Service-Oriented Computing for intelligent train maintenance

Boukaye Boubacar Traoré, Bernard Kamsu-Foguem, Fana Tangara, Xavier Desforges

Service-Oriented Computing for intelligent train maintenance

Boukaye Boubacar Traore[a,b], Bernard Kamsu Foguem[a], Fana Tangara[b] and Xavier Desforges[a]

[a]Laboratoire Génie de Production, Ecole Nationale d'Ingénieurs de Tarbes, Tarbes Cedex, France; [b]Faculté des Sciences et Techniques Colline de Badalabougou, Université des Sciences, des Techniques et des Technologies de Bamako (USTTB), Bamako, Mali

ABSTRACT

The purpose of this work is to apply the servicization of enterprise information systems in maintenance, particularly in the management of the maintenance process of the component parts of trains. Service Oriented Architecture (SOA) is an architectural approach that permits servicization since it provides a flexible set of design principles used during the modeling practices (abstraction and realization). With a view to supporting the model-driven engineering of software systems, Mode Driven Architecture (MDA) is a design approach delivering a set of guidelines for the configuring of specifications in systems development. Therefore, the combination of these two approaches can be fruitful to address the challenging issues the enterprise information system is facing today. Our study is based on a methodological approach using the MDA models for the automatic generation of web service. The case study concerns a Railways Maintenance Workshop (RMW) at Sidi Bel Abbes (Algeria). Finally, the information system for the management of maintenance of the component parts of passengers and baggage railcars, using the generated solution, is realized and deployed. This software helps to have better management of the RMW by the effective planning of interventions, improve performance by increasing reliability, traceability, and availability of the equipment (parts).

KEYWORDS

Servicization; enterprise information systems; Service Oriented Architecture (SOA); Model Driven Architecture (MDA); railways maintenance

Introduction

Maintenance services have a growing interest in Information and Communication Technologies (ICTs) assistant in order to help them in performing their function (Amadi-Echendu & Wit, 2015). There are some Computerized Maintenance Management Systems (CMMS) offered in the software market. CMMS is a software package used to track, schedule, organize and facilitate maintenance activities (Throop 2000). Maintenance service seeks to maintain or restore a property concerning a system or its components (e.g. equipment or parts) in a specified state so that it is able to provide a specific service. Maintenance is defined by (Dhillon 2002) as 'All actions appropriate for retaining an item/part/equipment in, or restoring it to, a given condition.' The information management systems must guarantee effective management of equipment in order to manage maintenance workshops and staff, manage stocks of equipment and spare parts, locate useful equipment at any one time, retrace the complete history of any equipment that has undergone maintenance, etc. Maintenance software has a considerable impact on a better management of the maintenance activities by an effective planning of

CONTACT Bernard Kamsu Foguem Bernard.Kamsu-Foguem@enit.fr; Boukaye Boubacar Traore boukaye@gmail.com Laboratoire Génie de Production, Ecole Nationale d'Ingénieurs de Tarbes 47, Avenue Azereix, BP 1629, Tarbes Cedex F-65016, France

interventions, ensuring equipment inventory at any time, improving performance by increasing reliability, traceability, and availability of the equipment (or parts) (Manzini et al. 2015).

Software parts can be implemented directly into the machines to get their current health status and to reduce, thanks to monitoring and diagnostics functions sending their data to CMMS, the lengths of downtime as it is proposed in (Desforges, Habbadi, and Archimède 2011). A more proactive approach is now developed by the Prognostics and Health Management (PHM) which mainly consists of predicting future failures of systems and in managing their maintenance according to their current and future health states (Xia et al. 2018), but also to provide decision supports to plan jointly productive and maintenance tasks aiming at better compromising between production and maintenance objectives (Desforges, Dievart, and Archimede 2017). Such considerations are also major concerns for Product-Service Systems (PSS) whose business core is to provide machine or system capability rather than product ownership. For such business, the main need is services that guarantee the availability of assets and that contribute to reducing the unscheduled downtime as much as possible and, therefore, to avoid contractual penalties (Silva Teixeira, Tjahjono, and Absi Alfaro 2012). Therefore, industrial maintenance becomes today as important as production activities in capacity intensive industries with considerable investments, such as railways, wafer fabrications, and airlines (Erkoc and Ertogral 2016). The success of any good computer system depends on a thorough and detailed study of the conceptual modeling of the application domain. One of the ways privileged nowadays is the search for servicization that is the process to distribute software functionalities within an organizational environment across a set of services. For this purpose, this work describes the implementation and deployment of an information system for the Railways Maintenance Workshop (RMW) at Sidi Bel Abbes (Algeria) using the servicization concept.

In practice, servicization is the modernization of legacy enterprise information systems based on web services that aim to guarantee product functionality over time, such as corrective, preventive and, more recently, predictive maintenance (Saccani, Visintin, and Rapaccini 2014). In recent years, enterprise information system architectures are increasingly complex and multidimensional causing some expensive processes in terms of development, maintenance and security (Kamsu-Foguem and Mathieu 2014). These aspects present a significant challenge for companies and will make activities more difficult to the information technology services from a business perspective. Given the exponential intensification in the volume of data and incidence of information exchanged by computer systems, there is a strong need for flexibility and interoperability of these systems (Neiva et al. 2016). The servicization approach allows dealing with the shortcomings listed above. Servicization is the process to partition software functionalities within an enterprise context into a set of services, so servicization will consider multiple dimensions of viewpoints, such as manageability and utility, to package a composition of suitable functionalities as a service (Lee, Shang-Pin, and Liu 2012).

Service Oriented Architecture (SOA) is one of the most practical solutions to these problems because it consists of a supple set of design principles employed in the stages of system development, migration and integration in information systems (Sweeney 2010). A software system based on SOA architecture (Figure 1) will offer a loosely coupled set of services that can be employed within numerous disjointed systems from various business domains (Bell 2008). SOA defines how to integrate largely heterogeneous applications (e.g. software from SAP (Systems, Applications, and Products for data processing), Oracle, Siebel, PeopleSoft, and International Business Machines Corporation (IBM)) for an Internet environment and employs numerous implementation platforms (Belete, Voinov, and Laniak 2017). An explicit implementation of SOA is the one founded on web services: modular applications, which are auto-descriptive Web Service Description Language (WSDL) that can be involved and transmitted through the Web following some recognized standards (characteristically Simple Object Access Protocol (SOAP) over Hypertext Transfer Protocol (HTTP)) (Barry. 2012). Web services and Model-driven software development have emerged as well as a set of technologies promising for the

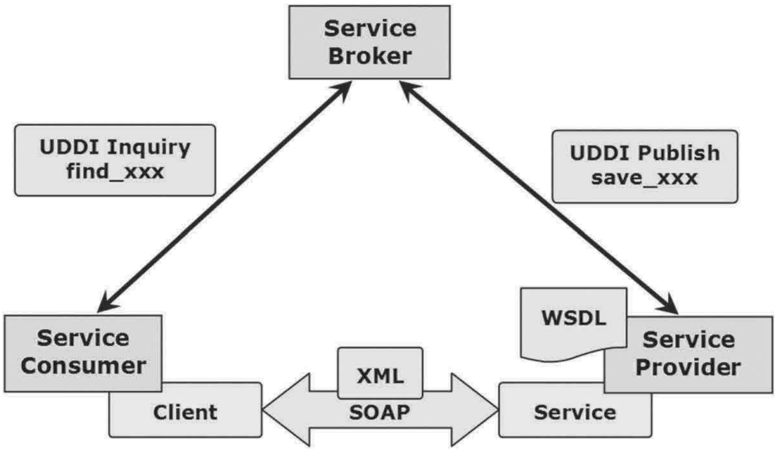

Figure 1. SOA (Service Oriented Architecture).

development, deployment and integration of innovative applications on the Internet and mobile platforms (Rieger and Kuchen 2018).

The development of Web Services is an interesting issue for servicization that suggests a shift from delivering software products to delivering integrated software products and services with chances and limitations in the generated business models (Hockmayr 2013). A relevant area of research for application development is to separate the independent and dependent aspects of the software platforms by a distinct description of their respective patterns of modeling. This trend is highlighted by the Object Management Group (OMG) with the Model Driven Architecture (MDA) approach that provides a guide for deriving value from models and architecture in support of the entire life cycle (from requirements to business modeling to technology implementations) of complex systems (OMG MDA, 2014). The establishment of a software design approach through MDA principles is increasingly becoming inseparable from the SOA at least for the automatic generation of Extensible Markup Language (XML) pivot formats: data schema, service interface with WSDL and orchestration with Business Process Execution Language (BPEL) (Gorton 2011). So the combination of these two technologies can be fruitful to address the challenging issues enterprise information systems are facing today. Our study is based on a methodological approach using MDA models for the automatic generation of web service, specifically the WSDL file for the servicization of enterprise information systems.

The remainder of this paper is organized into four sections. Section 2 provides the theoretical background about servicization of enterprise information systems with the MDA approach. Section 3 describes the approaches involved in the detailed steps of this methodology to pursue research objectives. In section 4, a real case study from a Railways Maintenance Workshop is presented with a discussion of the achieved and expected results. Section 5 highlights achieved an original contribution to the practice of the research work presented in this paper.

Theoretical background

Servicization of enterprise information systems: in many organizations, legacy information systems constitute a hard core of defining fundamental functions in business services; so for the modernization of these systems towards SOA, there is a need for suitable better methods to identify business value in large code bases. In this context, we are concerned with web service technologies and those of the MDA approach. The servicization can be performed in four methods namely (Almonaies, Cordy, and Dean 2010) (Table 1):

Table 1. Summary of servicization approaches.

Approaches	Advantages	Disadvantages
Replacement	Reduce maintenance Improve business functions	Time consuming Expensive Experienced resources needed
Wrapping	Fast	Inflexible Difficult maintenance
Redevelopment	Increase agility Flexibility Reduced cost	Source code needed Original requirements needed
Migration	Stable environment Tools availability	Time consuming Experienced resources needed Source code needed

- Replacement of Legacy Systems by the new service oriented system: is to rewrite the application of existing systems or replace them entirely with a new software product,
- Wrapping of the existing system functionality to provide the web service functions: provide a new interface to existing components and make them easily accessible as services to other software components,
- Reengineering or development of the existing system: using approaches of reverse engineering and re-engineering to add SOA functionality to systems,
- Migration: that moves the current system to the more flexible environment for SOA or cloud while preserving the original data and system functionality.

A general method for servicization can be described in four steps (Patel and Ragha 2013) (Figure 2): Architecture Recovery (identification of the legacy code architecture), Analysis (determination of key services), Mapping (correspondence between the new information system architecture and the legacy system architecture) and Transformation (migration of essential modules with potential addition of new functionalities or interfaces).

In fact, to preserve the original data and system functionality, we opted for the migration approach to modernize the legacy information system into a SOA stable environment, using MDA approach. The increasing interest in the MDA approach in the area of web services and other types of services is justified by the fact that the levels of models transformations proposed by the MDA approach are similar to any reasonable approach for implementing a SOA. Indeed, at the business level, the task to be undertaken is to determine a modeling to cover the specificities of the business conducted by the enterprise (including business process models). The logical level is to specify the functionality (services) implemented to meet the strategy and business needs. Finally, the technical level is the result of a projection of the logic level on a particular technological solution (Enterprise Application Integration (EAI), Enterprise Service Bus (ESB), Web service, etc.), in order to make the proposed architecture executable.

The methodology adopted focuses on the distinction between the specification of correspondences and the definition of transformations; this is considered particularly important for the implementation of the MDA approach to a SOA platform. The specification of correspondences is to specify the mappings between the elements of two meta-models and the definition of

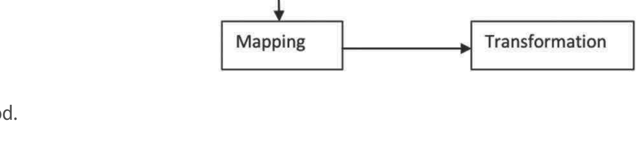

Figure 2. General method.

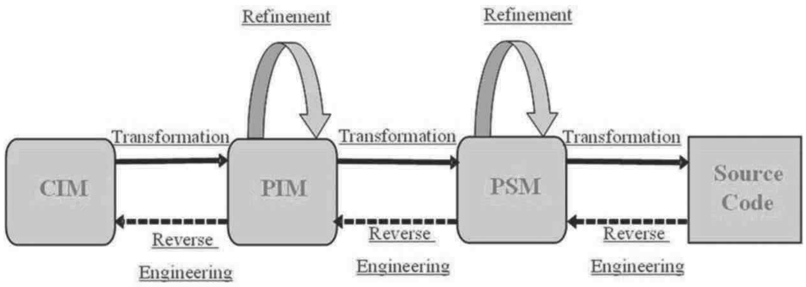

Figure 3. The MDA model refinement and transformation process.

transformations to describe in detail and perform the steps of transforming a model into another one in accordance with the specification of correspondences. MDA automates the production of a 'Platform Specific Model' (PSM) model (that is closer to the technology) from a 'Platform Independent Model' (PIM), which is nearer to business concepts and requirements The basic principle of the MDA is the development of different models, starting from an Independent Computer Model (CIM), transforming it into a Platform Independent Model (PIM) and finally the transformation of the latter into a specific Platform-Specific Model (PSM) for the concrete implementation of the system (Figure 3). The second level of translation from PIM to PSM is automate process by applying mapping technique (for instance: ATL – Atlas Transformation Language). However, this transformation can be enriched by refinement which consists in adding other technical information relating to the target execution platform and in specifying their content (OMG MDA, 2014). The PSM are thus refined and the information they contain is specified by successive enrichments. A PSM provides useful information for generating application code and is dependent on the execution platform. It is possible to create as many PSM as there are target platforms.

A transformation specification (which may also be a model) specifies how a PIM is transformed into a PSM based on parameters provided by developers (OMG MDA, 2014). The MDA Metamodel Description is presented in Figure 3 (adapted from (OMG MDA, 2001)) with PIM, PSM and mapping techniques that are based on metamodel articulated preferably with OMG core technologies like Meta-Object Facility (MOF), Common Warehouse Metamodel (CWM) or Unified Modeling Language (UML). A metamodel defines the structure and not the semantics of models conforming to this metamodel. A metamodel is a class diagram that defines the model's features, as well as the properties of their connections and their consistency rules. To be valid, each model must conform to its metamodel. This relation of conformity is essential in the MDA approach, it is thus possible to build tools able to handle the models (Figure 4).

Metameta-models is, so to speak, the metamodel of metamodels. And thus maintains with them the same relationship as a metamodel with his models. Metamodels are models that conform to their Metameta-models. In MDA, regardless of the meta level, all elements are considered as models. Therefore, metamodels and Metameta-models are also models. It could have continued to rise in the meta levels, but the fact is that the Metameta-model, used by MDA, defines itself. In other words, it is its own metamodel which brings to 04 levels of hierarchy. There is indeed only one Metameta-model used by MDA which is the standard MOF (Meta Object Facility) whose one of its features is to enable its self-definition.

State of the art

Distributed services growth led to developments of methodologies for the selections of the services against the specified requirements. Hence, the logic-based approach using formal

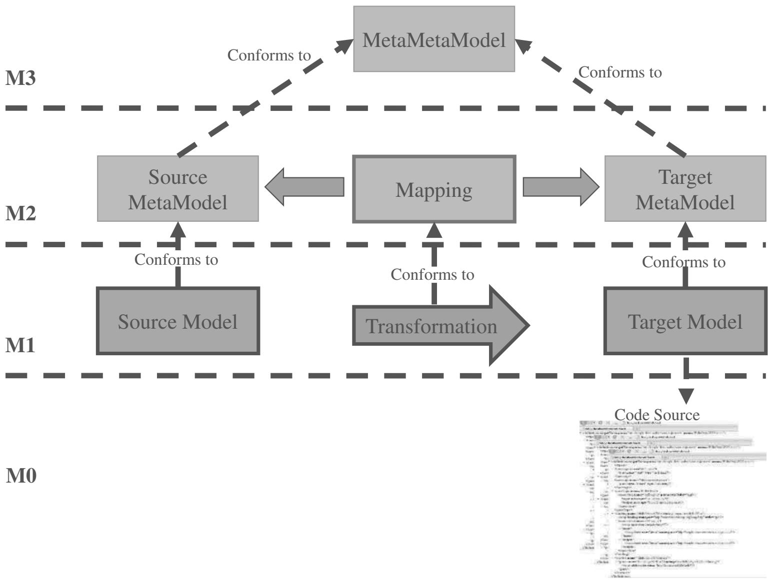

Figure 4. The MDA metamodel description.

verification and inference rules in the service workflow specification and composition is useful to automate compliance checking effectively (Viriyasitavat et al. 2018). Due to the increasing complexity of users' needs, other formal models provide mathematical specifications such as the Formal Concept Analysis (FCA) used for the reuse of service process fragments and improvement (e.g. time execution and reliability) of the composition process in order to deliver complete business processes (Mezni and Kbekbi 2018). According to the dynamic evolution of web services, the principles for establishing and modelling combinable relationship-based composition service network (CoRCS-Net) have been investigated and the concepts of combinable strength and variation of combinable strength are used to determine that CoRCS-Net is composed by few 'active services' and many 'silent services' (Tao et al. 2012). Moreover, with the speedy development of web-based technologies, the role-driven dynamic mechanism have been specified to support dynamic process representation in SOA and their usage and practicability are demonstrated in the setting of a travel reservation system application scenario (Tao et al., 2008). Specifically, within the context of the internet-of-things, aspect-oriented programming addresses transversal and cross-sectoral issues in application design, systems interaction, and integration (Cerny 2018).

The Mediation Information System Engineering have provided emerging collaborative situations with methods and tools to define and design a service-based platform, aimed at initiating and supporting the interoperability of collaborative networks (Benaben et al. 2015). The services in a service-oriented software as a service (SaaS) application can be implemented using different multi-tenancy patterns that are relevant to middleware and hardware components for scalable, multi-tenant aware and configurable information systems (Mietzner, Leymann, and Unger 2011). In service-oriented environments, a learning approach can be used to separate fair from unfair agents and estimate the quality of services for a better alignment between recommended service features and user preferences (Khoshkbarchi and Shahriari 2017). In the same vein, the extended service-oriented architecture can embed a reputation bootstrapping approach using an artificial neural network (ANN) to learn associations between features and performance of existing services in order to predict a reputation when assessing new and unknown services (Wu et al., 2015). The practical framework for the functioning of a complex system in a distributed environment requires networked service-oriented information that can be combined with progressive service matching approaches for supporting monitoring service management (Zhang et al. 2017).

Keizer and his colleagues presented a review of literature about Condition-Based Maintenance (CBM) policies for systems with multiple dependent components namely by the complexity of real-life systems. They provide an extended classification with multiple components subject to various dependencies (economic, structural, stochastic and resource) and investigate the implications of these dependencies ((e.g. structural dependence on a technical level or negative economic dependence) on the structure of the optimal CBM policy (Keizer et al., 2017). With the emergence of ICTs, Shin and Jun (Shin and Jun 2015) address the CBM implementation approach that makes a diagnosis of the asset status based on wire or wireless monitored data, predicts the assets abnormality, and executes suitable maintenance actions such as repair and replacement before serious problems happen. They reviewed the CBM approach from several viewpoints (definitions, advantages and disadvantages, related international standards, techniques, and procedures) with the introduction of related case studies (e.g. oil analysis, crack propagation analysis, field operation data analysis, and vibration analysis). Durazo-Cardenas and his colleagues propose a high-level architecture for a complex data-rich maintenance system based on data-fusion systems engineering principles. This architecture integrates railway technical and business drivers for optimized interventions and a proof of concept is validated by British rail network stakeholders (Durazo-Cardenas et al. 2018). Palau and his colleagues present a theoretical approach to collaborative prognostics for solving maintenance problems within the Social Internet of Things. They provide a conceptual proof showing that the potential implementation of such a system in a real industrial fleet may contribute to decrease maintenance cost compared to self-learning (Palau et al. 2018).

Li and Guldenmund present a broad overview of safety management systems (SMSs) from different angles and this includes a discussion on maintenance activities within the integrated management systems, and how maintenance satisfies system certification. Particularly, it is suggested that railway information system can improve near-miss data and combine safety data with Geographical Information System (GIS) data (Li and Guldenmund, 2018). Erkoyuncu and his colleagues place an emphasis on learning from feedback, testing and validating existing approaches for estimating corrective maintenance costs and availability at the Equipment Type level. They point out that although the adopted research methodology has been implemented within the military context, it can be adjusted across other industries such as commercial airlines or railways (Erkoyuncu et al. 2017). Sheng and Prescott focus on some types of maintenance actions called cannibalisation in which an unserviceable part in an inoperative platform is replaced by a serviceable part of the same type from another platform. They propose a hierarchical coloured Petri net model of a fleet operation and maintenance process that considers multiple maintenance levels and cannibalisation policies, maintenance scheduling and spare inventory management (Sheng and Prescott 2017). Carnero and Gómez analyse the choice of the most suitable maintenance strategy to be applied in electric power distribution systems responsible for supplying electrical energy to critical areas of the Health Care Organizations. The proposed multicriteria model integrates the Measuring Attractiveness by a Categorical Based Evaluation Technique approach with Markov chains to determine the predicted mean availability for different electric power distribution systems (Carnero & Gómez, 2017).

Madhikermi and his colleagues stress that the quality of enterprise maintenance services are closely linked to the quality of maintenance data reporting procedures. They propose the development of a maintenance reporting quality assessment (MRQA) dashboard enabling any company stakeholder assess/rank company subsidiaries in terms of maintenance reporting quality (Madhikermi et al. 2016). Roy and his colleagues present advanced computing and visualisation technologies (like 3D scan and computer tomography data analysis systems) that will improve efficiency of the maintenance and reduce through-life cost of the product. With these technological developments, there is an increasing progress of condition monitoring and prognostics for continuous maintenance within the context of Internet of Things (IoT) and industry 4.0 for a better understand the health of a product and plan maintenance (Roy et al. 2016). A diagnosis and prognosis approach has been proposed in the context of awareness for maintenance decision making by Galar and his colleagues. They propose a hybrid modelling approach integrating symbolic models (knowledge models from maintenance expertise) and numerical models (physical models of degradation and data driven models) to consider the consistency and the reliability of the measurement data obtained and evaluate the prognostic/diagnostic strategy of the system under examination (Galar et al. 2015). A similar approach is detailed in (Desforges, Dievart, and Archimede 2017) where the system model and computations are based on object oriented Bayesian networks and on components prognoses. It enables to assess the future availability of entities of interest of the systems (subsystems, functions…) and to point out components whose maintenance should be first undergone.

A CMMS is essential for planning, scheduling, and controlling the maintenance activities. Through effective reporting a CMMS can provide maintenance managers and engineers with the information needed for sound decision making to control and improve the maintenance process. However, its main disadvantage is that they do not perfectly match the particularities of each enterprise. Thus, many enterprises prefer to develop their own maintenance system including hardware, tracking and user-Augmented Reality interaction in industrial maintenance (Palmarini et al. 2018). In this context, we presented a computer service-oriented modeling methodology using MDA approach to modernize legacy systems or completely create new systems. For the validation of this methodology, we presented a case study about it was apply on train maintenance.

Methodology adopted

The adopted methodology (Lopes et al. 2005) in the context of the servicization is the migration approach of a computer application to a new stable environment; however, this is in line with the model driven engineering approach from earlier research. It is developed for mapping specification and generations of transformation definition in the MDA context and used in web services. The mapping specification is to establish the correspondences between the elements of two meta-models and the generation of transformation definition is to provide detailed descriptions and practical procedures for the steps of transforming a model into another one according to the mapping specification. This methodology is in line with the vision of the OMG to develop information systems, with valuable architectures that consistently separate specifications from their realizations. In fact, it allows the structured modeling and flexible implementation of software systems in various environments with traceability between the PIMs and the PSMs for interoperability and portability.

The main objective of this methodology is to automatically generate web services (WSDL file) that represent the functionalities of an information system that must be accessible to users. These web services may be concerned with an existing information system (servicization of enterprise information systems) or not. For our case study, this is a non-existent information system that has been implemented from the design to the implementation of the computer system using the web services generated in accordance with SOA principal.

The principle of a service-oriented architecture is to structure the enterprise information system as a set of services that expose their functional interface and communicate via messages. In an SOA, we basically have three main roles: the service provider, the service requestor, and the service registry. The service provider publishes the description of his service in service registers specially designed for the purpose of being interrogated by clients. Service clients (client applications) locate (find) their service requirements by performing service registry searches. Once the service is located, the client retrieves the description of the registry and based on the information provided in the service description, the client then interacts (bind) with the service in order to execute it and obtain the results (Figure 1). In this case, the service provider is known in advance and it is the management company of the organs park of Sidi Bel Abbes.

1.1. *The steps of the methodology*

The adopted methodology (see Figure 5) consists of eight steps that are described as follows:

Step 1 can be achieved by choosing UML or EDOC (Enterprise Distributed Object Computing) (OMG ECA, 2004) as PIM. Other approaches may opt for the creation of specific meta-models for a specific business, such as DSL (Domain Specific Language).

Step 2 is the choice of a pre-existing metamodel in the literature or the creation of a new specific metamodel to target platforms such as Web Services, Java Enterprise Edition (J2EE) (JSR-000316, 2009) and Microsoft.NET (Jorgensen 2002).

Step 3 is dedicated to the correspondence. It is to specify which PIM elements (e.g. UML) are mapped onto the PSM items.

Step 4 uses the correspondence specified in step 3 to generate the definition of the transformation in a transformation language. In fact, the correspondence specification can be treated as an abstract language. Abstract languages can specify the relationships between metamodels, but they are not executable. For example, abstract languages can be based on Object Constraint Language (OCL) (OMG OCL, 2014). Concrete languages like ATLAS Transformation Language (ATL) (Jouault et al., 2008) and Yet Another Transformation Language (YATL) (Patrascoiu 2004), are used to define the execution of transformations.

Step 5 is carried out by applying transformations of definitions for transforming an input pattern (PIM) in an output model (PSM).

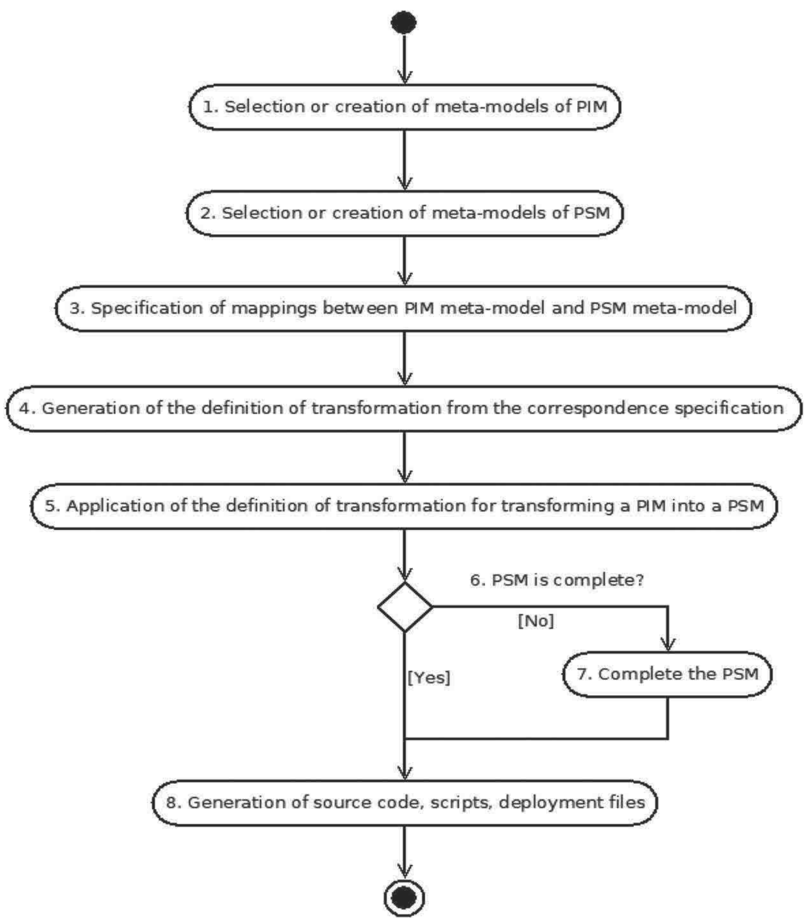

Figure 5. Methodology adopted: activity diagram for the application of the MDA approach.

Step 6 checks whether the model generated in the previous step is complete.

Step 7 requires the intervention of the developer to complete the PSM for instance the implementation of Java methods.

Step 8 is to generate source code, scripts and deployment files from the final PSM created in step 5 or 7. In this case, a type of model transformation for code generation is required.

This approach allows a better view of models specification and a good understanding during models transformation according to OMG. It emphasizes the need to make a distinction between matters relating to the technology itself and matters having to do with the use of the technology. In general, in MDA process, different transformations require refinements to enrich, filter or specialize the model as shown in Figure 3. This methodology is general and can be applied to generate any desired platform (eg .Net, Java, Corba . . .). The model of the desired platform is the PSM which is the third level of abstraction model of a computer application. A PSM depends on the execution platform and provides useful information for generating the code of the desired application. It is possible to create as many PSM as there are target platforms. In our study, this is the automatic generation of web services (WSDL files) by the MDA approach. The web services to be generated depend on the information system to be produced. In the MDA approach, computer applications are the entities to be modeled. The models represent the information, partial or total, necessary for the construction of these applications. In MDA, a computer application is represented by one or more models with different levels of abstraction.

Case study*: railways maintenance workshop management using servicization approach*

The case study concerns the RMW (Railways Maintenance Workshop) of Sidi Bel Abbes (Algeria), precisely the management of the maintenance of the component parts of passengers and baggage railcars. RMW has great incomes from maintenance and spare parts management. The implemented system also allows us to trace and store the physical maintenance flow (repaired works realized on the parts) with information flows (recorded properties associated with the parts). For the servicization of the legacy system to a new stable environment based on the proposed methodology using SOA, we present the UML metamodel and WSDL metamodel, the specification of correspondence between UML and WSDL, the definition of transformations from WSDL models to WSDL code, the conception of the management of case study and the servicization applied to the management of the maintenance of the components of passengers and baggage railcars.

1.2. *Presentation of the UML metamodel and WSDL metamodel*

(A) **Metamodel of UML**: among the elements of the metamodel, we have (Figure 6):

- **Package**: used to group multiple classes,
- **Class**: describes a set of objects with the same attributes, the same operations, and the same relations,
- **Interface**: describes the visible behavior of a class,
- **Attribute**: represents a characteristic or behavior,
- **Operation**: is a service that belongs to an object. It is described by a signature and can have parameters and return a result,
- **Parameter**: To specify the parameters entered and left
- **DataType**: Data types used by other elements such as attributes, parameter

(B) **Metamodel of WSDL**: composed the following elements (Figure 7)

- **Definition**: main element of this metamodel containing a set of Import, Type, Message, PortType, Binding, and Service,
- **Import**: allows the combination of a namespace (namespace) the location of an XML document,

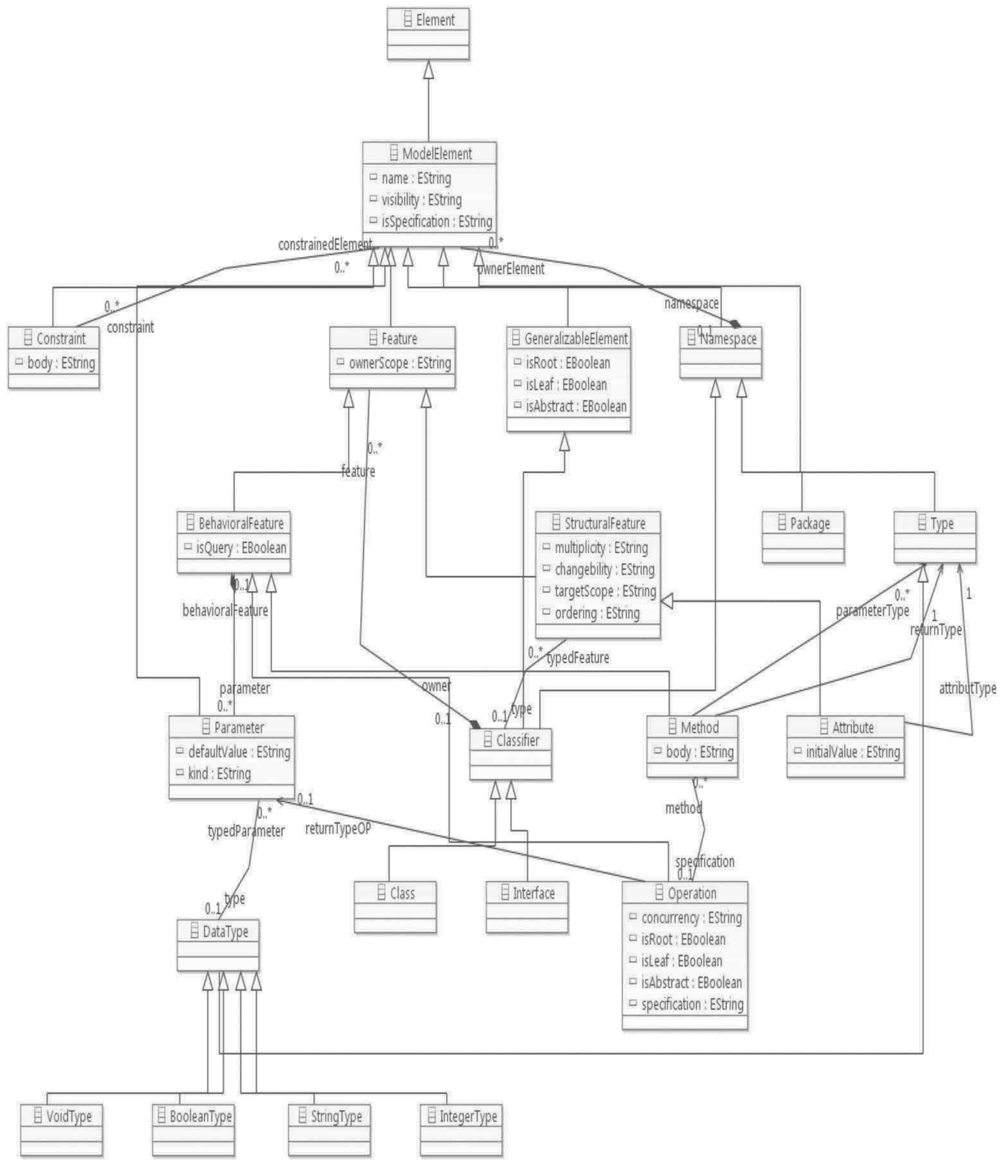

Figure 6. The UML metamodel.

- **Type**: used to define a type of simple or complex abstract data in accordance with an XML schema,
- **Message**: describes an abstract format of a particular message that the Web service sends or receives. It contains elements (e.g. parts) describing each section of a message,
- **portType**: defines the interface of services. It contains a set of operations that a service sends and/or receives. These operations are characterized by the Operation element that describes the types of calls in the abstract,
- **Binding**: describes a specific binding component of an interface with the communication protocol used, i.e. d. how a call can be completed in a concrete way (for example, using SOAP with HTTP). The BindingOperation element contains input, output, and faults described according to the concrete realization of the call,
- **Service**: describes a service, identifies its PortType interface and endpoint locations.

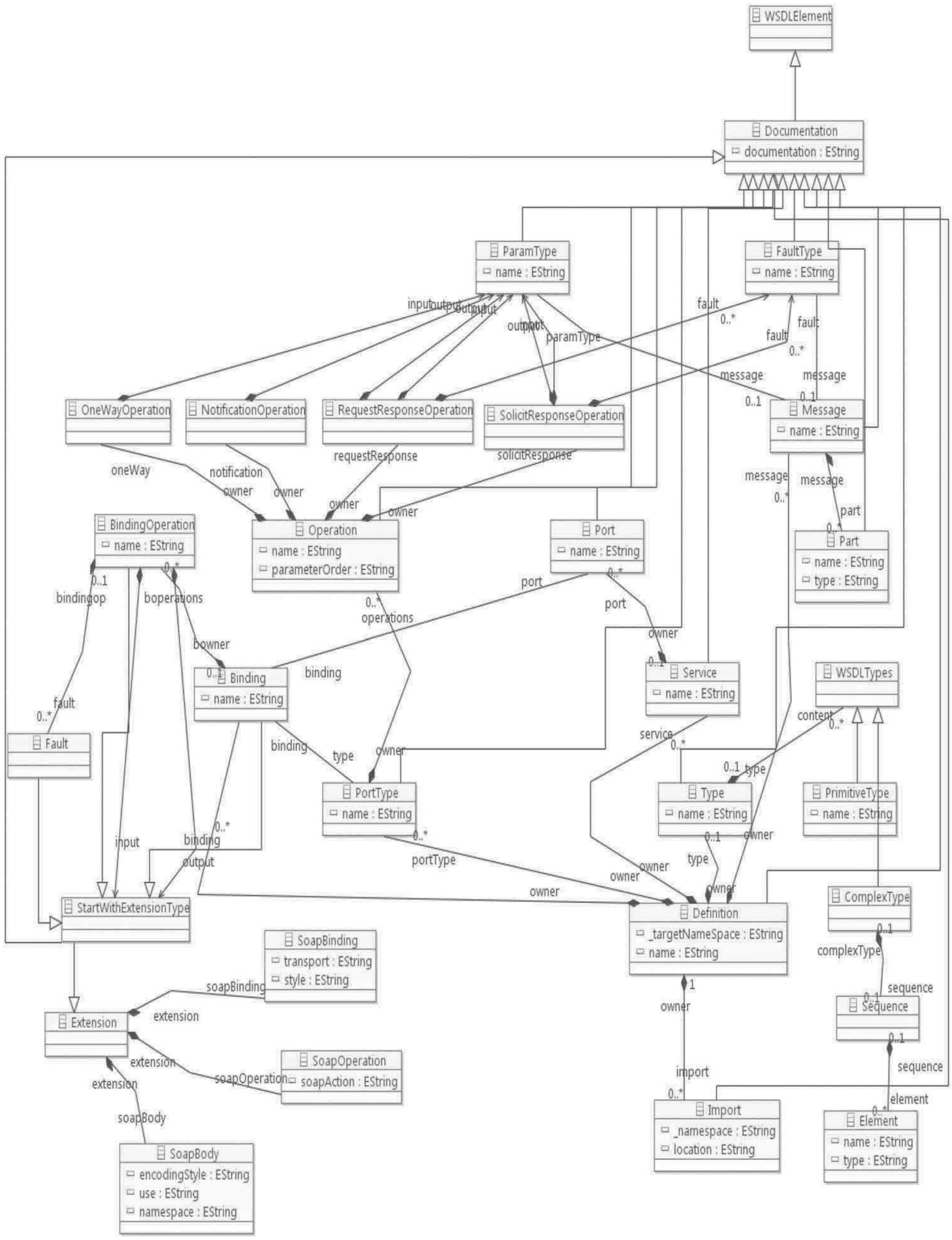

Figure 7. The WSDL metamodel.

1.3. *The specification of correspondence between UML and WSDL*

The specification of correspondences aims to specify the correspondences between the elements of two meta-models as with the example shown in Figure 8. This graphic notation highlights the inter-relationships between the different elements of these two metamodels. The source metamodel is located in the left part, the target metamodel in the right part, and the mapping is in between. A dashed and left-arrow line illustrates a reference to an element of the source metamodel. A double-dotted dashed line on the right illustrates a reference to an element of the target metamodel. The

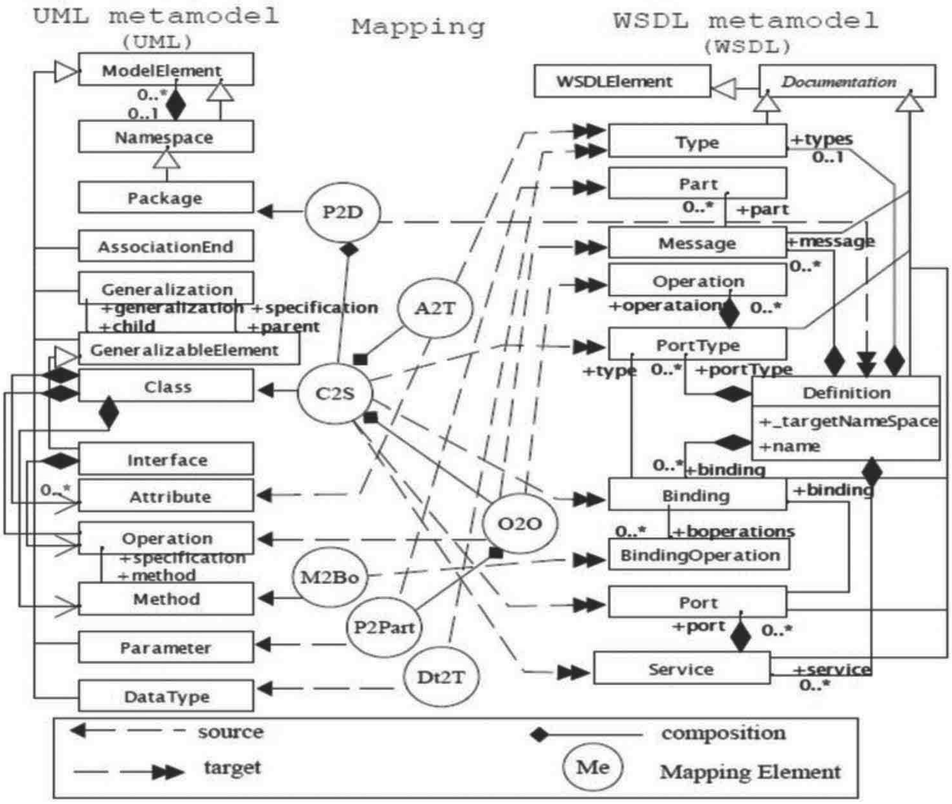

Figure 8. Specification of the correspondence between UML and WSDL (Lopes et al. 2005).

circle denotes a piece of correspondence identified by a name. The lines started by a diamond-shaped end represent the notion of UML composition: a correspondence element can be composed of other correspondence elements. Therefore, all elements are concerned with matching process. After that, the definition of transformations allows to write in detail and to execute the steps of the transformation of a model into another model respecting the specification of correspondences.

The Figure 8 shows a specification of correspondence between UML metamodel source and target WSDL metamodel. The matching element P2D connects the element of UML Package with the element Definition WSDL. The matching element C2S connects the UML Class element with the elements Service, Port, and Binding PortType WSDL. The matching element O2O connects Operation element UML with Operation Message WSDL elements. The P2Part matching element connects the UML Parameter element with the share component of WSDL. The matching element Dt2T connects the item DataType UML with the element type WSDL.

1.4. *The definition of transformation between UML and WSDL*

The language ATL (Atlas Transformation Language) was used to perform the definition processing between UML and WSDL manually based on specifying correspondence presented above. After the first model-to-model transformation to get the information from a UML model to a WSDL model. Then a WSDL transformation model to WSDL, i.e. the WSDL document, is required to complete the cycle of transformation and obtain the final platform. The definition of transformation ATL uses a query to trigger the transformation model-to-code, through the helpers to navigate and retrieve information from the WSDL model and write the WSDL document.

1.5. *The definition of transformations from WSDL models to WSDL codes*

```
File: WSDL2SourceCode_query.atlQuery WSDL2SourceCode_query = WSDL!
Definition.allInstances()->collect (x|x.toString().writeTo ('D:/
SourceCode/WSDL'+ x.name.replaceAll('.', '/') + '/' + x.name + '.wsdl'));
uses WSDL2SourceCode;
```

This request triggers the transformation model-to-code.

The allInstances () operation returns a collection consisting of all instances of the WSDL and definitions present in the WSDL model. The operation of collections is called collect (), then we can recover each element of this collection (represented by x) and invoke toString () operation on each of them.

This is a helper in ATL defined in WSDL2SourceCode library presented above. After the execution of the helper, the string is written in a '* .wsdl' file through the WriteTo () operation. The helper toString () defined in the context of WSDL! Definition also connects other calls to other helpers defined in the context of Service, Port, PortType and so on.

1.6. *Modeling of the case study*

The case study concerns an information system for maintenance of the component parts of passengers and baggage railcars of Sidi Bel Abbes (Algeria). This software helps to have better management of the RMW by the effective planning of interventions and to improve performance by increasing reliability, traceability, and availability of the equipment (e.g. critical parts). The RMW is responsible for the maintenance of Railways vehicles transporting persons. The RMW is a subsidiary of the Algerian National Rail Transport Company (abbreviated in French as 'SNTF') having other Railways vehicles transporting goods. A railcar is made up of a set of parts such as bogie, brake cylinder and brake devices, batteries, bar tables, doors, wings, chairs, windows, thermostats, floors, and ceilings etc.

The RMW includes a set of operational workshops which have the following tasks:

- Maintenance of the passenger railcars fleet;
- Maintenance of the railcar pneumatic, electrical, mechanical, air-condition systems;
- Rehabilitation and modernization of old passenger railcars;
- Supply management;
- Preparation of inspection and maintenance documents;
- Pricing and costs of maintenance;
- Training, learning, retraining and development.

The role of the maintenance of the component parts of passenger and baggage railcars is the proper management of replacement parts in order to make it as easy and efficient as possible. The passenger vehicle fleet of Sidi Bel Abbes is the main fleet and there are several secondary fleets scattered throughout the geographical regions of the country. The secondary fleets send failed parts to the main fleet in order to obtain new parts or parts in good operating conditions deliver in exchange by means of a justified request for an exchange with the code 3065. In return, the main fleet sends them such new parts or parts in good operating conditions, accompanied by a shipping bill with the code 3066. Once the failed parts are received, the main fleet is responsible for their distributions among different RMW workshops to undergo a meaningful repair, accompanied by a repair request with the code 3071. Where the failed parts are lost or irreparable for physical, technical or economic reasons, they should be replaced by new ones.

After their repair, the head of the workshop returns parts to the main fleet accompanied by a statement of delivery or a shipping bill. The fleets must always have parts located physically in the component parts of passengers and baggage railcars, available in a repair situation or in an

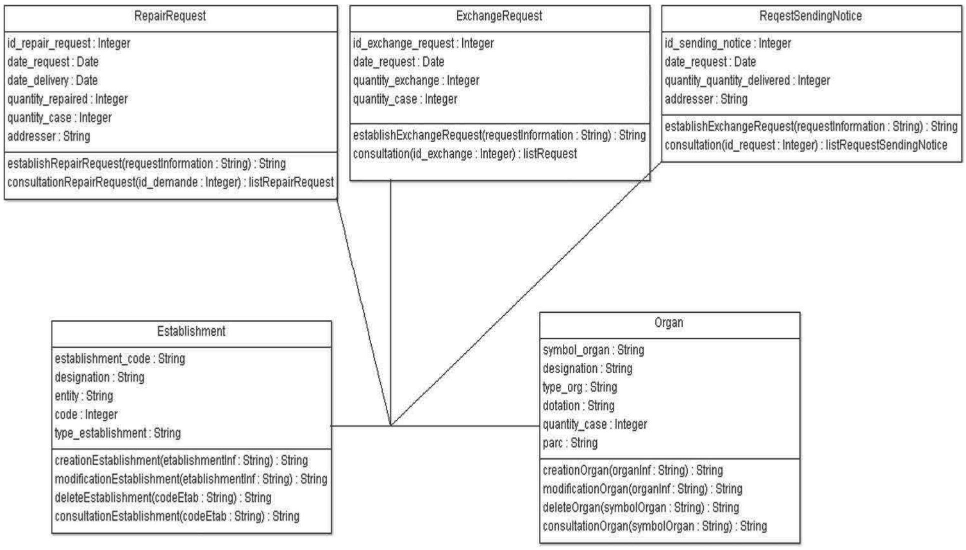

Figure 9. A class diagram of the management of the maintenance of the component parts of passengers and baggage railcars.

exchange process which respects the quantity of the parts defined by the nomenclature and inventories. Figure 9 presents a class diagram of the management of the maintenance of the component parts of passengers and baggage railcars.

The management of the Railways Maintenance Workshop (RMW) involves three main types of actors: the head of the principal maintenance of the component parts of passengers and baggage railcars, the head of the secondary maintenance of the component parts of passengers and baggage railcars and heads of workshops.

Head of the main maintenance of the component parts of passengers and baggage railcars

- Consultation of the requests for exchanges established by the heads of the secondary maintenance of the component parts of passengers and baggage railcars,
- Establishment of a shipping bill of the new or repaired parts to the secondary maintenance of the component parts of passengers and baggage railcars,
- Registration, Update, Delete, Consultation of the parts table,
- Establishment of a request for the repair of failed parts to workshops,
- Registration, Update, Delete, Consultation of the fleets table.

Head of secondary maintenance of the component parts of passengers and baggage railcars

- Establishment of an exchange request to send to the main maintenance of the component parts of passengers and baggage railcars,
- Consultation of parts table.

Head of the workshop

- Establishment of a shipping bill of the repaired parts of the main fleet,
- Consultation of requests for repairs at his workshop.

The Figure 10 presents a UML use case diagram for the interaction of the head of the main maintenance of the component parts of passengers and baggage railcars and the information system. Figure 11 presents an activity diagram that shows a flowchart of stepwise activities of the head of the main maintenance of the component parts of passengers and baggage railcars.

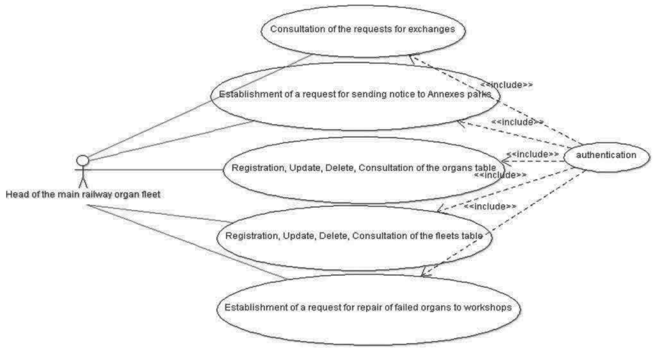

Figure 10. A use case diagram of the head of the main maintenance of the component parts of passengers and baggage railcars.

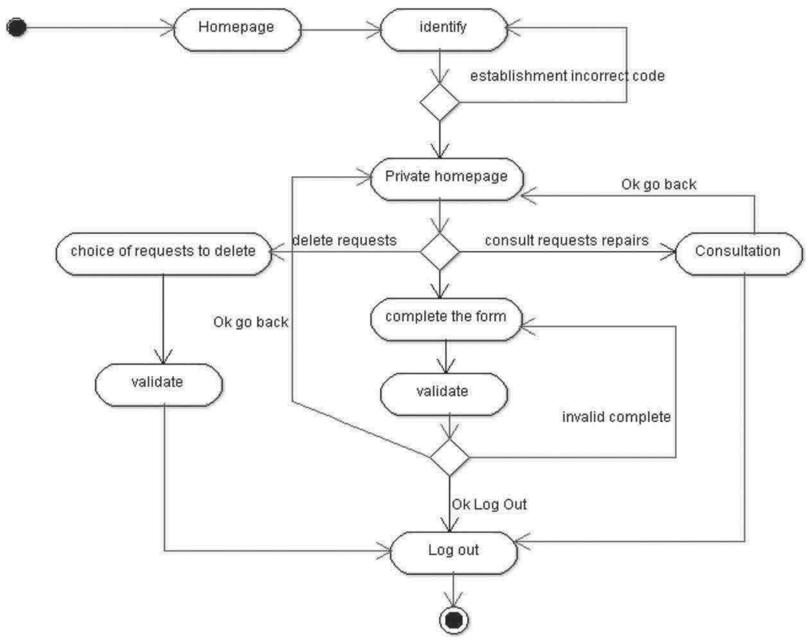

Figure 11. An activity diagram of the head of the main maintenance of the component parts of passengers and baggage railcars.

1.7. *Servicization applied to the management of maintenance of the component parts of passengers and baggage railcars*

For this purpose, we present the automatic generation of web service, specifically the WSDL file. This generation of web services is applied to the target case study that focuses on the management of the parts fleets in RMW. A web service allows to export and expose the features of a computer application and to make them accessible via standard protocols. These features define the different methods and their input and output parameters. However, the real implementation of these features is relayed to the computer application. In our case, the automatically generated web services are complete because they include all the information needed for their deployment: Import, Type, Message, PortType, Binding, and Service. Therefore, it is an automatic generation of web service. For the implementation of the computer system for the case study, we opted for the top down java bean Web service approach.

From our auto-generated web services (WSDL files) through the MDA approach, the top down java bean Web service approach generates java classes. The different methods of these java classes will be implemented by programmers to achieve the computer system.

The main tool used is the Eclipse Modeling Framework that supports the creation and the publication of metamodels in compliance with a core metamodel (Ecore). It also enables the edition of the models compliant to the principles of these metamodels. The MDA approach requires two metamodels to perform transformations from a source model to a target model, for our case it is the UML metamodel and the WSDL metamodel. Thus, the CIM and the PIM are modeled with UML. They must necessarily conform to the UML metamodel. A UML metamodel is a class diagram conforming to the MOF1.4 standard. For PSM modeling, MDA also recommends the use of UML Profile to achieve the metamodel of the target platform. This explains the need to have a WSDL metamodel to which the web service templates must conform. At this level, we have created some metamodels complying with UML and WSDL. To achieve the processing of model transformations, we used the language ATL (Atlas Transformation Language) integrated into the Eclipse EMF.

The methodology engaged in management of parts fleet includes the following steps:

- Creation of PIM models for the management of parts fleets in compliance with the UML metamodel
- Generation of target PSM models in compliance with the WSDL metamodel
- Generation of source code: services for the management of the parts fleet.

During the procedure established for the creation of PIM models compliant to the UML metamodel, we define the Package RMW that contains the *Class Organ (parts)*. Moreover, in this *class* we define the operations containing the input parameters and the output parameters; and then the specific methods applying to operations. In practice, maintenance packages could be a good added value to the considered information system (Kamsu-Foguem et al., 2015). Figure 12 presents the complete model of the *Class Organ* of Package RMW.

The generation of target PSM models in compliance with the WSDL metamodel is achieved via the import and treatment of ATL code defining the transformation between the UML metamodel and the WSDL metamodel. In this case, the inputs are the source models that conform to the UML metamodel and the outputs are target models that conform to the WSDL metamodel. Figure 13 describes an example of the generation of the WSDL model in the xmi format with the file called *UML2WSDL.ATL* for ATL Module and the file *organe.xmi* conforms to WSDL.

The realization of the information system of parts fleet management is made by following the final steps as detailed below:

- Importation of the engendered WSDL files into the integrated development environment Eclipse in order to generate the skeletons of operations;
- Provision of further information on these skeletons in order to achieve the server part of the information system;
- Deployment of fleet management services within the Tomcat/Axis server;
- Realization of web services for remote information review, knowledge sharing and flexible configuration of applications in advanced ways.

In this way, the information system guarantees the gathering and registration of the information essential to ensuring traceability of railways parts in maintenance workshops.

Discussions

The servicization of the considered information system provides interesting means for continuous improvement of organizational capabilities in the fleet management. One of the obstacles to find

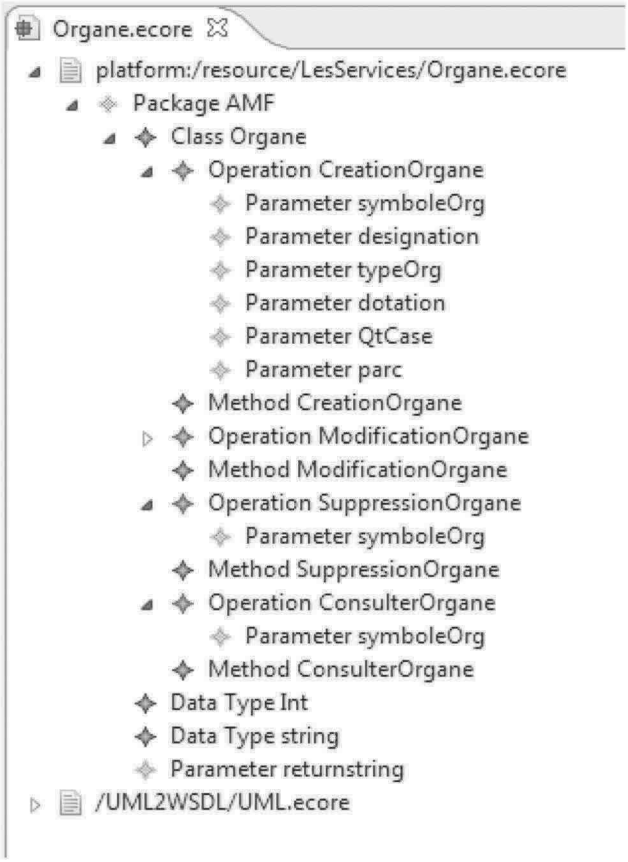

Figure 12. A complete model of the *Class Organ* of Package RMW.

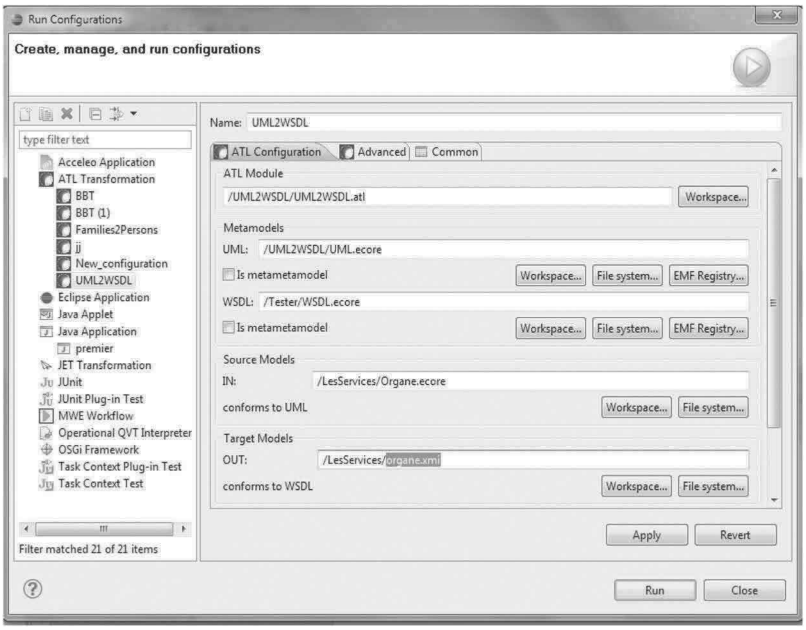

Figure 13. Generation of the WSDL model in the xmi format.

and share information on operational fleet safety management is the lack of evidence-based and consistent assessment means for proper implementation (Mitchell, Friswell, and Mooren 2012). A suitable system of information organization and management can make a substantial contribution to an optimized operational use and maintenance of fleets dedicated to transporting services (Lenain, Kechmire, and Smaha 1995). On the one hand, it is very common that analogous or identical railway parts in a fleet are being exposed to totally diverse working conditions for various tasks. On the other hand, railway parts that are accomplishing analogous tasks or that are in similar service periods may have comparable reliability and maintainability conditions. Based on such comparisons, part groups can be built as a rich information base representing diverse railways parts reliability and maintainability conditions (Lee, Kao, and Yang 2014).

Therefore, it is possible to take advantage of these characteristics to achieve a valuable fleet information system by collecting meaningful information from diverse cases and actions of maintenance (Potes Ruiz, Foguem, and Noyes 2013). In the same way, knowledge structuring schemes of fleets with semantic models based on ontologies can be engaged in the PHM domain for diagnostic purposes (Medina-Oliva et al. 2014). So, the formalization of past experiences in the considered fleet information system would generate ways and means to improve the obtainability of significant railways parts repair information and its integration into reliability and maintainability assessments for achieving the accessibility of maintenance services (Kamsu-Foguem and Noyes 2013). Furthermore, inspection and maintenance activities can be organized and adjusted from the identified patterns in the fleet levels with a potential support for knowledge capitalization and reuse (Potes Ruiz, Foguem, and Grabot 2014) (Medina-Oliva, Weber, and Iung 2015). Finally, the improved information and communication among the maintenance workshops, service management level and procurement management make the maintenance management more structured and comprehensible.

Quality management has become a way of life in every technical and business sector, whether service, logistic, building management, or any other sector, and its importance is broadly understood (Kiran 2017). A traceability system records and follows the trail of products, parts, materials, and services come from suppliers and are processed and ultimately distributed as final products and services (Campos and Miguez 2011). In the business networks, a specific and complete set of service guidelines can be provided for the development of intelligent automation control and execution of manufacturing systems (Giret, Garcia, and Botti 2016). One of the crucial characteristics to satisfy users is the availability of the system (Houssin and Coulibaly 2014). For that reason, all users of the system are secured by an authentication process. To improve data interoperability, availability, and sustainability, it is essential to introduce a new kind of intelligent products referring to communicating material paradigm to convey information between the different actors of the Product Life Cycle (PLC) (Kubler et al. 2013). In the railway domain, Model-Driven approach is useful for the evaluation of RAM (Reliability, Availability, Maintainability) by automatically generating formal models for the verification of railway applications (Bernardi et al. 2013). The maintenance information system must be able to provide accurate information about all maintain parts which allow reliability of information. The implemented information system takes into account the two maintenance techniques: time-based maintenance (TBM) and the most modern and popular CBM (Rosmaini & Shahrul, 2012). In the context of our study, we have focused on the maintenance policy defined by the enterprise, which aimed at optimizing the maintenance policy for a system is one of the most critical issues in operations management (Zheng et al. 2016). While our approach allows traceability of information about all parts of the collaborative maintenance system, by focusing on SOA to enhance the traceability services through a set of shared services dedicated to system developers (Kang and Lee 2013). This work contributes to improve performance by increasing reliability, traceability, and availability of the equipment (parts) through the complete realization of the company's computer system. Servicization using MDA approach is very profitable in replacement of legacy systems, migration the current system, wrapping and reengineering existing system functionality.

Conclusion and related works

The revolution of ICTs has facilitated the publication of enterprise functionalities and involves at the same time intra and extra communication business (B2B Business to Business) and B2C (Business to Costumer) (Liu et al. 2018). Flexibility and interoperability are two requirements in the development of enterprise information systems. In this way, servicization has been proposed to replace or supplement the legacy systems and middleware unable to efficiently respond to the requirement for interoperability on the Internet. SOA meets the interoperability issues of complex information systems that have a lot of challenges to face, especially concerning the complexity of information systems and high costs of technological migrations. Thus, the combination of service-centric strategies with MDA principles are interesting for the servicization of enterprise information systems.

The adopted methodology is deployed to support the considered information modeling framework for an application to maintenance management. This methodology is characterized by a formal specification of correspondences which gives a better overview of model properties and makes easier the definition of transformations. In addition monitoring/control model specifications combined with associated domain requirements and organizational features make easier the knowledge of the functional status of target systems. As a result, an efficient management of system modeling is a basic precondition for controlling operational processes and enables them to be configured in a way that guarantees flexibility and adaptability within a dynamic context having a variety of origins and consequences.

The case study concerns the parts fleet management system of RMW. The scheduling of maintenance activities is crucial in reliability and maintenance engineering (Manzini et al. 2015). In this context, the realization of the information system for the management of maintenance of the component parts of passengers and baggage railcars using the generated solution supports decision making process to plan interventions effectively in order to ensure equipment inventory at any time and improve performance by increasing the reliability, availability and traceability of equipment (parts) and their maintenance. For some RMW users, the information system developed is useful and ergonomic. However, there are some limitations and we will try to solve them in further works since this is the first version of the implemented tool, therefore few points should be improved such as user interface and the displays of generated reports. The distributed nature of transport services means that maintenance workshop policies cannot be considered in an isolated manner, but these workshops advance from sharing information and from delivering mutual supports in a fleet setting. So, the servicization of the information system of maintenance of the component parts of passengers and baggage railcars offers an opportunity for the maintenance management to deliver innovative services in a collaborative environment (Macbeth and Ibanez de Opacua, 2010).

This servicization has been implemented to accommodate technological evolution. Indeed, the servicization of the CMMS and its implementation to SOA enables its scalability. This scalability could be necessary to implement the emerging paradigm of PHM that requires the implementation of smart monitoring of critical components directly into the systems to maintain (Xia et al. 2018). These monitoring are able to assess the remaining useful life of those components in order to replace traditional maintenance policies such as corrective maintenance and systematic preventive maintenance by CBM (Tobon-Mejia et al. 2012). The servicization also facilitates reporting and decision-making. It enhances the traceability of the considered information system in maintenance management according to operating conditions of railway parts. In technical terms, the provided information system architecture permits the accessibility, openness, and interoperability of the delivered maintenance services that are integral to distant exchanges and collaborations in heterogeneous and distributed environments. Finally, the realization of web services for remote information review and knowledge sharing for the maintenance workshops of the maintenance of the component parts of passengers and baggage railcars will improve maintenance management in a structured, organized and controlled strategic fashion. However, there are also needs to cover the risk management dimension on fleet replacement and sustainable operations (Ansaripoor, Oliveira, and Liret 2014). The efforts towards combined model-driven and ontology-driven system development are showing promise in capturing the semantics of available services and exploring knowledge verification and sharing mechanisms (Chungoora et al. 2013). Finally, the joint use of ontology-based models (Nuñez and Borsato 2017) and deep learning methods (Khan and Yairi 2018) for prognostics and health management of complex systems seems as an interesting issue as future researches.

Highlights

- Stiervicization for flexible design principles in collaborative computing practices.
- Combining Service-oriented architecture (SOA) with Model-driven architecture (MDA).
- Methodology for the alignment between the business view and information system view
- Case study for the parts fleet management system of Railway Maintenance Workshops.
- Servitization improves maintenance with services based innovation in collaboration.

Acknowledgments

This work is supported by a funding within the grant of “Programme de Formation des Formateurs des Universités de Bamako et de Ségou”, delivered by the Malian government. We want to thank the Doctoral School Aeronautics and Astronautics of Toulouse University whose activities have stimulated the research works described in the content of the paper. Thanks are extended to Enterprise Information Systems’ reviewers for their constructive comments and suggestions during the entire review process.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Programme de Formation des Formateurs des Universités de Bamako et de Ségou (PFF), Convention N°15 - 050 -/ USJPB-R/PFF - [0137.919-B], GOUVERNEMENT DE LA REPUBLIQUE DU MALI.

References

Ahmad, R., and S. Kamaruddin. 2012. “An Overview of Time-Based and Condition-Based Maintenance in Industrial Application.” *Computers & Industrial Engineering* 63 (1, August): s135–149. doi:10.1016/j.cie.2012.02.002.

Almonaies, A., J. R. Cordy, and T. R. Dean, “Legacy System Evolution Towards Service-Oriented Architecture”, In Proceedings of International Workshop on SOA Migration and Evolution (SOAME) 2010, Madrid, Spain, March 2010, 53–62.

Amadi-Echendu, J. E., and F. C. P. De Wit. 2015. “Technology Adoption: A Study on Post-Implementation Perceptions and Acceptance of Computerised Maintenance Management Systems.” *Technology in Society* 43: 209e218. doi:10.1016/j.techsoc.2015.09.001.

Ansaripoor, A. H., F. S. Oliveira, and A. Liret. “A Risk Management System for Sustainable Fleet Replacement.” *European Journal of Operational Research* 237(2): s701–712. 1September 2014. doi:10.1016/j.ejor.2014.02.006.

Barry., D. K. *Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager’s Guide*, M. Kaufmann, Editor, 2 Édition. 248. 31 December, 2012. Burlington, MA: Morgan Kaufmann

Belete, G. F., A. Voinov, and G. F. Laniak. 2017. “An Overview of the Model Integration Process: From Pre-Integration Assessment to Testing.” *Environmental Modelling & Software* 87 (January): s49–63. doi:10.1016/j.envsoft.2016.10.013.

Bell, M. ““Introduction to Service-Oriented Modeling”.” *Service-Oriented Modeling: Service Analysis, Design, and Architecture*, Editor. John Wiley & Sons: Hoboken, New Jersey, United States. 11April, 2008. 384.

Benaben, F., M. Wenxin, N. Boissel-Dallier, A.-M. Barthe-Delanoe, S. Zribi, and H. Pingaud. 2015. "Supporting Interoperability of Collaborative Networks through Engineering of a Service-Based Mediation Information System (MISE 2.0)." *Enterprise Information Systems* 9 (5–6): s556–582.

Bernardi, S., F. Flammini, S. Marrone, N. Mazzocca, J. Merseguer, R. Nardone, and V. Vittorini. 2013. "Enabling the Usage of UML in the Verification of Railway Systems: The DAM-rail Approach." *Reliability Engineering & System Safety* 120 (December): s112–126. doi:10.1016/j.ress.2013.06.032.

Campos, J. G., and L. R. Miguez. 2011. "Standard Process Monitoring and Traceability Programming in Collaborative CAD/CAM/CNC Manufacturing Scenarios." *Computers in Industry* 62 (3, April): s311–322. doi:10.1016/j.compind.2010.09.003.

Carnero, M. C., and G. Andrés. "Maintenance Strategy Selection in Electric Power Distribution Systems." *Energy* 129: s255–272. 15June 2017. doi:10.1016/j.energy.2017.04.100.

Cerny, T. "Aspect-Oriented Challenges in System Integration with Microservices, SOA and IoT." *Enterprise Information Systems* Published online. 10 Apr, 2018. doi:10.1080/17517575.2018.1462406.

Chungoora, N., R. I. Young, G. Gunendran, C. Palmer, Z. Usman, N. A. Anjum, A.-F. Cutting-Decelle, J. A. Harding, and K. Case. 2013. "A Model-Driven Ontology Approach for Manufacturing System Interoperability and Knowledge Sharing." *Computers in Industry* 64 (4, May): s392–401. doi:10.1016/j.compind.2013.01.003.

Desforges, X., A. Habbadi, and B. Archimède. 2011. "Design Methodology for Smart Actuator Services for Machine Tool and Machining Control and Monitoring." *Robotics and Computer-Integrated Manufacturing* 27: 963–976. doi:10.1016/j.rcim.2011.03.004.

Desforges, X., M. Dievart, and B. Archimede. 2017. "A Prognostic Function for Complex Systems to Support Production and Maintenance Co-Operative Planning Based on an Extension of Object Oriented Bayesian Networks." *Computers in Industry* 86: 34–51. doi:10.1016/j.compind.2017.01.002.

Dhillon, B. S. 2002. "Engineering Maintenance – A Modern Approach." In: *CRC Press LLC, Boca Raton*. United States: Florida: Taylor & Francis Group.

Durazo-Cardenas, I., A. Starr, C. J. Turner, A. Tiwari, L. Kirkwood, M. Bevilacqua, A. Tsourdos, E. Shehab, P. Baguley, and X. Yuchun. 2018. "Christos Emmanouilidis. An Autonomous System for Maintenance Scheduling Data-Rich Complex Infrastructure: Fusing the Railways' Condition, Planning and Cost." *Transportation Research Part C: Emerging Technologies* 89 (April): s234–253. doi:10.1016/j.trc.2018.02.010.

Erkoc, M., and K. Ertogral. 2016. "Overhaul Planning and Exchange Scheduling for Maintenance Services with Rotable Inventory and Limited Processing Capacity." *Computers & Industrial Engineering* 98: 30–39. doi:10.1016/j.cie.2016.05.021.

Erkoyuncu, J. A., S. Khan, A. L. Eiroa, N. Butler, K. Rushton, and S. Brocklebank. 2017. "Perspectives on Trading Cost and Availability for Corrective Maintenance at the Equipment Type Level." *Reliability Engineering & System Safety* 168 (December): s53–69. doi:10.1016/j.ress.2017.05.041.

Galar, D., A. Thaduri, M. Catelani, and L. Ciani. 2015. "Context Awareness for Maintenance Decision Making: A Diagnosis and Prognosis Approach." *Measurement* 67 (May): s137–150. doi:10.1016/j.measurement.2015.01.015.

Giret, A., E. Garcia, and V. Botti. 2016. "An Engineering Framework for Service-Oriented Intelligent Manufacturing Systems." *Computers in Industry* 81 (September): s116–127. doi:10.1016/j.compind.2016.02.002.

Gorton, I. *Essential Software Architecture*, 2nd. Springer: Editeur. 27 avril, 2011. 242.

Hockmayr, B. *Chances and Limatations of New Business Model Generation by Servicization*. Editeur: Grin Verlag. 5 septembre, 2013. 24.

Houssin, R., and A. Coulibaly. 2014. "Safety-Based Availability Assessment at Design Stage." *Computers & Industrial Engineering* 70: 107–115. doi:10.1016/j.cie.2014.01.005.

Jorgensen, D. 2002. "Chapter 2 - Introduction to the Microsoft .NET Framework.." *Developing Net Web Services with XML* 29–65. Burlington, MA: Syngress. doi:10.1016/B978-192899481-7/50005-8

Jouault, F., F. Allilaire, J. Bézivin, and I. Kurtev. 2008. "ATL: A Model Transformation Tool." *Science of Computer Programming* 72(1–2): s31–39. 1 June 2008. doi:10.1016/j.scico.2007.08.002.

Kamsu-Foguem, B., and D. Noyes. 2013. "Graph-Based Reasoning in Collaborative Knowledge Management for Industrial Maintenance." *Computers in Industry* 64 (8, October): s998–1013. doi:10.1016/j.compind.2013.06.013.

Kamsu-Foguem, B., and Y. Mathieu. 2014. "Software Architecture Knowledge for Intelligent Light Maintenance." *Advances in Engineering Software* 67 (January): s125–135. doi:10.1016/j.advengsoft.2013.09.003.

Kamsu-Foguem, B., Y.F. Wandji, G. Tchuenté-Foguem, and D. Tchoffa. 2015. "Experienced Knowledge for the Description of Maintenance Packages." *Journal of Manufacturing Systems* 37: 448-455. doi:10.1016/j.jmsy.2014.11.005.

Kang, Y.-S., and Y.-H. Lee. 2013. "Development of Generic RFID Traceability Services." *Computers in Industry* 64 (5, June): s609–623. doi:10.1016/j.compind.2013.03.004.

Khan, S., and T. Yairi. 2018. "A Review on the Application of Deep Learning in System Health Management." *Mechanical Systems and Signal Processing* 107 (July): s241–265. doi:10.1016/j.ymssp.2017.11.024.

Khoshkbarchi, A., and H. R. Shahriari. 2017. "Coping with Unfair Ratings in Reputation Systems Based on Learning Approach." *Enterprise Information Systems* 11 (10). doi:10.1080/17517575.2016.1221999.

Kiran, D. R. 2017. “Chapter 34: ISO 9000 Quality Systems.” In: *Total Quality Management*, edited by Butterworth-Heinemann, 471–486. Oxford, United Kingdom: BSP Books Pvt. Ltd. Published by Elsevier . doi:10.1016/B978-0-12-811035-5.00034-9.

Kubler, S., W. Derigent, E. Rondeau, A. Thomas, and F. Kary. 2013. “Information Dissemination Framework for Context-Aware Products.” *Computers & Industrial Engineering* 66 (2, October): s485–500. doi:10.1016/j.cie.2013.07.015.

Lee, J., H.-A. Kao, and S. Yang. 2014. “Service Innovation and Smart Analytics for Industry 4.0 And Big Data Environment.” *Procedia CIRP* 16: s3–8. doi:10.1016/j.procir.2014.02.001.

Lee, J., M. Shang-Pin, and A. Liu. *Service Life Cycle Tools and Technologies: Methods, Trends, and Advances*, Editor. 356. Business Science Reference. 15 mars, 2012. Hershey, PA: Information Science Reference.

Lenain, P., M. Kechmire, and J. P. Smaha. 1995. “Monitoring Fleets of Electric Vehicles: Optimizing Operational Use and Maintenance.” *Journal of Power Sources* 53 (2, February): s335–338. doi:10.1016/0378-7753(94)01995-8.

Li, Y., and F. W. Guldenmund. 2018. “Safety Management Systems: A Broad Overview of the Literature.” *Safety Science* 103 (March): s94–123. doi:10.1016/j.ssci.2017.11.016.

Liu, Y., T. Foscht, A. B. Eisingerich, and H.-T. Tsai. 2018. “Strategic Management of Product and Brand Extensions: Extending Corporate Brands in B2B Vs. B2C Markets.” *Industrial Marketing Management* 71 (May): s147–159. doi:10.1016/j.indmarman.2017.12.016.

Lopes, D., S. Hammoudi, J. Bézivin, and F. Jouault, Generating Transformation Definition from Mapping Specification: Application to Web Service Platform, Lecture Notes in Computer Science - The 17th Conference on Advanced Information Systems Engineering (CAiSE’05), 13–17 June 2005, Porto, Portugal, 309–325 (ISSN: 0302–9743 (Print) 1611–3349 (Online), doi:10.1007/11431855_22).

Macbeth, D. K., and A. I. De Opacua. 2010. “Review of Services Science and Possible Application in Rail Maintenance.” *European Management Journal* 28 (1, February): s1–13. doi:10.1016/j.emj.2009.10.001.

Madhikermi, M., S. Kubler, J. Robert, A. Buda, and F. Kary 2016. “Data Quality Assessment of Maintenance Reporting Procedures.” *Expert Systems with Applications* 63: s145–164. doi:10.1016/j.eswa.2016.06.043.

Manzini, R., R. Accorsi, T. Cennerazzo, E. Ferrari, and F. Maranesi. 2015. “The Scheduling of Maintenance. A Resource-Constraints Mixed Integer Linear Programming Model.” *Computers & Industrial Engineering* 87 (September): s561–568. doi:10.1016/j.cie.2015.06.006.

Medina-Oliva, G., A. Voisin, M. Monnin, and J.-B. Leger. 2014. “Predictive Diagnosis Based on a Fleet-Wide Ontology Approach.” *Knowledge-Based Systems* 68 (September): s40–57. doi:10.1016/j.knosys.2013.12.020.

Medina-Oliva, G., P. Weber, and B. Iung. 2015. “Industrial System Knowledge Formalization to Aid Decision-Making in Maintenance Strategies Assessment.” *Engineering Applications of Artificial Intelligence* 37 (January): s343–360. doi:10.1016/j.engappai.2014.09.006.

Mezni, H., and M. Kbekbi. “Reusing Process Fragments for Fast Service Composition: A Clustering-Based Approach.” *Enterprise Information Systems* Published online. 13 Apr, 2018. doi:10.1080/17517575.2018.1460866.

Mietzner, R., F. Leymann, and T. Unger. 2011. “Horizontal and Vertical Combination of Multi-Tenancy Patterns in Service-Oriented Applications.” *Enterprise Information Systems* 5 (1): s59–77.

Mitchell, R., R. Friswell, and L. Mooren. 2012. “Initial Development of a Practical Safety Audit Tool to Assess Fleet Safety Management Practices.” *Accident Analysis & Prevention* 47 (July): s102–118. doi:10.1016/j.aap.2012.01.021.

Model Driven Architecture (MDA), Document – ormsc/2001-07–01

Model Driven Architecture (MDA): The MDA Guide Rev 2.0. Adopted by the OMG in Mid-2014 as the Official Description of the MDA. Document – ormsc/2014-06-01 (MDA Guide revision 2.0).

Neiva, F. W., N. José Maria, R. B. David, and F. Campos. 2016. “Towards Pragmatic Interoperability to Support Collaboration: A Systematic Review and Mapping of the Literature.” *Information and Software Technology* 72 (April): s137–150. doi:10.1016/j.infsof.2015.12.013.

Nuñez, D. L., and M. Borsato. 2017. “An Ontology-Based Model for Prognostics and Health Management of Machines.” *Journal of Industrial Information Integration* 6 (June): s33–46. doi:10.1016/j.jii.2017.02.006.

Object Management Group (OMG); Enterprise Collaboration Architecture (ECA) Specification, February 2004. Version 1.0, formal/04-02–01.

Object Management Group (OMG); Object Constraint Language (OCL), Version 2.4, OMG Document Number Formal/2014-02-03, February 2014.

Olde Keizer, M. C. A., S. D. P. Flapper, and R. H. Teunter. 2017. “Condition-Based Maintenance Policies for Systems with Multiple Dependent Components: A Review.” *European Journal of Operational Research* 261 (2, September 1): s405–420. doi:10.1016/j.ejor.2017.02.044.

Palau, A. S., Z. Liang, D. Lütgehetmann, and A. K. Parlikad. Collaborative Prognostics in Social Asset Networks. Future Generation Computer. Future Generation Computer Systems Available online 17 February 2018, doi:10.1016/j.future.2018.02.011

Palmarini, R., J. A. Erkoyuncu, R. Roy, and H. Torabmostaedi. 2018. “A Systematic Review of Augmented Reality Applications in Maintenance.” *Robotics and Computer-Integrated Manufacturing* 49 (February): s215–228. doi:10.1016/j.rcim.2017.06.002.

Patel, K., and L. Ragha. 2013. “Survey and Analysis on Migration of Legacy System to Service Oriented Architecture.” *IOSR Journal of Computer Engineering (IOSR-JCE)* 9 (2, Jan.-Feb): PP32–35.

Patrascoiu, O. YATL: Yet Another Transformation Language - Reference Manual Version 1.0, T echnical Report No. 2–04 March 2004, University of Kent, Canterbury, Kent CT2 7NF, UK.
Platform, J. 2009. “Enterprise Edition: (Java EE) Specification, V6.” *Sun Microsystem*, November 6: 264.
Potes Ruiz, P. A., B. K. Foguem, and D. Noyes. 2013. “Knowledge Reuse Integrating the Collaboration from Experts in Industrial Maintenance Management.” *Knowledge-Based Systems* 50 (September): s171–186. doi:10.1016/j.knosys.2013.06.005.
Quanwang, W., Q. Zhu, and L. Peng. 2015. “A Neural Network Based Reputation Bootstrapping Approach for Service Selection.” *Enterprise Information Systems* 9 (7): 768-784.
Rieger, C., and H. Kuchen. 2018. “A Process-Oriented Modeling Approach for Graphical Development of Mobile Business Apps.” *Computer Languages, Systems & Structures* 53 (September): s43–58. doi:10.1016/j.cl.2018.01.001.
Roy, R., R. Stark, K. Tracht, S. Takata, and M. Mori. 2016. “Continuous Maintenance and the Future – Foundations and Technological Challenges.” *CIRP Annals* 65 (I2): s667–688. doi:10.1016/j.cirp.2016.06.006.
Ruiz, P. P., B. K. Foguem, and B. Grabot. 2014. “Generating Knowledge in Maintenance from Experience Feedback.” *Knowledge-Based Systems* 68 (September): s4–20. doi:10.1016/j.knosys.2014.02.002.
Saccani, N., F. Visintin, and M. Rapaccini. 2014. “Investigating the Linkages between Service Types and Supplier Relationships in Servitized Environments.” *International Journal of Production Economics* 149 (March): s226–238. doi:10.1016/j.ijpe.2013.10.001.
Sheng, J., and D. Prescott. 2017. “A Hierarchical Coloured Petri Net Model of Fleet Maintenance with Cannibalisation.” *Reliability Engineering & System Safety* 168 (December): s290–305. doi:10.1016/j.ress.2017.05.043.
Shin, J.-H., and H.-B. Jun. 2015. “On Condition Based Maintenance Policy.” *Journal of Computational Design and Engineering* 2 (2, April): s119–127. doi:10.1016/j.jcde.2014.12.006.
Silva Teixeira, E. L., B. Tjahjono, and S. C. Absi Alfaro. 2012. “A Novel Framework to Link Prognostics and Health Management and Product–Service Systems Using Online Simulation.” *Computers in Industry* 63: 669–679. doi:10.1016/j.compind.2012.03.004.
Sweeney, R. *Achieving Service-Oriented Architecture: Applying an Enterprise Architecture Approach*, Editor. Hoboken, NJ: John Wiley & Sons. 18 June, 2010. 432.
Tao, F., H. Guo, L. Zhang, and Y. Cheng. 2012. “Modelling of Combinable Relationship-Based Composition Service Network and the Theoretical Proof of Its Scale-Free Characteristics.” *Enterprise Information Systems* 6 (4): s373–404.
Throop, W., Choosing Computerized Maintenance Management Systems for Facility Management, Engineering Tech Tips, Jun, 2000, 1e4.
Tobon-Mejia, D. A., K. Medjaher, N. Zerhouni, and G. Tripot. 2012. “A Data-Driven Failure Prognostics Method Based on Mixture of Gaussians Hidden Markov Models.” *IEEE Transactions on Reliability* 61: 491–503. doi:10.1109/TR.2012.2194177.
Viriyasitavat, W., L. Da Xu, B. Zhuming, and A. Sapsomboon. 2018. “Extension of Specification Language for Soundness and Completeness of Service Workflow.” *Enterprise Information Systems* 12 (5): s638–657.
Xia, T., Y. Dong, L. Xiao, D. Shichang, E. Pan, and X. Lifeng. 2018. “Recent Advances in Prognostics and Health Management for Advanced Manufacturing Paradigms.” *Reliability Engineering & System Safety* 178 (October): s255–268. doi:10.1016/j.ress.2018.06.021.
Zhang, F., Z. Zhou, Q. Liu, and X. Wenjun. 2017. “An Intelligent Service Matching Method for Mechanical Equipment Condition Monitoring Using the Fibre Bragg Grating Sensor Network.” *Enterprise Information Systems* 11 (2). doi:10.1080/17517575.2015.1048830.
Zhang, T., S. Ying, S. Cao, and J. Zhang. 2008. “A Modelling Approach to Service-Oriented Architecture.” *Enterprise Information Systems* 2 (3): s239–257.
Zheng, Z., W. Zhou, Y. Zheng, and W. Yongzhong. 2016. “Optimal Maintenance Policy for a System with Preventive Repair and Two Types of Failures.” *Computers & Industrial Engineering* 98: 102–112. doi:10.1016/j.cie.2016.05.007.