

# A Model-Driven Approach for Business Processes and Test Case Generation.

Himanshu Ghetia<sup>1</sup>, Shantanu Santoki<sup>2</sup> and Vairamuthu S.<sup>3</sup>

**Abstract---** model-driven development approaches, business process models are used at different levels in the development process. Business process modeling is one of the most important methods in business process management. It is used for better understanding of business concerns and communication between stakeholders. In this paper, we present a model-driven approach to generate business process models and the process interaction semantics from a language-independent process models. We employ UML 2.0 to deliver language-independent business process design with the low level syntax hidden from the business experts; the Web Services Business Process Execution Language (WS-BPEL) is utilized as the target business process modeling language. A WS-BPEL metadata model is defined to decorate UML 2.0 process model with WS-BPEL stereotypes; a semantics metadata model is defined for specifying process interaction semantics relevant stereotypes. With a set of model mapping patterns, we can generate a WS-BPEL business process model and a process interaction semantics file from each UML 2.0 process model. Our approach can also be applied for generating process models in other business process modeling languages.

**Index Terms---** Business process model, Process interaction semantics, Model-driven approach, Web Services Business Process Execution Language (WS-BPEL), Automatic testing, Modeling, Testcase

## 1. INTRODUCTION

Technology of Web Service plays an important role in constructing application via Internet. In the world of web there are many web services providers are available. Among them many of them provides good services but with some high cost. Services requester required more chipset and with their satisfiable requirement.

For that web services development using the Service Oriented Architecture (SOA)[5]. Web services provides the system integration by more and more system initiatives. This provides the reusability of software components with use of loosely couple architecture. For any business requirement is hardly to fulfill by only single web services. For fulfill customers requirement we requires composition of many web services. For that we make some web services business language standard, its describes the protocol which is known as Web Services Business Process Execution Language (WS-BPEL)[4].

- Himanshu Ghetia is currently pursuing masters degree program in computer science and engineering in VIT University, India, E-mail: @mail.com.
- Shantanu Santoki is currently pursuing masters degree program in computer science and engineering in VIT University, India, E-mail: shantanu.santoki@mail.com.
- Vairamuthu S. is currently Assistant professor in computer science and engineering in VIT University, India, E-mail: svairamuthu@vit.ac.in.

However, current technologies of business process creation based on only low level language and reuse of some available web services. It cannot use the high level of business process. So it cannot provide the dynamic web services selection and binding. For that reusability of web services we consider some critical challenges to focus on that, which are,

- 1) Transparency of low level syntax.
- 2) Independent of business process modeling language.
- 3) Process Model Transformation.[1]

Also we generate the test case generation using model-driven-approach toward generating executable test case for the given test cases. The approach is composite of three stages: defining a process under test based on the business process model, transforming from process under test to abstract test cases and from abstract test cases to executable test cases[3].

## 2. RELATED WORK

Basic aim of Model Driven Architecture (MDA) to mending the efficiency and quality of the software development. The basic idea behind that is to model transformation of the Platform independent models (PIM) to the platform specific models (PSM). PIM take care of the only business logics which can into web services and business process it don't considering the platform technologies. In the other side PSM can mainly concern about web services and business process

implementation which are related to some business process description language[2].

In the world of semantics web services give notations are to be used to dynamic web services selection and composition. As Aggarwal also propose METEOR-S, which is composition tool. These tools help the designer to binding the web services based upon the abstract process business restraint and generate executable process[6]. Zouzhile et also. [7] create a framework for web services selection and binding, and also create the design of normative languages to explaining about the semantics web services functionality and its business process requirements.

For test case generation we can use like platform independent test modelling and used the model driven test case generation methodology. Many of the researchers use the model driven approach in the web services to generating the test cases. For describing the BPEL, J. Garcia-Fanjul and C. de acquire SPIN techniques which generate the test suite specification of web services composition [1, 2]. Same SPIN method used by Shin Nakajima to validate the web service flow[3]; From the web services flow Zhang's research transforms that flow into the test model based which is based on UML 2.0 activity diagram and creates relevant test cases[8]; Main study during this period in test case generation is on formal and semi formal specification. Mainly generated test case based on the abstract test cases class, it means they cannot be run automatically.

In this paper we present the modern driven approach to dynamic web services selection and composition. Also we present the idea about how the test cases will be generated by using those services. There are many methods available to select and composition of services. From that techniques two way solution were suggested such as, "WSDL[9] + WS-BPEL" and second one was "OWL-S[10] + AI Planning" for composition. For increasing the business process industry smartly we use the model driven approach. Any business analyst understanding business process it used the business process modeling notation (BPEN)[11] and represent the all logic it terms of graphically notation based. As we discuss above that WS-BPEL for web services composition from UML, many of them also use the UML as extensible mechanism [1, 2]. BPEL is used to generate the test cases which work for business process the created test cases are to be translating into directly TTCN-3 test case. Those TTCN-3 test cases are to be compiled and run directly. The BPM (business process model) is based on the BPEL spec and UML2.0 activity diagram[3].

### 3. BASIC APPROACH

Our approach introduces a series of improvements to enable better automation of the business process generation and greater reusability of business process designs. Our model-driven approach involves three phases:-1) Design,2) Annotation,3) Generation, as shown in Figure 1[1].

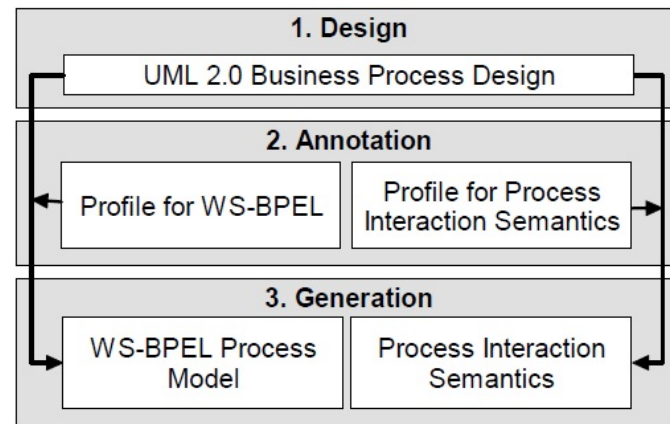


Figure 1. Approach Overview

In the Design phase, UML 2.0 generates the business process model, in that is not any specified target language like as WS-BPEL. In annotation phase by using the annotations it gives the set of information to specifying the target business process model. In final generation phase the process interaction semantics and the business process model in the target modeling language are created.

### 4. DESIGN PHASE

In the Business Process Model are described in to three models:-1) BPM, which discovered the relation between the business process and it's partners. 2) Data Model, which identified the business process model variables definition and its usage. 3) Control flow model, which informed us to sequences of task execution.

#### A) Business Process Model

In this model we discovered with uml 2.0 diagram. In that each type of interaction is done with the uml metaclass *collaboration*, which has many roles. Business process model described those roles via uml metaclass a port. If the collaboration has two roles, that role will be represented as business process role represented by circle and role of partners represented by an arc over circle.

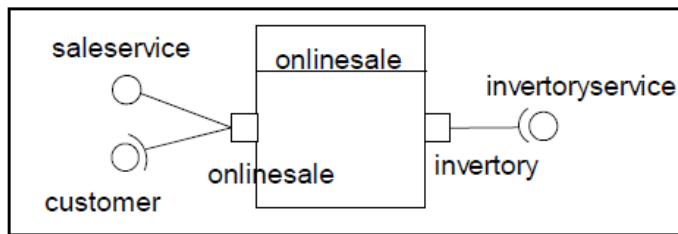


Figure 2. Business Protocol Model of OnlineSale Process

Example in Figure 2[1] as we consider the onlineselasprocess. In that process, business process model taken two type of interaction which is defined through OnlineSale port and Inventory port. The OnlineSale port contains two roles: SaleService, which is described about role of business process; and Customer, which is the described of the partner. The collaboration corresponding to the port Inventory only involves the role InventoryService. In this example compare with WS-BPEL we liked that Collaboration and Port correspond to PartLinkType and PartnerLink, respectively.

#### B) Process Data Model

In this model we described the UML class diagram; it describes the variable usage and its different activity. A class which containing attributes for process variables, properties, activities, and interaction messages which are declared as business process. Those attributes data types are simulated with other classes.

#### C) Process Control Model

In this model we described about activity diagram with UML 2.0. It described in which flow the execution of different activity should work. To make the business process correctly by using control flow model each and every activity diagram must be well formed.

#### Definition 1: Well-formed AD

An Activity Diagram is well-formed if and only if it satisfies the following conditions:-

- 1) There is exactly one entry node and one exit Node; 2) Every activity diagram has executable node which content exactly one incoming and one outgoing edge; 3) Every decision node and fork node has exactly one incoming edge and at least two outgoing edges;
- 4) Every merge node and join node has at least two incoming edges and exactly one outgoing edge;
- 5) Every node can be reached on a path from the entry node to the exit node.

These all are restrictions of well-formed activity diagram for taken the basic control flow patterns which defined in order to avoid the indirect method of full AD. However, these all restriction not made full semantically sound of well-formed AD. So that we declare the Separable AD.

#### Definition 2: Separable AD

An AD is separable if and only if it can be follow these rules:

- 1) A well-form AD with containing only decision nodes and merge nodes as main control elements is separable. We can use all part of node as well as edges from well-form AD as separable excludes entry and exit node as sequential region;
- 2) A well-form acyclic AD with only fork nodes and join nodes as control elements is separable. We can use all part of node as well as edges from well-form AD as separable excludes entry and exit node as parallel region;
- 3) If a1 and a2 are separable activity diagrams and a1 has an executable node e, the result of replacing e with a2 in a1 is separable.

From this definition of the activity diagrams we see that all the work done in industry are dissociable . However we also say that each type of parts in the development of the rules can be dissociable from other, so that from these two parts are separated each other, which helps us to business process mapping.

## 5. ANNOTATION PHASE

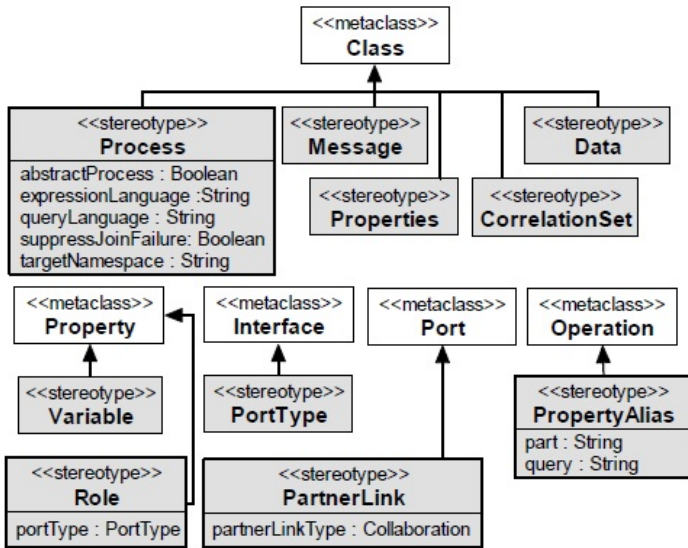
In the UML design phase, there are two new information generated which are to be add into the UML process model. In that, first information is for target business process modeling language such as WS-BPEL, but the target business language syntax and its meaning which doesn't know by the UML. The other set of information is about the process interaction semantics. That information discovered the signification of the business process interface with ontology. We set the two metadata models as UML profiles; these models discovered the two set of information to the UML process model.

In Figure 3[1] first part of figure defines the stereotypes which used to identify the business protocol. Those are PartnerLink, Role and PortType, and data flow, which continents Message, Properties, CorrelationSet, Data, Variable, and PropertyAlias. In the second part of figure discovers stereotypes. This gives the business process for control flow model. We have designed our algorithm to automatically detect control flow likes Flow, While, Switch and Sequence, so these are not covered in this profile. In this paper we also describe the meaning of the meta model for each process partner which take the process interaction meaningfully. For target process

interaction semantically representation we use the semantic meta model for that transformation[2].

description provides a easily human readable information of stereotype.

**Extensions for Class and Composite Diagram**



Definition of Stereotype <<Partner Semantic>>	
Base Class	Role (in composite diagram)
Description	Semantic requirement of a business process partner
Tagged Values	1) Semantic 2) Ontology domain
Constraints	This role is never referenced in the provided interface.
Definition of Stereotype <<Operation Semantic>>	
Base Class	Action (in activity diagram)
Description	Semantic requirement of a interaction operation
Tagged Values	1) Partner (reference of Partner Semantic) 2) Functionality: Input, Output, Precondition and Effect 3) Quality of Service
Constraints	This action should be the invoke action.

Table 1. Table Type Style.

**Extensions for Activity Diagram**

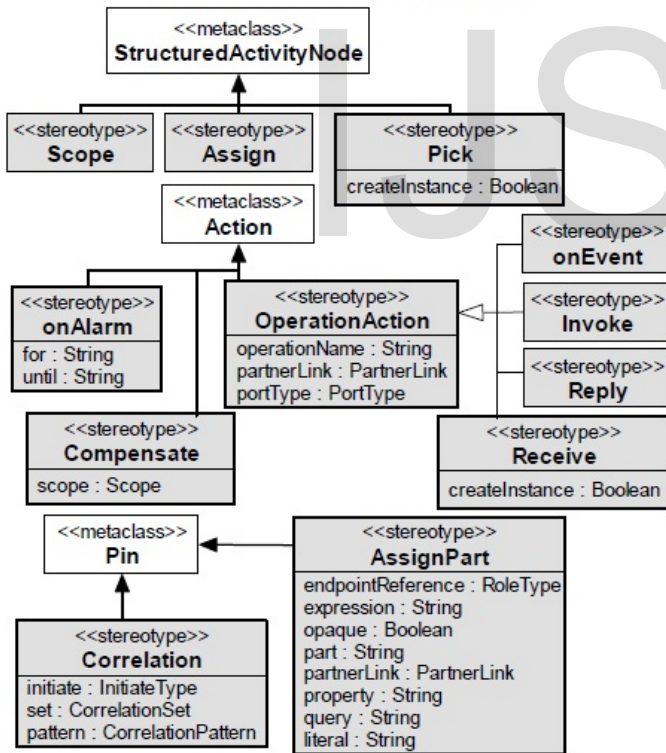


Figure 3. UML Profile for WS-BPEL

The following table gives the details about the UML notation with the two stereotypes semantic metadata model. In each stereotype, the base class is defined as UML class which can be applied to; the tagged-values, whose values can be changed during the UML diagram to edited; the constraints imposing rules and restrictions to the stereotypes; the

description provides a easily human readable information of stereotype.

**6. GENERATION PHASE**

In generation phase, it describes the process interaction meaning and the business process model in to target bpm (business process modeling) language is generated from the annotated UML business process model.

A. Transformation of Process Interaction Semantics:-

The creation of process interaction meaning depends on the two stereotypes, which are Partner Semantic, and Operation Semantic. From that semantics we can apply on the activity and composite diagrams. For that reason it's required two transformation rules.

- 1) Each Partner Semantic stereotype is mapped to a PPC entry.
- 2) The correlative Operation Semantic stereotypes are mapped to the Process Partner Contract Item (PPCI) entries contained in the PPC entry.

B. Transformation of Business Process Model:-

The creation of the process model for that it transforms the UML element in to the business process model in WS-BPEL. We first create the data elements into WS-BPEL variables based on the stereotype annotations. The data elements, such as the variable name and type; then we create the business protocols, such as the PartnerLinkType and PartnerLink; at

last we create the business process activities, in both terms atomic and compound.

For each atomic business process activity, it generates the stereotype business process, which was already commented in UML node, letting in the activity type and attributes. An atomic activity node, we first try to create the WS-BPEL activity element based upon its related activity type, and then the glossed activity attributes are repeated and placed in the newly created WS-BPEL activity element.

For composition of the web services, as we mention that above compound activity used rather than performing the process transformation depending all structural activity in WS-BPEL. It is also based on the trusting on the designer to create the well structured process model. Our main aim to focus on the analyze the graph oriented UML process model and it should be transfer it into the WS-BPEL. The sequential and parallel region as we mention above, they share their common characteristic of having a single starting edge to and a single ending edge from the region. These single-entry-single-exit concepts taken from compiler and also applied for workflow analysis.

From that decomposition of activity diagram the transformation was repeatedly done, starting from inner region to outer region. We are decomposition region to WS-BPEL activity. We use the five models which classify the different types of construction control flow, which describes in the Figure 4. In that we use the  $R_i$ , which to overcome the loss of generality, represent WS-BPEL code to the corresponding abstract node. For that transformation of each region is to find most appropriate region without considering the structure of its nesting ones. By identifying the region related method will be taken to create the WS-BPEL code for it and the region will be reduces to the abstract node.

On doing that transformation, for getting the more readable code we try to captured pattern of the construction of WS-BPEL. In our research there are some unstructured regions, which cannot be mapped by any pattern. For these unstructured regions we divided into two parts: unstructured sequential region and unstructured parallel region. For the unstructured sequential region is the sequential region which has either improperly nested Decision Nodes and Merge Nodes, or unstructured cyclic flows, it's may be both. So it cannot be mapped to any structured Sequence. For unstructured parallel region some kind of parallel region which has improperly nested Fork Nodes and Join Nodes, that can't be matching to Flow pattern. It is the challenging task to transformed unstructured region into structured region. We try to transformed unstructured sequential region to its equivalent WS-BPEL code. Also we try to transformed unstructured parallel region into nested nodes to activities

contained in a single WS-BPEL flow, and all that edges to be linked in this flow.

For our convenience, we considering the readability of the created WS-BPEL code we combined sequential pattern and links which are generated in WS-BPEL can make as many as sequential structure. In this way we can generate the WS-BPEL model with the process interaction.

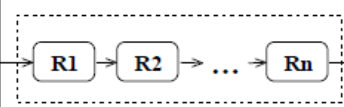
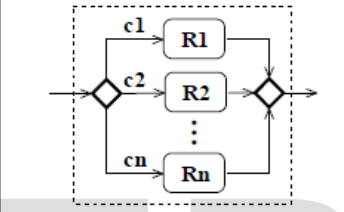
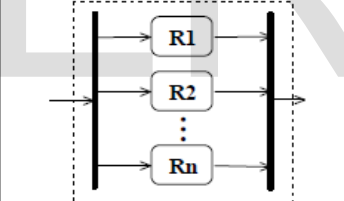
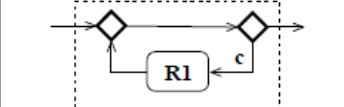
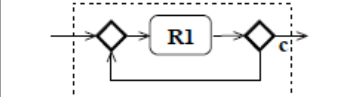
<p><b>Sequence Pattern</b></p> 	<pre>&lt;sequence&gt; code(R1) code(R2) ..... code(Rn) &lt;/sequence&gt;</pre>
<p><b>If Pattern</b></p> 	<pre>&lt;If&gt; &lt;condition&gt;c1&lt;/condition&gt; code(R1) &lt;elseif&gt; &lt;condition&gt;c2&lt;/condition&gt; code(R2) &lt;/elseif&gt; ..... &lt;elseif&gt; &lt;condition&gt;cn&lt;/condition&gt; code(Rn) &lt;/elseif&gt; &lt;/If&gt;</pre>
<p><b>Flow Pattern</b></p> 	<pre>&lt;flow&gt; code(R1) code(R2) ..... code(Rn) &lt;/flow&gt;</pre>
<p><b>While Pattern</b></p> 	<pre>&lt;while condition = "c"&gt; code(R1) &lt;/while&gt;</pre>
<p><b>RepeatUntil Pattern</b></p> 	<pre>&lt;repeatUntil &gt; code(R1) &lt;condition&gt;c&lt;/condition&gt; &lt;/repeatUntil&gt;</pre>

Figure 4. Mapping Patterns for Structured Regions

## 7. TEST CASE GENERATION

For the test case approach the platform independent test modelling and used the model driven test case generation methods. The business process model based on the BPEL specification and UML2.0 activity diagram defines the process under test (PUT). U2TP provides a framework necessary for modelling a test system with four parts

- 1) Test architecture
- 2) Test data
- 3) Test behaviour
- 4) Timing

The test case model based on U2TP framework and TTCN-3 key concepts defines the test behaviour and test data of the abstract test cases (ATC). PUT is visualized in activity diagram and ATC is visualized in sequence diagram.

In the test case generation approach there are two key transformations in our approach:-

Transformation from PUT to ATC and transformation from ATC to executable test case (ETC) in TTCN-3 which as defined in figure 5[3].

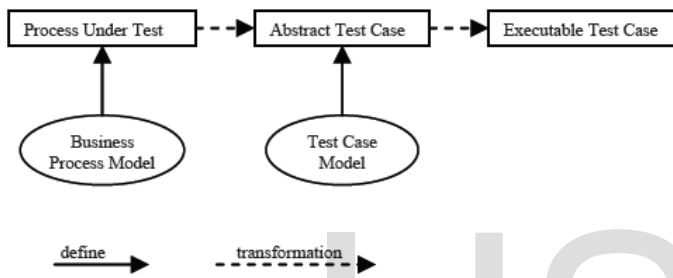


Figure 5. Test case generation methodology

**Definition 3:** - Business Process model

The business process model is composite of *Activity, Control Node, Client and Web Service*. There are four link which connect above four concept that are

*ActLink*:- specifies the relation between two *Activities*  
*ControlLink*:- defines the relation between an *Activity* and a *Control Node*.

*CliLink*:- defines the relation between a *Client* and an *Activity*.

*ParLink*:- defines the relation between a *Web Service* and an *Activity*.

There are seven type of primitive’s activity business process model as subclasses of activity: *put\_Receive, put\_Reply, put\_Assign, put\_Throw, put\_Exit, put\_Invoke and put\_Wait*. The primitive activities are defined with the same attributes with the corresponding BPEL basic activities. There are four type of control node *Join, Fork, Merge, and Decision*. They have the same semantics with the control nodes of UML2.0 activity diagram.

**Definition 4:** - Test case model

To describe an abstract test case, the following concepts are defined in the test case model:

□ *TestComponent*: provides the environment and mechanism for test case to implement its behavior. A test component has a set of ports via which it may communicate with other test components or with the SUT.

□ *TestCase*: is a specification of test behavior, including what to test with which input, expecting result and under which configuration.

□ *TestData*: denotes a group of data used in communicating between the test system and SUT.

□ *Verdict*: is the evaluation of the correctness of the SUT under specific test case. There are four verdicts: pass, fail, inconclusive and error.

□ *TestExecution*: a process of the execution of a test case.

□ *Message*: the messages interacted between two Test Components.

1) Transformation

Here we done 2 transformation for test case generation first was PUT to ATC and the second one ATC to ETC

A) PUT to ATC Transformation

**Definition 5:-** Test Case (TC):  $TC = \langle S, R \rangle$ , where

$$S = \{x \in A, input\},$$

$$A = \{put\_Invoke, put\_Receive\},$$

$$R = \{x \in B, expectation\},$$

$$B = \{put\_Invoke, put\_Reply\}.$$

S is stimulus, S has a two argument one of them is activities *put\_invoke* or *put\_receive* and second one is input, input is data which we have to given as a input and input is automatically generated by test case generation algorithm

R is response, R has alsotwo argument first one is activities like *put\_invoke* or *put\_replay* and second argument is expectation and expectation is expected output for response and expectation is set by user according to process specification and input data

There are two step to generate test case from PUT

- 1) Generate a test path from start to end
- 2) Generate a input associated with activity A

DFS Algorithm used for traverse activity diagram

In this paper we use boundary testing strategy to simple data point within the boundary, outside boundary, and exactly on line. In ATC(abstract test case) there are MTC(main test component) and SUT object, MTC is created for each test case by default. The stimulus will be sent to MTC, Once MTC terminated test case execution finish SUT is the process under test which accepts the stimulus and returns the responses.

To understand the complete test scenario, we add a Parallel Test Component and web service object. Web service invoked by the business process. PTC is started and terminated by MTC. It can collaborate with MTC to simulate the interaction between the process under test and web services invoked in the process.

*B) ATC to ETC Transformation*

One MTC in each and every test case, MTC always mansion in header part of test case, behaviour of test case is realized by MTC with SUT. Type of resultant of executed test case is type of VerdictType, given below table-1 show type definition and template definition

TC Model	TTCN-3
MTC	runsonComponent
TestCase	testcase
TestData	template
TestExecution	execute
testVerdict	getverdict/setverdict
Receive	receive
Invoke	call
Reply	reply

Table-2 Transformation mapping from ATC to TTCN-3

To transform TC model to TTCN-3 model required TTCN-3 code skeleton code file in that popular structure of code is define, with this code transform test case in sequence diagram into TTCN-3 test case, according to table-1 mapping.

**8. CONCLUSION & FUTURE WORK**

In this paper we proposed a model-driven approach for generating the WS-BPEL processes and web services composition using the WS-BPEL semantics. For using that business process modeling language we defining the notation of UML diagrams in easily human readable form into independent business process model. For generating the WS-BPEL code there are many methods available also for analyzing that that workflow structure. Also we make the test cases using mda (modern driven approach) for test the generated code will be quality code or not. It can cut the effort

in testing business process particularly when the process is evolving.

In future we can do model transaction approach to more general control flow models. It also we can do our sequencial and parallel approach into mixed region approach, OR-split and synchronized conditions. We also more refine our test case generation approach in future to get best solution for customers.

**ACKNOWLEDGMENT**

The authors would like to thank the School of Computer Science and Engineering, VIT University, for giving them the opportunity to carry out this research

**REFERENCES**

[1] Chenting Zhao, Zhenhua Duan, Man Zhang-2009 "A Model-Driven Approach for Generating Business Processes and Process Interaction Semantics", Eight IEEE/ACIS International Conference on Computer and Information Science.

[2] Chenting Zhao, Zhenhua Duan, Man Zhang-2009, "A Model-Driven Approach for Dynamic Web Service Composition", Eight IEEE/ACIS International Conference on Computer and Information Science.

[3] Qiulu Yuan, Ji Wu, Chao Liu, Li Zhang-2008, "A Model Driven Approach Toward Business Process Test Case Generation", IEEE International Conference on Computer and Information Science.

[4] T. Erl-2005, "Service-Oriented Architecture: Concepts, Technology, and Design", NJ, USA: Prentice Hall PTR.

[5] OASIS, Business Process Execution Language for Web Services (WS-BPEL) Version 2.0, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.

[6] R. Aggarwal, K. Verma, J. Miller and W. Milnor, "Constraint Driven Web Service Composition in METEOR-S", Proc. the IEEE International Conference on Service Computing, 2004, pp. 23-30.

[7] Zhile Zou, Zhenhua Duan, Jianli Wang, "A Comprehensive Framework for Dynamic Web Services Integration". Proc. The European Conference on Web Services, 2006, pp. 138-147.

[8] Zhang Guangquan, Rong Mei, Zhang Jun, "A Business Process of Web Services Testing Method Based on UML 2.0 Activity Diagram", Intelligent Information Technology Application Workshop, Dec. 2007, pp 59-65.

[9] W3C, Web Service Description Language (WSDL), <http://www.w3.org/TR/wsdl>, March 2001.

[10] W3C, Semantic Markup for Web Services (OWL-S), <http://www.w3.org/Submission/OWL-S>, March 2001.

[11] S.A.White, Business Process Modeling Notation Version 1.0,[http://www.bpmn.org/Documents/BPMNV1-0May\\_2004.pdf](http://www.bpmn.org/Documents/BPMNV1-0May_2004.pdf), May 2004.

IJSER