academicJournals

Vol. 9(18), pp. 814-825, 30 September, 2014
DOI: 10.5897/SRE2014.5932
Article Number: 83B0E2247847
ISSN 1992-2248 © 2014
Copyright©2014
Author(s) retain the copyright of this article
http://www.academicjournals.org/SRE

Scientific Research and Essays

Full Length Research Paper

Enhancing container transportation traceability based on intelligent product and web-services: Model driven approach

Mehdi NAJIB[1,2]*, Mohamed Yassine SAMIRI[1,2], Abdelaziz EL FAZZIKI[1] and Jaouad BOUKACHOUR[2]

[1]Cadi Ayyad University, Av Abdelkrim Khattabi, BP 511, 40000, Marrakech, Morocco.
[2]University of Le Havre, 25 rue Philippe Lebon, BP 1123, 76063, Le Havre cedex, France.

Received 14 April, 2014; Accepted 21 July, 2014

Containers transportation involves various intermediate supply-chain actors. These containers are subject to myriad risks that can occur at any moment. In addition, the absence of a specialized stakeholder for securing containers flow and the lack of communication among actors impedes identifying responsibilities when an adverse event arises. For this purpose, we advocate the use of traceability system to enhance containers transportation security. This objective is achieved through enhancing systems interoperability to collect necessary information for identifying risks. Thus, we opted for urbanizing a traceability system to meet the new requirements related to risk management. In this paper, we propose the use of web-services to overcome the problems related to the heterogeneity of systems used by supply-chain stakeholders. Thus, we proposed a development process based on model driven architecture (MDA) to automate the code generation of these web-services. Moreover, we used the concept of intelligent product (IP) to propose an intelligent container able to communicate with external systems and to manage information related to its traceability. This solution is validated by a real case study.

Key words: Intelligent product (IP), model driven architecture (MDA), service oriented architecture (SOA), business process (BP), traceability.

INTRODUCTION

Supply-chain activities are often subject to undesirable events, particularly in case of containers transportation. Establishing a risk management processes for securing containers flow is a tedious task, due to the heterogeneity and the distribution of actors involved in the supply-chain processes. This implies the participation of several actors who have different priorities concerning the establishment of risk management process (Erkut et al., 2007). This situation is compounded by the absence of an actor who has the visibility over the entire supply-chain activities. We have the conviction that traceability systems have the potential for improving containers transportation security

*Corresponding author. E-mail: mehdi.najib@edu.uca.ma, Tel: 00212 6 67 34 97 60.

(Boukachour et al., 2011). These systems allow the establishment of a process for collecting necessary information to indentify risk events. Furthermore, information related to containers traceability is the basis for investigating the responsibilities of supply-chain actors. To achieve this goal, the traceability system should consider the supply-chain specificity.

The reticular organization of supply-chain actors and their interventions in various networks hinder the cooperation for enhancing traceability. On the one side, the involved actors are reluctant to share information that describes their internal operations due to the risk of information disclosure to competitors. On the other side, there is a lack of interoperability among the traceability system and the heterogeneous systems used by these actors. However, despite these constraints, new regulations impose the adoption of a traceability system, particularly in case of hazardous materials transportation. Thus, adopting traceability should be considered as a competitiveness factor for supply-chain actors.

This work is motivated by supporting the implementation of the Container Safety Initiative (CSI) (Yang, 2011). This initiative aims to remedy the emergence of fraudulent activities, exploiting the vulnerabilities of multi-modal containers transportation. For this purpose, CSI imposes suspicious containers inspection in foreign seaports before goods shipping. However, inspection activities cost and the limited resources allocated for this mission impose the establishment of a decisional process for targeting high risks containers. Consequently, we envisage exploiting traceability information to improve the relevance of suspicious containers targeting.

The first objective is to propose a Service oriented architecture (SOA) for Geo-location Optimization and Secure Transport (GOST) traceability system, and the use of Intelligent Product (IP) concept to develop an intelligent container. The second objective seeks to ease systems adaptation for sharing traceability information based on the Model Driven Architecture (MDA). The MDA automates generation of the web-services foreseen to be interfaced with the GOST system.

RELATED WORKS

Traceability

According to ISO8402, traceability is associated with the ability to trace the history, the use and the location of an entity, based on recorded identifications (Aung and Chang, 2014). Accordingly, product traceability focuses on the identification of its origin, its location and the operations that altered product’s state based on the data provided by the information system of each supply-chain actor involved in the product's life cycle.

Traceability systems are based on a set of basic functionalities. According to Karlsen et al. (2012), traceability system must be able to trace both products and activities, by determining traceable unit and elementary activities. Regattieri et al. (2007) focused on the ability of the traceability system to describe and to communicate product characteristics, taking into consideration confidentiality constraints.

Traceability systems are used in various fields such as food industry (Jin and Zhou, 2014), quality control of products (Aung and Chang, 2014), improving supply chain visibility (Woo et al., 2009) and enhancing risk management in case of hazardous goods transportation (Planas et al., 2008). Bechini et al. (2008) have proposed a generic traceability model for controlling products quality during transportation in food industry. This model allows tracing both of the physical product and the activities executed by the actors involved in the food industry. Thus, it represents a solid basis for developing an efficient traceability system that minimizes recovery cost in case of food contamination. However, this model was not experimented in a real case study. Woo et al. (2009) have proposed a traceability system that enables products monitoring, taking into consideration the granularity of the traceable unit (container, pallet and product). In the MITRA project, traceability system was also used for monitoring the intervention for incidents related to transportation of hazardous materials in Europe (Planas et al., 2008). This project aims to improve the intervention of civil security centers in case of accidents related to hazardous goods transport using new technologies such as satellite navigation systems, geographic information systems and risk knowledge databases.

The aforementioned application proves that traceability systems are often used to enhance risk management in different domains. In this context, GOST traceability system was developed as part of an industrial partnership to enhance security and safety of containers transportation. This system is axed on real-time monitoring of containers carrying hazardous materials in a multi-modal transportation network, and taking into consideration the risk management issues (Boukachour et al., 2011). GOST functioning is axed on containers geo-location. Furthermore, it monitors transport conditions using a set of captors embedded on containers, and generates alerts when predefined rules are transgressed. For example, convoys of dangerous goods have the obligation to go through specific roads far from urban areas. Thus, the traceability system will verify if the convoys have passed through the predefined itinerary, or they broke this rule. Research on traceability field is more focused on technological issues and geo-location. However, only few works have tackled problems related to the interoperability of traceability systems with external systems, and how traceability information can be gathered in a collaborative way. The establishment of such solution will overcome the problems associated with

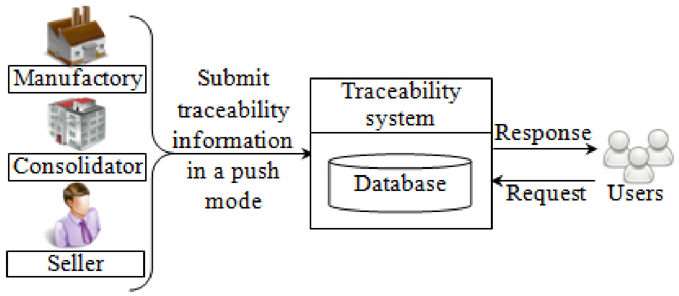


Figure 1. Current functioning mode of GOST.

the mass data of traceability and palliate the problems related to the confidentiality. Furthermore, process for developing a traceability system must lead to an agile and scalable solution, easy to align with the supply-chain's dynamic. This will facilitate the frequent adaptation of the traceability system to meet the new requirements of risk management, the changes of the involved stakeholders and their roles.

In this we will study the case of GOST traceability system. This system is based on centralized architecture and a push strategy for collecting traceability information. This information is archived in GOST database to respond to external requests. Figure 1 describes the current functioning mode of GOST traceability system.

The shortcomings related to this functioning mode manifest in a massive collection of information likely to be useless without an external request that requires the analysis of traceability information. Furthermore, the centralized architecture brings into focus the risk of information disclosure and the reliability of the entity that ensure this task.

Intelligent product

The development of technologies for products self-identification has emerged the concept of IP. The aim of this concept is to bridge the gap between products' physical and informational representation. According to McFarlane et al. (2003), an IP is characterized by an identifier, communication with the environment, information gathering, the ability to communicate its features and decision making. The IP's definition proposed by Kärkkäinen et al. (2003), is focused on the ability of a product to take decisions. Olli (2007) have focused on the ability of an IP to perceive its environment and how it responds to changes.

Meyer et al. (2009) have proposed an IPs classification according to three criteria; intelligence level, intelligence location, intelligence aggregation. The first criterion distinguishes the product that manages its information, the product that notifies problems occurrence and the product that analysis information to take a decision. The second criterion distinguishes the product with an embedded intelligence and the case of external intelligence through network. The third criterion distinguishes the case of simple IP considered as an atomic entity, and the case of composed IP that manages information related to its components. Wong et al. (2002) have proposed a simplified classification of IP axed on the intelligence level. Consequently, IPs are classified into products characterized by a communicative model or decisional model. The first model includes products that are able to gather information about themselves and the context where they are located. Moreover, these products are able to establish connectivity with external systems in order to communicate information. The second model encompasses products endowed with reasoning processes that exploits the gathered information to take decision concerning their destiny.

The implementation of IP concept is not limited to a single standard. The works we reviewed tackle the problems related to IP implementation from two points of view. The first one is axed on a theoretical approach and aims to propose a conceptual model of IP. The second one is axed on the experimentation of technologies for the realization of this concept.

From a theoretical point of view, Framling et al. (2003) have used the ID@URI concept to link the physical product to a software agent, which manages information related to its life-cycle. An IP model based on the combination of Holons and software agents was proposed by Valckenaers et al. (2009). The Holon concept was coined by Fischer et al. (2003) and covers both of the notions "whole" and "part". Accordingly, this concept consists of a fractal structure that is stable, coherent and that consists of several Holons as sub-structures and is itself a part of a greater whole. The software agent represents a program located in an environment and it is able to perceive and act on its environment. The agents display an amount of autonomy and are able to act in dynamic and uncertain environment (Fischer et al., 2003).

The IP model presented by Xiaoyu et al. (2009) is based on intelligence through network model. It is based on embedding data unit into product to be connected to services deployed by external servers. From an experimental point of view, Kiritsis (2011) has exploited this concept to enrich gathering information related to a product life-cycle, to ease the diagnosis and maintenance of gaming consoles. An application of IP was realized by Ngai et al. (2005) for tracking containers and forklifts inside warehouses.

IP concept presents a significant potential for ameliorating traceability systems functioning. It allows building an intelligent container able to perceive its environment, to manage its traceability information and to communicate with external systems. This goal can be achieved based on the technological solutions deployed by GOST system.

Model driven architecture

The frequent need to align the IT solutions used by an organization to its new business needs must be supported by a new approach to simplify this task. In addition, it is wise to consider the software maintenance phase in its development cycle (Liang and Des, 2009). To overcome this problem, the MDA approach has been proposed by the Object management group (OMG) to reduce the cost of development and adaptation of systems by automating the code generation from a system modeling.

The MDA approach is based on a software engineering idea which calls for the separation of the functional specification of a system and details related to the implementation platforms. This approach aims to separate the organizational and functional aspects of a system from the technological implementation decisions. MDA covers different phases of software development and promotes the automatic generation of a large part of the code. Moreover, the main objectives of this approach are reflected in the portability, the interoperability and the reusability, by the separation of architectural concerns (Estefan, 2008).

MDA is a modeling framework composed of multilevel models of abstraction. This allows capturing the business logic of the system and its functional and non-functional needs, along with a sufficient level of details needed for the automated code generation.

According to José-Luis and Pablo (2013), the key feature of this approach is its ability to define transformations for the passage from abstract models to more specific models. This is the mapping that establishes the link among the basic concepts of different models. Therefore, the modification of a model is reflected in the other models by a series of automated systemic changes which limit the system adaptation costs.

Business process (BP)

According to Muehlen and Indulska (2010), a BP is a collection of actions that address a set of input values to provide a result. Based on these definitions, we equate the BP to a structured set of activities performed by one or more actors and satisfying a set of constraints to achieve the goal set by its designer.

In a changing environment, BPs management and adaptation are key factors for organizations survival. Thus, the importance of processes improvement and understanding of their structures by unspecialized actors, have emerged the need for new modeling techniques. Vergidis et al. (2008) classified modeling techniques in three sets. The first technique is communicative and based on the use of diagrams modeling. The second technique is based on mathematical models to provide a precise specification of BPs. The third technique gives a linguistic description and simplifies the automation of the processes implementation.

There are several modeling standards for BPs such as Xml Process Definition Language (XPDL), Business Process Modeling Language (BPML), and Business Process Execution Language (BPEL). These are intended to simplify the implementation of BP by exploiting XML notation. However, these aforementioned standards do not have a graphical notation for BP modeling. Therefore, only a small number of specialized actors can operate the generated models. To overcome this problem, the OMG proposed the Business Process Modeling Notation (BPMN) which is a schematic standard simplifying the understanding of BPs. BPMN provides a visual and intuitive description of a BP (Vergidis et al., 2008). Chinosi and Trombetta, (2011) proposed a comprehensive study of this modeling notation, specification and relationship with other BP modeling tools.

BPMN allows BP modeling at several levels of details. This standard is suitable for modeling of internal and external organizational processes. For internal processes, it takes account of the internal activities of organizations and makes abstraction of interactions with external actors. Regarding the modeling of external BPs, it considers the collaborative aspect of process modeling activities that ensure the interaction between the organization and its environment. This aligns with the needs of activities modeling in a supply chain.

MATERIALS AND METHODS

Satisfying the inherent need for containers traceability in supply-chain's context is a tedious task. Accordingly, we adopt an approach structured into four steps to alleviate the complexity of this study. This approach is based on the use of these aforementioned concepts (MDA, Web-Service, and IP); in addition, it details the steps to follow from problem analysis to the solution implementation. Figure 2 illustrates the main steps of this approach.

The first step - objectives description - aims to present how we exploit the IP concept to propose an intelligent container able to manage the information related to its life cycle. In the second step, we propose a new distributed architecture for GOST based on the integration of the intelligent container and web-services to overcome the problems related to the centralization of information and traceability information sharing.

The second step - domain analysis - consists mainly of identifying the domain needs, specifying the characteristics of the system to implement and understanding its use context. This analysis simplifies the identification of actors involved in the process of containers transportation and details the role of each stakeholder. At first, we adopt a case oriented approach relying on the Use case diagram (UCD) of Unified modeling language (UML) to achieve a static view of system actors and functions. In a second step, we use the boundary, entity, and control (BEC) diagram (201)]. This diagram uses the functional requirements determined by the UCD to specify the classes to be implemented. To achieve a dynamic view of the system, we use the BPMN to model the execution sequence of the functions identified by the UCD.

The third step - web-service generation - deals with problems related to adaptation of the systems used by the supply-chain actors to enable traceability information sharing. For this purpose,

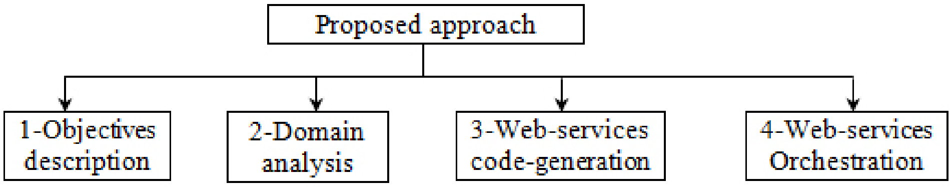


Figure 2. Principal steps of the proposed approach.

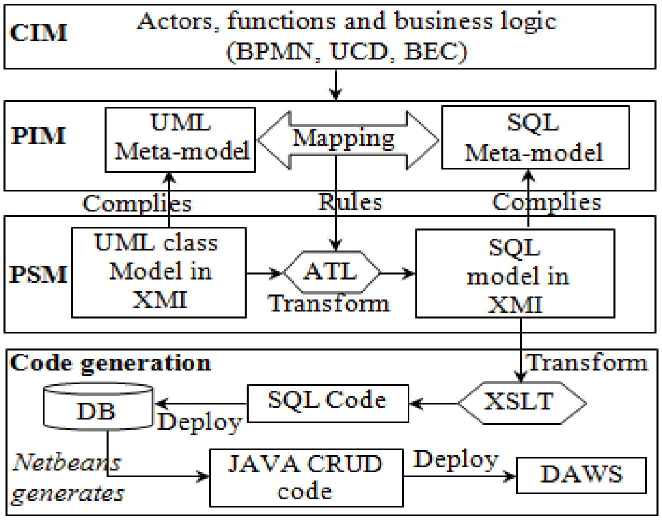


Figure 3. MDA process for generating web services.

we opted for web-services to collect information. This choice is supported by the ease of web-services invocation by the intelligent containers through internet. The aim of this step is to generate automatically web-services based on a MDA process. Figure 3 gives an overview of this process.

The MDA process starts by the specification of the Computation independent model (CIM) (Elfazziki et al., 2012). It deals with the conduct of a domain analysis to understand the mission of the studied system. The second stage of this process focuses on the specification of the Platform independent model (PIM) of the source model and the target model. It aims to define the UML class diagram meta-model and the Server query language (SQL) meta-model, to perform the mapping between their basic concepts as well as to ensure validity of the model to transform and the model to generate. The third step in this process is based on the specification of transformation rules from the mapping of the two meta-models using Object constraints language (OCL). The transformations are performed using the Atlas transformation language (ATL) (Elfazziki et al., 2012).

The ATL transforms the source Platform specific model (PSM) representing the class diagram expressed in XML Metadata Interchange (XMI), to a target PSM compliant with SQL expressed in XMI. The resulting PSM is transformed into SQL statements using the eXtensible stylesheet language transformations (XSLT) (Ming-zhe, 2013) to deploy a Database (DB). The last step of this process is based on the use of Netbeans development environment to generate the Create, Read, Update, and Delete (CRUD), JAVA code required for managing the DB. The JAVA classes generated for tables' consultation are deployed as Data access web-services (DAWS).

The last step of the proposed approach aims to orchestrate the generated DAWS according to the BP logic. This allows us to generate a new composite web-service that provides the information related to a container traceability based on the DAWS deployed by the supply-chain actors. To achieve this goal, we exploit the BPMN diagram expressed in the CIM. The BPMN is supported by the BPEL that provides a description of the interactions between the BP and the web-services. Thus, the implementation of the sequence of BP activities defines the order of granular web-services invocation to achieve the BP goal. Therefore, the execution of this process is equivalent to the invocation of a new composite web-service, based on the orchestration of granular web-services interfaced with the BP.

APPLICATION AND RESULTS

This section describes the implementation of each step of the proposed approach.

Objectives description

To improve the functioning of the GOST traceability system we propose a distributed architecture of this system based on web-services. This new architecture delegates traceability information gathering to the intelligent container. This latter is capable to collect and communicate information related to its life-cycle. This new solution allows container to get only its information at the appropriate time. Thus, it is based on a pull mode for collecting traceability information, and a decentralized architecture where each supply-chain actor ensures the storage of traceability information. Figure 4 gives an overview of the new functioning mode of GOST.

The proposed architecture imposes to supply-chain stockholders to deploy DAWS to share traceability information related to containers. These web-services are exploited by the intelligent container to generate a new composite web-service that provides the intervention description of each supply-chain actor involved in the container life-cycle.

The new functioning mode of GOST needs an IP model that overcomes the simple connection between physical product and its informational representation for a model able to interact with heterogeneous systems. For this aim, we propose an enriched model of IP to build an intelligent container able to exploit the deployed DAWSs.

The proposed model of the intelligent container is

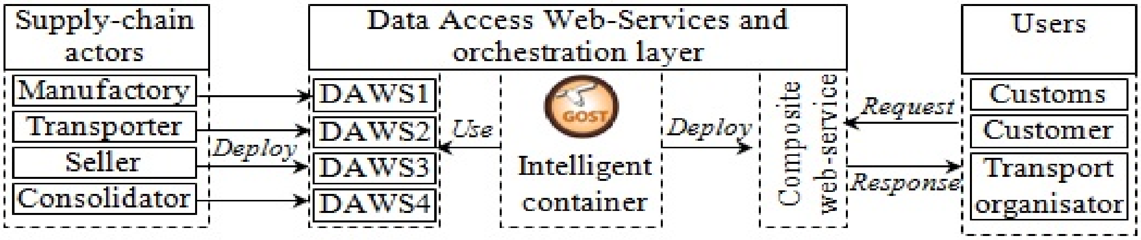


Figure 4. The proposed new functioning mode for GOST.

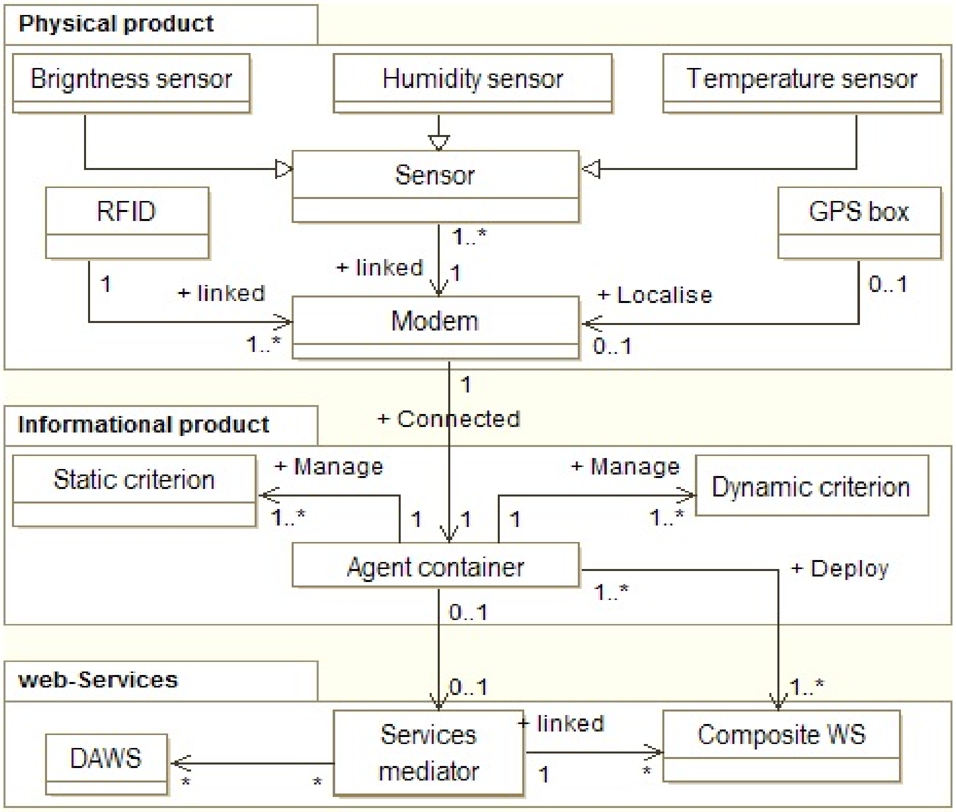


Figure 5. Enriched intelligent container model.

structured into three packages. The first package represents physical products and it consists of Radio frequency identification (RFID) tag for identification, GPS box for geo-location, a set of sensors to monitor container's internal state and a modem to communicate through network. The second package represents the informational product. It is composed of static information describing the products properties, dynamic information such as temperature evolution during the transportation and a container agent that manages this information. The third package represents the services layer. It is comprised of granular DAWS and a web-service mediator needed by container agent to deploy composite web-services. The composite web-service corresponds to the orchestration of the granular DAWS deployed by the supply chain actors that have intervened during container transportation. The container exploit this new web-service to gather information related to its life cycle. For example, Figure 5 shows the intelligent container model.

Domain analysis

The studied case treats the process of preparation and delivery of a Less-than-container-load (LCL). This container is based on the consolidation of parcels sent by different clients to the same destination. Each sender prepares his parcel and sends it to the consolidator,

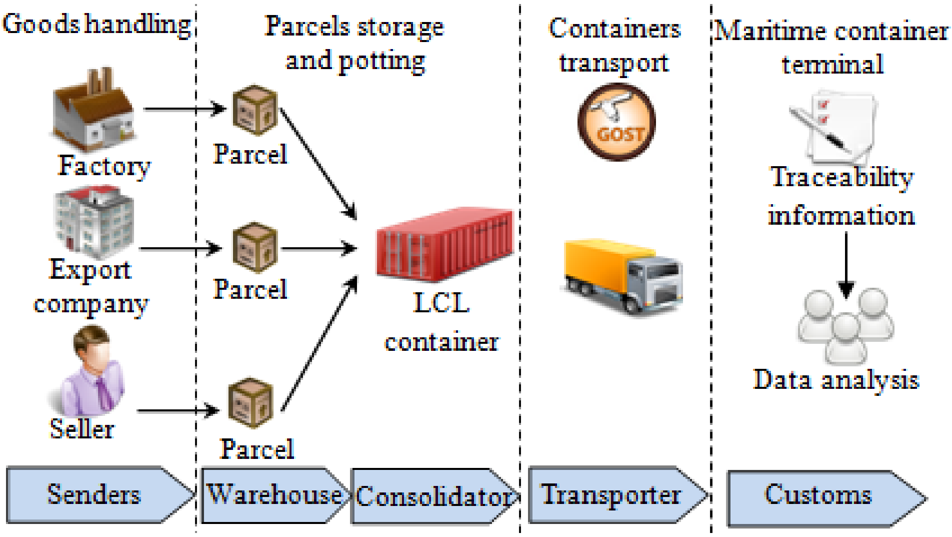


Figure 6. Case study: Less-than-container-load preparation and transport process

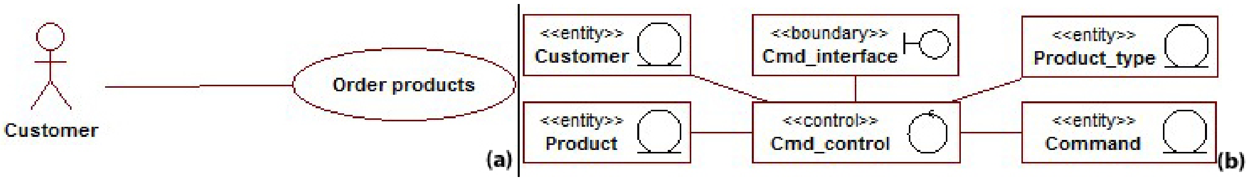


Figure 7. (a) order product use case diagram, (b) BEC diagram.

which provides storage of these parcels during the required time to receive enough parcels to fill the container. These parcels are packed in a container that will be transported to the maritime container terminal by a hauler. The consolidator provides information describing the nature and origin of goods. This information is used by customs to allow the container access to the terminal; Figure 6 illustrates this process.

The establishment of a traceability process must be preceded by an analysis to determine the information to be collected. To do this, the BEC diagram identifies persistent entities of the information systems used by the stakeholders. These entities carry information about the supply-chain actors' activities. We present a simple example of a seller making an order. BEC diagram (Figure 7) shows that the system used by the seller consists of interface and control classes, which represent the user interface and the entity that supervises operations performed by the seller. We focus on persistent entities of the system and we determine which entities are holding traceability information about the order preparation. Therefore, we must take account of persistent classes determined by the BEC diagram (Figure 7), which are command, customer, product and product type.

For a dynamic view of the preparation and delivery process of a LCL container to the seaport, we opted for a model based on BPMN. The modeling process is performed following the description level defined by Silver (2009). Thus, the following model provides a macroscopic view focusing only on the treatment of parcels sent by the seller, without detailed internal sub-processes of each stakeholder (Figure 8).

Web-services code generation

The process of generating web-services using the MDA approach starts with the specification of the CIM for capturing business requirements. This was detailed in the previous section - domain analysis - by specifying a static view of the main features of the system and a dynamic view by modeling the studied BPs.

The web-services generation is based firstly on the description of the traceability data shared by the supply chain actors, and secondly on the creation of new databases that can be accessed by DAWS. To do this, the actors give a detailed description of the traceability data. Then, these data will be refined to specify an UML class diagram. This class diagram is used for SQL code

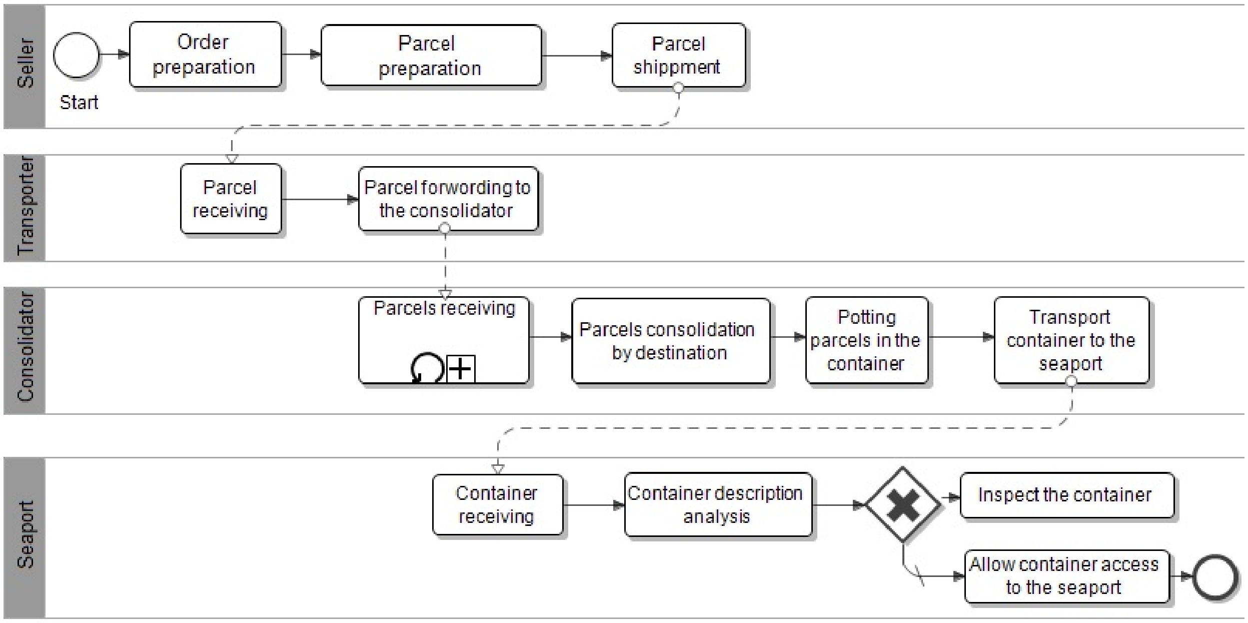

Seller
Start
Order preparation
Parcel preparation
Parcel shippment
Transporter
Parcel receiving
Parcel forwording to the consolidator
Consolidator
Parcels receiving
Parcels consolidation by destination
Potting parcels in the container
Transport container to the seaport
Seaport
Container receiving
Container description analysis
Inspect the container
Allow container access to the seaport

Figure 8. Case study: Less-than-container-load preparation and transport process.

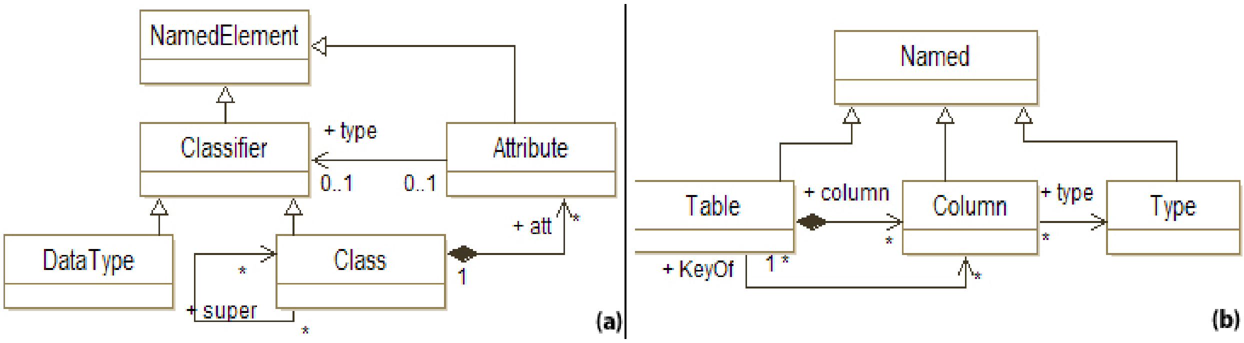


Figure 9. (a) UML class diagram meta-model (b) SQL meta-model.

generation, necessary to create the new database. This brings us to use in a first stage a class diagram meta-model to ensure compliance of the input model with the specifics of an UML class diagram, and in a second stage, a SQL meta-model that will guarantee compliance of the generated model to the specifics of the SQL implementation platform. Figure 9 shows a simple version of the meta-models used specifically for the case study.

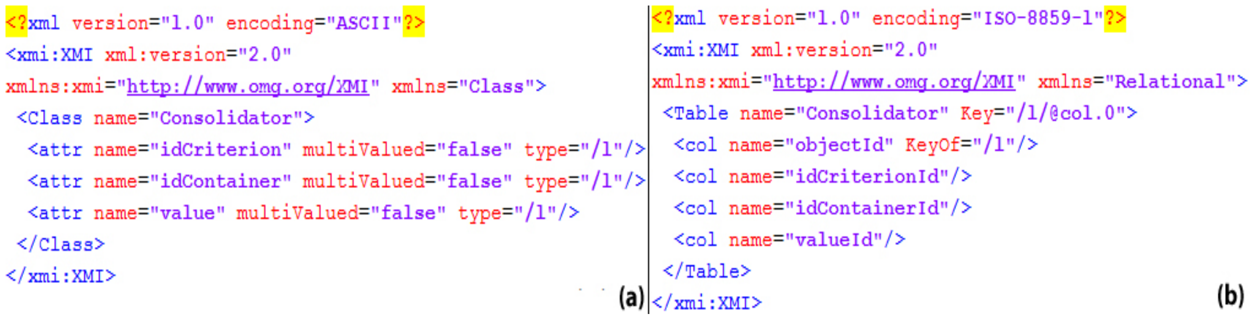
The transition from a class diagram to a SQL model is based on transformation rules between the two models. These transformation rules are derived from the mapping which determines the correspondence among the elements of these aforementioned meta-models. These correspondences are detailed as follows:

Named Element to Named: Establishes a link between the two models parent classes; we use parent classes in order to get main node during models specification using XML. Thus, all the classes composing a meta-model extend one main class.

DataType to Type: Binds attribute types of a class to columns types of a table, for example it binds the String to Varchar type; this guarantee that the parameters used by the deployed web-services correspond to au columns' type in databases.

Class to Table: Transforms the model classes into SQL table. Thus the generated SQL tables are used to deploy the database that will be interrogated by the web services.

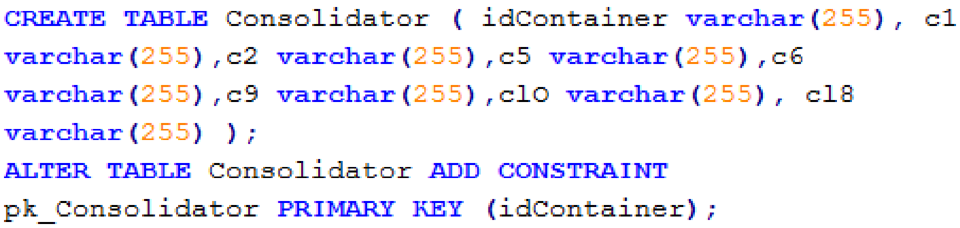
Attribute to column: Allows the transformation of the attributes of a class in columns in a SQL table; this is the



```
<?xml version="1.0" encoding="ASCII"?>
<xmi:XMI xml:version="2.0"
xmlns:xmi="http://www.omg.org/XMI" xmlns="Class">
 <Class name="Consolidator">
  <attr name="idCriterion" multiValued="false" type="/1"/>
  <attr name="idContainer" multiValued="false" type="/1"/>
  <attr name="value" multiValued="false" type="/1"/>
 </Class>
</xmi:XMI>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xmi:XMI xml:version="2.0"
xmlns:xmi="http://www.omg.org/XMI" xmlns="Relational">
 <Table name="Consolidator" Key="/1/@col.0">
  <col name="objectId" KeyOf="/1"/>
  <col name="idCriterionId"/>
  <col name="idContainerId"/>
  <col name="valueId"/>
 </Table>
</xmi:XMI>
```

Figure 10. (a) Input model class diagram (b) Output SQL model expressed with XMI.



```
CREATE TABLE Consolidator ( idContainer varchar(255), c1
varchar(255),c2 varchar(255),c5 varchar(255),c6
varchar(255),c9 varchar(255),c10 varchar(255), c18
varchar(255) );
ALTER TABLE Consolidator ADD CONSTRAINT
pk_Consolidator PRIMARY KEY (idContainer);
```

Figure 11. Generated SQL code: consolidator table.

basis for creating the tables for recording traceability data.

Association Column+: Transforms the connection between a class and its attributes to a table and its columns; this transformation rule brings together the generated columns relatively to the table that includes them.

Association KeyOf: Determines the primary key of the generated table. The key is used for identifying the databases records in unique manner.

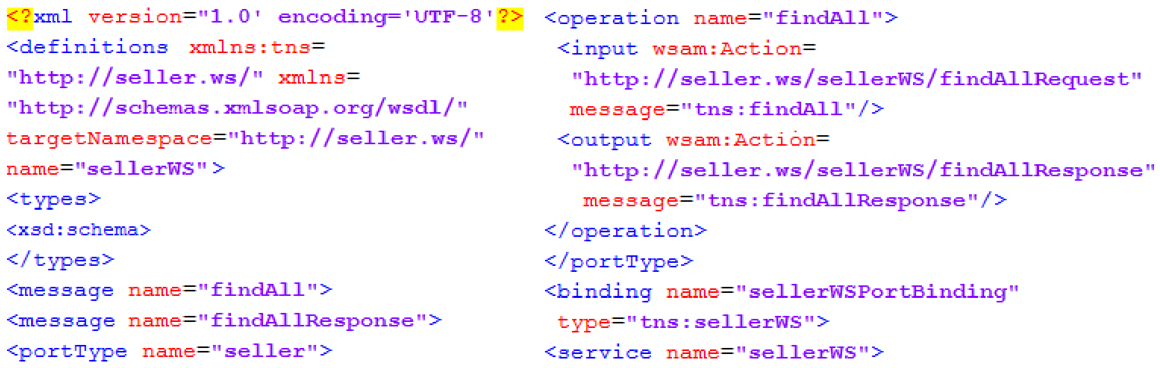
In the studied case, ATL provides the transition from a class diagram to a SQL database model. Therefore, the implementation of the defined specifications mapping covers the transformation rules in Object constraints language (OCL). For example, “DataType to Type” is the transformation rule that determines the relationship between an UML class attribute type and a SQL table column type. It is implemented as following:

```
rule DataType2Type {
from dt : Class !DataType
to out : Relational !Type( name<- dt.name )
  }
```

ATL requires the specification in ECORE of the two meta-models, SQL and class diagram. To do this, we used Eclipse modeling framework (EMF), which provides tools to generate ECORE meta-models from the diagrams shown in Figure 10. In addition, the ATL input model must be specified in XMI (XML Metadata Interchange). This is achieved using the EMF tools, which enables the generation of XMI code corresponding to a class diagram, along with ensuring the validity of the generated code with its meta-model. The Figure 10 shows the XMI input (a) and output (b) models.

The changes made by the ATL allow the passage from an input model to an output model -expressed in XMI- that meets the specifics of the implementation platform. The output model is not directly exploitable and requires a second transformation type "ModelToText" to generate SQL code for deploying the new DB. This step is completed by using the eXtensible Stylesheet Language Transformations (XSLT). It is a XML transformation language proposed by the W3C to convert XML documents into other languages. Thus, the transition from the generated XMI model to SQL is done by specifying a XSLT script that determines the transformation rules. These rules are executed using a XSLT processor. The following script shows the implementation code of the generated DB table that corresponds to the consolidator with the criteria that describe his interventions (Figure 11).

The second phase of the process of generating DAWS covers the use of the Netbeans IDE to generate the CRUD classes in Java, needed for the DB operation. We focus on classes of tables' consultation to meet the need of collecting traceability information. The Netbeans IDE provides tools for identifying entity classes from the deployed DBs. In this step, the generated classes are deployed as WS. The following script shows an example



```
<?xml version="1.0' encoding='UTF-8'?>
<definitions  xmlns:tns=
"http://seller.ws/" xmlns=
"http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://seller.ws/"
name="sellerWS">
<types>
<xsd:schema>
</types>
<message name="findAll">
<message name="findAllResponse">
<portType name="seller">
```

```
<operation name="findAll">
 <input wsam:Action=
  "http://seller.ws/sellerWS/findAllRequest"
  message="tns:findAll"/>
 <output wsam:Action=
  "http://seller.ws/sellerWS/findAllResponse"
   message="tns:findAllResponse"/>
</operation>
</portType>
<binding name="sellerWSPortBinding"
 type="tns:sellerWS">
<service name="sellerWS">
```

Figure 12. WSDL script describing the generated web-service.

of the WSDL file describing the deployed DAWS to retrieve information about a seller (Figure 12).

Web-services orchestration

In this step, we propose a solution to use DAWS for the deployment of a new composite web service. This latter is invoked by the intelligent container to meet the traceability requests. Thus, the solution is based on the use of BP logic to orchestrate the granular DAWSs.

To exploit the BPs modeling for orchestrating DAWS, we opted for the BPEL Designer proposed by Eclipse IDE. It allows modeling the invocation order of DAWS along with the implementation of activities that represent the BP shown in Figure 9. The BPEL Designer simplify the generation of BPEL code that will executes the BP. As a result, it enables automating the execution of the BPs by linking its activities to DAWS. This allows the generation of composite WS based on the orchestration of DAWS. The intelligent container will respond to external requests by operating the composite web-service. Figure 13 shows a model for the BPEL invocation of three DAWSs.

DISCUSSION

The objective of this work is to urbanise the GOST traceability system to meet the evolving data needs to improve the risk management during containers transport. For this purpose, we specified a new distributed architecture of GOST system based on web-services and the concept of intelligent container. Furthermore, a development process based on the MDA was proposed to automate the code-generation of web-services to facilitate the adaptation of systems used by supply-chain actors to share traceability information.

Our first contribution is the proposal of an enriched model of intelligent container through the specification of web-service layer. The aim of this contribution is to enhance the connectivity of the container with the external system and to delegate gathering the traceability data to each intelligent container. This solution allows us to avoid confidentiality problems related to the risk of data divulgation. Thus we differentiate from existing IP models axed on data gathering.

Our second contribution manifests in the proposal of a development based on MDA for web-service generation. The web-services were chosen because they allow to build more flexible architecture and to speed the integration of existing applications. Moreover, the proposed process minimizes the costs of adapting the heterogeneous systems to communicate with the intelligent container. The main advantage of this process is reflected in the separation of the system aspects in several models and automating the transition from one model to another using transformation rules as earlier specified. It was pointed out that this process is structured into many steps likely to be laborious; however it is based on open sources solutions and it is founded on well know standards. Thus, this approach contributes to minimize development times and can be adopted to accelerate software development cycle with other technologies.

As a perspective of this work, it is assumed that this solution can be improved by the use of Enterprise service web (ESB) (Ming-zhe, 2013). Thus, the connectivity among intelligent containers and externals systems can be ensured by the configuration of an ESB. Furthermore, we can avoid the generation of the DAWS by accessing the databases directly using the adapters proposed the ESB; however this solution required a special configuration for each database.

CONCLUSION AND PERSPECTIVES

This paper was aimed to enhance traceability system functioning to meet the needs of securing containers transportation. For this, a decentralized structure of GOST traceability system based on the exploitation of the

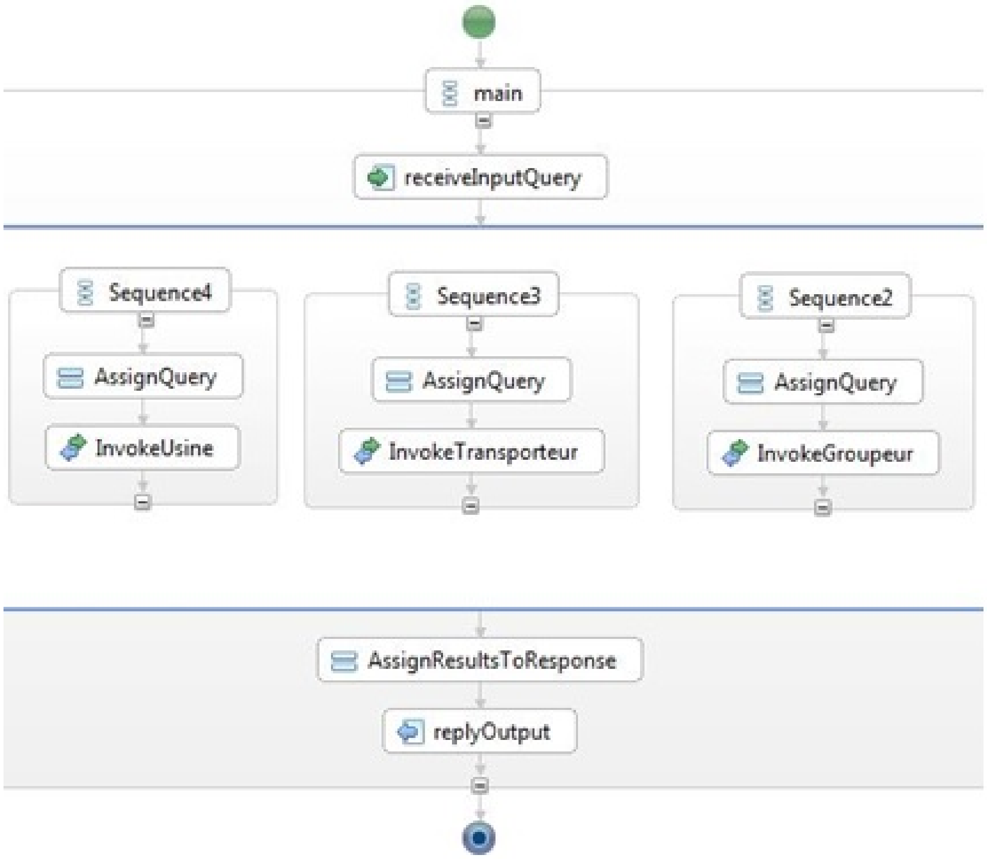


Figure 13. BPEL diagram for orchestrating the generated web services.

IP concept and web-services was proposed. The main contribution of this work is reflected in the proposal of a model-driven architecture approach and web-services to simplify the collection of traceability information. To this end, the ATL was used to automate the generation of web-services based on an UML class diagram. In addition, the BPMN diagram was used to orchestrate the generated web-services according to the business logic.

Work is currently underway for the proposal of a new solution to ease the collection of traceability information based on the configuration of an enterprise service bus. The main contribution of this alternative manifests in the availability of open source adapters for interfacing legacy systems with GOST traceability system.

Conflict of Interest

The authors have not declared any conflict of interest.

ACKNOWLEDGMENTS

This work is funded by the region of Haute-Normandie in France.

REFERENCES

Aung MM, Chang YS (2014). Traceability in a food supply chain: safety and quality perspectives. Food control 39:172-184. http://dx.doi.org/10.1016/j.foodcont.2013.11.007

BEC (2014). The Entity-Control-Boundary Pattern. http://www.cs.sjsu.edu/~pearce/ modules/lectures/ooa/analysis/ecb.htm.

Bechini A, Cimino MGCA, Marcelloni F, Tomasi A (2008). Patterns and technologies for enabling supply chain traceability through collaborative e-business. Information and Software Technology. 50:342-359. http://dx.doi.org/10.1016/j.infsof.2007.02.017

Boukachour J, Fredouët CH, Gningue MB (2011). Building an expert-system for maritime container security risk management. Int. J. appl. logistics. 2:35-56. http://dx.doi.org/10.4018/jal.2011010103

Chinosi M, Trombetta A (2011). BPMN: An introduction to the standard. Computer Standards & Interfaces. 34:124-134. http://dx.doi.org/10.1016/j.csi.2011.06.002

Elfazziki A, Lakhrissi H, Yetognon K, Sadgal M (2012). A service oriented information system: A model driven approach. Proceeding of the 8th international conference on signal image technology and internet based systems. 466-473.

Erkut E, Tjandra SA, Verter V (2007). Hazardous materials transportation. Handbook in OR & MS chapter 9. 14:539-62. http://dx.doi.org/10.1016/S0927-0507(06)14009-8

Estefan J (2008). Survey of Model-Based Systems Engineering methodologies, INCOSE-TD-2007-003-01.http://pdf.aminer.org/000/260/416/towards_a_unified_paradigm_for_verification_and_validation_of_systems.pdf

Fischer K, Schillo M, Siekmann J (2003). Holonic Multiagent Systems: A foundation for the organisation of multiagent systems. Proceedings of the First International Conference on Applications of Holonic and Multiagent Systems. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.1097

Framling K, Holmström J, Ala-risku T, Kärkkäinen M (2003). Product agents for handling information about physical objects. Espoo, ISBN: 951-22-6853-1. http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.80.6382

Jin S, Zhou L (2014). Consumer interest in information provided by food traceability systems in Japan. Food Quality 04.005

José-Luis HA, Pablo CB (2013). A model-driven approach to develop high performance web applications. J. Syst. Software. 86:3013-3023. http://dx.doi.org/10.1016/j.jss.2013.07.028

Kärkkäinen M, Holmstrom J, Framling K, Artto K (2003). Intelligent products - a step towards a more effective project delivery chain. Computers in industry. 50:141-151. http://dx.doi.org/10.1016/S0166-3615(02)00116-1

Karlsen KM, Dreyer B, Olsen P, Elvevoll EO (2012). Granularity and its role in implementation of seafood traceability. J. Food Eng. 112:78-85. http://dx.doi.org/10.1016/j.jfoodeng.2012.03.025

Kiritsis D (2011). Closed-loop PLM for intelligent products in the era of the internet of things. Computer-Aided Design. 43:479-501. http://dx.doi.org/10.1016/j.cad.2010.03.002

Liang X, Des G (2009). Adaptive Agent Model: Software adaptivity using an Agent-oriented Model-Driven Architecture. Information and Software Technology. 51:109-137. http://dx.doi.org/10.1016/j.infsof.2008.02.002

Mcfarlane D, Sanjay S, Jin LC, Wong CY (2003). Auto ID systems and intelligent manufacturing control. Engineering Applications of Artificial Intelligence. 16:365–376. http://dx.doi.org/10.1016/S0952-1976(03)00077-0

Meyer G, Framlin K, Holmstrom J (2009). Intelligent Products: A survey. Computers in Industry. 60:137-148. http://dx.doi.org/10.1016/j.compind.2008.12.005

Ming-zhe Y (2013). Design on enterprise service bus message conversion protocol based on XSLT. The journal of china universities of posts and telecommunications. 20:50-54. http://dx.doi.org/10.1016/S1005-8885(13)60249-6

Muehlen MZ, Indulska M (2010). Modeling languages for business processes and business rules: A representational analysis. Information Systems. 35:379-390. http://dx.doi.org/10.1016/j.is.2009.02.006

Ngai EWT, Cheng TCE, Auc S, Kee-hung L (2005). Mobile commerce integrated with RFID technology in a container depot. Decision Support systems. 43:62-76. http://dx.doi.org/10.1016/j.dss.2005.05.006

Olli V (2007). Intelligent products and systems technology theme final report. VTT Technical Research Centre of Finland, Edita Prima Oy, Helsinki, ISBN:978-951-38-7010-2.

Planas E, Pastor E, Presutto F, Tixier J (2008). Results of the MITRA project: Monitoring and intervention for the transportation of dangerous goods. J. Hazard. Mater. 152:516-526. http://dx.doi.org/10.1016/j.jhazmat.2007.07.032

Regattieri A, Gamberi M, Manzini R (2007). Traceability of food products: general framework and experimental evidence. J. Food Eng. 81:347-356. http://dx.doi.org/10.1016/j.jfoodeng.2006.10.032

Silver B (2009). BPMN method and style. Cody-Cassidy Press, Aptos, California. ISBN: 0982368100, 9780982368107. P. 236.

Valckenaers P, Saint-Germain B, Verstraete P, Van Belle J, Hadeli, Van Brussel H (2009). Intelligent products: Agere versus Essere. Comput. Ind. 60:217-228. http://dx.doi.org/10.1016/j.compind.2008.12.008

Vergidis K, Tiwari A, Majeed B (2008). Business Process Analysis and Optimization: Beyond Reengineering. IEEE transactions on systems, man and cybernetics part C: applications and reviews. 38:69-82. http://dx.doi.org/10.1109/TSMCC.2007.905812

Wong CY, McFarlane D, Zaharudin AA, Agrawal V (2002). The Intelligent product Driven supply chain, autoidcenter.org. http://dx.doi.org/10.1109/ICSMC.2002.1173319

Woo S H, Choi JY, kwak C, Kim CO (2009). An active product state tracking architecture in logistics sensor networks. Computers in Industry. 60:149-160. http://dx.doi.org/10.1016/j.compind.2008.12.001

Xiaoyu Y, Philip M, Seng KC (2009). Intelligent products: From lifecycle data acquisition to enabling product-related services. Comput. Ind. 60:184-194. http://dx.doi.org/10.1016/j.compind.2008.12.009

Yang YC (2011). Risk management of Taiwan's maritime supply chain security. Safety science. 49:382-393. http://dx.doi.org/10.1016/j.ssci.2010.09.019