations for finding the status of a business process object, the current operation, the events issued, the events registered, the operations invoked, the public data used and by which entity, etc. The main operations provided by this component are:

- BP object status information
- BP object event issued/registered information
- BP object operations invoked information

In the above information, start and end times are provided. Based on this information, mean-time statistical analysis about the BP objects can be produced. The main functional components of the BP controller and the relationships with other classes are depicted in Figure 7.

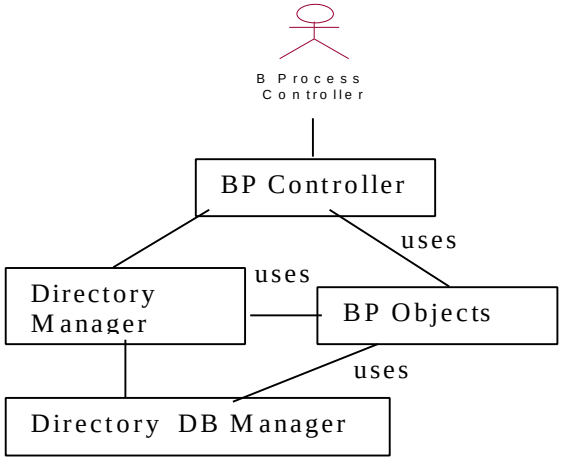

Figure 7: BP Controller Model

8. Conclusions

In this paper, we propose a virtual enterprise life-cycle model, a business domain architecture, and a set of Virtual Enterprise enabling services that support the establishment, management and execution of business processes across VE boundaries. Our approach is general and can be deployed for developing electronic commerce shared business processes. For the development of these services, we are utilizing object-oriented technologies and distributed component-based development practices so as to ensure easy integration, customization and high degree of interoperability.

Currently, we have analyzed, designed and specified the proposed VE building blocks using UML and the Rational Rose Case Tool [9,12,15]. The building blocks are being implemented and integrated into the CORBA architecture. In order to proof the validity of our concepts, we also developing a set of generic Electronic Commerce business processes, like ordering management. Following the VE life-cycle model, we will integrate, configure and customize them in order to form a typical VE scenario. We believe that one of the major reasons for building a VE in the near term future will be for Electronic Commerce purposes. This was also the major motivation behind this work.

Future activities of this work include the full development of the above, proposed VE building blocks and the integration of them into CORBA platform. Additionally, we will testify, evaluate and validate our concepts by applying Electronic Commerce practices in virtual enterprise environments. Finally, we also considering to propose an interoperability specification with emerging standards in this area like JavaBeans, DCOM and Business Objects proposed by OMG.

9. References

[1] I. Jacobson, etl. "The Object Advantage, business process re-engineering with object technology", Addision Wesley 1994, ISBN 0-201-42289-1

[2] T. Reenskaug, "Working with Objects, the OORAM Software Engineering Method", Manning 1996, ISBN 1-884777-10-4

[3] R. Gestner, "Using Objects for Workflow enabling of standard application Software", http://laser.cs.umass.edu/workflow/gestner.html , 1996

[4] Open Engineering, "Business Patterns", http://www.openeng.com/patterns.html, 1995-1998

[5] Virtual-organisation.net, "Newsletter", Vol. 1, No. 3-4, 1997

[6] M. Yogesh, "Enterprise Architecture: An Overview", @BRINT, http://www.brint.com/papers/enterarch.html, 1996

[7] Peter Lawrence, "The Business Impact of Workflow", Workflow Handbook, p33-p88, John Wiley and Sons Ltd., 0 471 96947 8

[8] OMG Common Business Objects and Business Object Facilities http://www.omg.org/ , 1997-1998

[9] "ORBIX 2 Programming Guide", IONA Technologies, 1996

[10] M. Fowler, etl. "Applying the Standard Object Modelling Language", Addison Wesley, ISBN 0-201-32563-2

[11] R. Orfali "The Essential Distributed Objects Survival Guide ", John Wiley and Sons, ISBN 0-471-12993-3 1996

[12] Rational Rose Case Tool, http://www.rational.com/

[13] Javabeans Specification Home page, http://java.sun.com/beans/spec.html

[14] DCOM architecture, Microsoft Corp. http://premium.microsoft.com/msdn/library/conf/pdc97/dcomarch-c.htm

[15] T. Quatrani, "Visual Modeling with Rational Rose", Addison Wesley 1998, ISBN 0-201-31016-3