Web Services based Collaborative B2B e-commerce in the Oil Industry

J. Gialelis[1], D. Karadimas[1], P. Chondros[1], A. Kalogeras[2], S. Koubias[1] D.Serpanos[2]

[1]Applied Electronics Laboratory
Electrical and Computer Engineering Dept.
University of Patras Campus
26500 Rio Patras, Greece
{gialelis, koubias}@ece.upatras.gr
{karadimas, pchondros}@apel.ee.upatras.gr

[2]Industrial Systems Institute
Science Park of Patras Building
Stadiou Str, 26506 Platani Patras, Greece
{kalogeras, serpanos}@isi.gr

Abstract. Collaborative B2B e-commerce provides enterprises with the necessary level of flexibility and efficiency to retain competitiveness under the increasingly turbulent business environment. XML-based frameworks may be utilized by enterprises to integrate enterprise level applications to provide B2B e-commerce without, though seamlessly integrating inter-enterprise processes with intra-enterprise processes to provide collaborative B2B e-commerce. This paper presents an innovative approach towards collaborative B2B e-commerce by utilizing the RosettaNet emerging standard in combination with Service Oriented Architecture and semantically enriched information in order to seamlessly integrate the inter-enterprise (public) and intra-enterprise (private) processes. An implementation of a collaborative B2B e-commerce business model in the oil industry is given as a use case example.

Keywords: Collaborative B2B e-commerce, Service Oriented Architecture, RosettaNet semantics.

1. Introduction

Economic pressure and turbulences in the global business environment make it increasingly more difficult for manufacturing enterprises to make predictable long-term provisions, as they radically change the way that organizations operate and interoperate with other enterprises in order to accomplish a business goal. In this context, electronic Business-to-Business interactions (B2B e-commerce) provide the necessary backbone for organizations to interact with their suppliers and customers in order to respond more quickly to changes. As a result, business competition is gradually becoming more value chain to value chain rather than enterprise-to-enterprise.

E-Business frameworks represent the state of the art in e-commerce; since framework-based B2B e-commerce effectively addresses the seamless linking of inter-enterprise (public) processes involving Enterprise layer's applications to those of business partners. These frameworks provide the necessary layers to achieve interoperability in e-commerce, since their interactions involve various business processes and business components, such as executable applications, systems, and their associated information [1].

In order to support the interchange of information among "public" modules, several standards and languages have been developed. EDI (Electronic Data Interchange), EDIINT (EDI over the Internet), ebXML (e-business eXtended Markup Language) and BizTalk are some generic standards addressing vertical as well as horizontal integration issues. Furthermore, standards such as RosettaNet, IOTP (Internet Open Trading Protocol), ICE (Information and Content Exchange) and OBI (Open Buying on the Internet) deal more with vertical market-oriented integration issues.

However, current framework-based developments consider inter-enterprise processes, involving only Enterprise layer applications, information, etc; without taking into consideration the intra-enterprise processes involving applications, information residing into the other enterprise layers such as Plant and Shop Floor layers thus, not assuring agility, flexibility and autonomy of the interacting partners [2]. The work presented in this paper, intents to overcome this drawback by proposing a set of technologies imposed over the components of a presented e-business frameworks' holistic architecture which extends the functionality of current frameworks in order to incorporate all enterprise layers into B2B e-commerce interactions thus elevating an innovative collaborative business model which seamlessly integrates the inter-enterprise (public) with the intra-enterprise (private) processes [3]. More specifically, section 2 describes the collaborative B2B e-commerce, section 3 depicts the structure of e-business frameworks and describes the proposed set of technologies imposed over the e-business frameworks' holistic architecture and section 4 presents our implementation of a collaborative B2B e-commerce business model in the oil industry according to the proposed approach.

2. Collaborative B2B e-commerce

Collaborative B2B e-commerce differs from basic B2B commerce, since collaborative commerce goes beyond on-line document exchanges, indicating that organizations adopt B2B networks to establish new collaboration mechanisms with channel partners. Thus, enterprises should establish collaborative B2B processes with cooperating manufacturing enterprises, in order to best exploit and gain the most out of the adoption of electronic commerce networks. Collaborative B2B processes are considered to be any

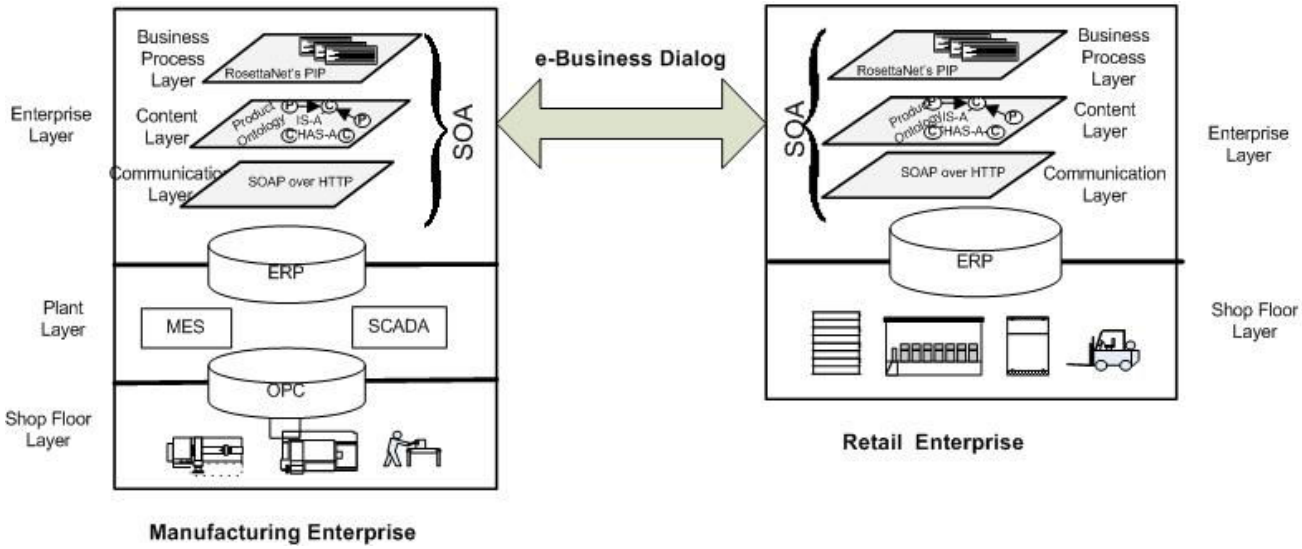

Fig. 1. Proposed e-business frameworks' holistic architecture.

end-to-end business processes, which results from the seamless integration of intra-enterprise processes with inter-enterprise processes.

A designated example of a collaborative B2B process is the Continuous Replenishment Planning (CRP) model on top of which one of the most efficient inventory management functions may be developed. The key characteristics of CRP are the sharing of real-time inventory data by retailers with manufacturers and continuous replenishment of retailers' inventory by manufacturers. This function, called continuous replenishment planning, is defined as the practice of partnering between distribution channel members that changes the traditional replenishment process from a warehouse/distributor centric process handling purchase orders, to an end-user/retailer centric replenishment process involving the manufacturing enterprise in the overall process utilizing actual and forecast data.

Even though, the CRP collaboration model has the potential to alter significantly production-scheduling methods and inventory management practices so that enterprises enjoy the above-mentioned benefits, survey results do not keep up with these expectations [4]. This drawback is caused, mainly, due to CRP deployment methodologies. Such methodologies are based on point-to-point architectures and on the utilization of proprietary solutions/technologies which present very limited tolerance to system/applications alterations and minimum scalability to value-network expansion.

In order to overcome these drawbacks and achieve both integration of heterogeneous systems/applications, at any layer in the enterprises they may reside in and unlimited expansion of the value-network, this paper proposes the deployment of specific set of technologies over both ERP (Enterprise Resource Planning) and OPC (Manufacturing Execution Systems and Supervisory Control And Data Acquisition) systems. These technologies are based on Internet ubiquity and emerging standards. Therefore, Service oriented Architecture (SoA) elements, RosettaNet's ontology and standard processes for the B2B transactions are utilized, in order to fully implement the CRP collaboration model. Moreover, SOAP is utilized as the transport message protocol in order to satisfy communication needs, while ontologies are used in order to standardize content semantics and WSDL (Web Service Description Language) is used in order to expose systems' functionality as Web Services. Figure 1 depicts the e-business frameworks' holistic architecture, based on the above mentioned principles which are further described in the next section.

3. e-Business Framework Holistic Architecture and Components Technologies

Several contemporary B2B interaction frameworks are based on the XML notation. They aim at overcoming some of the limitations of traditional electronic data interchange standards. XML-based frameworks for B2B transactions follow a hierarchical model, which comprises three layers: the Communication, the Content and the Business Process layer [5]. The Communication layer provides the appropriate means of communication for exchanging the required messages among the involved systems/applications, while the Content layer provides the necessary tools (languages and conceptual models) so that the involved entities agree upon the semantics of contents and the types of the exchanged documents. Finally, the Business Process layer is concerned with the conversational interactions among services. The following three sub-sections present the proposed imposed technology solutions for the three aforementioned layers.

3.1 Communication Layer

For the communication needs, the SOA paradigm is utilized by employing the use of Web Services (WSs) technology in order to open up enterprise systems, plant systems and production systems; thus making

available their functionalities. Therefore, the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL) are deployed at the communication layer. Universal Description Discovery & Integration (UDDI) is not integrated in our used case implementation.

3.2 Content Layer

The content layer provides the suitable means to describe and organize exchanged information in such a way that it can be understood and used by all the collaborating partners. Content interactions require that the involved systems understand document content semantic. Our work is based upon the development of an elaborated petroleum product ontology in order to achieve a uniform product terminology. Furthermore, existing work in different standards, such as RosettaNet Technical Dictionary (RNTD), is utilized wherever required.

3.3 Business Process Layer

Our work emphasizes on the RosettaNet framework in order to address business process layer needs. RosettaNet focuses on three key areas of standardization to automate B2B interactions. First of all, the vocabulary needs to be aligned. Secondly, the business process governing the interchange of the Business Messages themselves must be harmonized and specified. Finally, the way in which Business Messages are wrapped and transported should also be specified.

Next section describes in details an implementation example of a use case scenario of the proposed approach depicting a real world Continuous Replenishment Planning process application in the oil industry which involves gas stations or other enterprises as end users/retailers and a refinery as the manufacturing enterprise.

4. An illustrative Use Case

A use case scenario is presented in order to show a generic infrastructure of an online Oil-gas ordering Content Management System (OilCMS). The implementation presented in this section focuses on the specification and deployment of intra-enterprise (private) processes as well as the integration with the inter-enterprise (public) processes. However, there are some key characteristics of such an online system: (a) should be able to connect to any other available ERP system of the enterprise in order to update product and pricing information and (b) should be able to support enterprises with different products and pricing policies.

4.1 Scenario Description

Oil ordering and delivering is a demanding procedure, in terms of catching-up with the deadline dates, managing the paperwork and generally taking care of all the rest non-automated processes inside the Distributor Enterprise. Three entities are defined as follows: (a) the customer/user which makes the order(s), (b) the Distributor Enterprise which collects the orders and has the duty to deliver the order to the customers, accordingly and (c) the Supplier Enterprise which supplies the distributor with the appropriate quantities of the ordered products. The OilCMS automates this end-to-end online ordering infrastructure based on intra/inter-enterprise Web Services. Moreover, it is able to keep track and publish the different order states during the whole process.

An operational diagram of the implemented system is depicted in Fig. 2. The distributor enterprise exposes several Web Services identifiable by the UDDI protocol. These Web Services are implemented by the OilCMS software and are installed locally at the Distributor Enterprise's server. Figure 2 shows the three basic web services:

- `GetQuote()` – The customer can access information about available products, pricing and discounts policies of the Distributor Enterprise through this Web Service. The product catalogue is updated according to the enterprise's policy.
- `Order()` – The order Web Service allows the user to make an online order. The order is stored at the Distributor Enterprise local database and can be managed by the administrator through the OilCMS software.
- `TrackStatus()` – The different states of the order throughout the entire processing path are published through this Web Service.

Major role in the above process plays the intra-enterprise procedure which is implemented by the OilCMS software. Each OilCMS of a distributor company is connected to the local database of the corresponding enterprise. This database stores data relative to:

- User data – contains personal and technical information (i.e. a customer may receive the products by sheep, by land or both) about each user account.
- Product data – Contains information about available product types and costs, respectively.
- Order data – All the data associated with an order are stored in this structure. Some of these data could be order delivery date, customer's address, total cost, cost per product, information about pricing policies per user and discounts policies, etc.

All aforementioned data structures can be accessed by the OilCMS administrators. The OilCMS software is designed to support five different types of users that have special rights, defined as follows:

- **Unconfirmed Users** – An unconfirmed user has already made a registration request but has not yet been confirmed by the OilCMS administrators. Only when confirmed, he will be able to access the system.
- **Confirmed Users** – A confirmed user is able to access product and pricing information, place an

order and check his order status. Moreover, he can edit his account data in case of changes. The customer is represented by this type of user.

- **Banned Users** – A confirmed user can be banned by the administrators for several reasons (did not paid on time, etc.). A banned user cannot access the enterprise's system.
- **Administrators** – An administrator can manage users' accounts, product types and prices, update catalogs and ban or confirm potential customers/users. An administrator has also the duty to call the inter-enterprise service by uploading the orders to the supplier enterprise.
- **Super-Administrator** – The main difference between an administrator and the super-administrator, is that the latter has the privileges to manage administrators' accounts. The super-administrator is assumed unique in the proposed implementation.

4.2 The OilCMS Architecture Implementation

The OilCMS architecture has been designed according to object-oriented approach principles and implemented using PHP programming. Along with the basic classes, which are depicted in Fig. 3, come two auxiliary classes and several library functions. A configuration script is also provided and allows the installation of the software to be platform and technology independent. The system can support many users and administrators at the same time. Following there is a short presentation of the implemented classes, which are illustrated in Fig. 3.

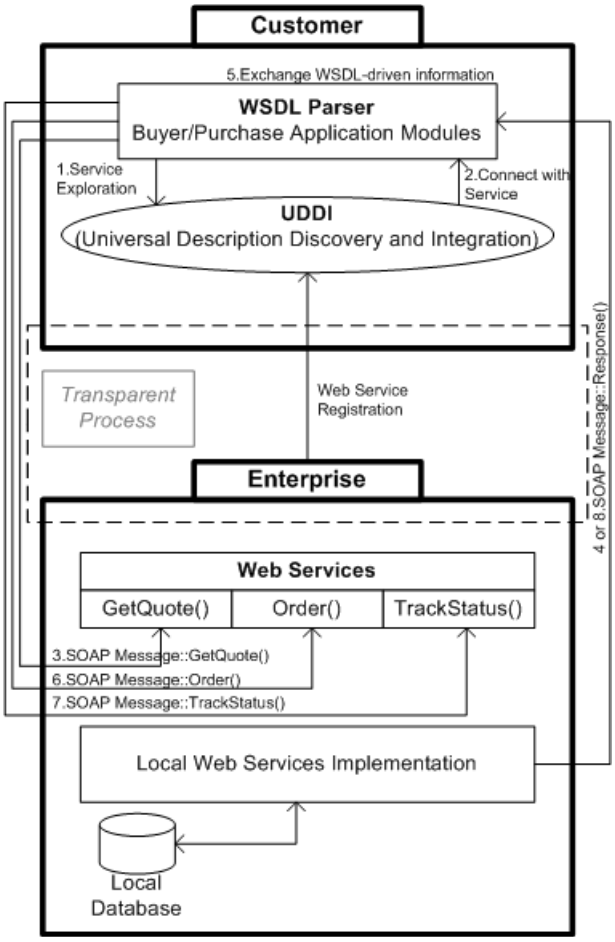

Fig. 2. Operational diagram of the online order management system based on web services.

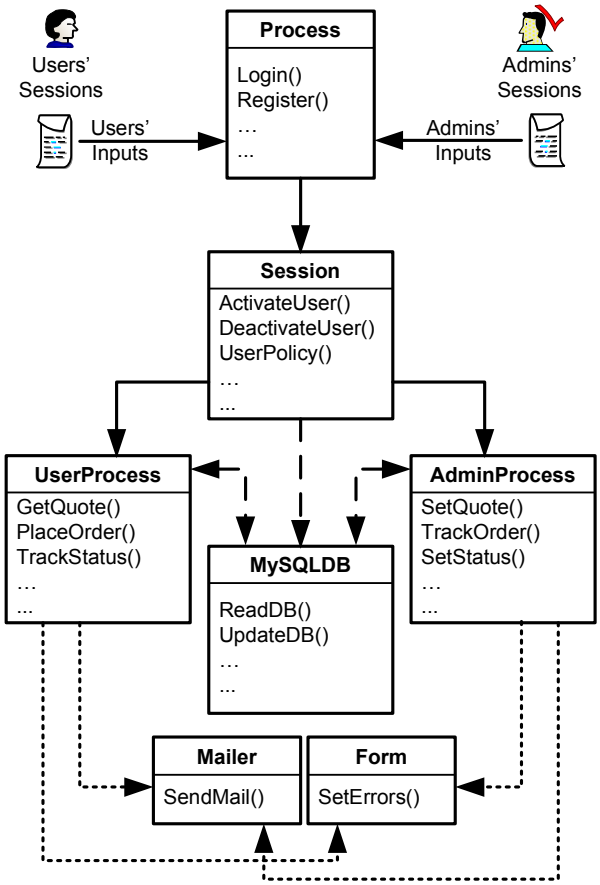

Fig. 3. Class Collaboration Diagram within the Distributor Enterprise Web Service (OilCMS).

- **Process** – manages the actions performed by a potential user.
- **Session** – keeps track of the OilCMS active users. Session is active until the user logs out from the system. It stores any information related to the online user such us user level, time of being active, last time the user entered the system, etc. One session object is generated each time a user enters the system.
- **UserProcess** – implements the navigation to the internal procedures of the OilCMS for confirmed users.
- **AdminProcess** – implements the navigation to the internal procedures of the OilCMS for the administrators and the super-administrator.
- **MySQLDB** – implements all methods employed to access the Distributor Enterprise local database.
- **Mailer** – auxiliary class used to send informative e-mails. A mail server needs to be properly configured on the Distributor Enterprise's side.
- **Form** – auxiliary class that manages all the errors caused by mistake on the user interface.

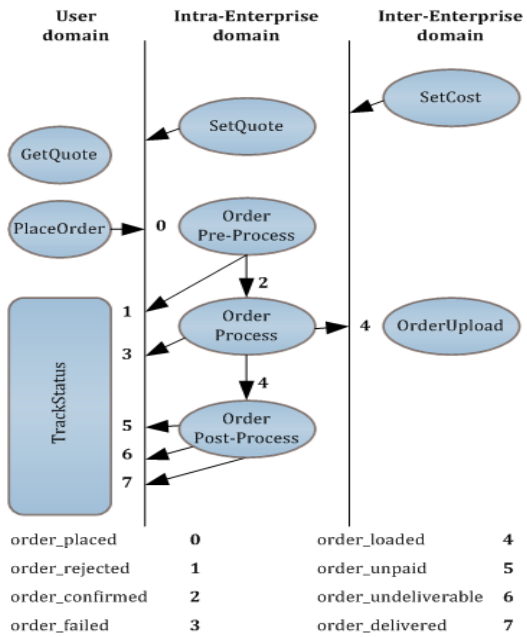

Fig. 4. Order state diagram versus implementation domains.

Many different security aspects have been taken into account so that the final architecture will be robust and tight enough in order to meet all the requirements from security point of view. In particular, every user is uniquely combined with his username and password, while the last one is strongly encrypted by employing hash functions. A second security layer is provided by employing the aforementioned logic of five different levels of users. The system is able to impart the appropriate privileges of a user level to each designated user, as all basic principles of object-oriented design have been followed. Finally, Session object is responsible for the security implementation for the entire Web Service as it grants users with the appropriate privileges. Figure 4 depicts the transactions that take place between the user, the Distributor Enterprise and the Manufacturing Enterprise (Supplier). Each of the three order sub-processes sets accordingly the order status, based on inputs provided by any of the administrators.

The order status can be accessed by the user through the `TrackStatus()` Web Service. The designed algorithm provides a real-time representation of the ordering status throughout the whole processing path.

4.3 Service Expandability

As already mentioned, the proposed replenishment planning process, aims to automate the intra-enterprise procedures by utilizing a Web Service. The basic components of this process include the Web Service implementation itself, the administration and management policy, the configuration and a local database, as depicted in one of the two cells at the left part of Fig. 5.

The standard scenario provides a potential buyer with the capability to employ the service offered by the Distributor Enterprise, while this service is able to transparently communicate with the corresponding application of the Supplier Enterprise. The Supplier Enterprise's application could be also implemented as a

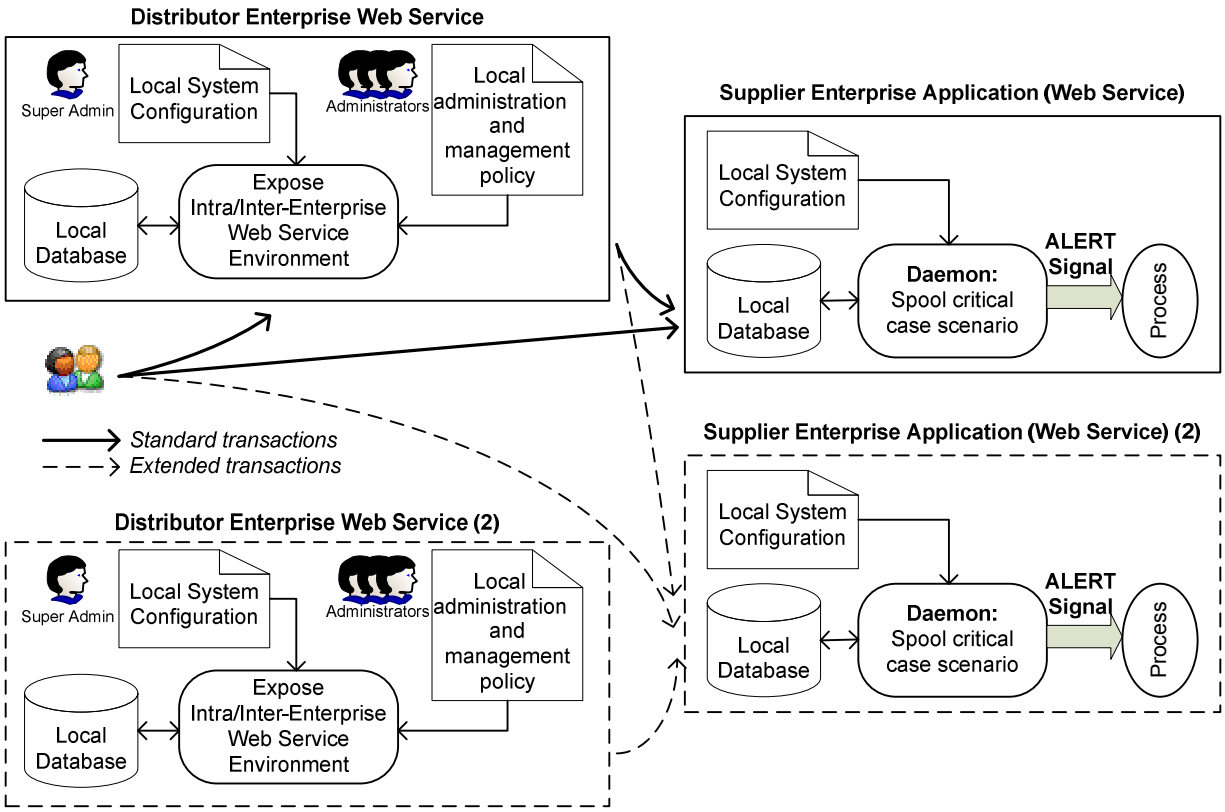

Fig. 5. Proposed Service Expansion Scenario.

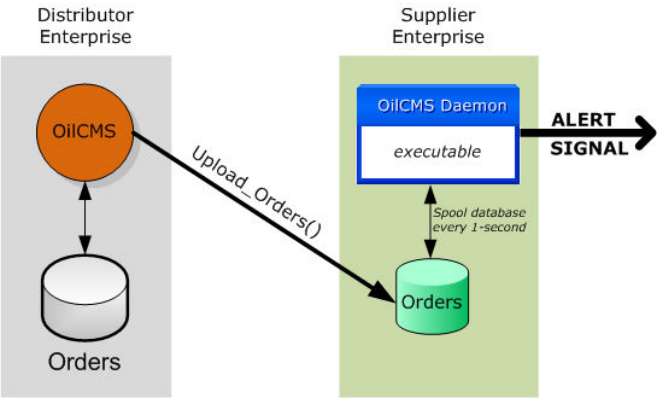

Fig. 6. OilCMS daemon process in details.

separate Web Service, which would be able to be exploited by other Distributor Enterprises and/or individual buyers, if they are granted with this privilege. Distributor Enterprise's transactions with the Supplier Enterprise's application or Web Service are secure and can be operated only by the administrators group of the last mentioned Enterprise. Transactions from the Distributor Enterprises Web Services are uploaded and finally stored in a local database next to the Supplier Enterprise application or Web Service.

The overall system architecture is able to provide many extra benefits, apart from the intra-enterprise automation of the ordering procedures within the Distributor Enterprise. Thus, the proposed implementation is suitable for also contributing to the automation of the production procedures within the Supplier Enterprise. The local database of the Supplier Enterprise could be implemented either as an extension of the possible existing ERP of the company or as a standalone database. In the first case, the use of an intermediate program, acting as a wrapper, in order to enable communication and data exchanging between the two databases is mandatory, while in the second case the local database of the Supplier remains very similar to the one of the Distributor Enterprise. Thus, the proposed replenishment and planning process does not repeal the utilization of any existing ERP within the Distributor or the Supplier Enterprise.

Moreover, in both cases, a special standalone process, could be deployed and utilized in order to take the advantages of the aforementioned local database of the Supplier Enterprise. Figure 6 depicts the idea of how this process is utilized, called oilcmsdaemon, while the main key functions of the daemon have been presented in Fig. 5. The major role of this daemon process is to spool the database at certain time intervals and make critical decisions, according to specific configuration scenarios that are embedded in the Supplier Enterprise application or Web Service. This process is able to produce a signal which can be appropriately driven and managed by the Supplier Enterprise Manufacturing Execution Systems or other systems, related with the production process chain.

Finally, Fig. 5 depicts that the proposed replenishment and planning process is well structured so as the Web Services of the Distributor and the

Fig. 7. User Interface from Distributor Enterprise Web Service.

Supplier Enterprise could be replicated in a more complex network of interactive Web Services.

4.4 OilCMS Implementation Details

This section provides the implementation details regarding the aforementioned use case scenario and concludes with a specific section of the exposed Distributor Enterprise Web Service, implemented for the need of use case scenario.

The Distributor Enterprise Web Service, named as OilCMS, has been developed with PHP/MySQL employing the Simple Object Access Protocol (SOAP), while its communication interface is based on the Web Services Description language (WSDL) technology. The OilCMS software is installed at the Distributor Enterprise's side and is designed to work on a PHP/MySQL/Apache server. Moreover, the installation of a suitable ODBC driver is required for the daemon process implementation at the Supplier Enterprise's side. Figure 7 depicts a screenshot from the user interface of Distributor Enterprise Web Service that presents the developed product ontology.

Finally, Fig. 8 depicts a segment from the final WSDL document that defines the implemented Distributor Enterprise Web Service. In particular, the presented segment refers to a specific method of the Web Service, named as CreateProduct(), while this method is part of a sub-service of the entire Web Service, named as AdminProcess().

5. Conclusions

This paper presents an innovative approach towards collaborative B2B e-commerce by utilizing the RosettaNet emerging standard in combination with Service Oriented Architecture and semantically enriched information in order to seamlessly integrate the inter-enterprise (public) and intra-enterprise (private) processes. An implementation of a special case of collaborative B2B e-commerce namely the Continuous Replenishment Planning process in the oil industry is given as a use case example.

The work presented in this paper has been partially supported by the General Secretariat for Research and Technology (GSRT) of the Hellenic Ministry of Development under the framework "Competitiveness" through the project No68, Measure 4.3, Action 4.3.6.2g

```
<?xml version='1.0' encoding='UTF-8'?>

<definitions name="Adminprocess" targetNamespace="urn:Adminprocess"
xmlns:typens="urn:Adminprocess" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/">

<!-- types -->
        <types>
                <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:Adminprocess">
                        <xsd:complexType name="Adminprocess">
                                <xsd:all/>
                        </xsd:complexType>
                </xsd:schema>
        </types>

<!-- message -->
        <message name="CreateProduct"/>
        <message name="CreateProductResponse"/>

<!-- portType -->
        <portType name="AdminprocessPortType">
                <operation name="CreateProduct">
                        <input message="typens:CreateProduct"/>
                        <output message="typens:CreateProductResponse"/>
                </operation>
        </portType>

<!-- binding -->
        <binding name="AdminprocessBinding" type="typens:AdminprocessPortType">
                <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
                <operation name="CreateProduct">
                        <soap:operation soapAction="urn:AdminprocessAction"/>
                        <input>
                                <soap:body namespace="urn:Adminprocess" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
                        </input>
                        <output>
                                <soap:body namespace="urn:Adminprocess" use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
                        </output>
                </operation>
        </binding>

<!-- service -->
        <service name="AdminprocessService">
                <port name="AdminprocessPort" binding="typens:AdminprocessBinding">
                        <soap:address location=""/>
                </port>
        </service>
</definitions>
```

Fig. 8. Part of WSDL document of the implemented use case scenario.

[6] and by the Network of Excellence "INTERMEDIA", [7].

References

1. S. Shim et al., "Business-to-Business E-Commerce Frameworks",IEEE Computer, pp.40-47, October 2000

2. T. Kim et al, "A modeling Framework for Agile and Interoperable Virtual Enterprises", Computers in Industry 57, pp. 204-217, 2006

3. J. Gialelis, A.P. Kalogeras, A. Kaklis, and S. Koubias, "Collaborative Continuous Replenishment Planning Process Implementation Utilizing Web Services, Product Ontologies and Workflow Management System" Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA2006 proceedings, pp. 574 – 581.

4. S. Lee et al., "Business Value of B2B Electronic Commerce", Electronic Commerce Research and Applications 2, pp. 350-361, 2003

5. B. Medjahed et al., "Business-to-Business Interactions: issues and enabling technologies", The VLDB Journal 12, pp. 59-85, 2003

6. http://www.gsrt.gr

7. European Project INTERMEDIA, "Interactive Media with Personal Networked Devices", http://intermedia.miralab.unige FP6 – IST- 38419.