EFFORTS: Service Oriented Architecture for Interoperable Port Systems

Pål Baltzersen, Wilhelmsen Maritime Services (Paal.Baltzersen@wilhelmsen.com)
Marianne Hagaseth, MARINTEK (Marianne.Hagaseth@marintek.sintef.no)
Beate Kvmstad, MARINTEK (Beate.Kvamstad@marintek.sintef.no)
Sven Mathes, Nielsen+Partner Unternehmensberater GmbH (mathes@NundP.com)

Abstract

Port Communities are characterized by being heterogeneous environments, with several actors and stakeholders and differing capabilities to handle information exchange and interoperability. To add complexity to this, changes in national and international legislations, for instance on security and customs, changes in customer requirements on tracking and tracing, and also changes in port organization put new requirements on the development and maintenance of interoperable ICT systems in Port Communities.

Another challenge when developing complex ICT systems is the discontinuity found between different phases in the development process. We want to close the gap between the specification, implementation, validation and roll-out phases to ensure that a consistent system specification is reflected throughout the implementation of the system, and throughout its life cycle.

This paper describes the EFFORTS methodology that supports a process model based approach to develop interoperable ICT systems in Port Communities. The methodology supports a development that clearly separates platform independent and platform specific parts, thus, it is suitable for specifying interoperable ICT systems.

The EFFORTS Methodology is implemented in Enterprise Architect (EA) as a Template Model. This Template Model contains Standard Business Cases (the EFFORTS Toolbox), a Specification Model of an actual port ICT system (specified using UMM as Business Domain View, Business Requirements View and Business Transaction View by the developer of the actual system), and also documentation of the Design Model and Deployment Model (Platform Specific Parts).

An advantage of the EFFORTS Methodology is that a large set of standard business cases and port processes (for example manifest distribution and berth allocation) is included in the EFFORTS Toolbox as actors, use cases, activities, classes, messages, and data elements. These elements can be used directly as-is when specifying the ICT system in a Specification Model, or they can be augmented and further changed to fit special requirements for the actual port community.

The objective of obtaining enhanced interoperability within a Port Community clearly points to an efficient way of exchanging information. Today, all research and experience concludes that the way to do this is by implementing systems following the principles of service oriented architecture where the actors within a community send/receive and publish/subscribe to information by invoking services. These services are loosely coupled and data is transported in messages. This is the architectural foundation in the EFFORTS Methodology.
1 Challenges in Port ICT Systems Development

1.1 Introduction

According to the European Transport Policy document [EUMarTransPol] almost 90% of the export goods traffic from the European Union and more than 40% of the internal trade goes by sea. This equates to about 3.5 billion tonnes of freight loaded and unloaded in EU ports every year, and a large number of people are involved in port related activities in total. The focus area of the European Transport Policy is how the commercial sea ports in Europe can be improved to meet these challenges. The Commission says that they are committed to help European ports to expand and become more competitive such that the ports can operate in a fair, open and efficient manner.

To maintain the position as an important player in the maritime sector, also on global basis, the European Commission has developed “An Integrated Maritime Policy for the European Union” [COM(2007)575 final] providing strategies to face the challenges within the European maritime community. Some areas have been given extra focus, and among them is the area of data and information. In short terms, the European Commission says:

- An establishment of an appropriate marine data and information infrastructure is very important.
- This data shall be made accessible as a comprehensive and compatible system.
- The European Commission will take steps towards an European Marine Observation and Data Network.

To make this happen, it is important that maritime actors and systems starts speaking the same language and have a common platform for development of interoperable systems. Unfortunately, there are several examples today that this is definitely not the case. Studies performed in the FP-6 DG Research integrated project EFFORTS (Effective Operations in Ports) [EFFORTS] revealed that most information exchange in small and middle sized European ports is still done by phone, fax, VHF, e-mail etc. Very small parts of the information exchange are transferred automatically between ports systems and actors. When the ports faces such internal challenges, one can only imagine the difficulties of exchanging data and information between ports, and between port communities and national or regional data information centres.

The benefit of achieving a common language and methodology platform for development of ICT systems is that one can easily integrate new maritime actors and provide a seamless exchange of information. Also, if a number of ports use the same platform, the interface towards external port actors will appear uniform and consistent. One example is that for a ship sailing from port to port within Europe, the information exchange would be more effective because it would be the same procedures to be followed in all port approaches. If one also could ensure that the communication channel was similar within each port area (e.g. digital VHF) [e-NavBWReq] the resemblance would be further improved. A second example is that by using a common platform one has a good starting point for building up regional systems (e.g. cooperation between a number of ports),

---

1 Section 3.2.3 Data and Information in [COM(2007) 575 final]
national systems (e.g. national maritime information centres) and even international systems (e.g. EU maritime information centre). In Norway a project is already ongoing for specification and implementation of such a national Maritime Information Centre, based on the EFFORTS methodology.

The rest of this paper contains the following:

- Section 2 presents the basic principles that need to be included when developing an interoperable ICT system and which have been the foundation for the EFFORTS methodology. This includes process modelling, Service Oriented Architecture (SOA), UN/CEFACT Modelling Methodology (UMM) and ARKTRANS.

- Section 3 presents the EFFORTS methodology

- An example of usage of the EFFORTS methodology is given in Section 4.

1.2 Heterogeneous Port Environment

One of the main challenges today is that the port communities are characterized by being extremely heterogeneous environments, with several actors, stakeholders and systems to handle information exchange and interoperability. To add complexity to this, changes in national and international legislations, for instance on security, customs, immigration and health services, changes in customer requirements on tracking and tracing of goods, and also changes in port organization put new requirements on the development and maintenance of interoperable ICT systems in port communities. Thus ports are truly complex systems requiring the involvement of quite distinct special experts. However, expertise in certain domains is not sufficient to manage communication and co-operation across border-specific disciplines. A way must be found to integrate all these experts and to create a common platform of understanding and to allocate port activities in a holistic and consistent way.

These challenges have been the starting point for the port process and ICT research and development work performed in the EFFORTS project. The proposed solutions are being developed in terms of a common port process map and a methodology for specification, implementation and validation of interoperable port systems. In addition to this, a tool has been developed in Enterprise Architect [EA] that implements the methodology, in addition to an ICT handbook describing how to use the EFFORTS methodology and tools. These results are further described in Section 3.

1.3 Support for Port ICT Systems Development

A well know challenge when developing complex ICT systems is the discontinuity between different phases in the development process. The first phase might include business experts developing nice business- and process models, which in the end are handed over to the SW developers responsible for implementation of the system. The SW developers interpret the business models and starts developing the system. The other way around, when the SW developers are communicating with and reporting back to the business developers and managers, they probably use a language that is not very well known among the receivers (e.g. UML), and information can easily be misinterpreted. In worst case, one might end up with a system quite different from the initial intentions.
The traditional way of developing an ICT system today is to write a specification document containing functional and non-functional requirements to the system, develop a SW design document containing different schemas and diagrams describing the system (often based on UML, [UMM]), writing test cases in a test description document, and in the end produce a test result document. The challenges with this type of documentation appear when a change in one of the requirements is required. It is difficult to catch all consequences of this change, and be able to adjust the correct sections within all the documents.

The tool for specification, implementation and validation of ICT systems developed in the EFFORTS project aims at closing the gap between the specification, implementation, validation and roll-out phases. This has been done by providing a strong link between the process model and a platform independent specification model, which again can be further detailed into a platform specific design and implementation model. This tool will make it easier to register and implement changes appearing from changes in user requirements and hence make the development process more efficient and errorless. The tool is also based on Service Oriented Architecture principles, helping the system to be interoperable and easy to integrate with other systems, see Section 2.2 and 2.4.
2 Foundation for Developing Interoperable Port ICT Systems

2.1 Introduction

As described in the introduction, the main challenges when it comes to development of port ICT systems are:

1. To agree upon a common language that connects all actors in the extremely heterogeneous port communities.

2. To ensure that functional and technical requirements of a port ICT system are consistent all the way from the process modelling phase to the validation phase.

To find a solution to the challenges in point 1, some important principles need to be followed. For instance, to be able to obtain enhanced interoperability within a Port Community, which again clearly points to the need for an efficient way of exchanging information, the principles of loosely coupled system modules and flexibility must be followed. Today, Service Oriented Architecture (SOA) is the most well known architecture providing these principles. In SOA, the actors within a community send/receive and publish/subscribe to information by invoking services. These services are loosely coupled and data is transported in messages. SOA has been selected as the architectural foundation in EFFORTS.

One solution to point 2 above is to develop a tool that ensures consistency from the business development phase to the implementation phase of an ICT system. A common language also needs to be established.

The EFFORTS project has developed a methodology and a tool to meet these challenges. They are characterized by being based on a process approach, and the tool is model based. The following sub-sections describe the architectural and principal foundation for the EFFORTS methodology and tool:

- Process modelling, Section 2.2.
- ARKTRANS (common language and role description), Section 2.3.
- Interoperability (SOA, networked enterprise), Section 2.4.

2.2 Process modelling

Process modelling is not a new concept within the business environment anymore. However, it is still not implemented in many organisations due to the different management systems, background of business, knowledge of human resources and different concepts offered by vendors etc. To enhance this situation, a port process platform has been developed in the EFFORTS project [EnterpriseMod] and [Zuesongdham]. This platform functions as a process library for the ports where they can exchange information about processes and their relevant objects (such as e.g. organisations, resources) involved in port operations. The process platform will provide the ports with a better understanding of port processes as well as a more efficient way to analyse their operations and identify the opportunities for improvements.
The EFFORTS port process map is a very important part of the EFFORTS methodology. Certain parameters in the process model are brought into the specification model implemented in the EFFORTS tool. This approach ensures the consistency between the business development and implementation phase when developing new port ICT systems. More information on the interface between the process model and specification model is given in Section 3.2.

2.3 ARKTRANS

To support the consistency between the EFFORTS port process map and specification model, the ARKTRANS role descriptions have been used to ensure a common understanding and description of the actors and/or resources involved in the port operations. ARKTRANS is also used by other frameworks and European projects such as MarNIS and FreightWISE [ARKTRANS] and [ITS].

ARKTRANS is the Norwegian system framework architecture for the transport sector. ARKTRANS is the result of a comprehensive study of all transport modes (road, sea, rail and air). The framework architecture is harmonised across all transport modes, meaning it is a multi modal architecture. ARKTRANS is also harmonised across freight and passenger transport. The pursuance of the work with ARKTRANS is ensured by ITS Norway.

By using ARKTRANS, the systems based on the EFFORTS methodology will speak the same language as other systems in the transport sector also based on ARKTRANS.

2.4 Interoperability

The last but not least foundation principle in EFFORTS is interoperability. The following sub sections describes different architectural aspects that needs to be involved to promote interoperability, such as networked enterprise, SOA and semantic interoperability. An example on SOA applied on a port community is also given.

[Vernadat] writes the following:

*Interoperable enterprise systems are essential components to build agile organizations using a mixed service and process oriented approach. Modern organizations, be they considered from the intra or inter-organizational point of view, need to be made interoperable both in terms of their business processes, their applications or IT systems, and even their human resources to face current business challenges.*

ICT application interoperability is becoming a reality thanks to recent advances in technology and standards, for instance on middleware. Many companies are getting away from tight application-to-application interfaces and are abandoning traditional enterprise application integration (EAI) approaches that have resulted in too monolithic systems. Instead, they are adopting more service-oriented, loosely coupled, message-based, and asynchronous techniques.

For the *single enterprise*, integration means that it is necessary to create a coherent information system architecture in which the various administrative and business processes, information stores, and systems are integrated so that they appear seamless from the point of view of the individual user. In other words, there is a clear need for an ICT systems interoperability platform if the goal is to achieve improved efficiency and effectiveness of internal processes and system operations, timely procurements and fast product delivery, easy and instant access to all required
job-relevant information by staff members, and enhanced reporting and monitoring facilities at the administrative level.

2.4.1 Port Community as a Networked Enterprise

*Networked organizations* are characterized by distributed control, inter-organizational business processes (i.e., business processes that cross enterprise boundaries and therefore do not belong to one enterprise), various producer–consumer supply chains, and shared information and knowledge. The main challenge is operations optimization via co-decision, co-ordination, and even negotiation mechanisms.

A port community can be regarded as a networked organization, or as a Networked Enterprise. Within the processes defined in the EFFORTS port process map we typically see this characterized as follows:

- Control is passed from one actor to the other triggered by events or information flow; for instance with Berth Allocation, control is passed from vessel to agent to terminal operator.
- The processes do not belong to one single enterprise; same as above. These three enterprises (vessel, agent, terminal operator) are autonomous organizational entities.
- Producer-consumer chains; resource owners (trucks, stevedoring, garbage collection etc) will typically represent such chains vs. the vessel.

A networked Enterprise is dynamic in nature, i.e. nodes or entities can be added or removed with minimum disturbance to the processes being active within the enterprise. This provides the necessary agility to sustain changing economic conditions and/or to implement new strategies. Thus, we will define a port community with its actors, processes and ICT systems as a Networked Enterprise.

Agility requires interoperable enterprise systems, i.e., reconfigurable systems that can work together. In this context, interoperability refers to the ability of a system (or process) to use information and/or functionality of another system (or process) by adhering to common standards.

2.4.2 SOA Principles

In [MarksBell], SOA (Service Oriented Architecture) is defined as follows:

"SOA is a conceptual business architecture where business functionality, or application logic is made available to SOA user, or consumers as shared reusable services on an IT network. “Services” in a SOA are modules of business or application functionality with exposed interfaces, and are invoked by messages”
Figure 1 - Service Oriented Architecture Components

Figure 1 shows the various elements that are part of a Service Oriented Architecture. In this context, **Application** means the set of systems within a community that either publishes or consumes services. A **Service Repository** can be part of SOA. Such repository, for instance implemented as UDDI (Universal Description, Discovery and Integration) [UDDI] or as an ebXML Registry [ebXML], makes it possible to search and load service descriptions by XML message interactions.

**Service Bus** is the set of mechanisms and components that makes it possible to transmit data between the application publishing a service and the corresponding consumer. A specific implementation of Service Bus would typically contain capabilities for transmission, monitoring, orchestration, storage, protocol and format adaptors, and data mapping. These capabilities are typically built into middleware products.

A **Contract** formulates the rules and conditions that are applicable when consuming services, e.g. authentication rules, mandatory data elements etc. A contract would typically be embedded in WSDL objects. **Implementation** is the physical occurrence of a service published for use within a defined domain, which can also be “public”. **Interface** defines the access mechanism, e.g. a Web Service/URL.

The Service itself can contain **Business Logic** like e.g. authentication, data validation and various generic conditions and constraints.

Figure 2 provides a picture of how service implementation is done in SOA.
A logical definition of a service indicates the business function that it performs. However, the service implementation may be a relatively complex process and depend on many sources of information to fulfil the functional requirements designed for the service. In addition, the technical design of the service implementation cannot make any assumptions about the context in which the service would be invoked. The service may be used in isolation, as part of a sequence of calls on behalf of a real-time transaction or as a subordinate component in an asynchronous Event Driven Application (EDA). As a result of this inherent complexity, most service implementations are likely to be relatively simple, self-contained and stateless. (All three principles — simplicity, isolation and statelessness — are the best practices of design for all distributed systems.)

In the context of a port community, with reference to Figure 2, one should note the concept of Wrapped Services. This is the situation where, in a port community, there is a demand to publish services where the actual functional business logic is embedded in legacy systems. In such cases, a service implementation object is written to connect the business logic to the service interface e.g. through an API.

Henrik Hvid Jensen and Erik Billington have in [HvidJensen/Billington] defined the following main drivers for investing in SOA implementation:

- The need for simplified application to application (A2A) integration
- Method of integration should not require large investments in technology or competence development
- One should use a pragmatic and evolutionary methodology
• Participants must be able to be part of a SOA with solutions based on existing platforms
• The objective of secure interoperability (integrity, ownership and access control) among information sources
• Loosely coupled applications
• The objective of increased total flexibility within an enterprise
• The implemented system must cater for sending, collection and reaction to events

2.4.3 SOA applied on a Port Community

According to the above descriptions of the port community as a networked enterprise and also applying definitions of SOA, we can define a set of basic capabilities to be contained in a port community to obtain an efficient level of interoperability:

1. Definition of processes; this is provided by the EFFORTS port process map (see Section 3.2.2)
2. Definition of actors; this is provided by associated ARKTRANS roles
3. Modelling capabilities; this is provided by EFFORTS EA Roadmap Framework based on UN/CEFACT Modeling Methodology (UMM) and Enterprise Architect.

Further, by applying SOA principles on Port Community as a Networked Enterprise we can create a model for enhanced interoperability. Some basic conceptual elements are:

• The Enterprise Service BUS (ESB) will become common for all actors, processes and systems in the port community.
• This again applies to the horizontal integration within the enterprise (other principles that might be applied in possible vertical integrations should be irrelevant in this context)
• To architect an enterprise from a SOA principle, three main concepts are essential to model the organizational; events, services and processes. In the context of EFFORTS, these events, services, and processes will exist in the Port Community as a whole, and not within a single entity being member of the port community. Consequently, to obtain enhanced interoperability, these three concepts are the SOA elements to model, see Figure 3.
In this example we see (ref. definition of SOA components in Section 2.4.2):

1. Four *applications* within the Port Community;
   a. Agent
   b. Port Authority
   c. Other Authorities, e.g. Customs
   d. Service Provider

2. Three *applications* outside the community
   a. Cargo Owner
   b. Vessel
   c. Vessel Operator
3. Five services with interfaces

4. The Port Community Enterprise Service Bus (ESB) (yellow)

5. A UDDI Service Repository which in the example depicts a shared repository for all European ports (upper left corner in Figure 3)

The example shows how different actors can access their local systems through services deployed on the Port Community ESB. For services (i) and (k), it shows how it further connects to the application associated with the Publication of the service (green dotted line).

As an example, Vessel Superintendent who is located outside the Port Community consumes information published through Service_i on the Community ESB. The green dotted line indicates that this service is published by a Service Provider, e.g. one which provides uploading of digital charts to the system used by a Superintendent.

2.4.3.1 Service Repository

A Service Repository serves the purpose of a “Yellow Pages” concept for services accessible by message interaction. That means one can send a XML message to the registry requesting the address/owner of those services which fit your search criteria embedded in the request message.

With the resulting address, e.g. an URL, the requester can fetch the description of a message, and then use this information to implement the interface necessary to consume the service. An example is shown in Figure 4.

![Service Repository Diagram](image)

**Figure 4 – Service repository example**

In the context of Port Community/SOA one can see that this concept is applicable at local, regional/national and also at European level. In a practical case, to implement a service registry at local level may not be feasible since the actors within that local community can share this information using much simpler mechanisms. Even at regional/national level, this can be the case, dependant on the size and complexity of the domain.
However, on a European level, creating such repository will be a very efficient way for a port community to build interoperable solutions, specially if those posting services to the registry all are using same standards. The economy of scale becomes obvious if all communities within this context apply same architecture (same SOA principles) and also the EFFORTS methodology for development and implementation.

2.4.4 Semantic Interoperability

ESB mechanisms ensure message-enabled integration and also ensure that application interoperability remains at the data and syntactic level. However, semantic interoperability concerns data/information integration and consistency. Semantic interoperability is the ability to share, aggregate, or synchronize data/information across heterogeneous information systems. In other words, it is about making sure that two communicating systems interpret the information the same way. A simple example is the use of “Master” and “Captain” which both mean the same thing in a shipping context; a description of the role “highest ranking officer onboard a vessel”.

Considering the number of databases and information systems in use in a networked enterprise, also considering that interpreting information within that enterprise is context-dependent, one can realize the complexity of this problem. This is a hard problem and unfortunately there is no packaged or out-of-the-box solution. It is up to each community to develop its own strategy and solution for semantic data interoperability. One solution is to build shared metadata repositories that describe the content and intent of data stored in the various information systems used in the enterprise or by partner companies. This can be achieved through many products that combines with ESB capabilities, e.g. from various middleware products from large vendors like Microsoft, IBM and many others.

Another solution is to build an ontology for enterprise interoperability, [UKMZ] and [WSP]. In ICT terms, an ontology is a specification of a conceptualization of the knowledge of some specific domain. It is used to formally describe every concept used in this domain as well as their relationships and associations using axioms, predicates, and formulae. Upper ontologies deal with universal concepts (of time, cost, quality, etc.) while domain ontologies deal with concepts specific to an application domain (e.g., power electricity, geometrical features, customer relationships, etc.). Ontology is used as a pivotal language to map concepts used in one system with concepts of another system and resolve the semantic impedance mismatch. It is probably one of the most promising perspectives in research to use enterprise models in an optimal way for enterprise architecture design. Ontology based semantic interoperability is a very active and long-term research objective.

One example of an ontology-based concept is developed in Norway and based on information modelling using methodology defined in ISO15926. This concept is implemented in a RDF based (Resource Description Framework, [RDF]) tool to support service and maintenance in the offshore industry. A Norwegian, government funded (Norges Forskningsråd) research project has looked at how to extend this to the shipping industry (Informasjonsmodell for Skip-Land kommunikasjon, [ISLK]).

With reference to the Commission policy document [COM(2007)575 final], development and implementation, i.e. using OWL (Web Ontology Language, [OWL]), of a European ontology-based information model for the domain as defined by EFFORTS would be a contribution to meet those objectives.
3 EFFORTS Methodology: Development of Interoperable Port ICT Systems

3.1 Introduction

The EFFORTS Methodology is implemented in Enterprise Architect (EA) as a Template Model. This Template Model contains Standard Business Cases (the EFFORTS Toolbox), a Specification Model of an actual port ICT system (specified using UMM as Business Domain View, Business Requirements View and Business Transaction View by the developer of the actual system), and also documentation of the Design Model and Deployment Model (Platform Specific Parts).

An advantage of the EFFORTS Methodology is that a large set of standard business cases and port processes (for example manifest distribution and berth allocation) is included in the EFFORTS Toolbox as actors, use cases, activities, classes, messages, and data elements. These elements can be used directly as-is when specifying the ICT system in a Specification Model, or they can be augmented and further changed to fit special requirements for the actual port community.

EFFORTS methodology supports:

- Model Driven Development.
- Development of interoperable systems due to the focus on the interfaces between modules and systems, and due to the top-down approach of the systems development.
- Integration of heterogeneous systems, since it supports direct use of standard messages and processes that are available to the system developer through the EFFORTS toolbox.
- Built on SOA.

3.2 EFFORTS Methodology

3.2.1 EFFORTS Roadmap

Basically, the EFFORTS methodology comprises of a procedure model (“Roadmap”), [ROADMAP]. Major function of this roadmap is to improve development of interoperating ICT solutions in the port sector. In order to accomplish this, the roadmap has to fulfil three main requirements:

- **Force a structured approach**: The main purpose of a procedure model for ICT solutions is to force a structured approach. This is achieved by means of defined phases with clearly specified input and output documents. As a consequence, the roadmap needs to ensure continuous support of all phases of the development process – mainly specification, implementation, validation and roll-out. This is achieved by means of a model based approach where all phases of the development lifecycle access one common model which is progressively expanded throughout the development process, [MDA]. Specification and design of interoperable systems requires involvement of numerous experts with a very different background – business experts, analysts, software architects, experts for quality management etc. This means that the underlying model which is used
throughout the process must be effective and easy to understand for all these parties involved.

- **Foster a process based approach:** Today’s data interchange standards like EDIFACT, ANSI X12 or SWIFT focus only on the specification of data formats and document types submitted. There is no relation to underlying processes linked to these documents. As a consequence, these standards allow a wide range for interpretation when they are mapped to “real-world-processes”. One simple example is that there are no checks on correct sequence of documents/messages exchanged (“choreography”).

  The EFFORTS roadmap shall be strictly based on a process based approach. This means in particular that the entire development process is based on a business process model specifying actors, use cases (processes) and interaction between actors and use cases, [Zuesongdham] and [ProcModHandbook]. Figure 7 shows the relation between the EFFORTS Port Process Map and the Specification Model described using the EFFORTS Roadmap.

- **Allow use of standard business use cases:** Another important advantage of a process based approach is that recurring business processes can be specified as “standard use cases”, [PortProcessMap]. This means that a certain set of functions can be re-used in an arbitrary number of applications.

From a technical viewpoint, this approach is implemented by service oriented architecture (SOA). SOA is particularly beneficial in heterogeneous distributed environments since the functionality is packaged as interoperable services made accessible via a network, [KrBaSl].

Basically, Service oriented architectures extrapolate the component based approach to interoperable systems communicating via wide area networks.

Use of generic services throughout a wide area network requires that the underlying approach must be platform independent. The roadmap must support a clear separation between **content** (information and transaction sequence – “what”) and **infrastructure** (transmission formats/protocols – “how”) into different layers. The services can be combined and re-used by arbitrary business processes.

The EFFORTS Roadmap is based on three different “views” on the target system with different levels of detail and abstraction, see Figure 5.

- The **Business Domain View (BDV)** has its focus on the business cases and actors involved. It defines the boundaries of the system in view.

- The **Business Requirements View (BRV)** deals with the processes defined in the BDV. It models the internal behaviour of the processes by means of activities and state transitions.

- The **Business Transaction View (BTV)** finally elaborates the sequence of activities and messages exchanged.
The sum of these views is called the **Specification Model**. The Specification Model in turn gives the input for the Test Model. Both Test Model and Specification Model form the basis for the subsequent phases of the Procedure Model – implementation, validation and roll-out, see Figure 6.

One major advantage of this approach is that it supports re-usability of generic components, called “Standard Business Cases”. These standard business cases are based on the process map, see Section 3.2.2. Elements defined within this process map can be used as “building bricks” to specify one specific interoperability solution in a specific port. The processes of the Port Process Map match very well the services within a Service Oriented Architecture.
3.2.2 EFFORTS Port Process Map

The EFFORTS port process map depicts different operational areas of a port irrespective to its cargo types, operational concentration and ownership model, [Zuesongdham]. Due to its generalisation, port operators therefore are able to retrieve information from this model and customize it to their requirements. The customization is done by including processes from the port process map (Phase 1 in Figure 7) into the Roadmap (Phase 2 in Figure 7) as standard business cases. The port process map is used as input to the specification model. The standardised port processes from the port process map can be used directly in the specification phase of the specification model, or they can be refined and supplemented by new processes specific for a certain port.

Figure 7 - Relation between Process Map (yellow) and Specification Model (green)

Five areas (process domains) are defined in the port process map, see Figure 8 [Zuesongdham], [PortProcessMap]:

- **Marine** includes processes associated with navigation and ship operations in ports.
- **Infrastructure and Resources** include processes regarding land and water infrastructure of ports.
- **Logistics** include processes associated with operation among ship, terminal and cargo/passengers.
• **Interfaces** include processes not directly linked to the port operation but affected or influenced by the processes executed in the port area.

• **Public Bodies** include processes directly related to public entities having influences on port operations.

---

**Figure 8 - EFFORTS Port Process Map**

3.2.3 Phases in the Roadmap

This section describes in more detail the phases in the EFFORTS Roadmap as shown in Figure 6:

1. Specification
2. Implementation
3. Validation
4. Roll-out

3.2.3.1 Specification

The specification phase, the green part of Figure 6, can be described as follows:

**Business Domain View - BDV**

The top-down EFFORTS approach begins with getting a clear understanding of the business domain. In particular, the following properties of the system in view are specified by means of the business domain view (BDV):
• boundaries of the system
• actors involved
• business processes, i.e. the interaction between actors and the system (“use cases”)

The major artefact of the BDV is the use case diagram (UCD) and the corresponding use case descriptions. Figure 13 shows an example of a use case diagram.

Each process is also documented by a use case description containing the following properties:

• Identifier/name of the process
• Description: Short textual description of the process
• Sequence: Sequence of activities carried out
• Trigger: describes the activation of the process
• Result: explains the results of the process including triggers to other processes
• Actors: description of the actors

Business Requirements View (BRV)

The objective of the BRV is to identify and specify requirements of the business processes defined in the BDV. This is limited to the interoperability requirements of the involved business processes. Internal functionality of the processes will only be modelled if necessary to describe the interoperability properties of the process in view.

Interoperating business processes, i.e. those involving two or more actors are called (business) collaborations in the UMM terminology [UMM]. Each business collaboration identified needs to be refined into a sequence of activities. Basically, the BRV serves to identify the following properties of collaborations:

1. The flow of activities locally (i.e. at one business partner/actor) as well as globally (i.e. between business partners).
2. Business entities involved in the collaboration.
3. State transitions affected by the collaboration.

Depending on the complexity of the collaboration in view, the “UML toolbox” offers different diagrams to model these properties:

• Activity diagrams to model the flow of activities
• Class diagrams to model entities and relations between entities
• State diagrams to model state transitions during the execution of a business collaboration.
Business Transaction View (BTV)

The BTV focuses on the atomic interactions identified within the BRV. The purpose of the BTV is to describe the choreography of these interactions and to specify the information exchanged; therefore it consists of the following sub-views:

1. The business choreography view to model the flow of business collaborations which may be composed of several interactions.

2. The business interaction view to specify all business transactions in detail.

3. The business information view to capture the information exchange within the business transactions.

3.2.3.2 Implementation

According to the EFFORTS roadmap, the implementation phase comprises of the following steps, see Figure 9:

- **Software design** – creation of the platform specific model (PSM): The input for the software design phase is the specification model, see Figure 6. The output is a platform specific model (PSM). The structure and detailed content of the PSM is dependent on the intended target system; in today’s typical target environments it will usually comprise the following elements:
  
  o Database dependent specifications like entity-relationship-models/table specifications specific to the target DBMS – Database Implementation Model

  o Packaging of the software delivery – Functional Implementation Model

  o Specifics of the applied middleware/messaging system – Integration Implementation Model

- **Software implementation**: The software implementation phase of the development process includes implementation of the database tables and programming of the functionality of the target system. Therefore, it is based on the Database Implementation Model and the Functional Implementation Model. The result – i.e. the implemented software packages – will be validated against the specification by means of unit tests. The packaging should be done so that each package can be validated independently. The software implementation concludes with successful unit tests.

- **Software integration**: This phase deals with integration of the implemented system into the target environment. It concludes with successful execution of the software integration tests (SIT) according to the specification model.
3.2.3.3 Validation
As outlined above, each phase of the described process is verified against the result of the previous phase. This ensures at the end of the implementation phase that the target system has been implemented “as specified”. Nevertheless, this does not guarantee that the target system meets the intended user requirements. This is the purpose of the validation phase, usually carried out by a set of User Acceptance Tests (UAT).

Following this approach, a test plan including test scenarios and test cases for the UAT is created in parallel to the specification process. The creation of this plan should be under the responsibility of a test manager and is done in close co-operation with the business experts. This test plan – also called the “test model” – is part of the specification model.

3.2.3.4 Roll-out
Input for the roll-out phase is the roll-out plan created during the implementation phase and completed by the validation phase. This is complemented by the validation protocol which is the resulting document from the validation phase. The roll-out phase consists of following issues:

- if applicable, data migration in the production system
- controlled commissioning of the target system
- if applicable, training of the users
- implementation of structures for maintenance and support
- implementation of release cycles
4 What’s in it for the Ports

The EFFORTS methodology has been developed based on communication with several European ports of different sizes and types. The feedback from the ports has been used to update and improve the methodology to ensure that it is as useful as anticipated. The following figure shows the benefits for the ports:

**Figure 10 – EFFORTS results and benefits for the ports**

It is quite clear that by introducing interoperability in the port community, all actors involved in port operations will have easier access to information. The today’s situation where one likely would have to make a phone call or send an e-mail to request for information would be avoided because the information could be either transferred automatically by point-to-point, publish-and-subscribe or request/reply models (see Section 2.4.1).

The automatic information exchange is clearly also more efficient than traditional information exchange, not only within, but also between different port communities. A national, or even regional (EU), service repository would give the ports access to public port services via one access point and be able to do fast implementation of port related services.

The ICT studies performed in the EFFORTS project also revealed that, especially in small ports, data are being manually transferred from system to system. People are using large parts of their working day for this kind of work and most of the time this process is problem-free. However, the chances of writing a wrong number, putting the commas on a wrong place, storing the data in wrong places are relatively large. When it happens, it might be almost impossible to correct errors again, and in worst cases the economic numbers of the port might end up being wrong. Using more automatic exchange of information would decrease the chances of putting wrong information in the wrong place.

The traditional approach of specification, implementation and validation of a new ICT system is to interpret requirements decided by e.g. business developers or management. The chances of misinterpretation exist, and the chances of ending up with a different system than intended are
definitely present. By using the EFFORTS methodology, which provides business and SW
developers with one common methodology, the chance of getting a system fulfilling the
requirements, is improved.

The last sections of this paper introduce an example on how the EFFORTS methodology can be
used. The case taken from an ongoing Norwegian project called MIS (Maritimt Informasjons
Senter). The objective of MIS is to offer existing and new maritime services to port related actors
on a national level. The case shows how an existing application (Resource Hub) can be integrated
in an interoperable ICT system and how its services can be made available to more than one port
at a time.

4.1 Resource Hub integrated in a Maritime Information Centre

MIS (Maritimt Informasjons Senter) is a Norwegian initiative to develop a maritime information
centre for actors in Norwegian ports and related transport modes. The focus is on developing a
‘single window’ solution for both authority and commercial information, see Figure 11.

Figure 11 - Overview of MIS

Resource Hub (RH) is a prototype system that fits well into the MIS concept. The idea of the
Resource Hub is to provide a database of services that can be booked by Service Users, for
example a service user representing a ship. Each service is made up of a set of resources, see
Figure 12. Each resource is owned by a resource owner, which can either be the Resource Hub
system itself or an external Resource Owner, that is, an external service provider system. Several
resource owners can cooperate to offer a certain service to the Service Users.

The services can be booked through the Resource Hub front end, or by interfacing a Service
Provider system directly to the Resource Hub.

One advantage with the Resource Hub is that several resource owners can cooperate to offer
services that are simple to use by the service users. The service providers can cooperate to offer
new services that they are not able to offer themselves.

Bookings can be easily done since the Resource Hub coordinates in a single system the resources
and services owner by several owners. An example is an arrival service, see Figure 12, containing
a truck resource, a lineman resource, fuel filling resource among several other resources.
The rest of this section shows how the EFFORTS Methodology can be used to model some aspects of the Resource Hub system. We will focus on the overall requirements specification (BRV) and the description of the interoperability between several resource owners (service provider systems).

Figure 12 - Services and Resources in Resource Hub

Figure 13 shows a high-level Use Case Diagram of the Resource Hub. This is part of the BDV – Business Domain View – that describes the boundaries of the system and also gives an overall view of the processes in the system. The actors are Shipping Agent, Ship, Terminal Operator, and Resource Owner. The ovals indicate processes that must be fulfilled when some Service User (Ship/Agent) orders a service.

Figure 14 shows an activity diagram that is part of the BRV – Business Requirements View. It shows how the handshaking between the Resource Hub and several external Resource Owner systems can be done for a service booking containing several resources with several owners.

First, the booking is set to Reserved in the Resource Hub. Then, each of the resource owners must be asked if this booking is OK for them, since we have supposed that the external resource owners hold the master data. If all of them answer OK, a commitment can be sent to all participating systems, and the booking can be confirmed. If one of the systems does not answer, or sent a negative answer, all reservations regarding this service must be cancelled.

In this case, we realize that an activity diagram is an efficient tool for describing the interoperability between the various port related systems cooperating with the Resource Hub.
The request service order can be sent both from the vessel and from the shipping agent. The shipping agent is the appointed ship representatives.

Invite external resource owners to take part of the resource hub or give contact information to external resource owners.

Figure 13 - Resource Hub: Business Domain View (BRV) of Service Bookings
Figure 14 - Resource Hub: Business Requirements View (BRV) of Booking Confirmation
5 Conclusions

The Port Community as an Enterprise is a target for SOA. It does not imply that all systems within that community itself must be based on SOA. However, all legacy systems participating in building enhanced interoperability within the same community must develop interfaces to expose business logic as services within their system. Obviously, they must also provide mechanisms to consume services exposed by other parties.

An important factor is that some elements of SOA must be implemented on the Enterprise level. The Enterprise Service Bus is the most important. That raises some organisational issues related to information ownership, security, integrity, and system operations, among others.

When building such systems, on all levels as depicted, the EFFORTS Methodology has proven to be an efficient tool. Through the work in EFFORTS and other projects, we find that a model driven development methodology is a very important element in making such projects more efficient. The EFFORTS project deliverables comprising the EFFORTS Roadmap and the Port Process Map implemented in Enterprise Architect support the objectives of enhanced interoperability.

In Norway, we intend to bring these results further when working on the Maritime Information Centre (MIS) research project which includes the Resource Hub concept.
6 References

[ARKTRANS] http://www.arktrans.no/?nid=6917&lcid=1044


[EFFORTS] www.efforts-project.org


[EnterpriseMod] Zuesongdham, P., Combined approach for enterprise modelling: CIMOSA and SOA as dynamic architecture in maritime logistics

[EUMarTransPol] DG TREN/European Commission, Office for official publications of the European communities, 2006


[ITS] www.its-norway.no


[MarksBell] Eric A. Marks and Michael Bell: Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology (Hardcover)


[OWL] http://www.w3.org/TR/owl-features/


http://www.w3.org/RDF/  


http://www.oasis-open.org/presentations/uddi_v3_webcast_20050222.pdf  

Mike Uschold, Martin King, Stuart Moralee and Yannis Zorgios: The Enterprise Ontology http://mis.uoa.gr/Portals/57ad7180-c5e7-49f5-b282-c6475c67ee7/Temp/98-ker-ent-ontology.pdf  


and UMM User Guide  


Lawrence E. Whitman, Danny Santanu, and Hervé Panetto: AN ENTERPRISE MODEL OF INTEROPERABILITY http://hal.archives-ouvertes.fr/docs/00/08/20/37/PDF/366a.pdf  