IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 13, 2019, accepted June 28, 2019, date of publication July 9, 2019, date of current version July 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927785

A Novel Framework to Automatically Generate Executable Web Services From BPMN Models

IQRA ZAFAR, FAROOQUE AZAM, MUHAMMAD WASEEM ANWAR, BILAL MAQBOOL, WASI HAIDER BUTT, AND AIMAN NAZIR

Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Islamabad 44000, Pakistan

Corresponding author: Muhammad Waseem Anwar (waseemanwar@ceme.nust.edu.pk)

ABSTRACT Enterprise resource planning (ERP) is a business process management system in which integrated applications are used to manage business processes in a shared data environment. ERP systems usually deal with the two types of business processes, i.e., exchange and conversion. In the exchange process, economic resource, such as product, exchanges to another economic resource, such as the sales process. In a conversion process, an enterprise consumes resources in order to produce new resources, such as the distribution process. Generally, the communication between ERP applications, based on the conversion and exchange processes, is accomplished through Web services. In this context, the implementation of Web services in ERP systems is a complex task. To manage this, the business process model and notation (BPMN) are frequently utilized to simplify the development of ERP applications. However, state-of-the-art BPMN approaches usually deal with the modeling of exchange processes without considering the conversion process. Furthermore, the model transformation solution to automatically generate Web services from the BPMN models are hard to find in the literature. Therefore, in this paper, a novel framework is proposed that supports the modeling of both exchanges as well as conversion processes through BPMN. Particularly, a modeling approach is introduced to represent the ERP processes through BPMN concepts. Subsequently, the rules are developed to convert source BPMN models into target Service-oriented architecture Modeling Language (SoaML) models. Finally, transformation rules are developed to generate fully functional executable Java Web services from SoaML models. As a part of the research, a complete open-source BPMN to Web services transformation (B2W) tool is developed to automatically generate the Web services from the high-level BPMN models. The proposed framework is validated through multiple case studies. The experimental results prove that the proposed framework accurately generates Web services from the BPMN models, which eventually helps in developing the ERP systems with simplicity.

INDEX TERMS Model-driven engineering (MDE), business process model and notation (BPMN), service oriented architecture (SOA), ERP systems, service generation.

I. INTRODUCTION

Business models are economic models that describe the way and reasons why and how the organization share values. These business models are based on Resource, Event and Agent (REA) ontology [1]. This ontological framework captures economic phenomena in a shared business environment. This approach conceptualizes business activities in terms of an enterprise information system. In REA framework [2], resources can increase or decreases through the exchange or conversion process.

According to REA framework, exchanges processes are those in which enterprise receives economic resources from economic agents and provides other resources in return. In the Sale & Distribution process, a sale is an exchange process [4]. It exchanges economic resource, for example, a product for another resource like cash. According to REA framework, conversion processes are those in which organization economic resources are consumed for the production of a new economic resource. In the Sale & Distribution process, distribution is an example of the conversion process [5]. It only modifies or changes a feature of the same resource e.g. the product's location etc.

From state-of-the-art (**Section II**), it can be analyzed that BPMN modeling is an emerging trend in the area of ERP System development. In ERP System, exchange and conversion processes are commonly required to be developed.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhangbing Zhou.

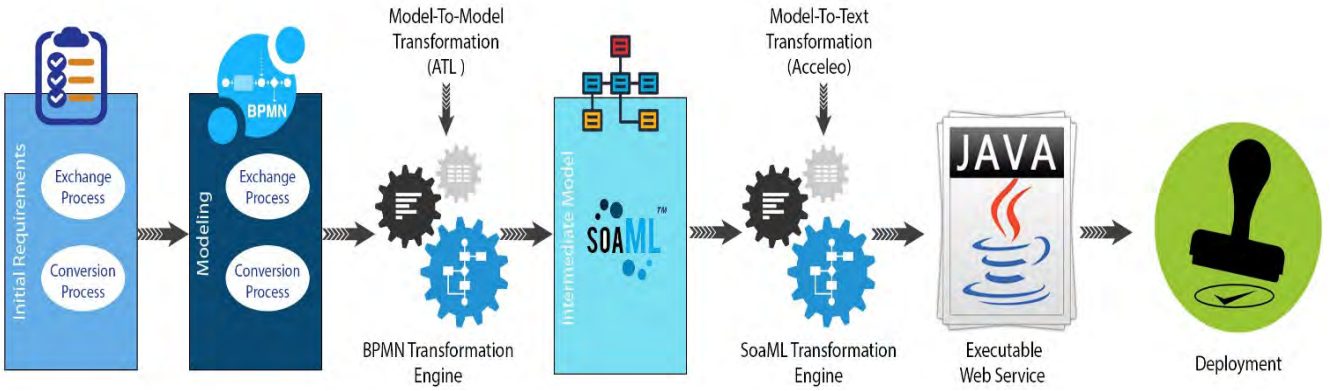


FIGURE 1. Research overview.

In this context, the implementation of web services in ERP systems is a complex task. To manage this, Business Process Model and Notation (BPMN) is frequently utilized to simplify the development of ERP applications. However, state-of-the-art BPMN approaches usually deal with the modeling of exchange processes without considering the conversion process. Furthermore, the model transformation solution to automatically generate web services from BPMN models is hard to find in the literature. However, there exist few studies where SoaML is used for web service specification of the exchange business process. In this regard, web service specification and web services generation for conversion business processes is not targeted yet.

Due to the growing importance and complexity of web services, manual development of services becomes a time, resource and cost consuming process. Identifying web services, which encapsulates the discrete functionality of business processes of the enterprise information system, has become increasingly challenging. Moreover, changes in business processes cause changes in the development of web services. These changes must be tested for error-free services. Additionally, there is no way to trace the code back to their specification, thereby making the system evolution costly and time-consuming. Therefore, there is a strong need to develop a framework that provides the modeling of both processes (exchange as well as conversion) through BPMN. Furthermore, it also provides a complete transformation engine to automatically generate the web services from the high-level BPMN models. Therefore, in this article, a complete framework is developed to achieve aforementioned gaps of exiting solutions. An Overview of research is shown in **Figure 1**. The major contributions of the article are as follows:

1. The development of approach to model both exchange as well as conversion processes in BPMN.
2. A complete open source **B**PMN to **W**eb services transformation (B2W) tool is implemented by utilizing the concepts of both Model-to-Model (M2M) and Model-to-Text (M2T) transformation approaches. Particularly, M2M approach is implemented to generate SoaML models from BPMN models. Furthermore, M2T approach is used to generate executable web services in Java language from SoaML models.
3. The proposed framework is validated through three benchmark case studies.

This paper is organized as: **Section II** provides the background of current research. The web services generation (B2W) approach is proposed in **Section III**. The details of B2W transformation engine are described in **Section IV**. The application of the proposed approach is demonstrated in **Section V** through three case studies. Significant aspects of B2W transformation engine are discussed in **Section VI.** Finally, the paper is concluded in **Section VII**.

II. PRELIMINARIES AND RELATED WORK

A. ERP SYSTEM AND PROCESS

Business processes are dealing with Enterprise Resource Planning (ERP). According to REA ontology, there are two types of ERP System processes i.e. Exchange and Conversion. Sales & distribution are exchange and conversion type business processes respectively. For example, distribution is a conversion process because it only changes a feature of the same resource e.g. the product's location etc.

B. BPMN FOR ERP SYSTEMS

Business Process Model and Notation (BPMN) [6] was standardized by OMG in March 2010. It acts as a supporting pillar for the business process modeling and implementation. The event-driven process value chain is also being used for modeling business requirement [28], [29]. It has widely been adopted since then. The essential part of BPMN is providing notations that are used for business process modeling and further execution and equally understandable by all business analyst and developers. BPMN is a mean of communication between different business users, customers, suppliers and business process implementer across the word [7]. Furthermore, it provides support for understanding the internal business process of an organization and models them in a graphical notation. BPMN also provides a mapping to an execution language of BPM Systems (WSBPEL) [8]. We target both exchange and conversion ERP processes using BPMN.

TABLE 1. Research gap.

REFERENCE #	MODELLING LANGUAGE	TOOL SUPPORT	FUNCTIONALITY ASPECTS	TARGETED PROCESS
DELGADO, ANDREA, ET AL. [12] - 2014	BPMN	PARTIAL	MODELLING	EXCHANGE
BOCCIARELLI, PAOLO, ET AL. [7] - 2015	BPMN	PARTIAL	MODELLING	EXCHANGE
CHAÂBANE, AYMAN, ET AL. [14] -2010	BPMN	PARTIAL	MODELLING	EXCHANGE
BOCCIARELLI, PAOLO, ET AL. [15] - 2010	BPMN	PARTIAL	MODELLING	EXCHANGE
BAZOUN, HASSAN, ET AL. [11] - 2014	BPMN	PARTIAL	MODELLING	EXCHANGE
DE CASTRO, VALERIA, ET AL. [5] - 2009	BPMN	PARTIAL	MODELLING	EXCHANGE
DELGADO, ANDREA, ET AL. [9] - 2011	BPMN, SOAML	N/A	MODELLING	EXCHANGE
BLAL, REDOUANE, ET AL. [4] - 2017	BPMN, SOAML	N/A	MODELLING	EXCHANGE

A brief description of some of the targeted constructs of BPMN is given in **Section III-B** because these constructs are mandatory for both exchange and conversion ERP processes modeling. From literature, we analyzed that BPMN does not cover conversion process modeling. We found only exchange BPMN processes such as Sales.

C. SOAML FOR ERP SYSTEMS

Service-oriented architecture Modeling Language (SoaML) [9] is standardize by Object Management Group (OMG) that provides a meta-model web services specification and a UML profile for the specification of services through SOA framework. It specifies the consumers and providers of services and defines the ports through which services are provided by a provider organization that is requested by consumers. A brief description of some of the targeted constructs of SoaML is mentioned in **Section III-C**. For ERP, exchange and conversion service extraction is not covered in previously done researches through the SoaML service model. So we targeted particular SoaML constructs that cover both types of services and generate the required SoaML service model for further service code generation.

D. MODEL-DRIVEN ARCHITECTURE (MDA)

Model Driven Architecture (MDA) [6], [7] is an OMG standard approach for modeling, development and implementation of the software. The main objective of MDA is to deliver instructions for software specification structure in the form of models. Business objects and application functionalities are being separated from platform technology is the major focus of MDA. Application's model becomes platform independent when designed under model-driven architecture using OMG standard modeling language e.g. UML in any platform that are Web Services, .NET, CORBA R, J2EE, and others [8]. These platform-independent models are then used to generate target artifacts by utilizing the concepts of model transformations [10] for documentation of business objectives and behaviors of the systems without bothering the technology specific implementation. Instead of separation of concern of both business objectives and system functionalities, they can evolve with a period. In this context, BPMN is the dynamic process modeling language, standardized by OMG in 2005 [7]. BPMN is used to model those business requirements in a flow chart type diagram that depict the functionalities and behavior of the business process of ERP Systems. In this context, the implementation of conversion and exchange processes in ERP systems is a complex task. To manage this, Business Process Modeling Notation (BPMN) is frequently utilized to simplify the implementation complexity of ERP applications under MDA technological approach.

E. RESEARCH GAP

In ERP System, exchange and conversion processes are commonly required to be developed. However, developing their specifications, and further developing their code are complex and time-consuming activities. **Table 1** depict the research gap. BPMN modeling is used for modeling of business processes in order to describe their business flows through a business pattern approach and partial tool support is available just for BPMN modeling [7], [12], [13]. In [15], the author performs business process simulation through model transformation. It is found that web service specification of the exchange business process through SoaML modeling language has been done [4]. Web service generation for exchange process is not covered in previous researches. Web Service specification and web services generation for conversion business processes is not targeted so far [11], [4].

In this context, the implementation of conversion and exchange processes in ERP systems is a complex task. However, state-of-the-art BPMN approaches usually deal considering the conversion process. Furthermore, the model transformation solution to automatically generate web services from BPMN models is hard to find in the literature. Therefore, there is a strong need to develop a framework that provides the modeling of both (exchange as well as conversion) processes through BPMN as well as to provide a complete transformation engine to automatically generate the web services from the high-level BPMN models.

The proposed work includes a fully automated approach to generate java web services code from exchange and conversion business processes that are modeled in BPMN. ERP System has two major business processes i.e. exchange and conversion. Transformation engine includes three types of transformation; BPMN Model transformation, XMI Model transformation and SoaML Model transformation. Web services generation framework provides the advantage of automated code generation of exchange and conversion business of ERP system.

In order to obtain the complete Java code for the web services for exchange and conversion processes, Both M2M and M2T transformation is applied on BPMN and SoaML model. First of All, Model-to-Model (M2M) transformation, using ATL transformation language [24], take place using BPMN and SoaML Meta-models that are discussed and shown in **Section III**. BPMN XMI of ERP targeted processes is taken as an Input model for M2M transformation. Mapping of BPMN XMI model based on BPMN Meta model is define for SoaML Meta model constructs. After the M2M transformation, SoaML XMI model obtained as an output of this transformation. Final artifact generated from M2T transformation on the SoaML model is a fully executable Java web services.

III. PROPOSED APPROACH

This section contains details of the proposed methodology. **Section A** discusses the targeted business processes, **Section B** discusses the targeted BPMN model constructs, **Section C** discusses the targeted SoaML model constructs and **Section D** provides a detailed proposed solution.

A. TARGETED BUSINESS PROCESSES

BPMN processes modeling is standardized a way to model the business activities of any organization. For detailed discussion and specification of business processes, we have to define particular concepts for business terms. For example, these terms include economic resources, agents, and economic events. Many business ontologies are used for business terms. REA [1], e3-Value [16] and Business Process Ontology (BPO) [3]. According to REA framework, key business abstractions contain business actors that are called business agents or economic agents, economic resources and events. These resources that are exchanged between them create a relationship between different agents [17]. In this research work, we chose REA ontology for business terms that are widely getting attention nowadays. However, our approach can be applied on any of the aforementioned business ontologies.

In REA framework, resources can increase or decreases through the exchange or conversion process [18], [19]. **Figure 2** shows both types of processes. The details of these processes are given in subsequent sections.

- ***Exchange Process***: According to REA framework, exchanges processes are those in which enterprise receives economic resources from economic agents and provides other resources in return. In the Sale & Distribution process, the sale is an exchange process. It exchanges economic resource, for example, product for another resource, cash.
- ***Conversion Process***: According to REA framework, conversion processes are those in which enterprise consumes economic resources in order to produce new and modified resources. In the Sale & Distribution process, distribution is a conversion process. It changes or modifies the only feature of the same resource, for example, the product's location.

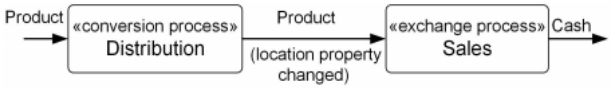


FIGURE 2. REA processes - exchange & conversion.

B. TARGETED BPMN CONSTRUCTS

BPMN meta-model [20] provides the semantics and structure of constructs. BPMN meta-model comes with two packages. Core package contains main BPMN concepts whereas the Extensions package contains some enhanced characteristics that make the application more interactive. The basic purpose of introducing extensions is to make the application more expressive, increase the readability and to make the elements less abstract. We have only targeted the core concepts because without these core constructs ERP processes will not be correctly modeled with service requirements offered by ERP systems. A brief description of some of the core concepts of BPMN is given as follows.

- **Swimlane:** Swimlane is used to group modeling elements by using any of the two elements. 1) Pools 2) Lanes.
- **Flow Objects:** In order to define the behavior of the business process, flow objects are the major elements used to define these behaviors. Three flow objects are used for this purpose. 1) Events 2) Activities 3) Gateways. The first element is ''Event'' that is happening of business processes behaviors. The second element of flow object is ''Activity'' that is a task that will be performed by business personnel working under any organization. The third element that is present in the table is ''Gateway'' that is used for controlling business processes. Join, fork, merge and branches will be under control of this element.
- **Connecting Objects**: Flow objects are connected by Connecting Objects. There are four connecting objects. 1) Sequence Flows that are used to define the flow of communication between different objects. 2) Message Flows define the message flow between flow objects or between swimlane and vice versa. 3) Associations define the relationship between flow objects. 4) Data Associations show the association between data objects and flow objects like events and tasks.

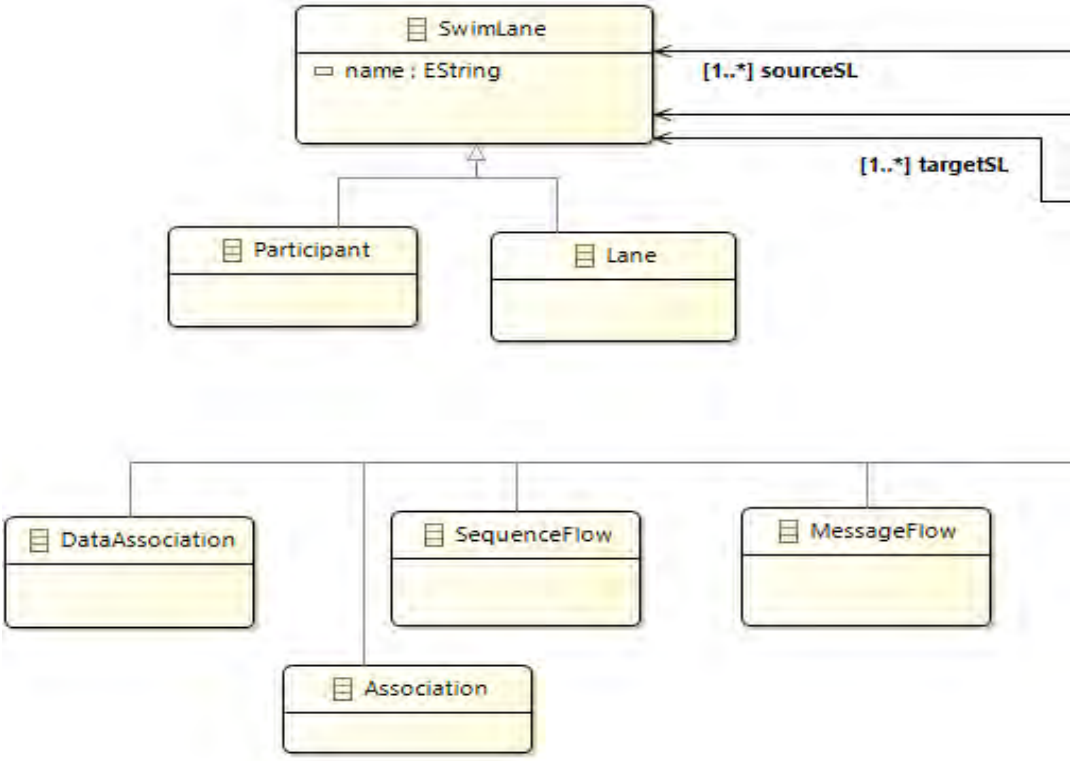


FIGURE 3. BPMN meta-model (diagram 1 of 3).

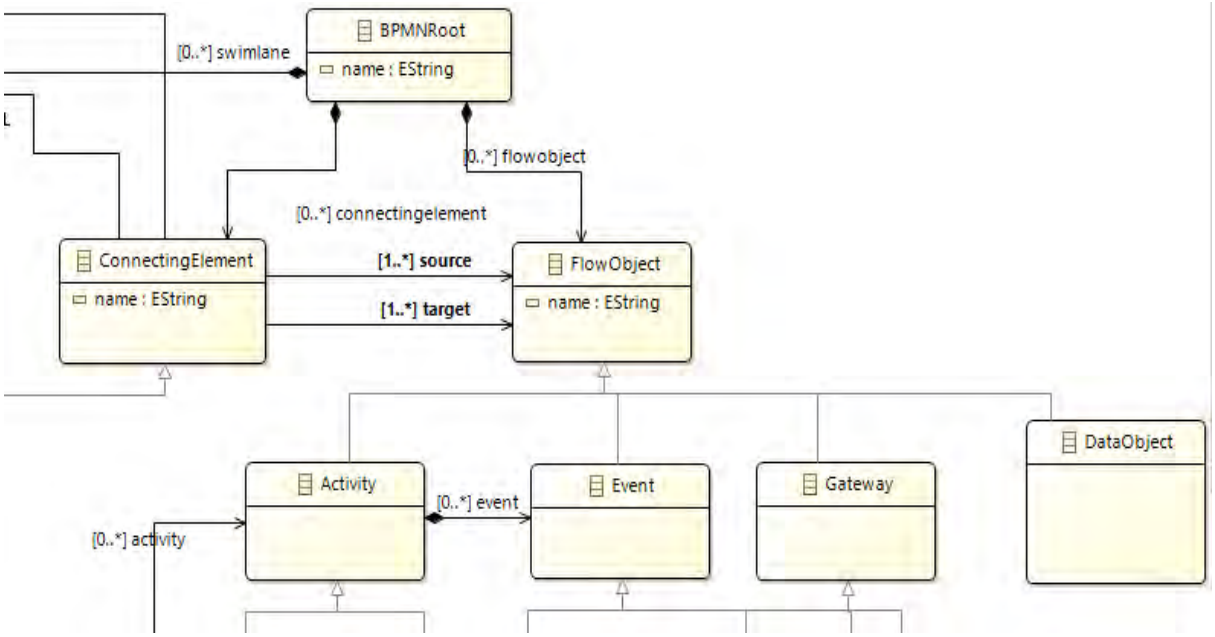


FIGURE 4. BPMN meta-model (diagram 2 of 3).

FIGURE 3, 4 & 5 shows the BPMN Metamodel of targeted constructs. Our designed Meta model covers all the core BPMN constructs that are essential for accomplishing proposed research objectives. Exchange and conversion services are achieved using this Metamodel. Due to the complexity of Metamodel, we divided it into 3 parts.

C. TARGETED SOAML CONSTRUCTS

Service-oriented architecture Modeling Language (SoaML) [21] is widely used for specification of web services from the business process value chain. One of the major advantages of SoaML is its wide scope. Many technology-specific tools are available to develop web

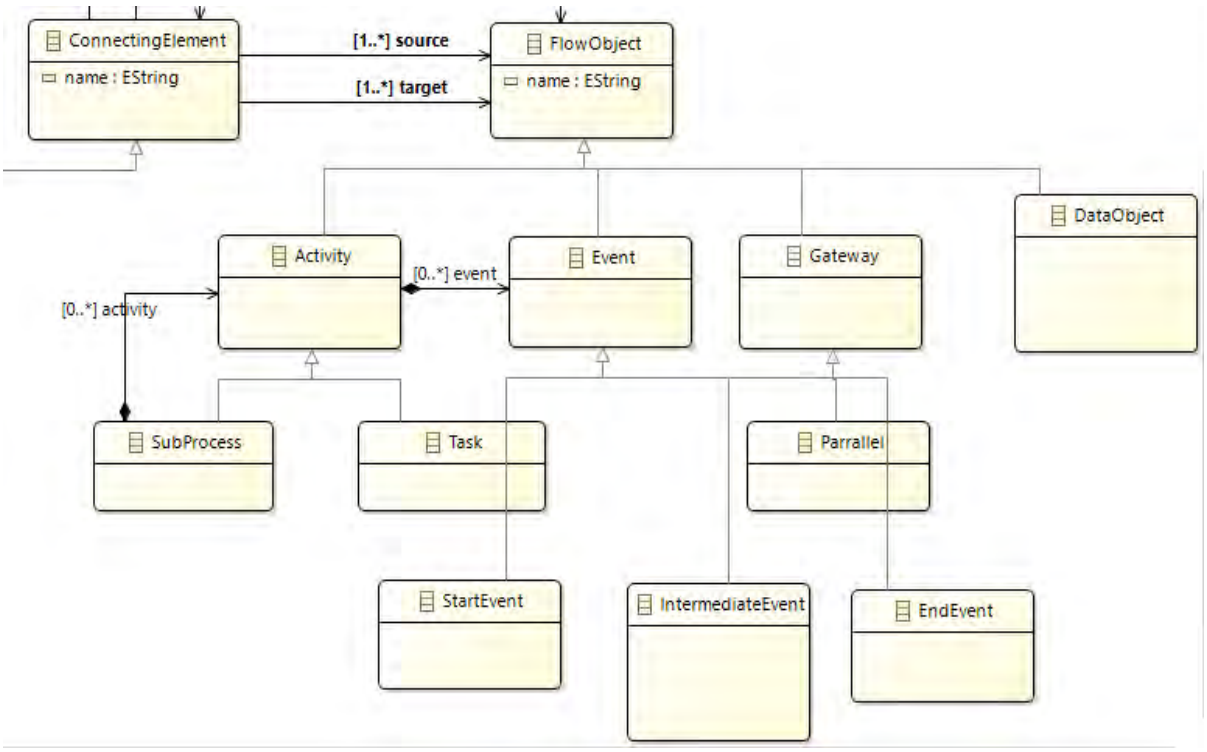


FIGURE 5. BPMN meta-model (diagram 3 of 3).

services [22] but these tools do not detail with high-level concepts that must be defined through service-oriented architecture. This architecture shows the services, its participants, and their relationship through which business values achieved. Due to hundreds or thousands of daily services provided by the organization, SoaML facilitates these process services. We targeted some constructs of SoaML to achieve our research purpose.

A brief description of some of the core concepts of SoaML is given as follows:

- <<**Services Architecture**>>**:** SOA high-level view is defined as the work done in a group by the participants to achieve share or common purpose of using and providing services. It is also depicting model architecture with their own behaviors. Service Architecture in inherited from Collaboration, which is also called BPMN Root.
- <<**Participant**>>**:** Service requesters and service providers are the participants of service architecture. It is extended from UML class.
- <<**Service Port**>>**:** Service typed is offered with a service interface through service port.
- <<**Request Port**>>**:** Service typed is used with a service interface through request port.
- <<**Service Contract**>>**:** Condition between users and providers for using the service is specified.
- <<**Message Type**>>**:** It is also extended from UML Class and is used to identify data objects that travel among participants of service architecture. It may be a data type or signal sent between service consumers and provide and act as a way of communication.
- <<**Connecting Elements**>>**:** These elements are used to show the sequence flow between different classes that can be provider, consumer, interface or any participant of service architecture. Connecting elements in SoaML interface diagram can be depicted through realization relationship or usage relationship. Both show the connection between service participants.
- <<**Service Interface**>>**:** Service Interface represents the connection point between the service provider and service consumer. Through this connection point, service participants interact with each other.

Figure 6 & 7 show the SoaML Metamodel of targeted constructs. Our designed Meta model covers the entire core SoaML constructs that are essential for accomplishing proposed research objectives. Exchange and conversion services are achieved using this Metamodel. Due to the complexity of Metamodel, we divided it into 2 parts.

A brief description of metal model constructs and their division for exchange and conversion services are mentioned below. This SoaML Metamodel is designed according to the identification of the service from BPMN Process Model. By joint implementation of Both BPMN Metamodel and SoaML Metamodel, we are able to get desired Service model that fulfill our ERP processes.

Constructs for Exchange SoaML service model are mention below:

- Service Interface
- Participant

Constructs for Conversion SoaML service model are mention below:

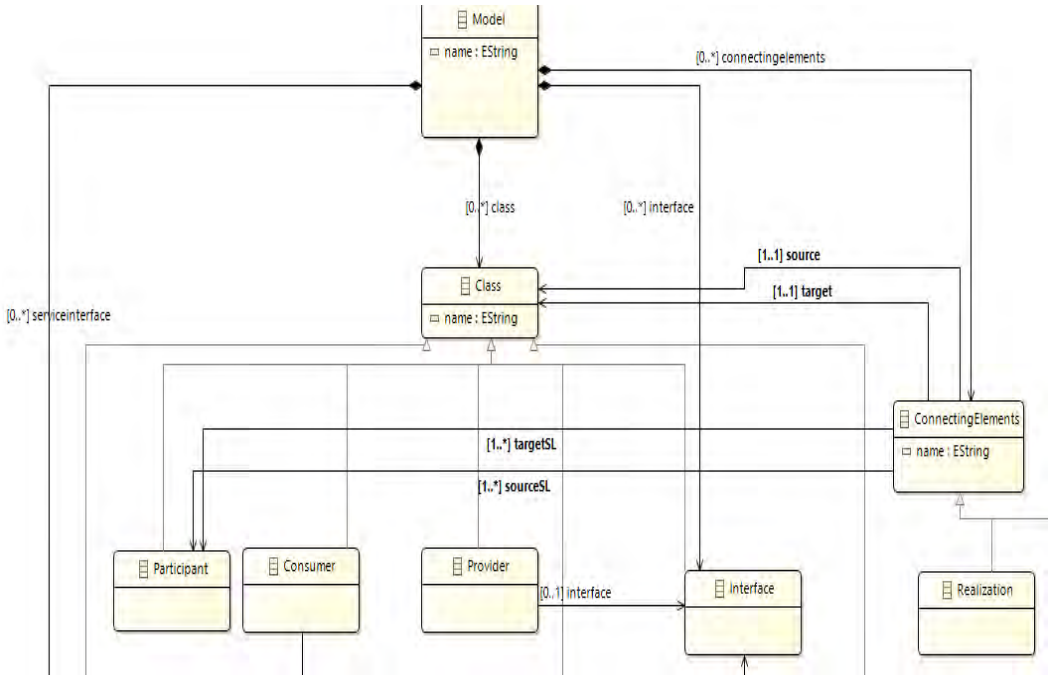


FIGURE 6. SoaML meta-model (diagram 1 of 2).

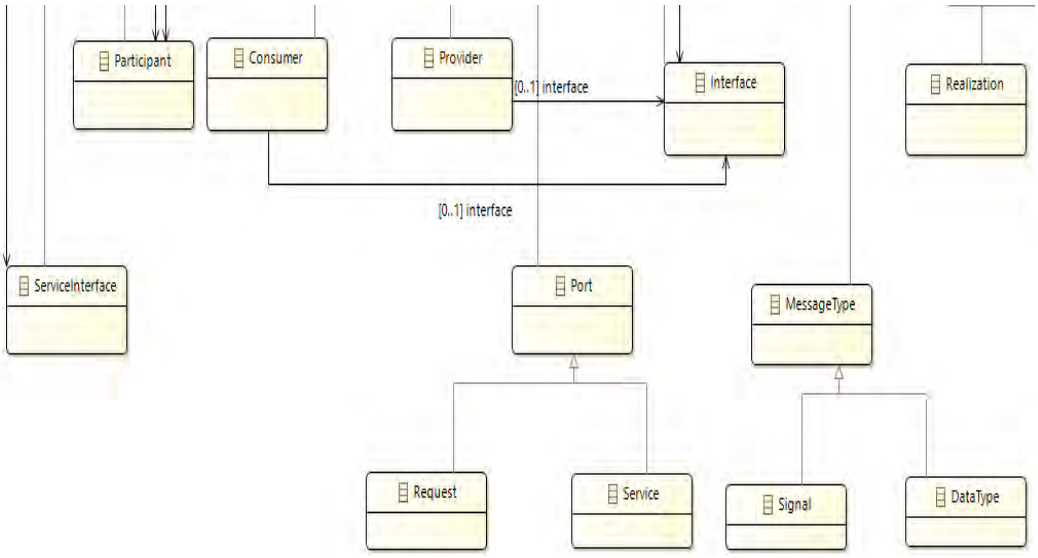


FIGURE 7. SoaML meta-model (diagram 2 of 2).

- Service Interface,
- Participant,
- Message Type
- Message Attachment
- Service Contract

These constructs are also used for exchange service modeling. Other constructs that are different in the Conversion processes SoaML model are; Message Type, Message Attachment and Service contract. Details are already placed in the above section where we define targeted SoaML model constructs.

D. PROPOSED SOLUTION

We have proposed an approach based on Model Driven Software Engineering (MDSE). MDSE involves transformations for obtaining the code or target model from the source model. Our approach used a BPMN model for exchange and conversion business processes of ERP system. SoaML model is used for web services specification of selected business processes. The approach takes both models as input and applies M2M and M2T transformations in order to obtain desired outcomes. The transformations result in automated code generation for web services. Two types of transformations can

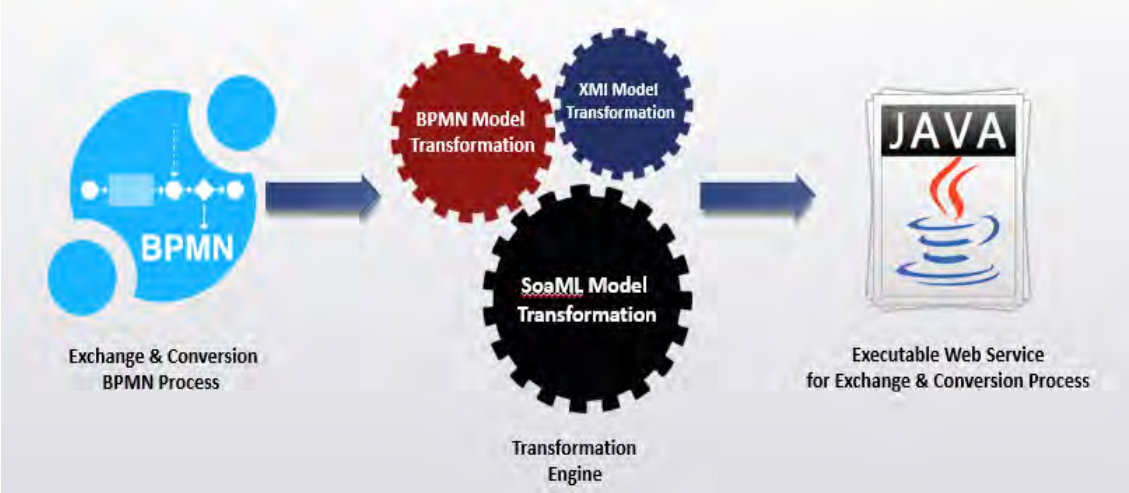


FIGURE 8. Proposed methodology.

be performed in MDSE. The first type of transformation is Model to Model (M2M) transformation [22] in which a target model is obtained from a source model. The second type of transformation is called Model to Text (M2T) transformation [23] in which code or text is generated from the target model. The source models can be of any design type i.e. UML model and BPMN core model. **Figure 8** shows our proposed methodology.

In order to obtain the complete Java code for the web services for exchange and conversion processes, Both M2M and M2T transformation is applied on BPMN and SoaML model. M2M transformation is required to get SoaML service Model and M2T transformation is required to get desired executable services from service Model. First of All, Model-to-Model (M2M) transformation, using ATL transformation language [24], take place using BPMN and SoaML Meta-models that are discussed in Section III-B and Section III-C. BPMN XMI of ERP targeted processes is taken as an Input model for M2M transformation. Mapping of BPMN XMI model based on BPMN Meta model is define for SoaML Meta model constructs. BPMN Meta model contains ''BPMN Root'' that is mapped with SoaML ''Model'' construct. Participant of the BPMN model is mapped to Participant of SoaML using BPMN XMI model. After the M2M transformation, SoaML XMI model obtained as an output of M2M transformation. Final artifact generated from M2T transformation on the SoaML model is a complete Java web services. SoaML model is used as an input model for M2T transformation, using Acceleo transformation language [25]. SoaML is the extension of UML2 means it is the profile of UML2. Mapping of service constructs to Java code has been done through standard language.

<<Service Interface>> of SoaML model is mapped to the interface of JAVA. <<Service Consumer>> and <<Service Provider>> are mapped to Java classes. Other service constructs have been mapped according to the transformation rules mention in **Section IV.**

The most important contribution made by the proposed approach is the automated generation of web services from conversion and exchange business processes of ERP system. Exchange and Conversion process BPMN models are transformed into a SoaML model, which contains the services, constructs as service architecture and the connections between them are also represented using connecting objects such as realization and usage for web service specification. The resulting model can be opened in SoaML papyrus tool [26] for verifying the output model. Then, this model is used to transform into Java code for web services; hence, providing a fully automated approach. Generated web services are verified using a web service test client application.

IV. IMPLEMENTATION

A. ARCHITECTURE

The architecture of our transformation engine is described in **Figure 9**. We have implemented a transformation engine based on model-based web services framework. This transformation engine fully automates the designing and coding phase of SDLC by providing web services code from BPMN exchange and conversion processes. The tool used for transformation of BPMN and SoaML models to web services in Eclipse ATL and Acceleo modules. Our transformation engine is composed of three main parts. The first part does BPMN modeling; second part does Model-To-Model transformation and other parts do Model-To-Code transformation. Major components of transformation engine are Main Interface, ATL and Acceleo Transformation and BPMN model, SoaML model and Java web services code for selected business processes of ERP system as transformation output artifacts. Details of functionality performed by each component are explained below.

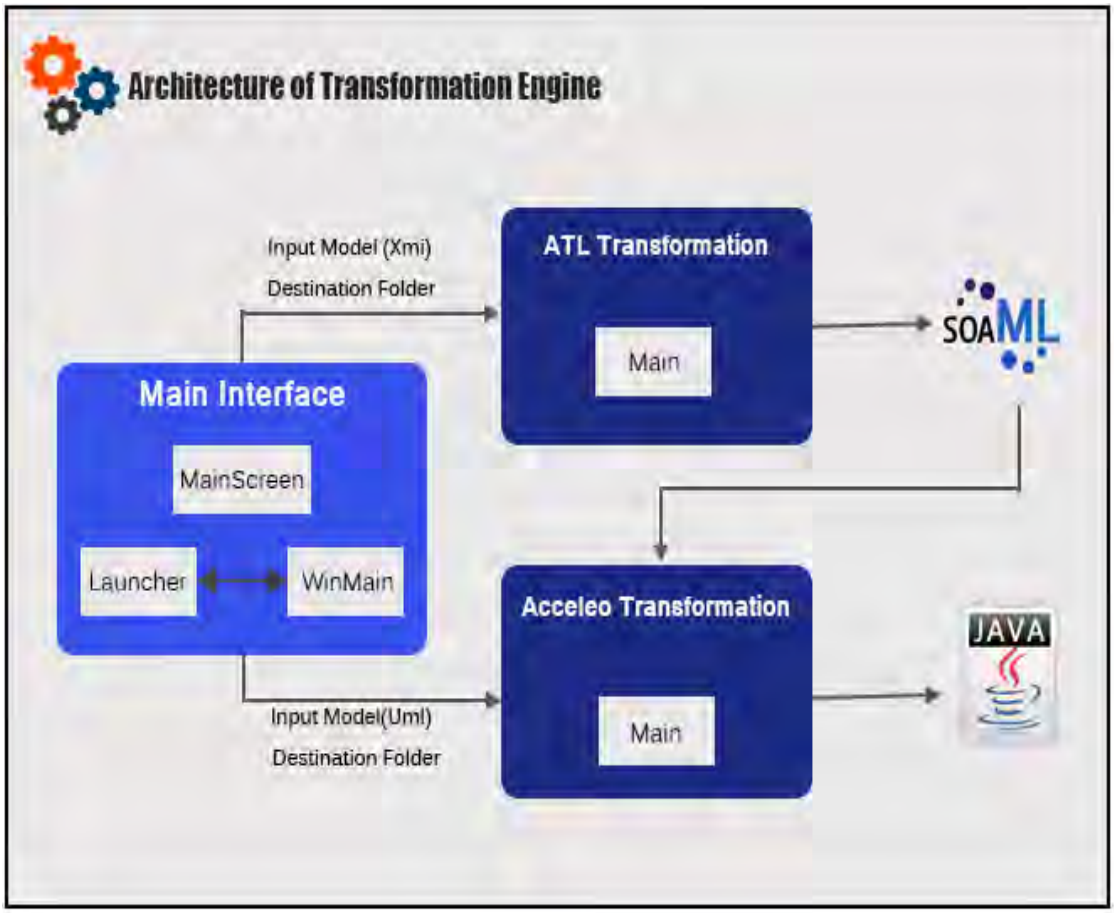


FIGURE 9. Transformation engine architecture.

1) MAIN INTERFACE

Main Interface component consists of three classes i.e. MainScreen, Launcher and WinMain. These three classes are used in the development of the graphical user interface of our tool. MainScreen is a class, which provides execution, and WinMain and Launcher contain its actual functionality. When the main screen of the tool is opened (**Figure 10**) it provides us with BPMN modeling, SoaML model generator and web service generator options. BPMN modeling open BPMN editor where BPMN process can model. Through the SoaML generator option, transformation engine takes BPMN XMI model and generate the SoaML model. For Web service generator option, transformation engine takes transformed BPMN XMI Model of exchange and conversion ERP processes (SoaML web service model) and model to code transformation applied on SoaML model and web services generated (**Figure 11**). If some error has occurred while transformation, a Console bar is shown. Reset button, empties all the fields i.e. input models' path, destination folder path, and status. Close button closes the interface from the screen.

2) ATL TRANSFORMATION

Eclipse ATL project takes BPMN XMI model as input and passes them to ATL Transformation. Foremost files

FIGURE 10. BPMN to web service transformation engine.

included in ATL transformation are BPMN.ecore and SoaML.ecore containing major constructs of BPMN, SoaML and transformation.atl containing atl transformation code. These two files work together to produce the SoaML model (.uml) which contains all the service related concepts captured from BPMN XMI model of exchange and

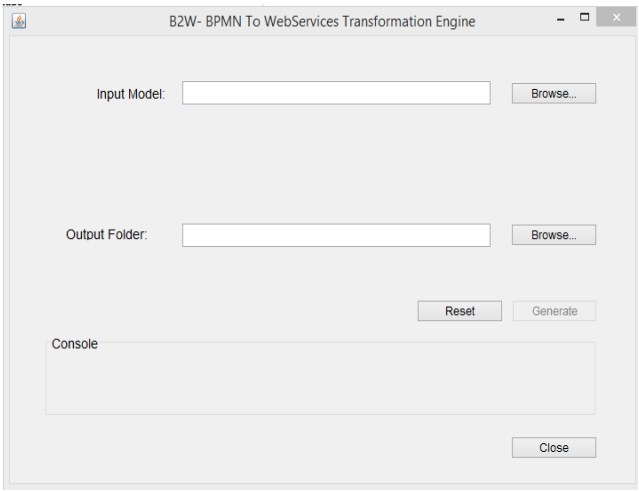


FIGURE 11. B2W-BPMN to web services transformation engine.

conversion processes. Transformation rules map exactly what required for service specification in the form of SoaML (.uml) model.

3) ACCELEO TRANSFORMATION

The Main interface takes the SoaML model as input and passes them to Accelo Transformation. Foremost files included in Accelo Transformation are Main.java containing java code for transformation and generate.mtl containing acceleo transformation code. These two files work together to produce a web service file (.java) which contains all the classes and interface related concepts captured from SoaML model. After a successful transformation, the transformation engine provides a SoaML model for service specification and then, Secondly, it provides executable web services java code as an output of this engine. SoaML model that is returned in the destination folder after successful model-to-model transformation is then used as an input model for the model to code transformation. Then after this transformation, web services executable code is then returned as output in the destination folder. All the mapping for the model to model and model to text transformation done exactly according to the transformation rules that are mentioned in Section IV-B. The executable B2W tool is publically available [41] along with user manual and sample case studies for further evaluation.

B. TRANSFORMATION RULES

In this section, we have described the transformation rules in detail, which we have defined in order to transform the BPMN model to BPMN XMI model, BPMN model to SoaML model than web service code artifacts. Section **1)** contains rules for BPMN Model to BPMN XMI Model. **Section 2)** defines rules for BPMN XMI Model to SoaML Model. **Section 3)** contains rules for SoaML model to Web Services.

1) BPMN MODEL TO BPMN XMI MODEL TRANSFORMATION RULES

SoaML model that is service specification model is obtained through BPMN model but M2M transformation needs BPMN XMI model as an input model. BPMN XMI model is also being used for business process specification in the form of XML Metadata Interchange (XMI). SOAML only takes BPMN in XMI model format. This specification is not specified yet in OMG BPMN specification [30]. Therefore, there is a dire need to convert BPMN Process model to XMI Metadata model for model-to-model transformation using ATL transformation language.

In this section, we have explained the mapping rules used for our transformation of BPMN Model to BPMN XMI Model. XMI is XML Meta-model Interchange [27] model just like XML. The transformation rules defined in **Table 2** are used to transform the concepts from the BPMN model to the BPMN XMI model.

Parent construct of BPMN model is mapped to parent instance of BPMN XMI model. Likewise, child of the BPMN model is mapped to child instance of BPMN XMI model. The root of BPMN Model is mapped to collaboration in BPMN XMI Model. Lane is mapped to Participant of BPMN XMI model. Flow Objects are mapped to Activity, Event, Gateways, and Data Associations in BPMN XMI. Activity is mapped to task or sub-process and its corresponding name. Likewise, Data Objects are mapped to Data objects of XMI Model. Connecting elements are mapped Sequence flow, Message flow, Data Association and Association in BPMN XMI.

2) BPMN XMI MODEL TO SOAML MODEL TRANSFORMATION RULES

Mapping rules used for transformation of BPMN model components into their respective SoaML model are provided in this section. Swimlane has two extension types i.e. Participant and Lane. The transformation rules defined in **Table 3** are used to transform the web services components from BPMN XMI model to SoaML model. We have not included other components of BPMN model for transformation because they did not capture the SoaML model details.

BPMN *CompleteModel* of type *Package* is mapped to its respective *Model* in the SoaML model. Each *Process and Lane* are mapped to SoaML Model *Participants*. *The activity* of BPMN is mapped to *Service Port* or can be mapped to *Request Port* or with *Class* according to the model situation. *Service Task (Provider)* is mapped to *Service Port* or with *Class. Task Consumer* is mapped to *Request Port* or with *Class* according to the case study under observation. *Collaboration* is mapped to *Service Model. Collaboration Participant* is mapped to *Service Class. The message* is mapped to *Message Type class* of SoaML model . *MessageRef is* mapped to the *parameter (In/Out). Data Object* is mapped to *Class. MessageFlow* is connecting the element of the BPMN

TABLE 2. Transformation rules for BPMN Model to BPMN XMI Model.

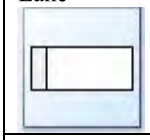
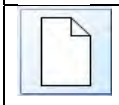
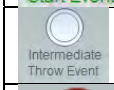
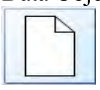
BPMN Model	Corresponding BPMN XMI Model	Transformation Description
Root	Collaboration	Root →Collaboration Root-name →Collaboration-name
Lane 	Participant	Lane →Participant Lane-name →Participant-name
Flow Object 	Activity	Flow Object →Activity Flow Object-name →Activity-name
	Event	Flow Object →Event Flow Object-name →Event-name
	Gateway	Flow Object →Gateway Flow Object-name →Gateway-name
	Data Object	Flow Object → Data Object Flow Object-name → Data Object -name
Activity 	Task	Activity →Task Activity -name → Task -name
	Sub-Process	Activity → Sub-Process Activity -name → Sub-Process- name
Event 	Start-Event	Event →Start-Event Event-name →Start-name
	Intermediate-Event	Event → Intermediate-Event Event -name → Intermediate -name
	End-Event	Event→End-Event Event-name →End-name
Data Object 	Data Object	Data Object → Data Object Data Object -name → Data Object -name
Connecting Element 	Sequence Flow	Connecting Element → Sequence Flow Connecting Element-name → Sequence Flow-name
	Message Flow	Connecting Element → Message Flow Connecting Element-name → Message Flow -name
	Data Association	Connecting Element Data Association Connecting Element-name → Data Association - name
	Association	Connecting Element → Data Association Connecting Element-name → Data Association - name

TABLE 3. Transformation rules for BPMN XMI model to SoaML model.

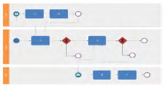
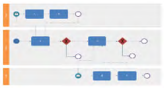
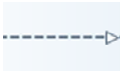
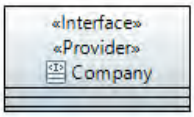
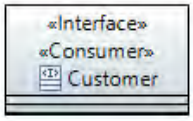
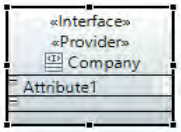
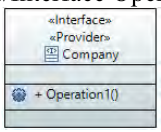
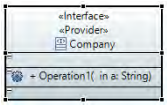
BPMN Model	Corresponding SoaML Construct	Transformation Description
Definition (Complete Model)	Model	BPMN Model definition is mapped to the SoaML Model definition BPMN Model--name → SoaML Model--name
Process/ Participant 	Consumer/Provider	Process is mapped to service Consumer/Provider Process—name → Consumer/Provider - name Process—attribute → Consumer/Provider -- attribute Process – operation → Consumer/Provider -- operation
Lane/Participant 	Consumer/Provider	Lane is mapped to service Consumer/Provider Lane—name → Consumer/Provider -- name Lane —attribute → Consumer/Provider -- attribute Lane – operation → Consumer/Provider -- operation
Activity 	Service Port	Activity is mapped to Service Port Activity —name →Service Port-- name
	Request Port	Activity is mapped to Request Port Activity —name →Service Port-- name
	Class	Activity is mapped to Class Activity—name →Participant -- name
Service Task (Provider) 	Service Port	Service Task is mapped to Service Port Service Task—name →Service Port-- name
	Class	Service Task is mapped to Class Service Task—name →Class -- name
Task (Consumer) 	Request Port	Task is mapped to Request Port Task—name →Service Port-- name
	Class	The task is mapped to Class Task—name →Class -- name
Collaboration 	Services Model	Collaboration is mapped to Service Model Collaboration —name → Model -- name
Collaboration Participant 	Model	Collaboration Participant is mapped to Model Class Collaboration Participant —name → Model Class-- name
Message	Message Type	Message is mapped to Message Type Class Message —name → Message Type -- name
		
MessageRef In/Out	Parameter In/Out	MessageRef is mapped to Parameter(In/Out) MessageRef —name → Parameter -- name
Data Object 	Class	Data Object is mapped to Class Data Object—name → Class-- name
Message Flow (SourceRef=Task, TargetRef=ServiceTask) 	Message Type	Message Flow is mapped to Message Type Message Flow —name → Message Type -- name
	Interface	Message Flow is mapped to Interface Message Flow —name → Interface-- name
	Operation	Message Flow is mapped to Operation Message Flow —name → Operation -- name
	Parameter In/Out	Message Flow is mapped to Parameter(In/Out) Message Flow —name → Parameter -- name
	ServiceContract	Message Flow is mapped to ServiceContract Message Flow —name → ServiceContract -- name

TABLE 4. Transformation rules for SoaML <<Service Interface>> to the Web Service Code.

SoaML Model	Corresponding Java Code	Transformation Description
<<Interface>> << Provider>> {Class} 	Code contains Java Interface for provider Stereotype	*Interface* → Interface name with Visibility mode.
<<Interface>> << Consumer>> {Class} 	Code contains Java Interface for consumer Stereotype	*Interface* → Interface name with Visibility mode.
Class/Interface Attribute 	Code contains Java Attribute or Property	*Attribute* → Attribute name with data type and visibility mode.
Class/Interface Operation 	Code contains Java Opertion	*Operation* → Operation name with return type and visibility mode.
Class/Interface Operation Parameter 	Code contains Java Opertion with parameter	*Operation Parameter* → Parameter name + input type

model that is either mapped to *Message Type class* of SoaML model or *Interface* or *Operation*, *Parameter (In/Out)* and *ServiceContract*.

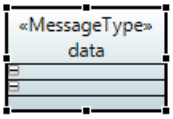
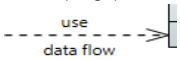
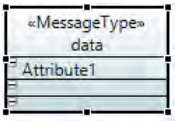
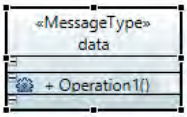
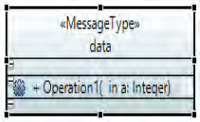
3) SOAML MODEL TO WEB SERVICE CODE TRANSFORMATION RULES

In this section, we have explained the mapping rules used for our transformation of SoaML model into the executable Java web service code. The transformation rules defined in **Table 4** are used to transform the concepts from SoaML

<<Service Interface>>model to the Web Service code. **Table 5** is used to transform the concepts from SoaML<<Message Type>> Model to Web service code. Both tables show the transformation of SoaML model constructs and produce Java code according to the input models. We have not included other components of SoaML model for transformation and hence their rules are not included because they did not lie under our area of focus. We have described the details of the applicability of these rules in **Section V.** Three case studies have been used to verify the transformation rules mentioned above.

<<ServiceInterface>> is the SoaML model construct for modeling services. The interface is used to request or provide services through Request or Service Port, respectively. <<Interface>><<Provider>> is mapped to Java Interface with its visibility mode. <<Interface>><<Consumer>> is mapped to Java Interface with its visibility mode. Class/Interface Attribute is mapped to Class Attributes with its visibility mode. Class/Interface Operation is mapped to Class Operation with its return type and visibility mode. Class/Interface Operation Parameter is mapped to Class Operation Parameter with its name and parameter name. <<*MessageType*>> is the SoaML model construct for modeling services message data. MessageType is used to send the request of service from service

TABLE 5. Transformation rules for SoaML <<MessageType>> to the Web Service Code.

SoaML Model	Corresponding Java Code	Transformation Description
<<Message Type>> {Class} 	Code contains Java Class for Message Type Stereotype	*Message Type* → Java Class name with Visibility mode. *Message Type* → Java Data Type name with Visibility mode.
Usage {Edge} 	Code contains Parameter for Usage Edge	*Usage* → usage name with data values as operation parameters.
<<Message Type>> Attribute 	Code contains Java Attribute or Property	*Attribute* → Attribute name with data type and visibility mode.
<<Message Type>> Operation 	Code contains Java Operation	*Operation* → Operation name with a return type and visibility mode.
<<Message Type>> Operation Parameter 	Code contains Java Operation with parameter	*Operation Parameter* → Parameter name + input type

consumers and send confirmation message form service provider after successful completion of service request. *<<MessageType>>* is mapped to either Java Class or Java *Data Type* with its visibility mode. *Use* connecting element is mapped to *Data Parameters* with its visibility mode. *<<MessageType>>Attribute* is mapped to *Class Attributes* with its visibility mode. *<<MessageType>>Operation* is mapped to *Class Operation* with its return type and visibility mode. *<<MessageType>>Operation Parameter* is mapped to *Class Operation Parameter* with its name and parameter name.

V. VALIDATION

In this section, the applicability and validity of our proposed approach are presented with the help of three detailed case studies. Details and validation of Amazon.com case study are given in **Section A**. Financial loan system case study is mentioned in **Section B** and **Section C** contains Hiring process case study.

A. AMAZON.COM CASE STUDY

Amazon.com case study has been explained and validated using four sections. Amazon.com contains both types of ERP processes that are exchanged (sale) and conversion (distribution). **Section 1)** covers the requirement specification for Amazon.com BPMN model. **Section 2)** presents the BPMN modeling for this case study in Eclipse editor using its BPMN2 plugin. **Section 3)** covers the equivalent BPMN XMI Model. **Section 4)** presents the transformation of BPMN XMI model to code generation. Lastly, verification of the case study modeled in BPMN has been provided in **Section 4)**.

1) REQUIREMENT SPECIFICATION

This section describes the modeling of a simple case of Amazon.com case study where a customer can order a book from Amazon.com order service and payment will be done through credit card.

- ***COLLABORATION PROCESS:*** BPMN model contains the collaboration process in which two or more

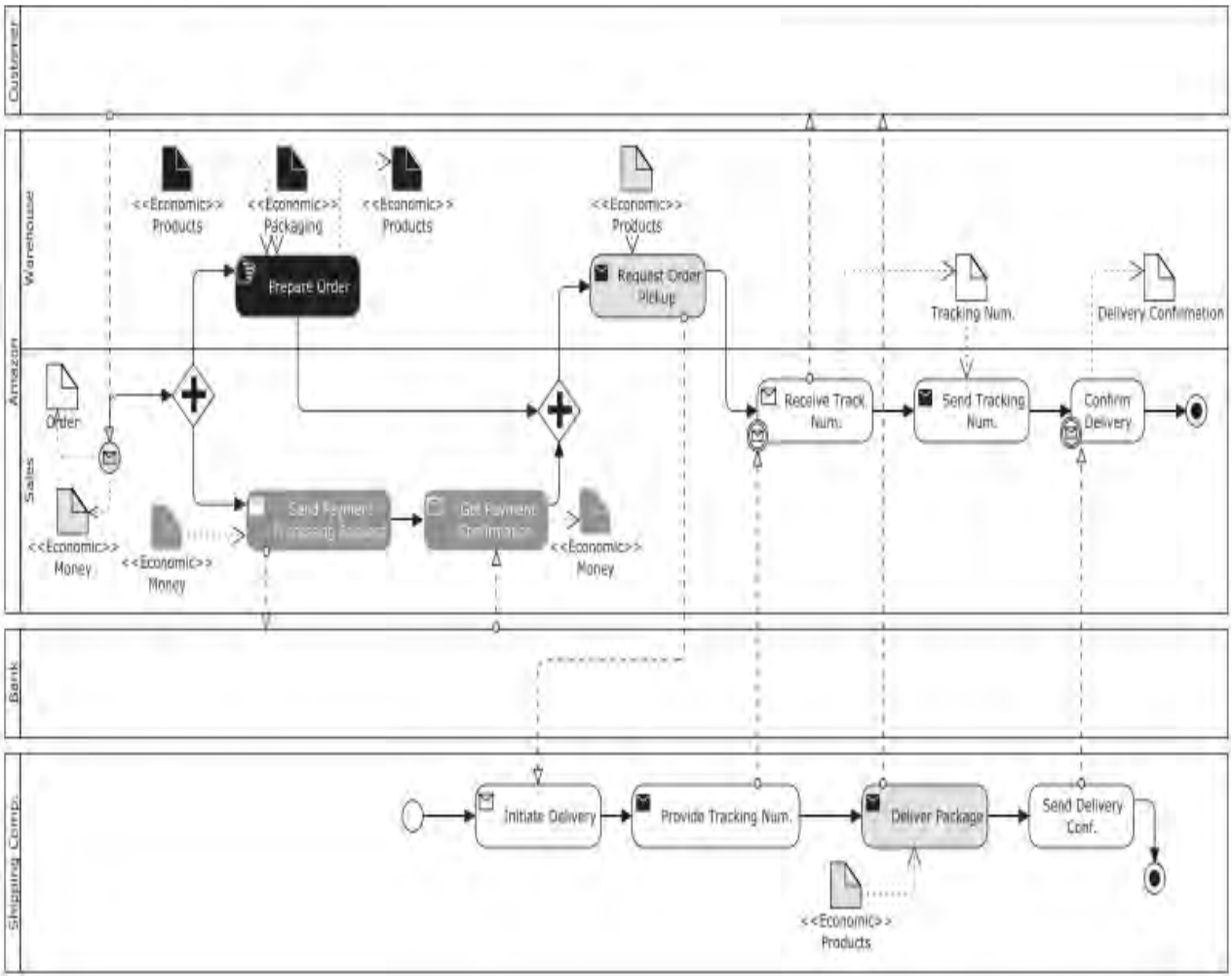


FIGURE 12. Amazon BPMN model.

participants can interact with each other using message conversation. The flow of conversation between processes either can reach to process boundary or within the boundary of other process or interact any event or activity of the process.

- ***PARTICIPANT:*** BPMN model contains participant of business processes that collaborate with each other in order to fulfill the goal of the business.
- ***START EVENT:*** There is a start event in any process, which initiate the whole process. After this event, a process running in its defined activity flow.
- ***END EVENT:*** Business process model contains the end event in order to complete the process. End event shows the finishing line of the business process flow.
- ***TASK:*** BPMN Model contains activities or tasks that are the functions of the process, which should be performed in order to complete the whole process. These are the set of activities, which perform business daily routine to provide business participants with a complete functionality.
- ***CONNECTING OBJECTS:*** Business process model contains connecting objects that define the flow of the business process. It should contain the source reference task and target reference task. Connecting objects either be sequence flow or message flow or data association or simple association relationship between source and target objects.
- ***DATA ITEM:*** the Business process may contain data item that is passed between tasks for successful completion of the task. For example, message, input or output parameters are the data items that are passed during process execution.

2) MODELLING

In **Figure 12,** Amazon BPMN model is designed in Bizagi BPMN Editor. Requirement Specifications is defined in **Section 1)** lead us directly to the modeling of the BPMN model.

The customer initiates Amazon procurement process when he or she selects product online from Amazon.com and sends

an order request by clicking the order button. Amazon sales department get the customer order and prepare product by giving the responsibility to the warehouse. Here, the sale process is the exchange process of the ERP system. After enclosing the product in shipping box and shipping address labeled on the box. The shipping process is an example of conversion business process of ERP. At the same time, online payment is processed via a customer credit card. Both of these activities of order and payment are taking place in parallel, the product is shipped to the customer, and a tracking number is provided to the customer. Delivery confirmation sent to Amazon by shipping company after a parcel is delivered to customer and process ends with successful order completion.

3) EQUIVALENT BPMN XMI MODEL

Amazon BPMN process is converted into equivalent XMI model. XMI model is XML Metadata Interchange and just like XML. Here, BPMN XMI model is used as an input model for the model to model transformation in order to get the SoaML model as output model. **Figure 13** shows the equivalent BPMN XMI model that is mapped to the BPMN process.

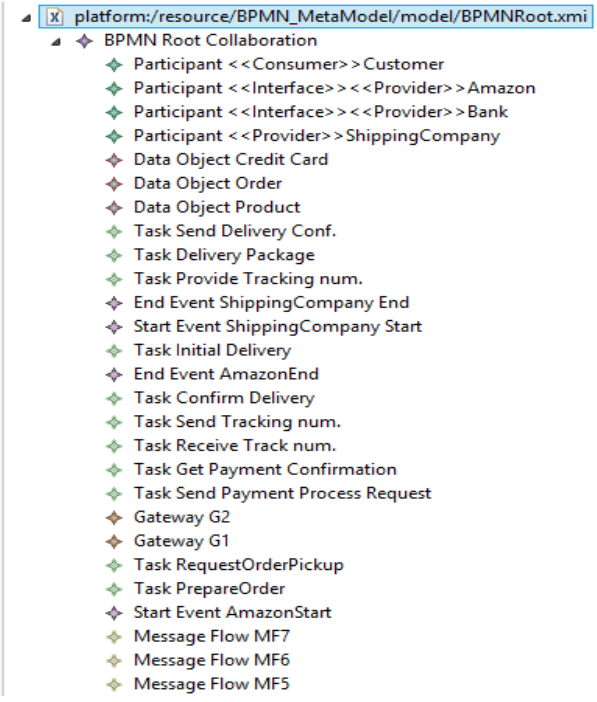


FIGURE 13. Amazon.com BPMN XMI model.

4) B2W TRANSFORMATION ENGINE

FIGURE 14 shows the BPMN to Web services (B2W) Transformation Engine. BPMN Generator is used for BPMN modeling of Amazon process. SoaML Generator is used to generate the SoaML model of Amazon. Through Web service Generator, Web services generated. For Web service

FIGURE 14. B2W transformation engine.

generation, SoaML model of Amazon.com is given as input (**Figure 15, 16**).

We have provided a folder on desktop as a destination. On clicking the Generate button, Web services for Amazon process are generated in the destination folder. Then these generated web services Java codes are used in standalone deployment web project and integrate it with web services through Tomcat Server (**Figure 18**). Their following artifacts are produced through B2W tool:

- BPMN model editor modeled the Amazon process. This model is the basic input of our transformation engine.
- SoaML generator generates SoaML model with (.uml) extension. SoaML model contains a service specification of the Amazon process. This model contains all the service specifications of Amazon.com BPMN process as services and provides interfaces and service participants against their particular mapping from BPMN model. SoaML model is the result of model-to-model transformation through ATL transformation.
- Web service Generator generates Java code for web services that are extracted through the SoaML model. Generated files in the destination folder are shown in **Figure 17.**

The executable B2W tool is publically available [41] along with user manual and Amazon sample case study for further evaluation.

B. FINANCIAL LOAN SYSTEM CASE STUDY

The financial loan case study has been explained and validated using four sections. Financial loan system contains exchange and conversion business process of ERP system.

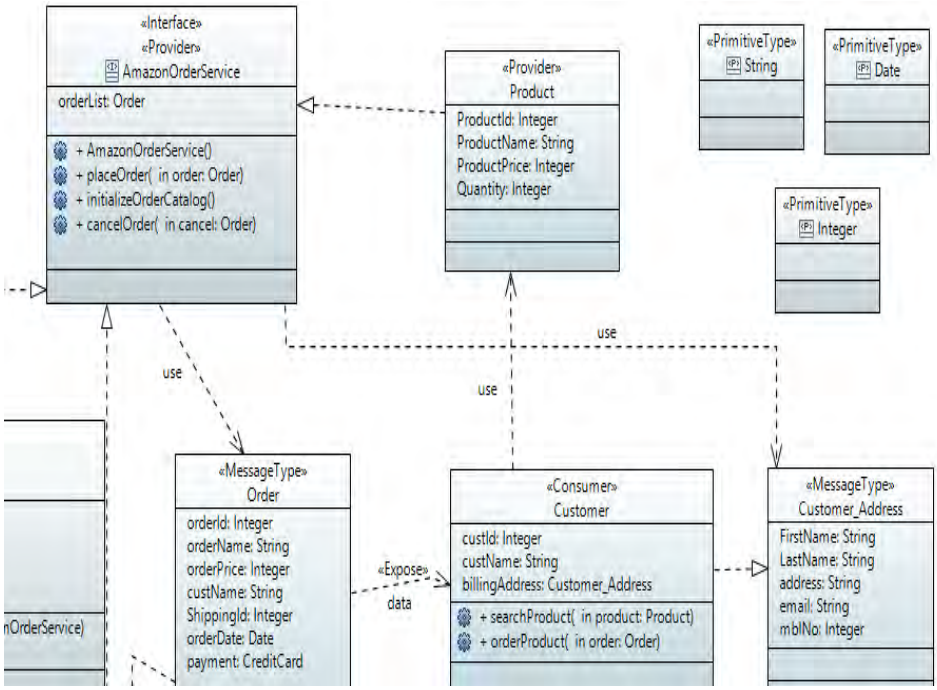


FIGURE 15. Amazon.com SoaML model (diagram 1 of 2).

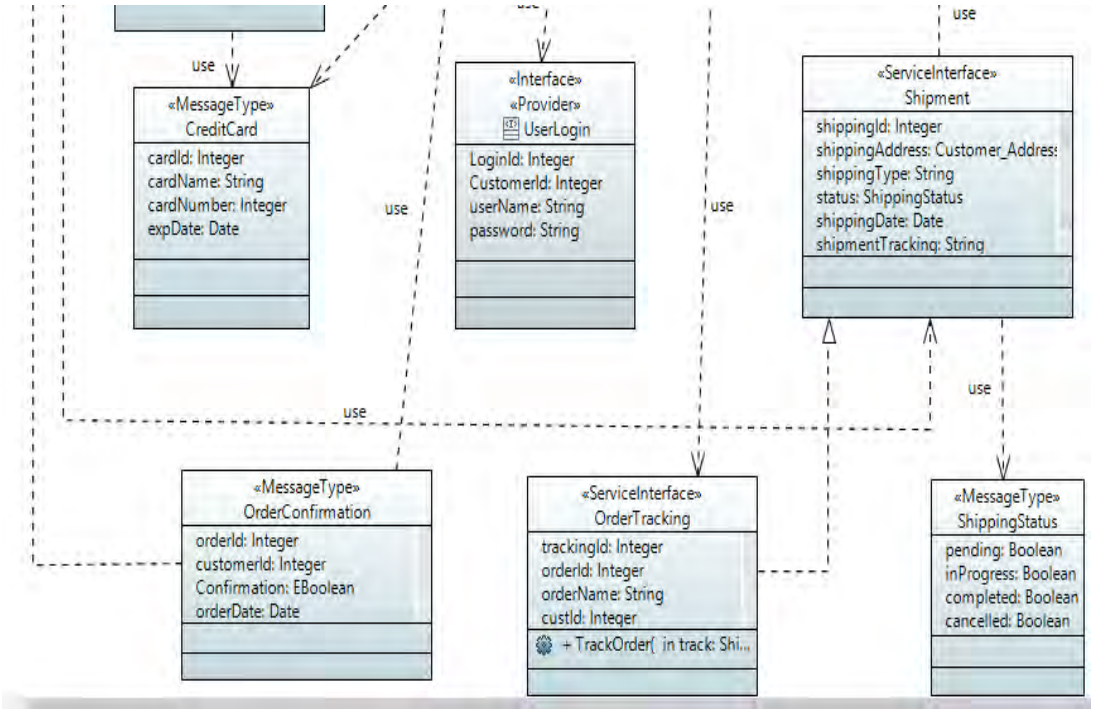


FIGURE 16. Amazon.com SoaML model (diagram 2 of 2).

Section 1) covers the requirement specification for the financial loan BPMN process. **Section 2)** presents the BPMN modeling for this case study in Eclipse editor using its BPMN2 plugin. **Section 3)** covers the equivalent BPMN XMI Model. **Section 4)** presents the transformation of BPMN XMI model to code generation. Lastly, verification of the case study modeled in BPMN has been provided in **Section 4).**

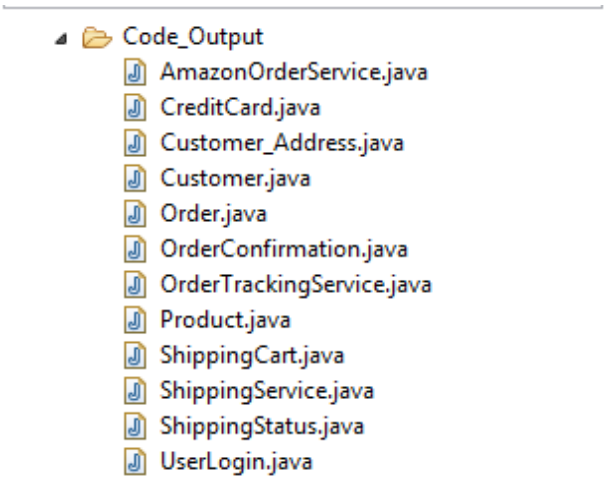


FIGURE 17. Web services code files.

1) REQUIREMENT SPECIFICATION

This section describes the modeling of a simple case of financial loan case study where a customer can send loan request and request will be fulfilled through acceptance of loan. Brief description of each requirement is mention in Amazon case study (**Section A-1**).

2) MODELLING

In **Figure 19**, the Financial BPMN model is designed in Bizagi BPMN Editor. It is mandatory to use BPMN constructs to model business process constructs because of the wide scope of BPMN. Requirement Specifications is defined in **Section 1)** lead us directly to the modeling of the BPMN model.

Financial loan process initiated with the need for a loan from company side for due to some financial matters. The quotation process is conversion business process of an ERP system. Loan request sent to the bank through a request for quotation event. Bank accept request after reviewing request for quotation. Then, Quotation is created from the bank and send to the company. Quotation process started in company and partner selection done by the company. Loan contract signed by company and agreement process is completed between bank and company. Cash disbursement is exchange business process. Cash disbursement process takes place from the bank side and loan is given to the company for a given time limit. The company takes a loan from the bank and agreed to return that loan with interest within the specified returning time and process is successfully completed from both sides, Company and bank.

3) EQUIVALENT BPMN XMI MODEL

Financial loan BPMN model is also converted into equivalent XMI model. XMI model is XML Metadata Interchange and just like XML. In our research work, BPMN XMI model is used as an input model for the model to model transformation in order to get the SoaML model as output model. **Figure 20** shows the equivalent BPMN XMI model that is mapped to the BPMN model.

4) B2W TRANSFORMATION ENGINE

FIGURE 21 shows the generated SoaML model for financial loan services. This model contains all the service constructs that are necessary for automatic executable web services generation. This SoaML model acts as input in M2T transformation. Details of all the artifacts generated by B2W

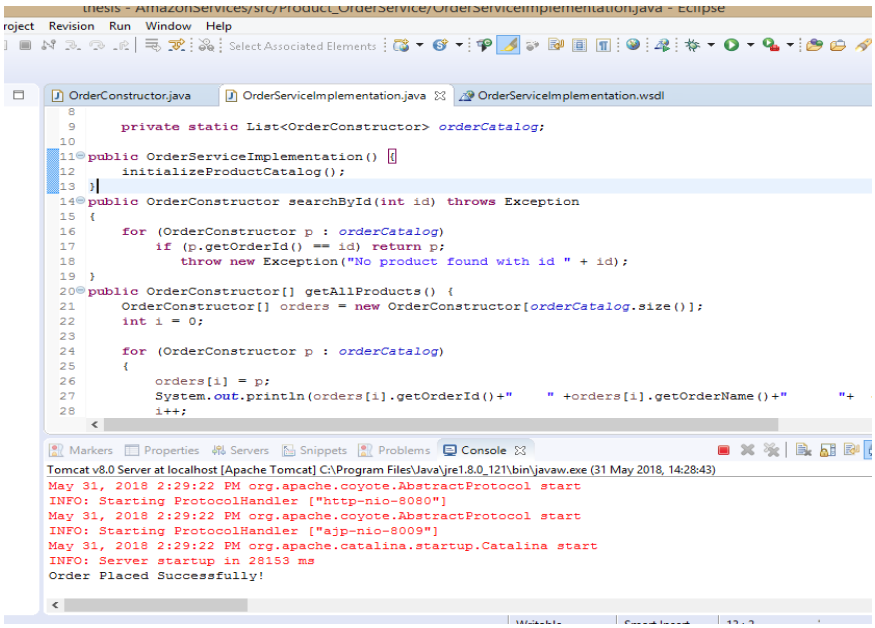


FIGURE 18. Order web service deployment.

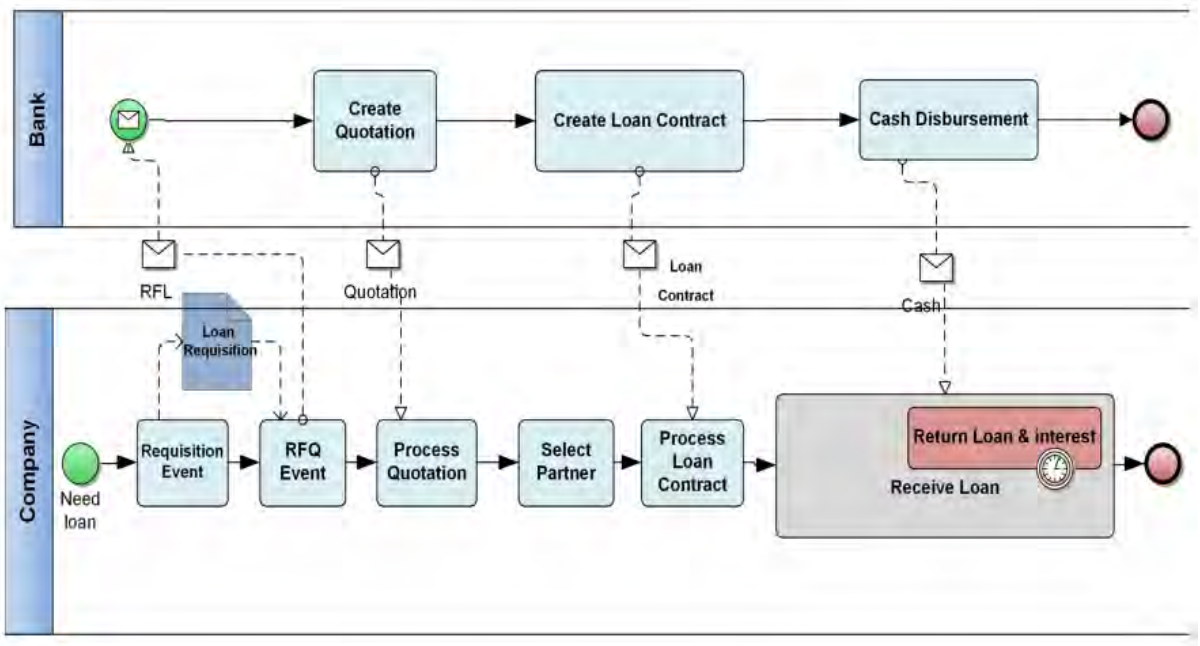


FIGURE 19. Complete financial loan BPMN model.

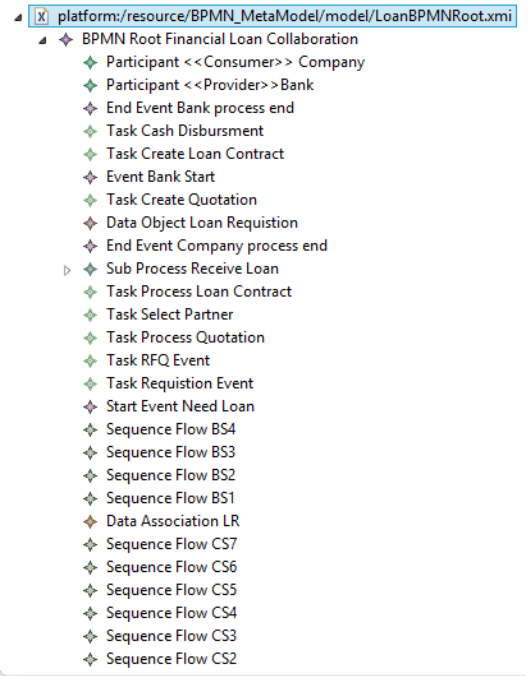


FIGURE 20. Financial loan BPMN XMI model.

transformation engine are already described in the Amazon case study (**Section V-A-4).**

C. HIRING PROCESS CASE STUDY

Hiring Process case study has been explained and validated using four sections. Hiring system contains exchange and conversion business process of ERP system. **Section 1**) covers the requirement specification for Hiring the BPMN process. **Section 2**) presents the BPMN modeling for this case study in Eclipse editor using its BPMN2 plugin. **Section 3)** covers the equivalent BPMN XMI Model. **Section 4**) presents the transformation of BPMN XMI model to code generation. Lastly, verification of the case study modeled in BPMN has been provided in **Section 4**).

1) REQUIREMENT SPECIFICATION

This section describes the modeling of a simple case of hiring case study where candidate can send his/her resume for a particular job opening. After the interview, Particular candidate hires for the job and vacant space is filled. Brief description of each requirement is mention in Amazon case study (Section A-1).

2) MODELLING

In **Figure 25**, Hiring BPMN model is designed in Bizagi BPMN Editor. It is mandatory to use BPMN constructs to model business process constructs because of the wide scope of BPMN. Requirement Specifications is defined in Section 1) lead us directly to the modeling of the BPMN model.

The first step in any hiring process is to determine whether the position is, in fact, needed in your company. If any position is vacant then the company create a Job opening for that particular seat. Resume of Candidates is required for the hiring process. Candidates send their resume to company HR personnel. Selected Candidates called for interview. After the interview, Particular candidate hires for job and job is offered to a particular candidate. After Job acceptance, Company will give Contract to the hired candidate.

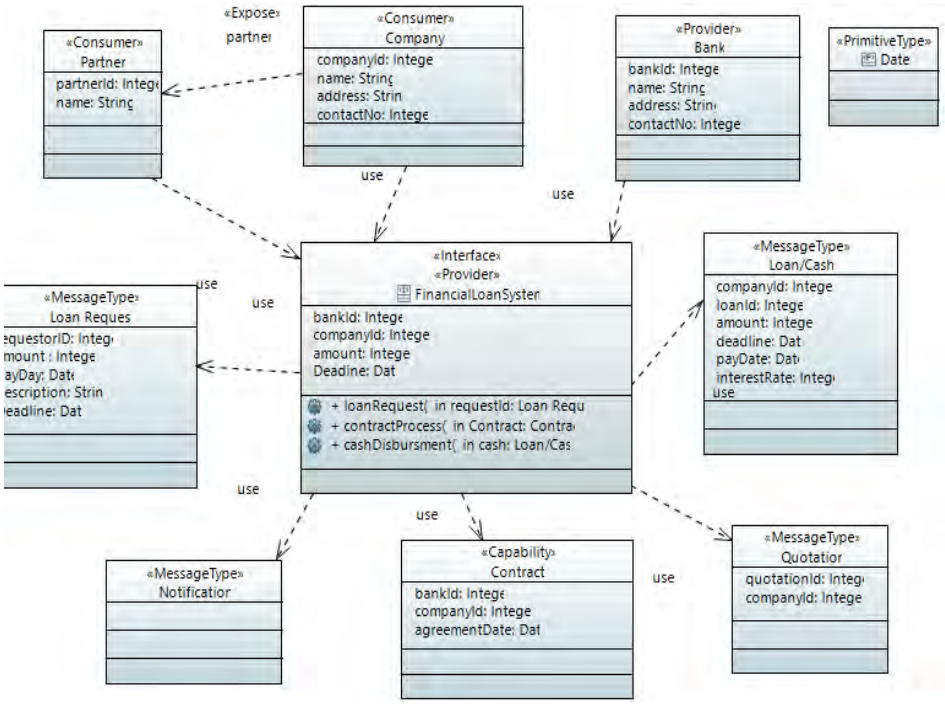


FIGURE 21. Financial loan SoaML model.

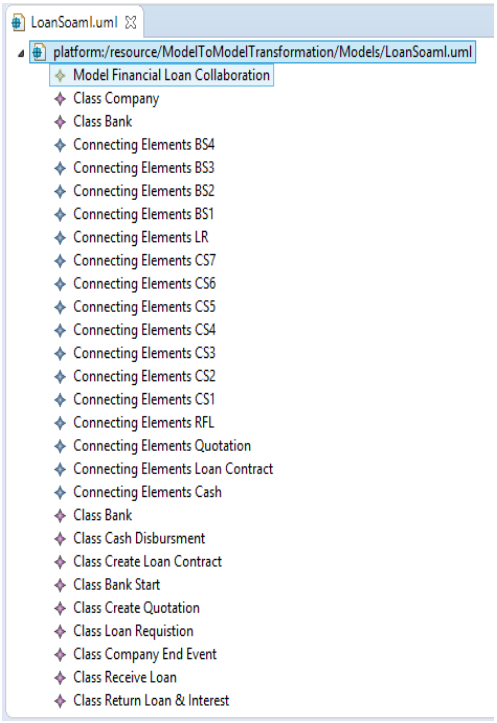


FIGURE 22. Tree view of loan SoaML model.

3) EQUIVALENT BPMN XMI MODEL

Hiring BPMN model is also converted into equivalent XMI model. Equivalent BPMN XMI model is mapped from BPMN model just done in pervious case studies.

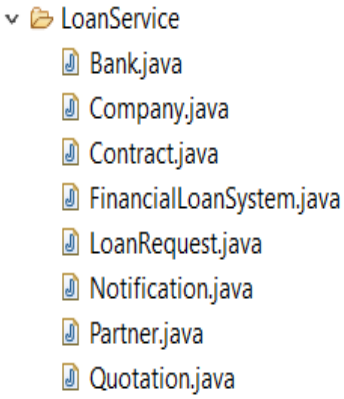


FIGURE 23. Web services code files.

4) B2W TRANSFORMATION ENGINE

FIGURE 26 shows the generated SoaML model for Hiring process services. This model contains all the service constructs that are necessary for automatic executable web services generation. This SoaML model acts as input in M2T transformation. Details of all the artifacts generated by B2W transformation engine are already described in the Amazon case study (**Section V-A-4**).

VI. DISCUSSION

In ERP System, Exchange and Conversion processes are commonly used [4]. However, developing their specifications, and further developing their code are complex and time-consuming activities [15]. Moreover, there is no way

```
1  package com.loan;
2  import java.util.*;
3  public class LoanServiceImp
4  {
5      private static List<LoanData> loanCatalog;
6
7      public LoanServiceImp() {
8          initializeLoanCatalog();
9      }
10     public LoanData searchById(int id) throws Exception
11     {
12         for (LoanData p :loanCatalog) if (p.getId() == id) return p;
13         throw new Exception("No record found with id " + id);
14     }
15     public LoanData[] getAllLoanRecord() {
16         LoanData[] loan = new LoanData[loanCatalog.size()];
17         int i = 0;
18
19         for (LoanData p : loanCatalog)
20         {
21             loan[i] = p;
22             System.out.println(loan[i].getId()+"    " +loan[i].getCompanyName()+"
23             i++;
24         }
25
26         return loan;
27     }
28
29     public void insertLoanData(LoanData loan) {
30         loanCatalog.add(loan);
```

FIGURE 24. Web service deployment.

to link the code back to the specification, thereby making the system evolution costly and time-consuming. Hence, this research work is to provide early web services specification for ERP processes and generation of corresponding web services code, so that the code generation remains cost-effective, traceable to specifications and easy to evolve of a period of time [4]. Our proposed approach uses BPMN modeling for exchange and conversion business processes and provides us with complete BPMN to web services generation framework after successful transformations on BPMN targeted models. Meanwhile, it also provides a tool for web services specification and code generation.

Business Process Model and Notation (BPMN) [6], [30] was standardized by OMG in March 2010. It acts as a supporting pillar for the business process modeling and implementation. It has widely been adopted since then. Other MDA approach like UML is also used for modeling but not specifically for business processes. The essential part of BPMN is providing notations that are used for business process modeling and further execution and equally understandable by all business analyst and then developers. BPMN is a mean of communication between different business users, customers, suppliers and business process implementer across the word [1]. Manual development of business processes is a difficult task for any organization. BPMN modeling provides modeling of business processes in standardized notations that will be easily understandable for all business analyst and then developers. Service-oriented architecture Modeling Language (SoaML) [9] is standardize by Object Management Group (OMG) that provides a meta-model web services specification and a UML profile for the specification of services through SOA framework. According to this language, services can be specified using a simple interface, service interface or through service contract based on the interface-based or contract-based approach [34]. It specifies the consumers and providers of services and defines the ports through which services are provided by a provider organization that is requested by consumers [31] [38] [40]. Through SoaML modeling, web service specification and generation became easy and web services evolve with a period of time when new functionality required for web services.

We have currently select core meta-model elements of BPMN i.e. Swimlane, Flow Objects, Connection Objects, Data Objects etc. and a few important extension meta-model elements i.e. Data Association, Message and Gateways etc. [29] and for SoaML we select i.e. Service Interface, Participant, Service Contract, Message Type, Message Attachment [38]. Two types of transformations are proposed in our research i.e. ATL Transformation and Acceleo Transformation. Executable web services for exchange and conversion business processes generated from transformation engine [31].

The approach generates SoaML service specification model and Java code using the model-based technique on BPMN modeling of the exchange and conversion business processes of the REA ontology framework in ERP system.

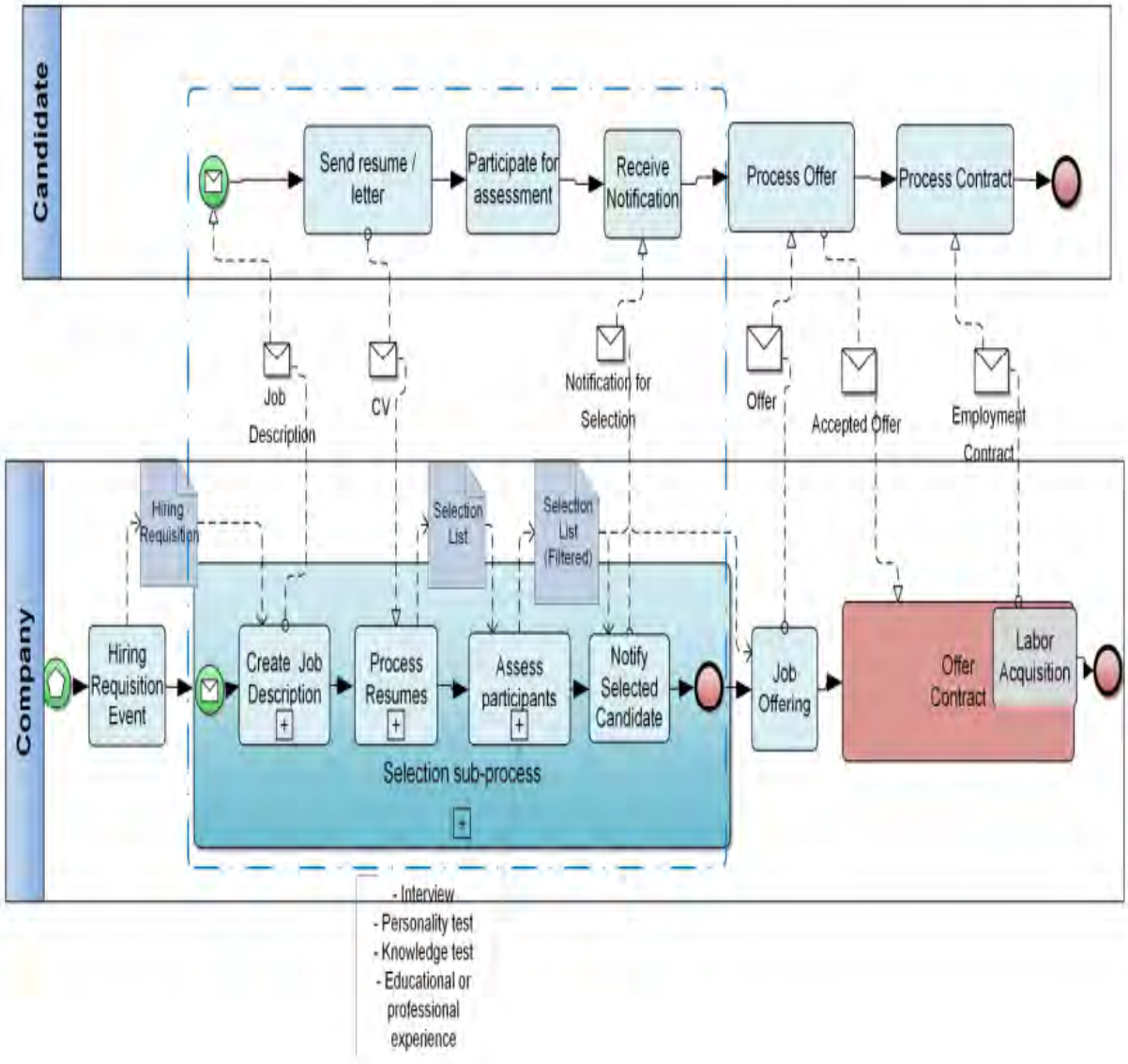


FIGURE 25. Complete hiring BPMN model.

The tool we have built produces all the real artifacts that a developer will need in order to develop the right product meeting the user specified BPMN model and a tester that test client application will encounter while dealing with the testing of published web services. The motivation behind this work is to provide early web services specification and Java implemented code so that quality can be built inside the application, which eventually proved out to be a cost and time efficient approach. The generated Java code provides complete and detailed functionality of web services extracted from BPMN models, for example, how the customer will order any product using Order web services and invoice will be generated after successful completion of the order produced by the tool. The Java code focuses entirely on the main functionality provided by BPMN process of ERP system. Furthermore, SoaML model shows all the participants and their interaction using connection objects i-e Realization and Usage relationship. BPMN has been inspired by Event-Driven Chain (EPC) that is traditional process modeling language [32]. BPMN is a dynamic process modeling language and overcomes the limitation of the EPC modeling language. Many tools are available that provide BPMN modeling. Even these tools are not mature enough and there were still some problems encountered during BPMN modeling. One of the many issues is that BPMN is not using standard execution semantic for functional testing and execution for different software used by the different user for validation [33]. There should be both levels of modeling i-e Abstract or High level for normal user and low level or more detailed design for business analyst. One of the major issue seems in BPMN modeling tool is that BPMN model does not convert into XMI model [35]. BPMN language is evolving and a lot of work is being done

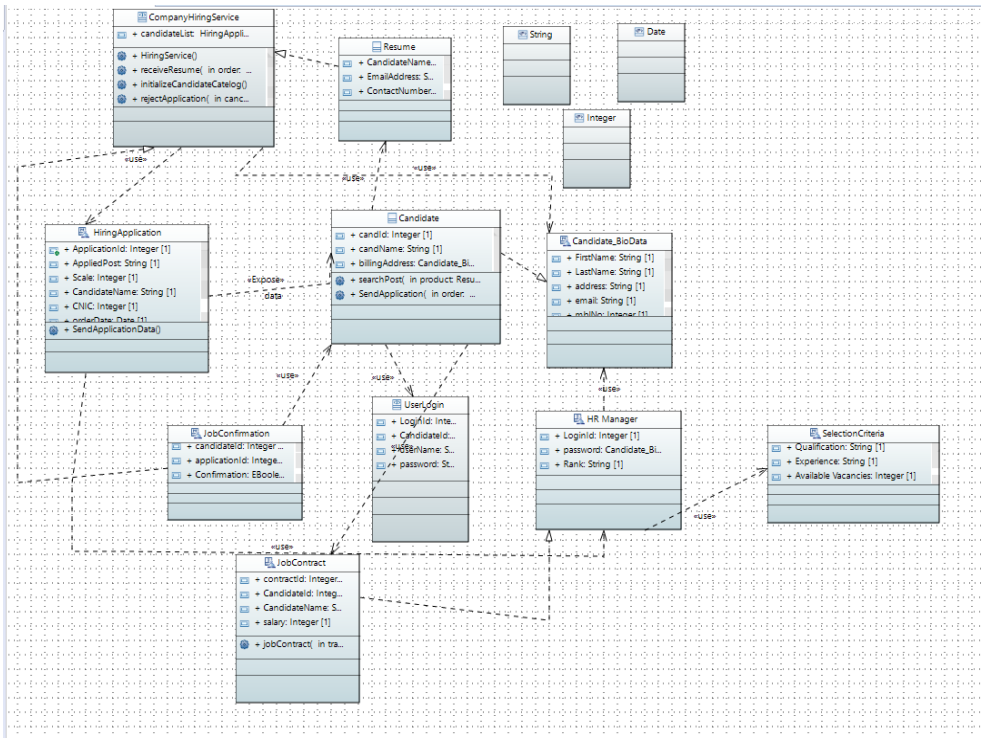


FIGURE 26. Hiring process SoaML model.

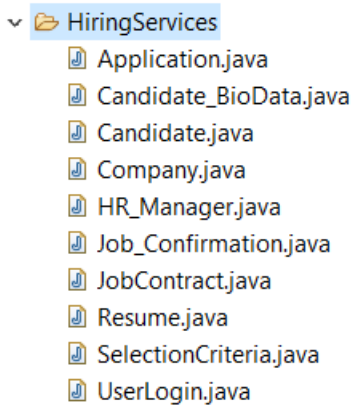


FIGURE 27. Web services code files.

on its tool support, we are hoping that these issues will be resolved soon. The first case study we selected was a very detailed case study on Amazon.com, which involves more than 15 subprocesses, task and connecting objects. The purpose of considering this case study was to validate the testing mechanism for web services of exchange and conversion business processes of ERP systems. Financial loan system and hiring process case study also validates our proposed approach. It includes exchange and conversion business services and involves many sub-processes until the process came to its completion.

The experimental results prove that the proposed approach significantly support ERP systems in generating web services from BPMN models which eventually helps in building the right product at the right time at a comparatively lower cost. From this research, it has been analyzed that there is a very limited amount of research work done on automatic execution of BPMN models in the area of model-based web services and the available research work did not capture complete tool that provides both functionality, modeling and code generation. Most of the work has been done only for business process modeling through BPMN and not targeting service specification and web service generation [36]. Our proposed approach is the first step toward automated execution of web services from BPMN models.

As we have taken the first step to automation execution for BPMN, there are a few limitations to our work. BPMN has a lot of features but due to the limited amount of time and resources, we have currently only selected limited core meta-model elements i.e. *Swimlane*, *Flow Objects*, *Connection Objects, Data Objects* etc. and a few important extension meta-model elements i.e. *Data Association*, *Message* and *Gateways* etc. There are many BPMN constructs that we have not yet considered, for example, *Signals*, *Time*, *Complex Gateways* and *Activity loop* etc. [37] on which we intend to work in future. The applicability of the proposed automated BPMN execution approach on other modeling languages i.e. SysML may be explored [39].

VII. CONCLUSION

This article presents a fully automated framework to generate executable Java web services from **B**usiness **P**rocess

Model and **N**otation (BPMN) models. Particularly, a modeling approach is introduced to represent ERP processes through BPMN concepts. Subsequently, the rules are developed to convert source BPMN models into target SoaML (**S**ervice-oriented architecture **M**odeling **L**anguage) models. Finally, transformation rules are developed to generate fully functional executable Java web services from SoaML models. As a part of research, a complete open source **B**PMN to **W**eb services transformation (B2W) tool is developed to automatically generate the web services from the high-level BPMN models.

The applicability of proposed framework is demonstrated through three benchmark case studies. The experimental results prove that the proposed framework is capable of generating Java web services accurately from BPMN models. Consequently, the proposed framework significantly simplify the development of ERP systems which eventually helps in building the right product at the right time at a comparatively lower cost.

Several future enhancements are possible in the proposed framework. For example, a one important improvement is to include other important BPMN constructs like, Signals, Time, Complex Gateways and Activate looping etc. in the proposed framework in order to support the modeling and transformation of complex and time dependent exchange and conversion processes. Similarly, existing framework can be extended to generate the web services in languages other than Java e.g. C# etc. Finally, it would be very interesting to apply the concept of proposed framework through other renowned modeling languages e.g. SysML etc.

REFERENCES

[1] G. L. Geerts and E. W. McCarthy, "The ontological foundation of REA enterprise information systems," in *Proc. Annu. Meeting Amer. Accounting Assoc.*, Philadelphia, PA, USA, vol. 362, 2000, pp. 127–150.

[2] W. E. McCarthy, "The REA accounting model: A generalized framework for accounting systems in a shared data environment," *Accounting Rev.*, vol. 57, no. 3, pp. 554–578, 1982.

[3] A. Osterwalder and Y. Pigneur, "An ontology for e-business models," in *Value Creation from E-Business Models*. London, U.K.: Elsevier, 2004, pp. 65–97.

[4] R. Blal and A. Leshob, "A model-driven service specification approach from BPMN models," in *Proc. IEEE 14th Int. Conf. e-Bus. Eng. (ICEBE)*, Nov. 2017, pp. 126–133.

[5] V. De Castro, E. Marcos, and R. Wieringa, "Towards a service-oriented MDA-based approach to the alignment of business processes with IT systems: From the business model to a Web service composition model," *Int. J. Cooperat. Inf. Syst.*, vol. 18, no. 2, pp. 225–260, 2009.

[6] B. Elvesaeter, D. Panfilenko, S. Jacobi, and C. Hahn, "Aligning business and IT models in service-oriented architectures using BPMN and SoaML," in *Proc. 1st Int. Workshop Model-Driven Interoperability (MDI)*, 2010, pp. 61–68.

[7] P. Bocciarelli, A. D'ambrogio, A. Giglio, E. Paglia, and D. Gianni, "A transformation approach to enact the design-time simulation of BPMN models," in *Proc. IEEE 23rd Int. Conf. (WETICE)*, Jun. 2014, pp. 199–204.

[8] H. Leopold, J. Mendling, and A. Polyvyanyy, "Generating natural language texts from business process models," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Berlin, Germany: Springer, 2012, pp. 64–79.

[9] A. Delgado, F. Ruiz, I. García-Rodríguez de Guzmán, and M. Piattini, "Business process service oriented methodology (BPSOM) with service generation in SoaML," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Berlin, Germany: Springer, 2011, pp. 672–680.

[10] M. W. Anwar, M. Rashid, F. Azam, and M. Kashif, "Model-based design verification for embedded systems through SVOCL: An OCL extension for SystemVerilog," *J. Des. Automat. Embedded Syst.*, vol. 21, no. 1, pp. 1–36, 2017.

[11] H. Bazoun, Y. Bouanan, G. Zacharewicz, Y. Ducq, and H. Boye, "Business process simulation: Transformation of BPMN 2.0 to DEVS models (WIP)," in *Proc. Symp. Theory Modelling Simulation-DEVS Integr.*, 2014, p. 20.

[12] A. Delgado, "Towards pattern-based generation of services to support business process execution," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2014, pp. 171–178.

[13] A. Delgado, "QoS modeling and automatic generation from SoaML service models for business process execution," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2015, pp. 522–529.

[14] A. Chaâbane, S. H. Turki, A. Charfi, and R. Bouaziz, "From platform independent service composition model in BPMN4SOA to executable service compositions," in *Proc. 12th Int. Conf. Inf. Integr. Web-Based Appl. Services*, 2010, pp. 653–656.

[15] P. Bocciarelli, A. D'Ambrogio, A. Giglio, E. Paglia, and D. Gianni, "Empowering business process simulation through automated model transformations," in *Proc. Symp. Theory Modelling Simulation-DEVS Integr.*, 2014, p. 39.

[16] J. Gordijn and H. Akkermans, "Designing and evaluating e-business models," *IEEE Intell. Syst.*, vol. 16, no. 4, pp. 11–17, Jul. 2001.

[17] J. Zdravkovic and T. Ilayperuma, "Designing consumer-aligned services using business value modelling," *Int. J. Org. Des. Eng.*, vol. 25, nos. 2–3, pp. 317–342, 2012.

[18] J. S. David, "Three 'events' that define an REA methodology for systems analysis, design, and implementation," in *Proc. Annu. Meeting Amer. Accounting Assoc.*, Dallas, TX, USA, 1997, pp. 1–35.

[19] J.-H. Koh, N. Weber, and A. V. Virkar, "Synthesis of lithium-beta-alumina by various ion-exchange and conversion processes," *Solid State Ionics*, vol. 220, pp. 32–38, Jul. 2012.

[20] B. Korherr and B. List, "Extending the EPC and the BPMN with business process goals and performance measures," in *Proc. ICEIS*, no. 3, 2007, pp. 287–294.

[21] A. Berre, "Service oriented architecture modelling language (SoaML)-specification for the UML profile and metamodel for services (UPMS)," Object Manage. Group, U.K., Tech. Rep. ptc/2009-12-09, 2008.

[22] G. Alonso, *Web Services*. Berlin, Germany: Springer, 2004, pp. 123–149.

[23] K. Czarnecki and S. Helsen, "Classification of model transformation approaches," in *Proc. 2nd OOPSLA Workshop Generative Techn. Context Model Driven Archit.*, vol. 45, no. 3, 2003, pp. 1–17.

[24] W. Kintsch and T. A. van Dijk, "Toward a model of text comprehension and production," *Psychol. Rev.*, vol. 85, no. 5, p. 363, 1978.

[25] J. Bézivin, E. Breton, G. Dupé, and P. Valduriez, "The ATL transformation-based model management framework," INRIA, Nantes, France, Tech. Rep. 03.08, 2003.

[26] J. Mtsweni, "Exploiting UML and acceleo for developing semantic Web services," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, Dec. 2012, pp. 753–758.

[27] J. Kovse and T. Härder, "Generic XMI-based UML model transformations," in *Proc. Int. Conf. Object-Oriented Inf. Syst.* Berlin, Germany: Springer, 2002, pp. 192–198.

[28] A. Amjad, F. Azam, M. W. Anwar, W. H. Butt, and M. Rashid, "Event-driven process chain for modeling and verification of business requirements—A systematic literature review," *IEEE Access*, vol. 6, pp. 9027–9048, 2018.

[29] B. Maqbool, "A comprehensive investigation of BPMN models generation from textual requirements—Techniques, tools and trends," in *Information Science and Applications* (Lecture Notes in Electrical Engineering), vol. 514, K. Kim and N. Baek, Eds. Singapore: Springer, 2019, pp. 543–557.

[30] *UML Profile for BPMN Processes*. Accessed: Feb. 2019. [Online]. Available: https://www.omg.org/spec/BPMNProfile/About-BPMNProfile/

[31] M. Geiger, S. Harrer, J. Lenhard, and G. Wirtz, "BPMN 2.0: The state of support and implementation," *Future Gener. Comput. Syst.*, vol. 80, pp. 250–262, Mar. 2018.

[32] F. Hasic, J. De Smedt, and J. Vanthienen, "Augmenting processes with decision intelligence: Principles for integrated modelling," *Decis. Support Syst.*, vol. 107, pp. 1–12, Mar. 2018.

[33] M. A. López-Campos, A. C. Marquez, J. F. G. Fernández, "Modelling using UML and BPMN the integration of open reliability, maintenance and

condition monitoring management systems: An application in an electric transformer system," *Comput. Ind.*, vol. 64, no. 5, pp. 524–542, 2013.
[34] F. Corradini, C. Muzi, B. Re, L. Rossi, and F. Tiezzi, "Global vs. local semantics of BPMN 2.0 OR-join," in *Proc. Int. Conf. Current Trends Theory Pract. Inform.* Cham, Switzerland: Edizioni della Normale, 2018, pp. 321–336.
[35] F. Corradini, A. Ferrari, F. Fornari, S. Gnesi, A. Polini, B. Re, and G. O. Spagnolo, "A guidelines framework for understandable BPMN models," *Data Knowl. Eng.*, vol. 113, pp. 129–154, Jan. 2018.
[36] A. J. Berre, S. Huang, H. Murad, and H. Alibakhsh, "Teaching modelling for requirements engineering and model-driven software development courses," *Comput. Sci. Educ.*, vol. 28, no. 1, pp. 42–64, 2018.
[37] S. Banerjee, S. Bajpai, A. Sarkar, T. Goto, and N. C. Debnath, "Ontology driven meta-modelling of service oriented architecture," *J. Fundam. Appl. Sci.*, vol. 10, no. 4S, pp. 205–210, 2018.
[38] I. Zafar, F. Azam, M. W. Anwar, W. H. Butt, B. Maqbool, and A. K. Nazir, "Business process models to Web services generation: A systematic literature review," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 789–794.
[39] P. Yustianto and R. Doss, "A unifying structure of metamodel landscape," *J. Model. Manage.*, to be published.
[40] M. Ozkaya, "Analysing UML-based software modelling languages," *J. Aeronaut. Space Technol.*, vol. 11, no. 2, pp. 119–134, 2018.
[41] *BPMN to Web Services (B2W) Tool, CEME, NUST.* Accessed: May 2019. [Online]. Available: https://ceme.nust.edu.pk/ISEGROUP/B2W/index.html

IQRA ZAFAR received the M.S. degree in software engineering from the Computer and Software Engineering Department, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan, in 2018. She is currently a Senior Researcher in the field of software engineering. Her research areas are business process automation, model-driven software engineering (MDSE), and web services automation.

FAROOQUE AZAM is currently an Adjunct Faculty Member with the Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. He has been teaching various software engineering courses, since 2007. His research interests include model-driven software engineering, business modeling for web applications, and business process reengineering.

MUHAMMAD WASEEM ANWAR is currently pursuing the Ph.D. degree with the Department of Computer and Software Engineering, CEME, National University of Sciences and Technology, Pakistan. He is also a Senior Researcher and an Industry Practitioner in the field of model-based system engineering (MBSE) for embedded and control systems. His major research area is MBSE for complex and large systems. His profile can be viewed at http://ceme.nust.edu.pk/ISEGROUP/seniormembers/waseem.html.

BILAL MAQBOOL received the M.S. degree in software engineering from the Computer and Software Engineering Department, College of Electrical and Mechanical Engineering (CEME), National University of Sciences and Technology (NUST), Pakistan, in 2018.

Since 2017, he has been a Senior Member of the Intelligent Software Engineering (ISE) and Model-Driven Software Engineering (MDSE) Research Groups, CEME, NUST. He is currently a Senior Researcher in the field of software engineering. His research areas are business process automation, model-driven software engineering (MDSE), natural language processing (NLP), and web services automation.

WASI HAIDER BUTT is currently an Assistant Professor with the Department of Computer and Software Engineering, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan. His research interests include model-driven software engineering, and Web development and requirement engineering.

AIMAN NAZIR received the M.S. degree in software engineering from the Computer and Software Engineering Department, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Pakistan, in 2018.

She is currently a Senior Researcher in the field of software engineering. Her research areas are model-driven software engineering, data engineering, knowledge discovery, and decision sciences both in academia and industry.