A Service Oriented Architecture for Managing Operational Strategies

Nektarios Gioldasis, Nektarios Moumoutzis, Fotis Kazasis, Nikos Pappas, and Stavros Christodoulakis

Laboratory of Distributed Multimedia Information Systems & Applications,
Technical University of Crete – TUC/MUSIC
{nektarios,nektar,fotis,nikos,stavros}@ced.tuc.gr

Abstract. In this paper we present a case study where an e-business platform (OPERATIONS) has to be built in order to manage the operational strategies of a high volume coastal company. The platform is intended to support multiple operational scenarios that have to be executed when emergency situations or sudden changes of scheduled events unfold. In order to properly integrate all the involved parties that may take part in such a scenario, the system must support multiple interaction patterns according to the communication channel and the device that is used. Thus, its design must take into account multi-channel delivery of functionality and information. The design of the system follows a Service Oriented Architecture (SOA) in order to properly integrate back-office data sources and legacy systems with new required application extensions providing a complete e-business platform. The core business logic tier of the system is a Process Oriented Platform (POP) that is able to dynamically generate the appropriate workflows, to orchestrate the constituent services and to execute complex operational scenarios interacting with involved users and systems in a multi-channel fashion. Finally, appropriate knowledge bases have been defined to support the storage, evaluation, and evolution of the executed scenarios and to provide valuable Business Activity Monitoring information to the company's administration.

Keywords: Service Oriented Architecture, SOA, Process Oriented Platforms, POP, Multi-channel delivery, Service Orchestration, Business Process Integration & Management

1. Introduction

The ultimate goal of the OPERATIONS platform is to provide a state-of-the art and reliable e-business environment supporting a regional coastal company. The company has a fleet of eleven modern and luxurious ships operating in the Aegean, Ionian and Adriatic Sea. In an attempt to improve its internal processes, to provide better services to its customers, and to capture and exploit its internal knowledge, the company asked for a requirements analysis and system architecture that satisfies its needs. The proposed e-business platform will be used to deploy and monitor in an extensible manner all the operational strategies and related processes of the company using multi-channel delivery of services and standard descriptions of operational scenarios. These scenarios represent procedures of major significance that today are either not considered at all or done in an empirical basis and it is consequently difficult to evaluate their impact on the economic indicators of the company. Such scenarios that are going to be managed by the OPERATIONS system are:

- Management of emergency situations (e.g. mechanical breakdowns) that obstruct normal voyage.
- Management of urgent events that happen on-board (e.g. man on the sea, sudden death of a passenger, incidents that need special medical treatment).
- Management of changes on scheduled events (e.g. mass cancellations of reservations, cancellation of a scheduled trip).
- Management of passenger and/or vehicle overbooking due either to company policies or external events.
- Optimized cargo fitting dynamically, taking into account various constraints, and coordination of involved personnel.

It should be stressed that the above scenarios refer to polymorphic user roles that are impossible to manage at a strategic level without a system like OPERATIONS. Moreover they encapsulate numerous legal, spatial, temporal, economic and technical constraints arising from both the actual situations in which these scenarios unfold and the external environment (including e.g. legal framework for voyage safety) of the company. The dynamic nature of this setting calls for innovative technical solutions in order to reduce risks, increase flexibility, maintain low adaptation-times to changes in the external environment and increase the capability of the company to analyze and improve its performance by monitoring and reengineering its operational strategies.

One of the most important parts of this project is the software architecture, which is presented in this paper. The platform will be built on a multi-tier architecture that clearly distinguishes the application logic (presentation tier), the business logic (services or middleware tier), and the underlying IT infrastructure of the company (back-office tier). Knowledge bases are used to store scenarios' descriptions, user roles, as well as scenarios' instances. A well defined interface is used for the communication with the back-end systems of the company. The platform will be built following a Service Oriented Architecture. We strongly believe that this approach is the most suitable for such a system. According to the Gartner, by 2007 SOA will be the dominant strategy (more than 65%) of developing information systems. Moreover, the core business logic tier of the architecture will be a Process Oriented Platform (POP) in order to dynamically set the appropriate workflows of primitive services according to operational scenarios. Operational scenarios are well-defined manifests that drive the construction and execution of service oriented workflows.

The rest of this paper is organized as follows: In section 2 we mention the current state-of-the-art of the main technologies which will be utilized in the development of the OPERATIONS platform, while in section 3 we provide two typical scenarios that are to be supported by the proposed system. Section 4 illustrates the core platform architecture and section 5 concludes the paper.

2. State-of-the-Art

Service Oriented Architectures [1] has been proposed by W3C as reference architectures for building Web-based information systems. Service-Oriented Architecture (SOA) refers to an application software topology according to which business logic of the applications is separated from its user interaction logic and encapsulated in one or multiple software components (services), exposed to programmatic access via well-defined formal interfaces. Each service provides its functionality to the rest of the system as a well-defined interface described in a formal markup language and the communication between services is platform and language independent. Thus, modularity and re-usability are ensured enabling several different configurations and achieving multiple business goals.

Web Services technology is currently the most promising methodology of developing web information systems. Web Services allow companies to reduce the cost of doing e-business, to deploy solutions faster and to open up new opportunities. The key to reach this new horizon is a common program-to-program communication model, built on existing and emerging standards such as HTTP, Extensible Markup Language (XML) [4], Simple Object Access Protocol (SOAP) [5], Web Services Description Language (WSDL) [2] and Universal Description, Discovery and Integration (UDDI) [6]. However, the real Business-to-Business interoperability between different organizations requires more than the aforementioned standards. It requires long-lived, secure, transactional conversations between Web Services of different organizations. To this end, a number of standards are under way. Some of these (emerging) standards are the Web Services Conversation Language (WSCL) [7], the Web Services Flow Language (WSFL) [8], the Business Process Execution Language for Web Services (BPEL4WS) [9], the Web Services Choreography Interface (WSCI) [10], the Web Services Coordination Specification (WS-Coordination) [11], the Web Services Transaction Specification (WS-Transaction) [12], and the Business Transaction Protocol (BTP) [13]. With respect to the description of service's supported and required characteristics the WS-Policy Framework [3] is under development by IBM, Microsoft, SAP, and other leading companies in the area.

On the other hand, other no "standard" languages and technologies have been proposed for composing services and modeling transactional behavior of complex service compositions. Such proposals include the Unified Transaction Modeling Language (UTML) [14], [17] and the SWORD toolkit [15]. UTML is a high level, UML-compatible language for analyzing, composing, designing, and documenting extended transaction models based on a rich transaction meta-model. Transaction nodes can be implemented by web services that exhibit specific transactional characteristics. The final design can be exported in XML format and thus it can be easily transformed to a web service description. SWORD is a toolkit that facilitates the process of service matchmaking in order to develop composite value – added services.

In the multi-channel delivery area, the main technologies referenced in this paper are the following:

- WAP (Wireless Application Protocol) [19] is a secure specification that allows users to access information instantly via handheld devices.

- Short Message Service (SMS) is the transmission of short text messages, usually to and from a mobile phone. Messages must be no longer than 160 alpha-numeric characters and contain no images or graphics.
- Multimedia Message Service (MMS), a store-and-forward method of transmitting graphics, video clips, sound files and short text messages over wireless networks using the WAP protocol. Carriers deploy special servers, dubbed MMS Centers (MMSCs) to implement the offerings on their systems.
- E-mail service is a simple and widely used service that is supported by the majority of the devices (including the hand-held) and could be used by a system that aims to implement multi-channel access. Technical specifications of SMS and MMS services can be found at [18].
- With the introduction of the Java 2 platform Micro Edition (J2ME) [20] the role of mobile wireless devices was enhanced from voice-oriented communication devices with limited functionality into extensible Internet-enabled devices. The use of Java Technology makes the application adequate for dynamic delivery of content, provides satisfactory user interactivity, ensures cross-platform compatibility and allows the offline access. Especially the offline access that allows the applications to be used without active network connection is of great importance since it reduces the transport costs and alleviates the impact of possible network failures.

3. Operational Scenarios

The OPERATIONS platform will be used in real operational scenarios such as management of emergency situations and urgent events that happen on-board, management of changes on scheduled events, overbooking situations, and dynamic optimized cargo fitting. These envisaged scenarios represent complex procedures that involve company's personnel, clients, and other external parties. The final decisions for the successful implementation of such scenarios including a detailed action plan will be supported by the OPERATIONS e-business platform in two complementary ways: First by coordinating all necessary flow of information and messages to the involved parties so that all actions are taken in correct order and by the most appropriate actor. Second, by retrieving, processing and presenting critical information from the company's extended IT Infrastructure so that alternatives could be efficiently evaluated and actions could be taken efficiently. Moreover, the execution history of the scenarios are finally filled in the OPERATIONS Knowledge Base for further off-line analysis and future use in similar situations, thus providing a formal way of expressing and reusing corporate knowledge that today more or less exists in a tacit manner.

To make things clear, consider the scenario of figure 1. An emergency or urgent event takes place on-board (step 1 of the scenario) and requires immediate treatment. A preliminary evaluation of the situation is done on-board by the authorized personnel (step 2) and immediately three complementary procedures are triggered:

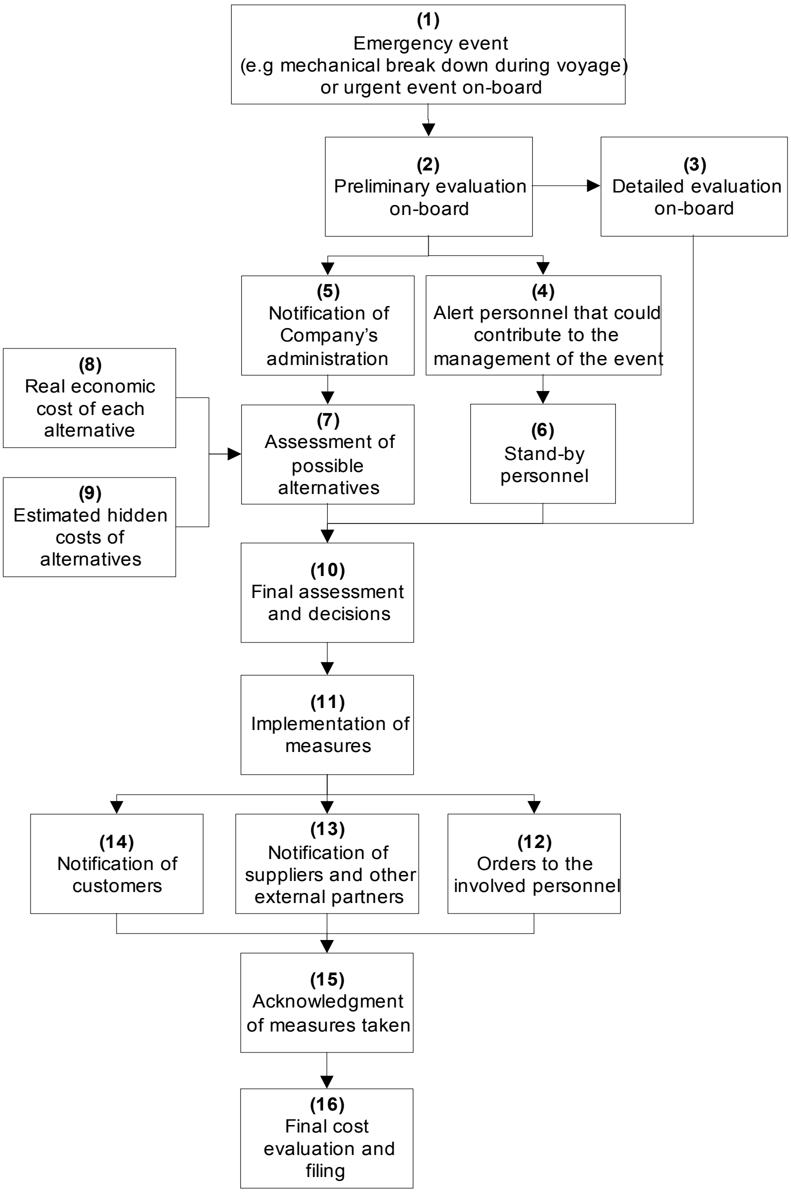

Fig. 1. A scenario concerning urgent or emergency situations happening on-board

- A detailed evaluation is initiated on-board (step 3) to investigate what are the details of the problem.
- The administration of the company is notified (step 5) about the event and starts working on possible solutions. The authorized people start to elaborate on a detailed assessment (step 7) on possible alternatives based on real cost estimations of each alternative (step 8) as well as hidden costs estimations (9) that refer to

various factors including issues that could not be immediately quantified such as the credibility of the company. The alternatives considered at this time remain at a higher level, waiting for more detailed information about the exact problem and the readiness of the personnel that is necessary to handle the alternative solutions.

- Appropriate alerts are issued to the personnel considered as critical for the resolution of the problem and the implementation of the possible measures that could be taken (step 4). The involved persons may be on-board, on the land (at ports or in company's headquarters) or in other ships of the company that sail nearby. Some time later, the information about the readiness of the involved personnel starts coming and the personnel found to be ready to respond is brought into stand-by mode (step 6).

The information about possible alternatives, available personnel, and detailed evaluation of the problem is gathered to help find the optimal solution considering costs, company's policies and other critical constraints such as time, availability of supplies, customer satisfaction (or dissatisfaction), other related problems that could arise etc. This final assessment guides the final decision (step 10) that refers to a set of necessary measures along with a detailed schedule that takes into account all dependencies among the needed tasks. Then, all corresponding measures are taken (step 11) by issuing proper orders to the involved personnel (step 12), notification messages and corresponding acknowledgements to suppliers and other external parties involved (step 13) and possible notification messages to customers (step 14) that include both passengers on-board and customers that have made reservations for the next days that should be changed. The authorized personnel supervises the activities and finally, after the successful finish of the scenario, a final cost could be computed and the whole process could be filed (step 16) for future consideration and analysis.

This envisaged scenario is handled today in an empirical basis and in fact some of the steps illustrated in figure 1 are not at all taken or done in an ad-hoc way. For sure, there is no final evaluation and filing of the process (step 16) that leads to repetition of possible missteps in the future. Neither detailed cost estimations (steps 8 and 9), nor detailed monitoring of the implementation of the measures decided (steps 11, 12, 13, 14) is done due to high overheads and the great number of message exchanges induced.

This will change dramatically with the introduction of the OPERATIONS e-business environment which will be capable for handling the necessary message exchanges, exploiting multiple communication channels, processing the big amount of information necessary to evaluate alternatives, developing detailed action plans, coordinating the implementation of the measures and finally evaluating the actual costs endured, and storing the evolution of the scenario for future analysis and use. Note that this filing feature can be used for more knowledgeable future management of similar scenarios as it can provide information for cost estimations (steps 8 and 9), and action plans (steps 10 and 11), thus increasing responsiveness and overall efficiency of the company. It could also provide other valuable information such as concrete indicators of the efficiency and performance of personnel and external partners.

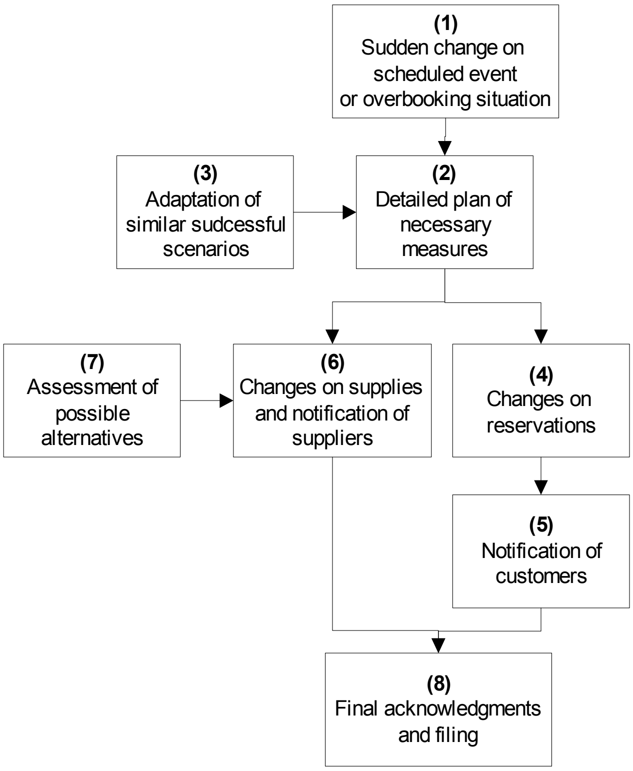
(1)
Sudden change on scheduled event or overbooking situation
(3)
Adaptation of similar sudcessful scenarios
(2)
Detailed plan of necessary measures
(7)
Assessment of possible alternatives
(6)
Changes on supplies and notification of suppliers
(4)
Changes on reservations
(5)
Notification of customers
(8)
Final acknowledgments and filing

Fig. 2. A scenario concerning a sadden change of a scheduled plan or an overbooking situation

Figure 2 illustrates yet another class of possible scenarios that are distinguished from the previous one because they do not require immediate involvement of the company's administration. These scenarios refer to sudden changes on scheduled events and overbooking situations. As soon as such a situation is detected (step 1 of the scenario in figure 2), the authorized personnel elaborates a detailed plan of necessary measures (step 2). This plan takes into account similar situations that have happened in the past (step 3) and currently is handled empirically based on tacit knowledge and experience (the plan today usually exists in the mind of some of the most efficient employees of the company). Having decided on how to handle the problem, proper changes are made to reservations (step 4) triggering some notification messages to interested customers. At the same time, suppliers are communicated to handle necessary changes (step 6) after evaluating possible alternatives (step 7). The scenario finishes on reception of acknowledgements about the measures taken and leads to filing of the whole process (step 8).

One could make similar comments as the ones made in the previous scenario with respect to the current management of such kind of situations: empirical management, high overheads, no final recording of the evolution of the scenario, dependence on experienced personnel to coordinate the activities.

Using the OPERATIONS e-business environment and corresponding multi-channel communication (affecting mainly steps 4, 5, 6, and 8) a detailed action plan on the optimal alternative could be elaborated and efficiently deployed, monitored and filed at the end. Successful scenarios could be re-used or properly adapted in the future (used as input to steps 4, 6 and 7) reducing processing times and possible risks. In addition, scenario analysis could provide concrete indicators of customer satisfaction as well as of the efficiency and performance of personnel and external partners.

4. System Architecture

The OPERATIONS platform adopts a multi-tier Service Oriented Architecture that clearly distinguishes the application logic (presentation tier), the business logic (services tier), and the underlying company's extended IT infrastructure (back-office tier). This approach ensures the company's investment in IT infrastructure and middleware tiers (through appropriate application integration using a service based approach) and supports modular deployment of both device specific user interfaces through multi-channel delivery of services and new or adapted operational processes and strategies. All connectivity interfaces are based on standard specifications. The architecture provides mechanisms for receiving, managing and coordinating information or service requests and corresponding messages, through the appropriate middleware components residing at the presentation tier at the one side and to the back-end systems of the company at the other. In addition the architecture incorporates sophisticated knowledge bases where scenarios and user roles are stored.

Figure 3 illustrates this architecture and clearly separates the presentation logic, the business logic, and back-office infrastructure. The service-based business logic components implement the various scenarios that reside in the Knowledge Bases using the information from the company's IT infrastructure and coordinating all flow of information and messages through multi-channel delivery mechanisms.

4.1 Application Logic and multi-channel delivery

A variety of devices (e.g. mobile phones, PDA's, laptops, PC's) and external partner servers are the client devices of the proposed system, exchanging formatted data via standard protocols. The external partner servers will have immediate access to the public business logic web services. The users will be using different devices that run device specific software (special applications, e-mail client, or web browsers). For example the cellular phone users could access the OPERATIONS system, using the SMS service or use mobile services (WAP, e-mail) or run special J2ME/MIDP applications, connecting to the server using a dialup or the more advanced GPRS connection (for a detailed description of these alternative see section 4.5 of this document). Special logic will be developed in the server-side presentation layer that dynamically discovers the device capabilities and redirects the information to the appropriate application logic adapters that, according to the capabilities of the device, will handle the interactions with the user, adapt the content and deliver it via the appropriate gateway. Thus, the presentation layer provides essentially a multi-channel information delivery that is a key feature of the system. It will be the frontend of the company's infrastructure, web servers, application servers and legacy systems, so that the same content could be efficiently delivered to different user devices.

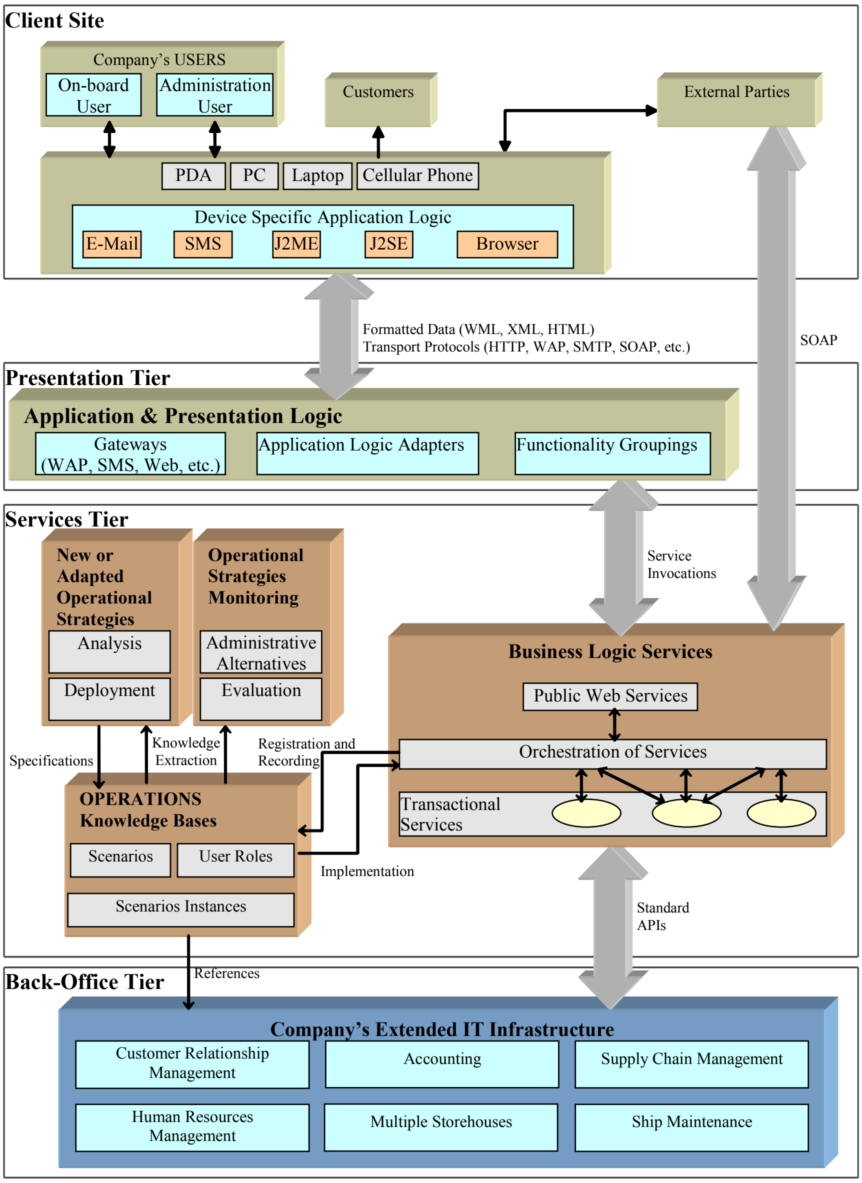

Fig. 3. The OPERATIONS Platform Architecture

4.2 Business Logic and scenarios support

The business logic is captured and implemented in the middleware through utilization of transactional services and complex orchestrated services guided by operational scenarios residing in the knowledge bases. This way, Process Oriented Architecture (POP) is followed which distinguishes between primitive organizational activities, complex operational processes and operational scenario specifications, monitoring and evaluation.

4.2.1 OPERATIONS Knowledge-Bases

These are sophisticated components capable of storing standard descriptions of operational scenarios as well as scenarios instances. The scenarios stored are the driving force for the business logic services and especially for the execution and coordination of complex orchestrated services. They are used as well-defined and set down manifests for specific orchestrations of primitive transactional services. These scenarios are referencing the data and database transactions of the components of company's IT extended infrastructure so that they can fully exploit them. Finally, the knowledge bases contain standard descriptions of user roles (in the context of the available scenarios), providing this way behavioural patterns for all involved users in a particular scenario.

4.2.2 Operational strategies creation, adaptation and monitoring

This is one of the major system components. Using OPERATIONS' Knowledge-Bases, the company can monitor its day-to-day operation and evaluate various alternatives with respect to adaptation of existent or the introduction of new operational strategies. It can thus closely evaluate the performance of its personnel, its external partners or the response of customers to existing products and services. It can also deploy adapted or create new operational strategies by creating or modifying the corresponding descriptions of scenarios and user roles. This is done with high-level tools and methodologies that hide the complex implementation details encapsulated by the business logic components of the system. Scenarios will be modelled and described with the Unified Transaction Modeling Language (UTML) [17] which is a high level language for analyzing, modeling, designing and documenting advanced transaction models and workflows for web applications. UTML is a UML based language providing a flexible and extensible transaction meta-model along with a UML compatible notation system for visualizing the design process of web transactions and services. The notation system utilizes and extends UML finite state machines and state-chart technologies to describe the execution scenarios of the designed transactions. The UML compatibility of the language makes it a candidate for describing complex transactional scenarios. This holds due to the visual nature of UML and its multi-level (conceptual, specification, and implementation modelling) usage which greatly facilitates the description of complex business or software processes. A software tool is available that supports the design process with UTML and provides correctness checking based on built-in well-formedness and well behaving rules. It is an extension of the Rational Rose industrial UML design tool and provides description of the final UTML design into XML format. The description is based on an XML schema which has been developed to support transparent transaction description.

4.2.3 Transactional services

The transactional services component integrates and provides the functionality and information of the back-office resources and applications in a platform and data independent manner. It does so, by providing primitive transactional services which directly interact with the underlying functionality and resources wrapping them accordingly, providing this way a transparent layer of interoperability. When called, this component retrieves the service descriptors, sets up the execution environment and issues appropriate execution requests to the resource providers.

Transactional services are used by the service orchestration component to compose new complex services. Transactional services are considered as the primitive services of the system that have access to the back-end systems. The purpose of these services is to perform standard transactional processing. These transactions are mapped to specific operations for the back-office systems, using the vendor specific programming interfaces. Thus, messages, sent through the multi channel access, are finally executed through transactions mapped to functions of the back-office systems.

4.2.4 Orchestration of services

In this component, primitive transactional services are orchestrated accordingly to support involved business processes of the company. An embedded workflow engine is used to handle and execute compositions of existing services provided by the system and supports long-lived transaction processing. The orchestration done by this component is guided by the operational scenarios residing in the knowledge bases. In order to support the implementation and orchestration of complex operational scenarios, a mapping is required that transforms the scenario descriptions in the knowledge bases to appropriate service-based execution specifications (e.g. WSFL, BPEL4WS, BPML, etc.). Moreover, the notion of compensation is also supported, to allow the engine to reverse actions that have taken place if something goes wrong with the process.

4.2.5 Public Web Services

This component promotes orchestrated or primitive services as autonomous public web services to the outside world. These services will constitute the public interface of the company to the outside world and will be accessed by external users and partners. There are two important reasons for separating private from public business logic when building systems like OPERATIONS:

- The first is that businesses obviously do not want to reveal all their internal data management, decision making and processing logic to the outside world.
- The second is that, by separating public from private logic provides to businesses the freedom to change private aspects of the process implementation without affecting the public protocols and interaction patterns.

4.3 Company's IT Extended Infrastructure

The base layer of the architecture consists of the core IT Infrastructure of the company including the extensions and enhancements to be realised in the context of the project. It is essentially the back-office component of the system consisting of a number of different modules including:

- Human Resource Management
- Customer Relationships Management
- Supply Chain Management
- Ship maintenance
- Multiple Warehouses' Management
- Accounting

This is the major pool of information and functionality for transactional services that communicate through standard Application Programmatic Interfaces (APIs).

5. Summary

In this paper we presented the architecture for an e-business platform that will be used to support the operational strategies of a high volume coastal company. The followed approach uses a Service Oriented Architecture in order to provide a reliable, integrated, configurable and extensible environment that will be able to serve multiple business goals and operational scenarios.

Additionally, multi-channel deliver of information and functionality has been taken into account and appropriate components have been defined that will ensure ubiquitous access of system's functionality through multiple communication channels and terminal devices.

The core business logic tier of the system is a Process Oriented Platform inspired by the SOA principles. It utilizes primitive services, dynamic workflow generation, and orchestration of services to serve multiple business goals in dynamic way.

The implementation of the OPERATIONS platform starts with the wrapping of the company's IT infrastructure and continues with the development of the primitive transactional services that will export this functionality, along with new extensions, to the rest of the system. The next step is the development of the Knowledge Bases and the scenario modeling and representation. Afterwards, the service orchestration will take place and the definition of the public web services. During these steps, prototype interfaces will be developed to facilitate the iterative system testing and evaluation. Then, the presentation tier will be finally developed to support all the multi-channel interaction models that are needed in the envisaged environment.

6. References

1. "Web Services Architecture", W3C, http://www.w3.org/TR/2002/WD-ws-arch-20021114/

2. Web Services Description Language (WSDL) Version 1.2, W3C, http://www.w3.org/TR/wsdl12/
3. Web Services Policy Framework, IBM, BEA, MICROSOFT, SAP, http://www.ibm.com/developerworks/library/ws-polfram/
4. eXtensible Markup Language (XML), W3C, http://www.w3.org/XML/
5. Simple Object Access Protocol (SOAP), W3C, http://www.w3.org/TR/SOAP/
6. Universal Description, Discovery and Integration, OASIS, http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm
7. Web Services Conversation Language (WSCL), W3C, http://www.w3.org/TR/wscl10/
8. Web Services Flow Language, IBM, http://www.ibm.com/developerworks/library/
9. Business Process Execution Language for Web Services (BPEL4WS), IBM, BEA, Microsoft, SAP, ftp://www6.software.ibm.com/software/developer/library/ws-bpel11.pdf
10. Web Services Choreography Interface, W3C, http://www.w3.org/2002/ws/chor/
11. Web Services Coordination (WS-Coordination), IBM, http://www.ibm.com/developerworks/library/ws-coor/
12. Web Services Transaction (WS-Transaction), IBM, http://www.ibm.com/developerworks/library/ws-transpec/
13. Business Transaction Protocol (BTP), OASIS, http://www.oasis-open.org/committees/download.php/1184/2002-06-03.BTP_cttee_spec_1.0.pdf
14. UTML: Unified Transaction Modeling Language, Nektarios Gioldasis, Stavros Christodoulakis, in the proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE 02), Singapore, December 2002
15. SWORD: A Developer Toolkit for Web Service Composition, Shankar R. Ponnekanti, Armando Fox, The 11th International WWW Conference 2002
16. Dynamic Composition of Workflows for Customized eGovernment Service Delivery, Soon Ae Chum, Vijayalakshimi Atluri, and Nabil R. Adam, 2nd Conf. on Digital Government, 2002
17. UTML: Unified Transaction Modeling Language, Nektarios Gioldasis, Doctoral Poster, VLDB 02, August 2002, Hong Kong, China
18. The European Telecommunications Standards Institute (ETSI). http://www.etsi.org
19. The WAP Forum http://www.wapforum.org
20. The Java 2 Platform Micro Edition. http://java.sun.com/j2me