Automatic Detection of Semantic Inconsistency Between BPMN Process Model and SBVR Rule Model

Akanksha Mishra

Computer Science

Indraprastha Institute of Information Technology, Delhi (IIIT-D), India

A Thesis Report submitted in partial fulfilment for the degree of

MTech Computer Science

22 August 2015

1. Prof. Ashish Sureka (Thesis Adviser)

2. Prof. Somitra Kumar Sanadhya (Internal Examiner)

3. Mr. Anjaneyulu Pasala (External Examiner)

Day of the defense: 22 August 2015

Signature from Post-Graduate Committee (PGC) Chair:

Abstract

Business Process and Model Notation (BPMN) is a graphical model of business processes and Semantic of Business Vocabulary and Business Rules (SBVR) provide business rules corresponding to that business process. Though both these formalism should have no inconsistencies, but that is often present. Our research interest lies in converting SBVR rules to graphical representation and using sub graph-isomorphism to detect instances of inconsistencies between BPMN and SBVR model. We propose a framework and multi-step process to identify the instances of inconsistencies between the two models. We first generated XML of BPMN diagram and apply parsing and extracted tags relevant to us. We then used Stanford NLP Parser to generate parse tree of Rules and detailed information about the parse tree is stored in the form of Typed Dependency which represent grammatical relation between words of a sentence. We utilized this grammatical relation to extract triplet (actor-action-object) of a sentence. We find node-induced sub graph of all possible length of nodes of a graph and apply VF2 Algorithm to detect instances of inconsistency between sub graphs. Finally we evaluate the proposed research framework by conducting experiments on a synthetic dataset to validate the accuracy and effectiveness of our approach.

I dedicate my M.Tech Thesis to my parents for their endless love, support and encouragement. This work is also dedicated to Richa Sharma for her immense encouragement and valuable guidance. I am grateful for her constant support and help.

Acknowledgements

First and Foremost, I offer my deepest gratitude to my advisor, Dr. Ashish Sureka, whose valuable support and consistent guidance has helped me complete my thesis. Without his guidance and support, this thesis would not have been possible. His excellent guidance and motivation has always been the source of inspiration whenever I got stuck in the work.

Besides my advisor, I would like to deeply thank my esteemed committee members Prof. Somitra Kumar Sanadhya and Mr. Anjaneyulu Pasala for agreeing to evaluate my thesis.

Next, I would like to thank Richa Sharma for her constant support, encouragement and help during the course of my thesis. I also thank her for her valuable hints during the work.

Last but not the least, I would like to thank all my family members who encouraged and kept me motivated throughout the thesis.

Declaration

This is to certify that the thesis titled **"Automatic Detection of Semantic Inconsistency Between BPMN and SBVR"** submitted by **Akanksha Mishra** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under my guidance and supervision at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Professor Ashish Sureka
Indraprastha Institute of Information Technology, New Delhi

Contents

List of Figures **ix**

List of Tables **xi**

1 Research Motivation and Aim **1**

1.1 Business Process Modeling 1

1.1.1 BPMN Process Model: An Overview 2

1.1.1.1 Basic BPMN Modeling Elements 4

1.1.1.2 BPMN Modeling Styles 5

1.2 Business Rule Modeling 5

1.2.1 SBVR Rule Model: An Overview 6

1.3 Research Problem 7

1.3.1 Instances of Semantic Inconsistency 7

1.3.1.1 Vehicle Reservation System 8

1.3.1.2 Order Fulfillment Process 11

1.4 Research Aim 12

2 Related Work and Research Contributions **13**

2.1 Related Work 13

2.1.1 Integration of BPMN and SBVR 13

2.1.2 Triplet Extraction from Sentence 14

2.1.3 Graph Matching Algorithm 14

2.2 Novel Research Contributions 15

3 Research Framework and Solution Approach **17**
3.1 XML based BPMN Graph 18
3.2 Typed Dependency Based Triplet Extraction 24
3.3 Subgraph Graph Algorithm: VF2 Algorithm 29

4 Experimental Analysis and Results **32**
4.1 Experimental Dataset 32
4.1.1 Travel Reservation 32
4.1.2 Order Fulfillment 33
4.1.3 Employee Reimbursement Request 33
4.1.4 Cab Booking 33
4.2 Experimental Results 34
4.2.1 Consistent Scenario : Cab Booking Result 34
4.2.2 Inconsistent Scenario : Order Fulfillment 38
4.2.3 Experimental test-bed result 41

5 Limitations and Future Work **43**

6 Conclusion **44**

References **45**

List of Figures

1.1 BPMN Example 2
1.2 An Instance of BPMN Process Model for Vehicle Reservation System . . 8
1.3 Subset of SBVR Rule Model for Vehicle Reservation System 9
1.4 An Instance of BPMN Process Model for Vehicle Reservation System . . 10
1.5 Subset of SBVR Rule Model for Vehicle Reservation System 10
1.6 An Instance of BPMN Process Model for Order Fulfillment Process . . . 11
1.7 Subset of SBVR Rule Model for Order Fulfillment Process 12

3.1 Research Framework 17
3.2 Sample BPMN Process Diagram for Travel Reservation 19
3.3 Subset of Generated XML file for Travel Reservation and represents Activity of the BPMN Process Model 20
3.4 Subset of Generated XML file for Travel Reservation and represents MessageFlow of the BPMN Process Model 21
3.5 Subset of Generated XML file for Travel Reservation and represents Transitions of the BPMN Process Model 22
3.6 Adjacency Matrix for BPMN Process Model 23
3.7 Graph Representation of BPMN Process Model for Travel Reservation . 23
3.8 Generation of Typed Dependency for a English Sentence 26
3.9 Classification of Rule (Action-Oriented) 27
3.10 Classification of Rule (Structure-Oriented) 28
3.11 Graph Representation of SBVR Rule Model 28
3.12 Graph V 30
3.13 Graph V′ 31

4.1 BPMN Diagram for Cab Booking 34
4.2 BPMN Graph for Cab Booking 36
4.3 SBVR Graph for Cab Booking 37
4.4 BPMN Diagram for Order Fulfillment 38
4.5 BPMN Graph for Order Fulfillment Process 40
4.6 SBVR Graph for Order Fulfillment Process 41

List of Tables

1.1 Node Label of the BPMN Example Diagram 3

3.1 List of Triplet Extracted from Business Rules for Cab Booking 29

4.1 Collection of Synthetic Data set of Consistent and Inconsistent Scenarios 34
4.2 List of Activity (Nodes) of Cab Booking Scenario 35
4.3 List of Message Flow and Transition (Edges) of Cab Booking Scenario . 35
4.4 List of Edges of Cab Booking Scenario 36
4.5 List of Triplet Extracted from Business Rules for Cab Booking 37
4.6 List of Activity (Nodes) of Order Fulfillment Process 39
4.7 List of Message Flow and Transition (Edges) of Order Fulfillment Process 39
4.8 List of Edges of Order Fulfillment Process 40
4.9 List of Triplet Extracted from Business Rules for Order Fulfillment Process 41
4.10 Collection of Synthetic Data set of Consistent and Inconsistent Scenarios 42

1

Research Motivation and Aim

In recent years, business process modeling and rule modeling are increasingly been used by the organization for the effective and efficient delivery of the IT system. Some of the high level use of such integration can be found in [1]. Both are equally important for the better quality and reduce cost pressure during the development of a highly efficient IT system.

1.1 Business Process Modeling

Business process modeling [2] bridge the gap between business analyst who design the IT system and technical developers who develop the IT system. There are many business process modeling techniques developed which are used for the business process improvement and restructuring of the business process during the development of the IT system. Organization need process-modeling for various interests like

1. Business process integration and management
2. Evaluation and re-engineering of the processes

Various process-modeling standard like Business Process Modeling and Notation(BPMN), Event Process Chain(EPC)[1] and Unified Modeling Language(UML)[2] have been developed in the recent past. In our research, we use BPMN process-modeling standard.

[1] $https://en.wikipedia.org/wiki/Event-driven_process_chain$
[2] $https://en.wikipedia.org/wiki/Unified_Modeling_Language$

1.1.1 BPMN Process Model: An Overview

Business Process Model and Notation [3, 4] is the state-of-the-art standard for business process diagrams which is developed and maintained by the OMG[1]. The main purpose of BPMN is to provide a notation that is easily understandable by all business users of different expertise who design, manage and monitor the business process. It make things easy for business analyst who design the process, technical developers who understand the design and implement the process, and to the business people and managers who manage and monitor those processes.

Also, BPMN acts as a standard bridge reducing the communication gap that frequently occurs between the business process design and process implementation. Other modeling-oriented standards lack sufficient expressiveness as compared to BPMN. They are also less suited for direct use by the humans for the creation of business process. Figure 1.1 shows the BPMN process diagram for the scenario of travel reservation. Please refer table 1.1 for the text of BPMN diagram.

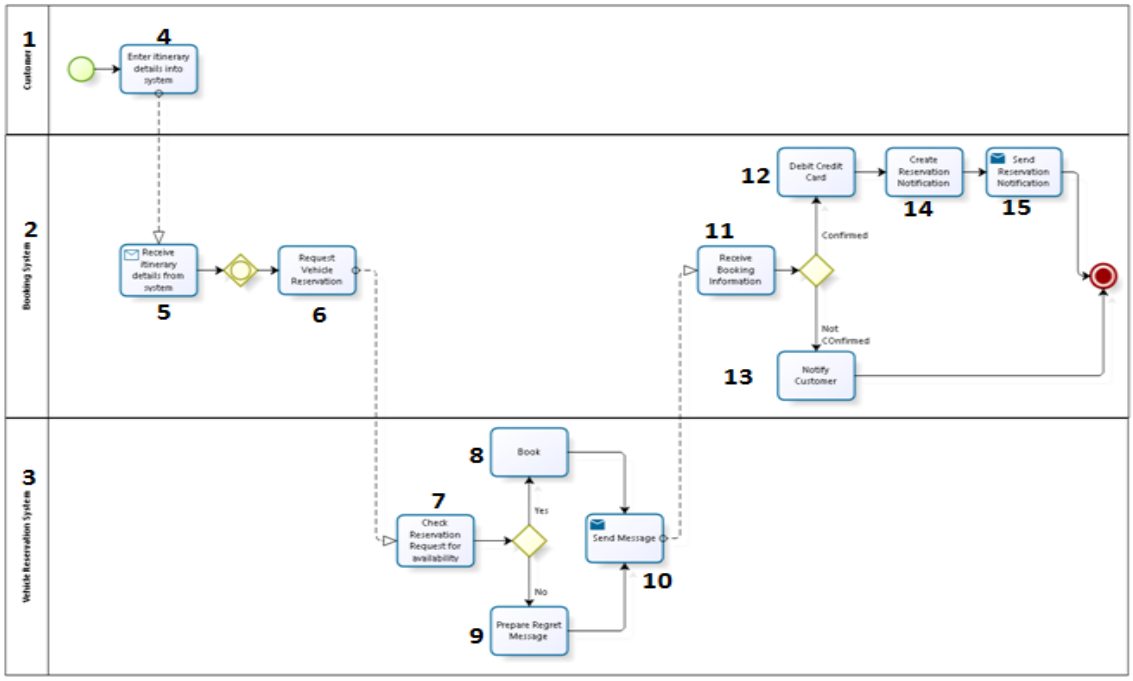


Figure 1.1: BPMN Example

[1]http://www.omg.org/

Table 1.1: Node Label of the BPMN Example Diagram

Node Number	**Node Label**
1	Customer
2	Booking System
3	Vehicle Reservation System
4	Enter itinerary details into system
5	Receive itinerary details from system
6	Request vehicle reservation
7	Check Reservation Request for availability
8	Book
9	Prepare Regret Message
10	Send Message
11	Receive Booking information
12	Debit credit card
13	Notify Customer
14	Create Reservation Notification
15	Send Reservation Notification

1.1.1.1 Basic BPMN Modeling Elements

To understand the BPMN diagrams, it is very important to have knowledge about the set of notations and basic types of elements that is widely used by the creator of the BPMN diagrams. Four basic categories [3, 4] are discussed below:

1. Flow Objects : These are the main graphical elements within BPMN responsible for the behavior of the business process. Three core flow objects are:
    - Event : It denotes something that happens during the process and affects the flow of the process and usually have a trigger or result within a circle to differentiate between trigger or result. Start, Intermediate and End Event constitute the events.
    - Activity : It describes the kind of work that is to be performed during the process. Task and Sub-process are the type of activities that are part of the process model.
    - Gateway : It is used to control the convergence and divergence of the sequence flow in a process and indicate type of behavior expressed.
2. Connecting Objects : They are used to connect Flow Objects to each other. Four connecting objects are as follows:
    - Sequence Flow : It shows the order in which activities are performed in a process.
    - Message Flow : It shows the flow of message between two pools or participants in a process.
    - Association : It is used to associate a text or artifact to a flow object. Directional arrowhead represents an input or output and non directional represents its association with sequence flow or message flow.
3. Swimlanes : It is visual mechanism of organizing and categorising activites based on cross functional flowcharting. Two ways of grouping the modeling elements:
    - Pool : It represents major participants in a process.
    - Lane : It is used to organize and categorize activities within a pool according to a function or role.

4. Artifacts : They are used to provide additional information about the process. Group and Text Annotation are the two standard artifacts.
    - Group : It is used to group different activities within a process but does not affect flow of the process.
    - Text Annotation : It is a mechanism to provide additional text information about the BPMN diagrams.
5. Data Object : It provide information about the data required or produced in an activity.

1.1.1.2 BPMN Modeling Styles

We should follow some of modeling styles during the creation of BPMN diagram as it will lead to better layout and understanding of BPMN diagrams [4, 5]:

1. It is always better to put in some time and optimize the layout of BPMN diagram so that we can avoid crossing flows in the BPMN diagram.
2. Naming Conventions to be followed in the creation of BPMN diagrams:
    - Each activity name must be Verb-Noun Phrase.
    - Each participant name must be Noun.
    - XOR Gateways should be labeled with a question.
3. Higher degree of understanding is possible with the better layout and symmetric structure of a BPMN diagram.
4. Prefer to use equal task sizes as it leads to avoid confusion in the understanding of the BPMN diagram.

1.2 Business Rule Modeling

Business rule modeling is very important for the effective business communication by the business people among the organization. It helps to define business concepts and business vocabulary in an unambiguous manner to make it understandable for the

business people. It focuses on maintaining consistency among all the terms and its meaning from business point of view among the business people.

Business rule[6] defines-

- Semantic of business concepts
- Reactions to business events
- Conditions and preconditions to tasks and activities
- Rights and obligations of business actors

An instance of a business rule in context of License Inspection System is "A driver of a vehicle must have a valid Driver's License".

Business vocabulary[7] contains set of specialized terms and definitions of concepts that a given organization or the community uses in their talking and writing in the course of doing business. Term "driver" in the context of License Inspection System is an instance of business vocabulary.

1.2.1 SBVR Rule Model: An Overview

SBVR business vocabulary and business rule [8, 9] expresses business knowledge in a Controlled Natural Language to ensure that the representation of business concepts and rules is unambiguous and understandable to humans as well as to computer systems. According to the Object Management Group (OMG) , SBVR defines vocabulary and rules for documenting the semantic of business vocabularies, business facts and business rules for the exchange of business vocabularies and business rules among organizations and between software tools. Together vocabulary and rule constitute a shared domain model with the same expressive power of standard ontological languages.

Rules play a very important role in defining business semantics, and influencing the behavior of the business. Rule based approach aims to address two different kinds of user - business community and IT professionals.

In SBVR [9], meaning is separate from expression;Rules are built on Fact types (Verb concepts) and Fact types are built on Terms (Noun Concepts).

SBVR rule statements [9] require elements of model logic as formalization and are expressed either using alethic modality or denotic modality. SBVR **Structural Business rules** use two alethic modal operators:

- *It is possible that...*
- *It is necessary that...*

SBVR **Operative Business Rules** uses two denotic modal operators:

- *It is permitted that...*
- *It is obligatory that...*

1.3 Research Problem

The business process diagram and the business rule are designed in an organization to develop an IT system. The quality of process diagram and business rule is of up most importance to the organization - particularly their consistency. The notion of consistency refers to the absence of modeling faults within a model. As these are human created artifacts, so there is a possibility of inconsistency between the two developed model for the same scenario. Inconsistency comprises of syntactic inconsistency and semantic inconsistency. In an organization both the models are developed using a standard editor for the model so there is no possibility of syntactic error within a model. But it might be possible that the developed IT system does not show an expected behavior. In our research, we are mainly focusing on the detection of instances where the developed IT system does not produce an expected behavior.

BPMN is a graphical model of business processes and SBVR rules provide business rules corresponding to that business process. Though both these formalism should have no inconsistencies, but that is often present. Our research interest lies in converting SBVR rules to graphical representation and using sub graph-isomorphism to detect instances of inconsistencies between BPMN and SBVR model.

1.3.1 Instances of Semantic Inconsistency

We came up with the following 2 real life scenarios that demonstrates the instances of semantic inconsistency between BPMN Process Model and SBVR Rules.

1.3.1.1 Vehicle Reservation System

In this section, we define 2 different instances of semantic inconsistency between process model and rules. Figure1.2 shows an instance of semantic inconsistency for the scenario of vehicle reservation system. Please refer table 1.1 for the text of figure 1.2 BPMN process model shows the process of vehicle reservation by the customer. As this process model represent how the customer perform booking for the vehicle. SBVR Business Vocabulary and Business Rules are written to define vocabulary and rules for the vehicle reservation system. In this BPMN process model, customer of any age can perform reservation. Figure1.3 shows subset of SBVR Rule Model for Vehicle Reservation System. But in SBVR rule, it is mentioned that it is necessary for the customer to be of at least 18 years old to perform reservation. So when both are applied together on the same scenario they will not produce expected behavior.

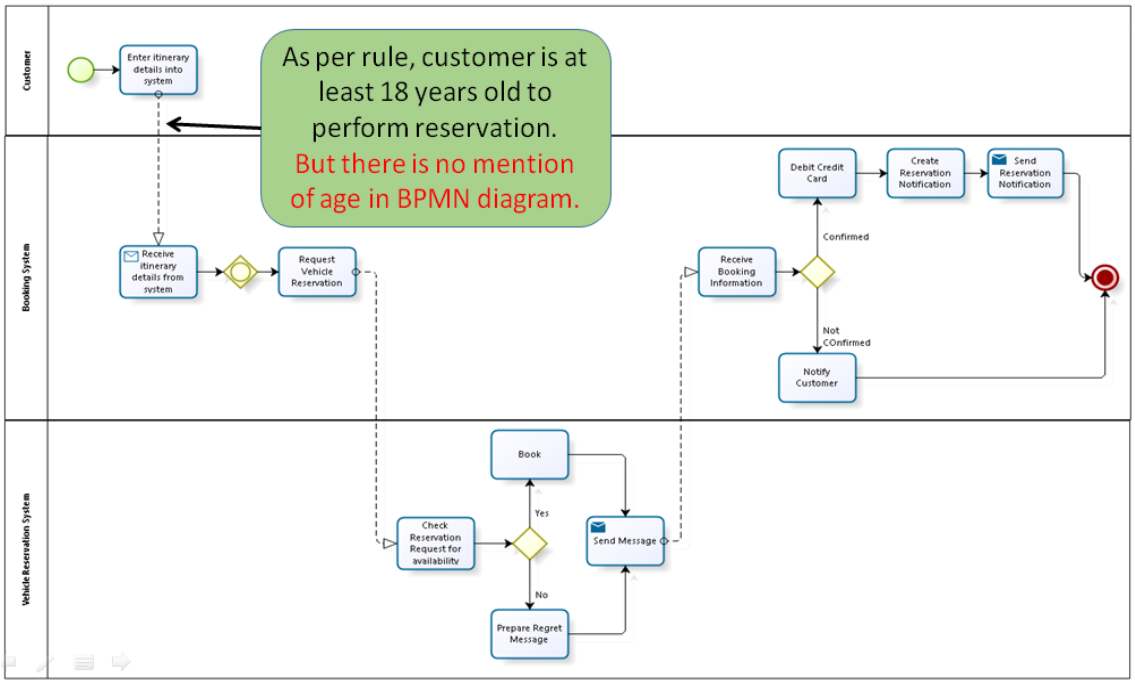


Figure 1.2: An Instance of BPMN Process Model for Vehicle Reservation System

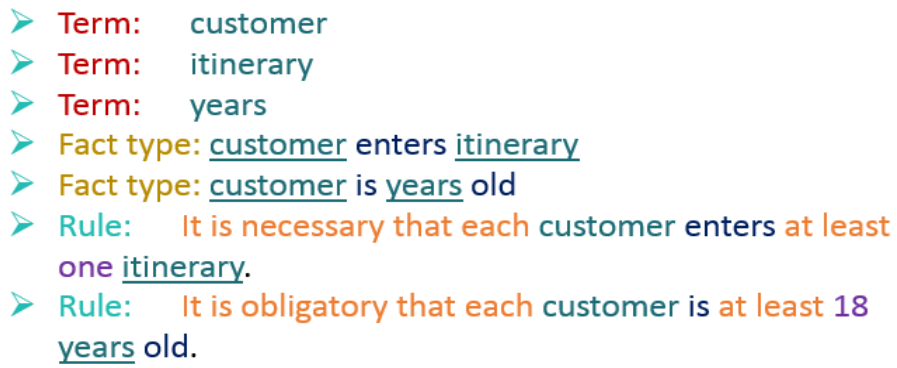


Figure 1.3: Subset of SBVR Rule Model for Vehicle Reservation System

Figure1.4 shows another instance of semantic inconsistency between process model and rules. Please refer table 1.1 for the text of figure 1.4. BPMN Process Model demonstrate the activity involved in the scenario of vehicle reservation system. Figure1.5 shows subset of SBVR Rule Model for the Vehicle Reservation System. In SBVR rule, it is mentioned that there can be different specification of the customer namely gold or silver. And according to the specification of customer, they are eligible for the discount in the payment. But in BPMN Process Model, customer specification is not defined and hence they will not be eligible for discount in the payment.

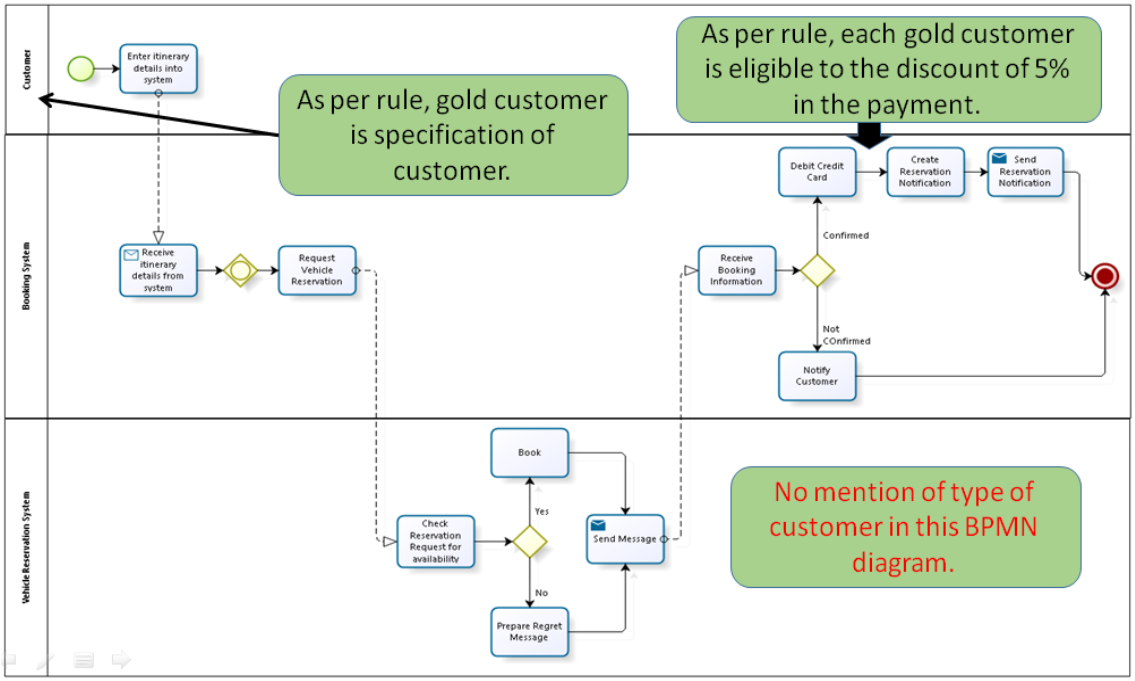


Figure 1.4: An Instance of BPMN Process Model for Vehicle Reservation System

- Term: customer
- Term: gold customer
- Term: payment
- Term: discount
- Fact type: gold customer is specification of customer
- Fact type: gold customer is eligible to discount of five percent in payment
- Rule: It is obligatory that each gold customer is specification of the customer.
- Rule: It is necessary that each gold customer is eligible to the discount of five percent in the payment.

Figure 1.5: Subset of SBVR Rule Model for Vehicle Reservation System

1.3.1.2 Order Fulfillment Process

In this section, we demonstrate the instance of semantic inconsistency between BPMN Process Model defined for the Order Fulfillment Process and SBVR rule specified for the same scenario. Figure1.6 shows the BPMN Process Model for the Order Fulfillment Process. BPMN Process Model depicts the complete flow of activities that are involved in the order fulfillment process. Figure1.7 shows the subset of SBVR rules for the same scenario. In SBVR rule, it is mentioned that it is necessary to check the previous record of the customer before the acceptance of order. But there is no mention about checking how the relationship of customer is with the system.

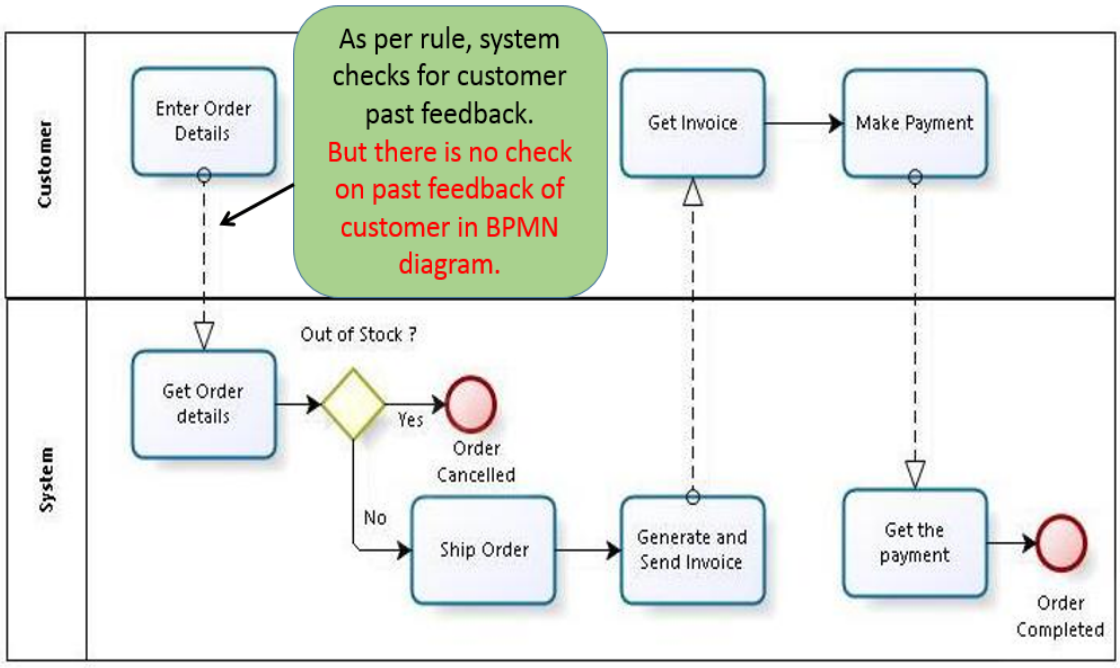


Figure 1.6: An Instance of BPMN Process Model for Order Fulfillment Process

- Term: system
- Term: customer
- Term: order
- Fact type: system receives order
- Fact type: system cancels order
- Fact type: system checks for customer past feedback
- Fact type: system has bad feedback about customer
- Fact type: system checks for order availability if customer has good feedback
- Rule: It is obligatory that the system receives the order and checks for the customer past feedback.
- Rule: It is necessary that the system checks for the order availability if the customer has good feedback.
- Rule: It is possible that the system has bad feedback about the customer and cancels the order.

Figure 1.7: Subset of SBVR Rule Model for Order Fulfillment Process

1.4 Research Aim

The research aim of the work presented in the thesis is the following:

1. To automate the process of detecting semantic inconsistency violation across BPMN Process Model and SBVR rule model.

2. To extract Triplet from a English Statement efficiently and accurately.

3. To develop system to detect faults within models which are human created artifacts.

4. To validate the developed system we conducted experiment on synthetic dataset related to many real world scenarios.

2

Related Work and Research Contributions

In this chapter we discuss previous work closely related to our research and list the novel research contributions of our work in context to already existing work.

2.1 Related Work

We divide related work into following three lines of research:

2.1.1 Integration of BPMN and SBVR

Integration of process model and rule model attracted researchers of research labs several decades ago. One of the first research done related to this field in the year 1991 by Krogstie et al[10]. Muehlen et al[11] proposed a framework relevant to academics and industry but there was a lackof validation in the framework. Sheen et al [12] investigated automatic way to transform business rules to business process and analysed the impact of mode driven technologies. Habich et al[13] proposed an SBVR annotated approach for the modeling of rule and process together. From SBVR annotated approach they developed BPMN process definitation with OCL constraints. Cheng et al[14] discussed about the semantic and structural aspects of BPMN and SBVR. Skersys et al[15] mapped meta models of BPMN with SBVR and related different elements of BPMN with different elements of SBVR. Skersys et al[16] observed that there is a gap between business process modeling and business vocabulary and business rules. They

have proposed a set of principles for the integration of BPMN and SBVR. Sharma et al[17] reviewed business rule meta-model and evolved process template from the same. Skersys et al[7] proposed a method to extract business vocabularies from BPMN process model and they confirmed that it is accurately possible for semi transformation if the process model is developed following defined modeling practices.However, in case of fully transformation there can be affect on extraction efficiency and success rate. Mickeviciute et al[18] observed that business process model should be designed in line with the business rules so that it is possible to transform business process models to business rules. And business vocabulary can be used for the modeling of both the process model and rule model.

2.1.2 Triplet Extraction from Sentence

Rusu et al[19] proposed an approach for the extraction of subject-predicate-object triplet from a sentence using four open-source parsers, namely Stanford NLP, OpenNLP, Link Parser and Minipar. Parser generates a parse tree and set is extracted from the parse tree using parser dependent techniques. Dali et al[20] proposed machine learning approach to extract subject-predicate-object triplets from English sentences. They used SVM to train the model using human annotated sets and then used multiple parsers to evaluate the eifficency of the approach. They calculated scores using different feature types. Zin Thu Thu Muint et al in [21] observed that other techniques developed are using one or the other parser affecting processing time for each sentences. They proposed an approach to extract triplets from the sentence with the help of domain specific ontology and machine readable dictionary WordNet

2.1.3 Graph Matching Algorithm

Ullmann[22] proposed the matching algorithm for the graph isomorphism and sub graph isomorphism. This algorithm is based on backtracking procedure to reduce space search with an effective look ahead function. It is one of the most widely used algorithm as it is independent on the topology of the graph so it can be easily scaled.

Another algorithm related to sub graph isomorphism is proposed by Cordella et al[23]. They introduced a unique representation for matching process known as state space representation. Each state of matching process contains partial mapping solution and each transition from one state to another state add one new pair of matched nodes

to the partial mapping solution. They also introduced a set of feasibility rules in this algorithm which helps in pruning the states from partial matching solution which do not satisfy the graph isomorphism.

Cordella et al[24] proposed another work about the performance of the graph matching algorithm popularly known as VF Algorithm[23]. They have done theoretical analysis of the algorithm about the computational complexity and memory requirements.

Cordella et al[25, 26] proposed an improved version of graph matching algorithm to solve graph isomorphism and graph-sub graph isomorphism problems on attribute relational graphs. They have focused on the semantic information of the graph and tried to organize search space in such a way to reduce memory requirements for large and medium sized graphs. This algorithm is popularly known as VF2 Algorithm developed to improving the VF Algorithm[23] by reducing spatial complexity to O(N) in best and worst case.

2.2 Novel Research Contributions

In context to existing work and to the best of our knowledge, the study presented in this thesis makes the following novel contributions:

1. There have been many research work done with the integration of BPMN and SBVR but none of them have focused on the model quality characteristic i.e. consistency. Our work is the first in which we are focusing on the notion of consistency between the BPMN Process Model and SBVR Rule Model.

2. While there has been work done in the area of triplet (actor-action-object) extraction, our work is the first that used grammatical relations between words stored in the form of Typed Dependency to extract the actor, action and object of a sentence.

3. While there has been work done in the integration of BPMN and SBVR, but there is no dataset available in combination of both. In our work, we have proposed a synthetic dataset for the research problem in integration of BPMN Process Model and SBVR Rule Model.

4. We propose a novel research framework to detect the instances of semantic inconsistency between BPMN Process Model and SBVR Rule Model. We demonstrate the effectiveness of our approach, discuss the strengths and limitations of our tool based on conducting experiments on synthetic dataset.

3

Research Framework and Solution Approach

Figure 3.1 shows the high level architecture diagram of the proposed solution approach. The proposed approach is a multi-step process primarily consists of 3 phases: conversion of BPMN diagram to graph, conversion of SBVR rule set to graph and apply graph isomorphism algorithm on the two generated graph.

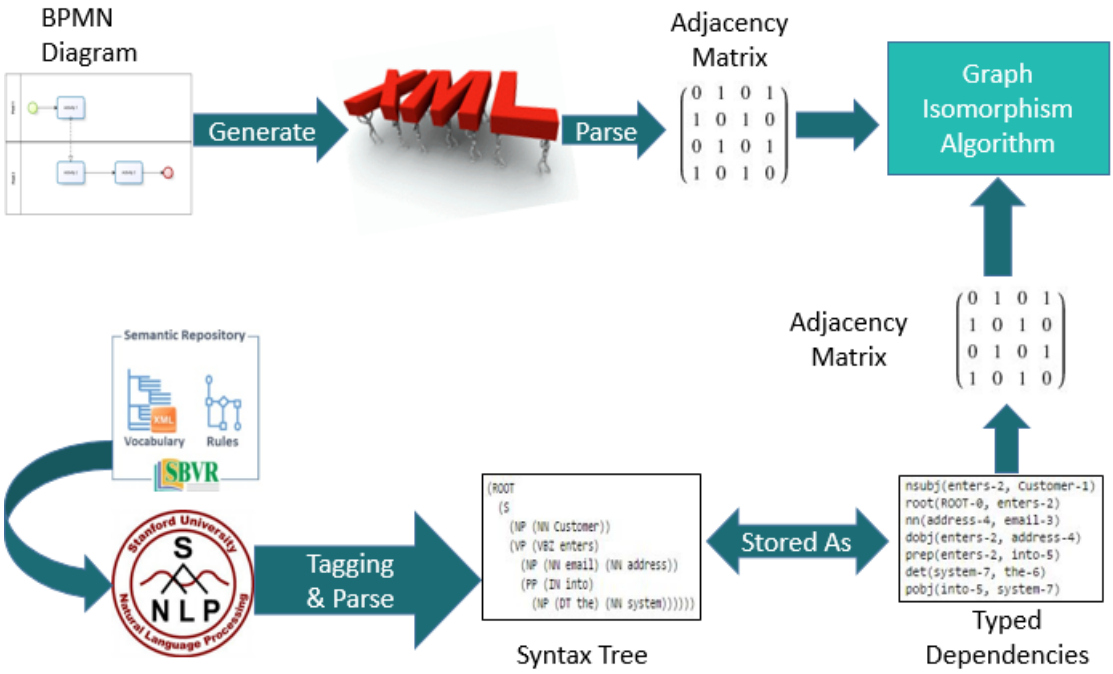


Figure 3.1: Research Framework

3.1 XML based BPMN Graph

First phase involves the generation of XML representation of the BPMN diagram using one of the popular modeling tool for BPMN namely Bizagi Modeler[1]. We are parsing generated XML file and extracting values of tags namely Activity, Message Flow and Transitions. We make use of these extracted values to identify nodes and edges of the graph. Activity are considered as nodes of the graph. And, Message Flow and Transitions are considered as edges of the graph.

The pseudo-code for XML based BPMN Graph algorithm is shown in Algorithm 1. The input to the Algorithm is an XML file containing complete information about BPMN diagram. This XML file comprises of various tags like Pools, Lanes, Activity, Message Flow, Associations, Artifacts, WorkFlowProcesses, Transition, and many more. Each tag comprises of several attributes like Name, Id, From (optional) and To (optional) . We are focusing only on the tags namely Activity, Message Flow and Transition. The algorithm returns nodes and edges of the graph in the form of adjacency matrix which is the representation of the graph.

Algorithm 1: XML based Adjacency Matrix

Data: XML of BPMN Diagram
Result: Adjacency Matrix

1 Parse XML file and Extract Name and Id of the tags
2 Identify all the Activity in a process - these are nodes of the graph
3 Identify the Message Flow between pools and the Transitions within pools - these are edges of the graph
4 Adjacency Matrix has activities as the nodes of the matrix
5 Each entry in the matrix will be 1 if there is Message Flow or Transitions between two activities representing row and column. Otherwise corresponding entry will be 0.

Figure 3.2 depicts the BPMN diagram for the travel scenario. Please refer table 1.1 for the text of figure 3.2. In this, Message Flow represents the flow of message between activity of two different pools or participants in a process. Transitions represents the flow of message between activity of same pool or participant in a process.

[1]http://www.bizagi.com/

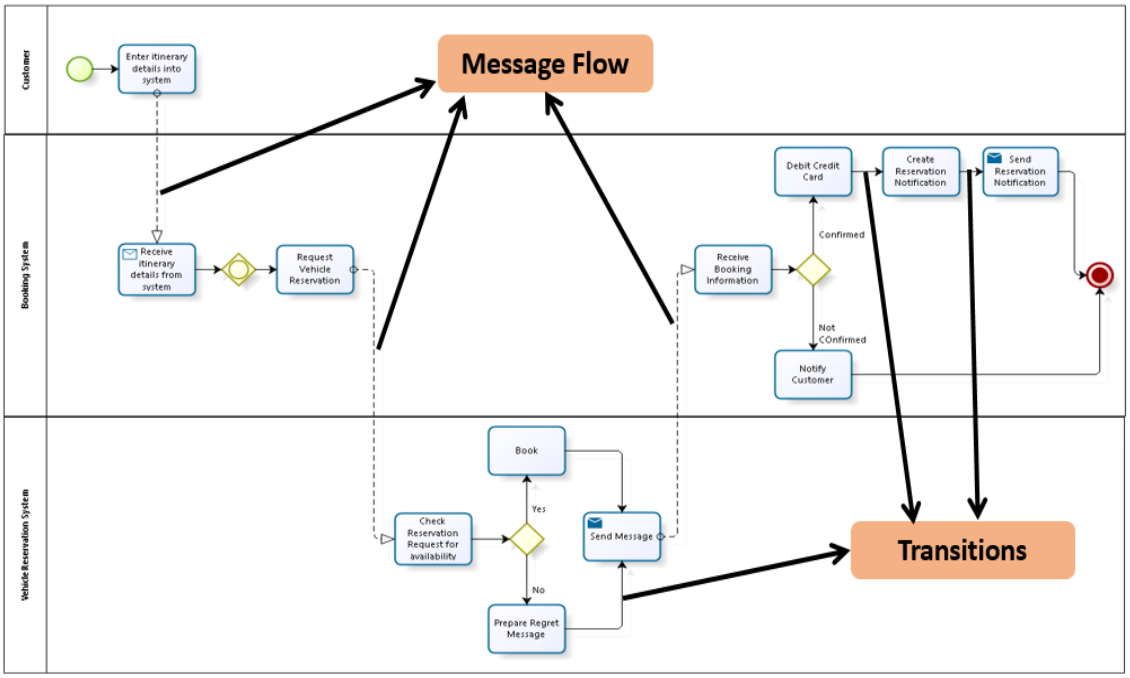


Figure 3.2: Sample BPMN Process Diagram for Travel Reservation

Figure 3.3 depicts the subset of generated XML file for the Travel Reservation BPMN Process Model. It represent activity of the BPMN Process Model which are considered as nodes of the graph.

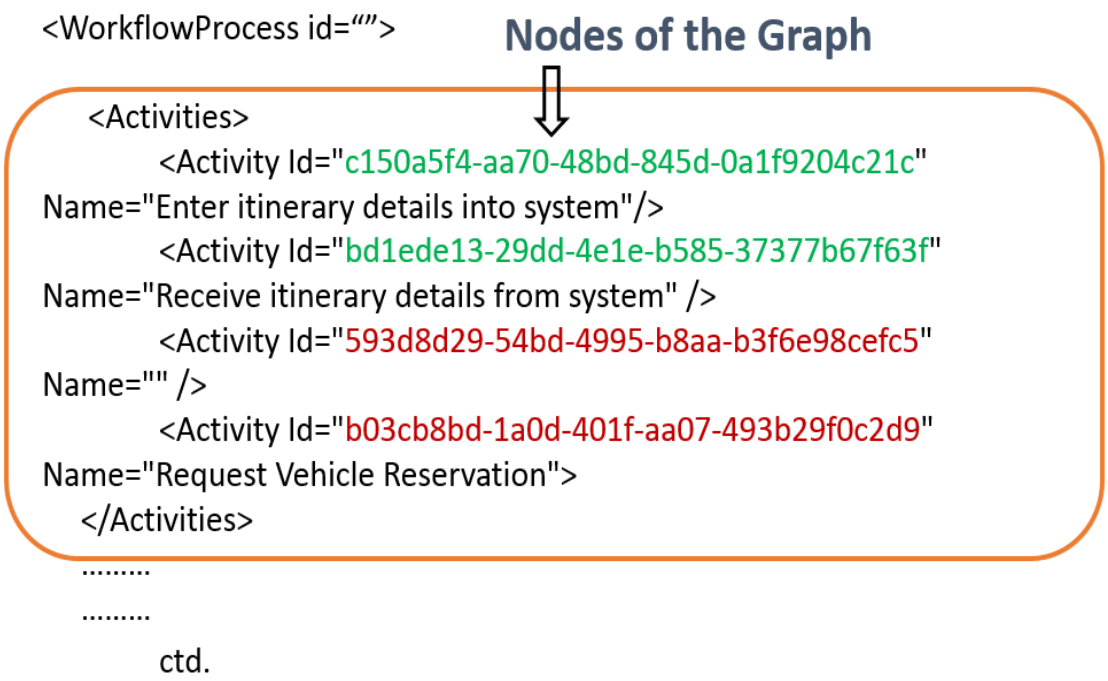


Figure 3.3: Subset of Generated XML file for Travel Reservation and represents Activity of the BPMN Process Model

Figure 3.4 and 3.5 indicates the subset of generated XML file and shows the MessageFlow and Transitions of the BPMN Process Model which are considered as Edges of the graph.

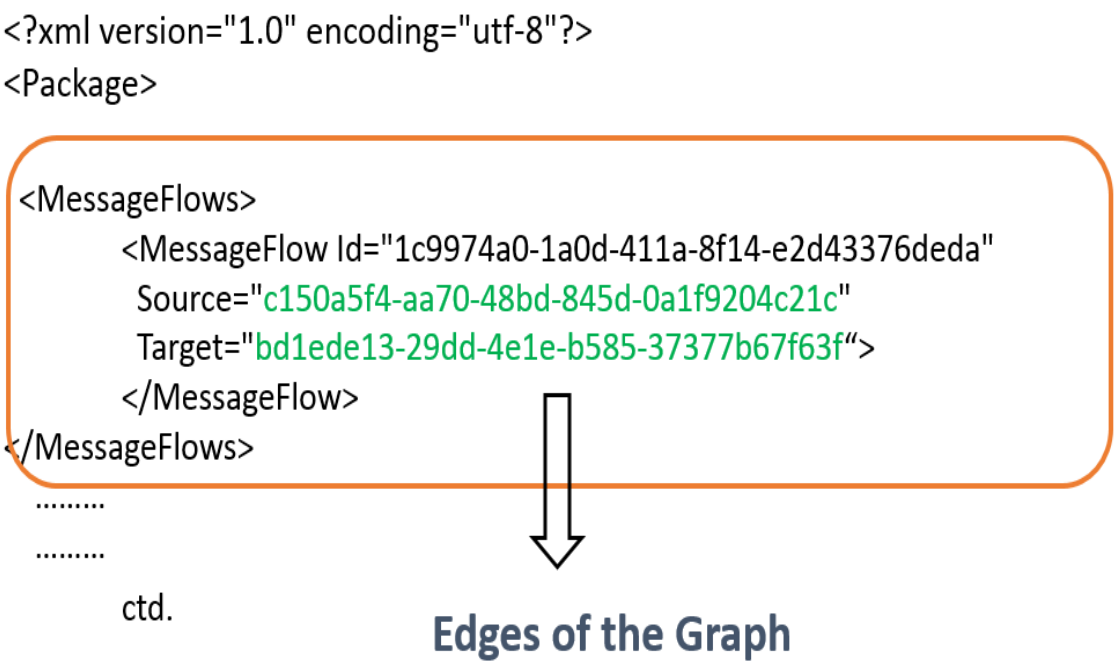


Figure 3.4: Subset of Generated XML file for Travel Reservation and represents MessageFlow of the BPMN Process Model

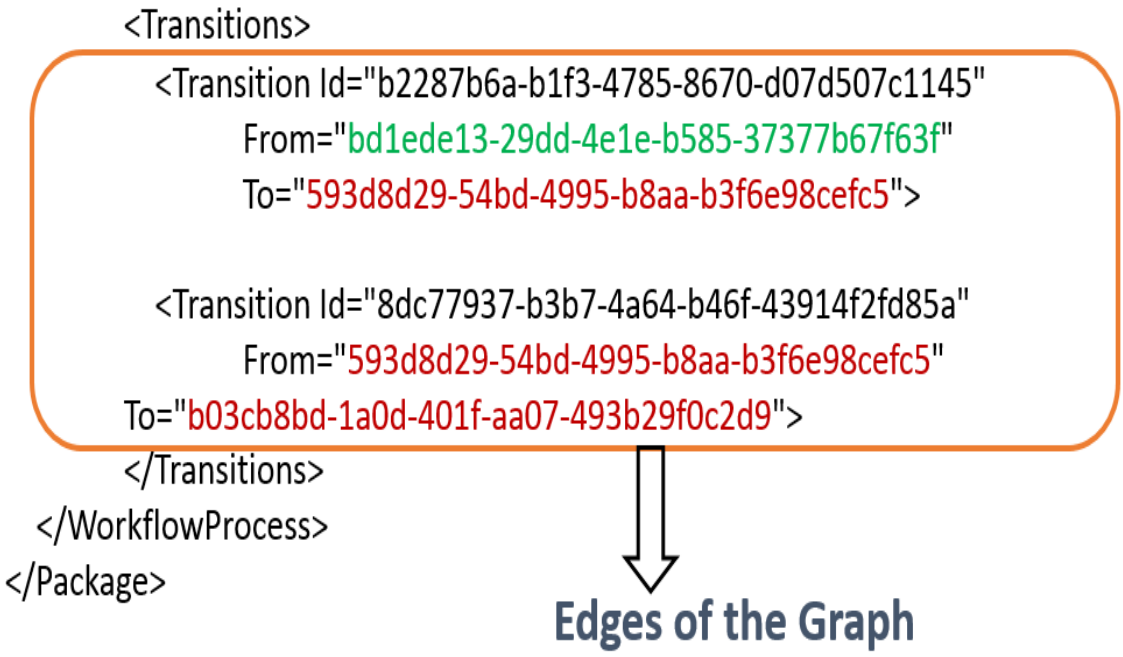
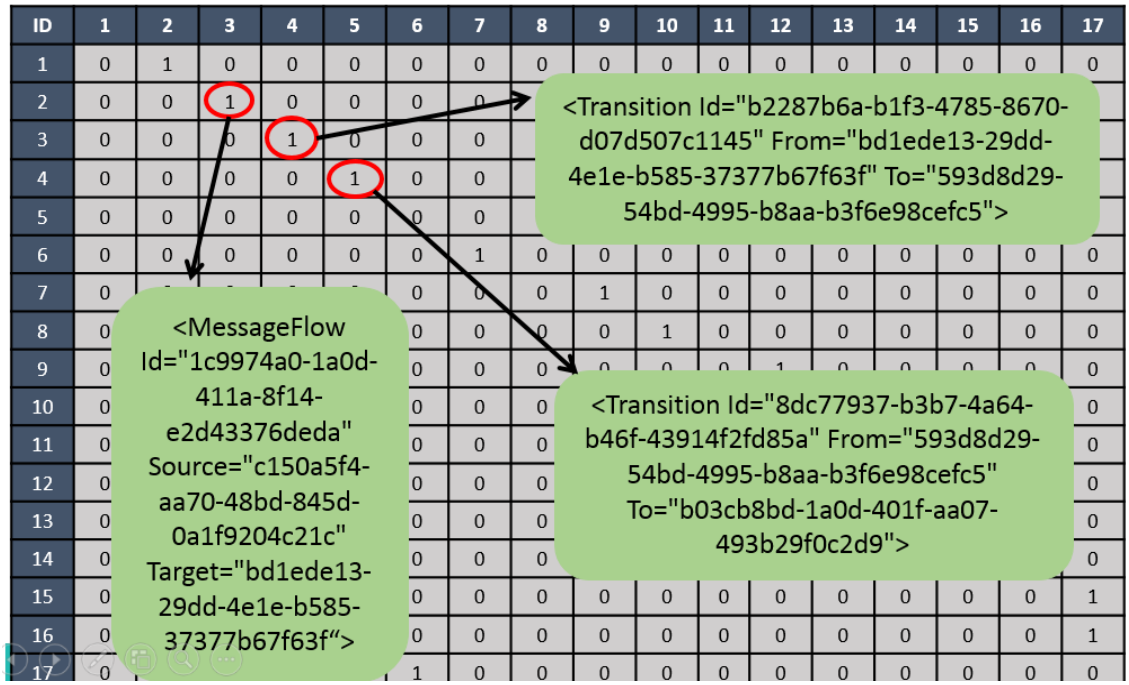


Figure 3.5: Subset of Generated XML file for Travel Reservation and represents Transitions of the BPMN Process Model



ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0										
3	0	0	0	1	0	0	0										
4	0	0	0	0	1	0	0										
5	0	0	0	0	0	0	0										
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
7	0					0	0	0	1	0	0	0	0	0	0	0	0
8	0					0	0	0	0	1	0	0	0	0	0	0	0
9	0					0	0	0									0
10	0					0	0	0									0
11	0					0	0	0									0
12	0					0	0	0									0
13	0					0	0	0									0
14	0					0	0	0									0
15	0					0	0	0	0	0	0	0	0	0	0	0	1
16	0					0	0	0	0	0	0	0	0	0	0	0	1
17	0					1	0	0	0	0	0	0	0	0	0	0	0

Figure 3.6: Adjacency Matrix for BPMN Process Model

Figure 3.6 depicts the adjacency matrix representation of graph for the BPMN Process Model. An entry in the matrix as 1 and 0 represent presence and absence of edge between the two nodes. Figure 3.7 demonstrate the graph representation of BPMN Process Model for the BPMN diagram of Travel Reservation as shown in figure 3.2.

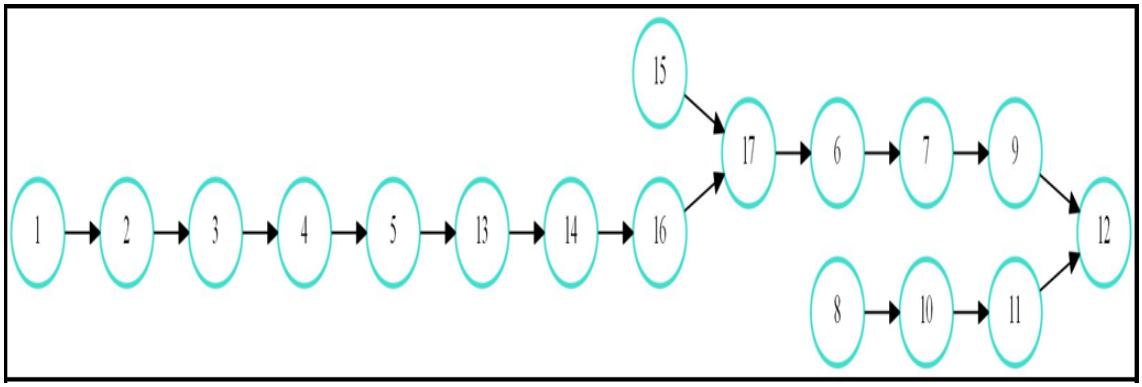


Figure 3.7: Graph Representation of BPMN Process Model for Travel Reservation

3.2 Typed Dependency Based Triplet Extraction

Second phase of our proposed solution approach focus on generation of graph from SBVR rule set. SBVR rule broadly consists of action-oriented and structure-oriented rules. We are mainly focusing on the action-oriented rules. SBVR rules are written in a natural language. We are considering only those rules which are action-oriented. We generated phase structure parses of the rule using the Stanford NLP parser[1]. Then, we extracted typed dependency parses of English sentences which represent the dependency between individual words of a sentence. Also, a typed dependency parse labels these dependencies with the grammatical relations. We are extracting triplet (Subject-Verb-Object) from a rule using the mentioned grammatical relation. Identification of rule as action-oriented or structure-oriented is done on the basis of whether the verb used in a rule is transitive or intransitive.

This phase of our solution approach consists of further three sub phases: (1) Triplet Extraction from sentences based on Typed Dependency, (2) Classification of rule as action-oriented or structure-oriented based on Triplet Extracted, (3) Generation of Graph using action-oriented rules

First sub-phase of second phase involves the generation of parse tree of rule. The information of parse tree is stored in the form of Typed Dependency which represents grammatical relations between words of a sentence. The pseudo-code for Triplet Extraction from a rule based on Typed Dependency of Stanford NLP Parser is shown in Algorithm 2. The input to the algorithm is the file comprising of rules on a scenario. The algorithm returns the triplet (Actor-Action-Object) of each rule.

Figure 3.8 depicts how the Typed Dependency are generated for an English Sentence. As shown in the Figure, there is a rule written in Natural Language. We used Stanford Parser, a Natural Language Parser API to tag English Sentence, then generate a parse tree and that parse tree information is stored in the form of Typed Dependency which represent grammatical relations between two words of a English Sentence.

[1]http://nlp.stanford.edu:8080/parser/

Algorithm 2: Typed Dependency based Triplet Extraction

Data: Rule written in Natural Language
Result: Triplet Extraction from Sentences

```
1  foreach Rule in File do
2      foreach Object in List do
3          if DependencyName not equals "mark" then
4              if DependencyName equals "root" then
5                  add Node String to Verb
6              if DependencyName equals "nsubj" or "nsubjpass" then
7                  if DependencyName preceeded by "compound" then
8                      Concatenate compound and Node String
9                      add Node String to Subject
10                 else
11                     add Node String to Subject
12             if DependencyName equals "dobj" then
13                 if DependencyName preceeded by "compound" then
14                     Concatenate compound and Node String
15                     add Node String to Object
16                 else
17                     add Node String to Object
18         else
19             if DependencyName equals "nsubj" then
20                 add subject governer to SubjectGoverner
21                 if DependencyName preceeded by "compound" then
22                     Concatenate compound and Node String
23                     add Node String to Subject
24                 else
25                     add Node String to Subject
26             if DependencyName equals "dobj" then
27                 add object governer to ObjectGoverner
28                 if SubjectGoverner equals ObjectGoverner then
29                     add ObjectGoverner to Verb
30                 if DependencyName preceeded by "compound" then
31                     Concatenate compound and Node String
32                     add Node String to Object
33                 else
34                     add Node String to Object
```

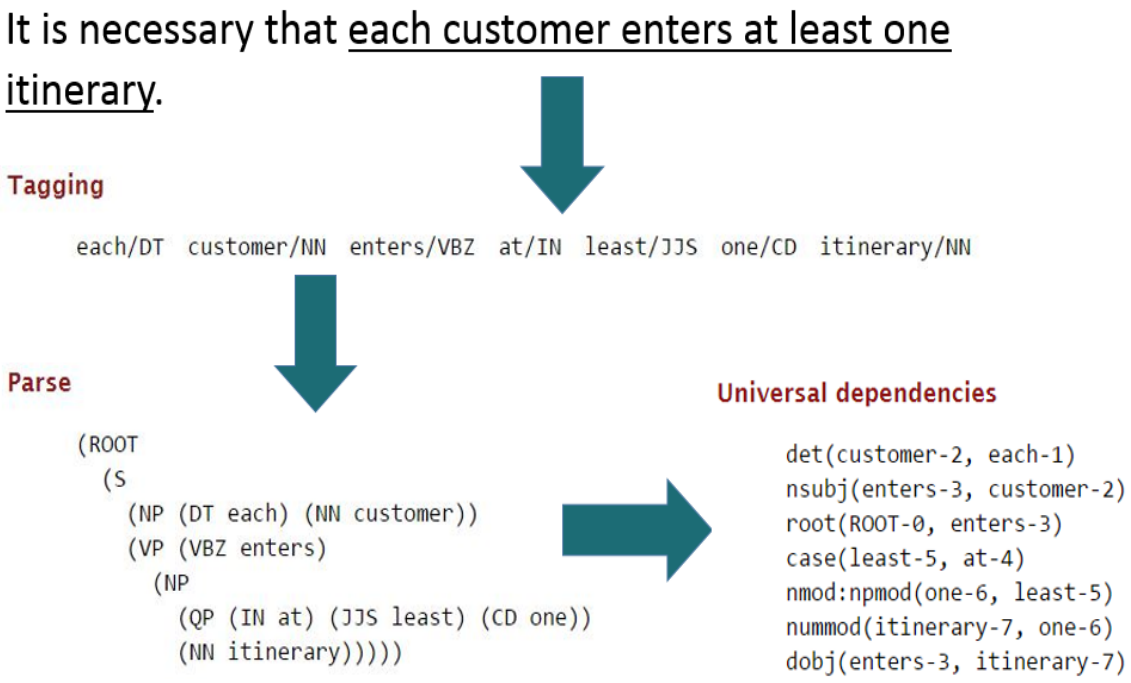


Figure 3.8: Generation of Typed Dependency for a English Sentence

Second sub-phase of second phase involves the classification of a rule as action-oriented or structure-oriented on the basis of the verb used in a rule is transitive or intransitive. The pseudo-code for the classification of rule is shown in Algorithm 3. The input to the algorithm is the triplet of a rule. The algorithm returns triplet of rules which are action-oriented.

Figure 3.9 depicts the classification of a rule as action oriented or structure oriented depending on the type of verb. As in this example, we extracted subject, verb and object using Dependency Name namely nsubj, root and dobj respectively. There is direct object and action-verb used in this rule, hence the verb is classified as transitive Verb. Thus, this rule participate in the business process.

Algorithm 3: Triplet based Rule Classification

Data: Actor-Action-Object of a rule

Result: Classification of a rule

```
Read a file
foreach Rule in File do
    Create an arraylist for an Actor, an Object and an Action
    if Actor != Null and Verb != NULL and Object != NULL then
        add Actor, Verb and Object to arraylist
        Transitive Verb
        Rule participate in Business Process
    else
        Intransitive Verb
        Rule do not participate in Business Process
```

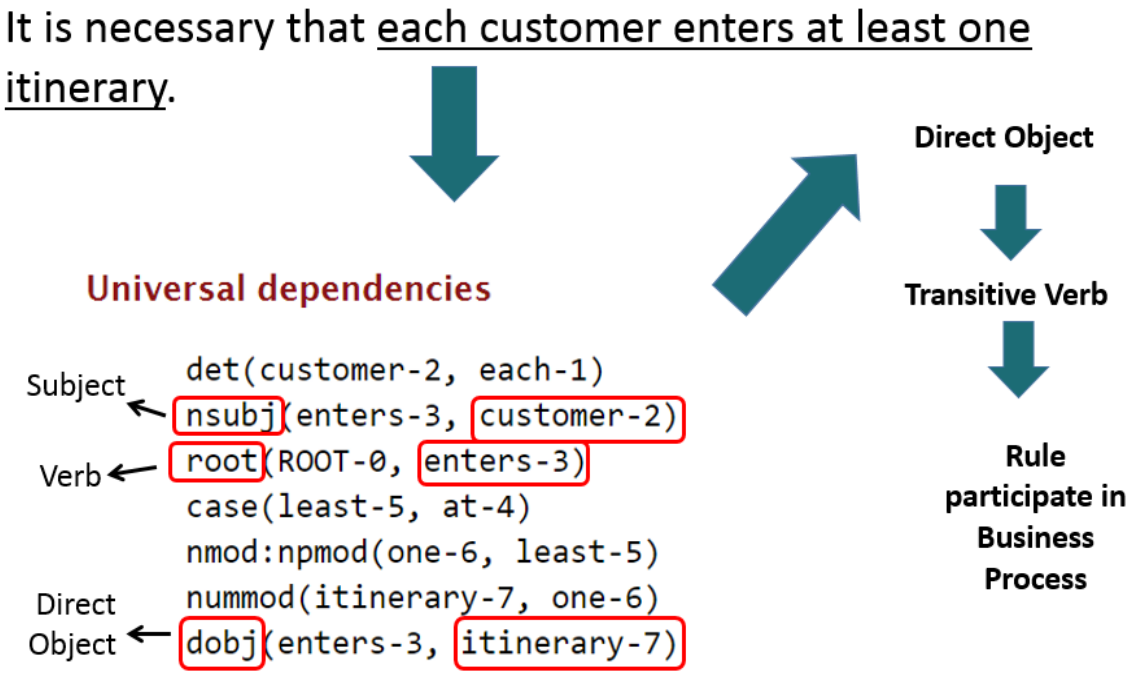


Figure 3.9: Classification of Rule (Action-Oriented)

Figure 3.10 depicts the classification of structure oriented rule. As in this example there is no direct object for the verb to act upon, hence this verb will be classified as intransitive verb. Thus, this rule do not participate in the business process.

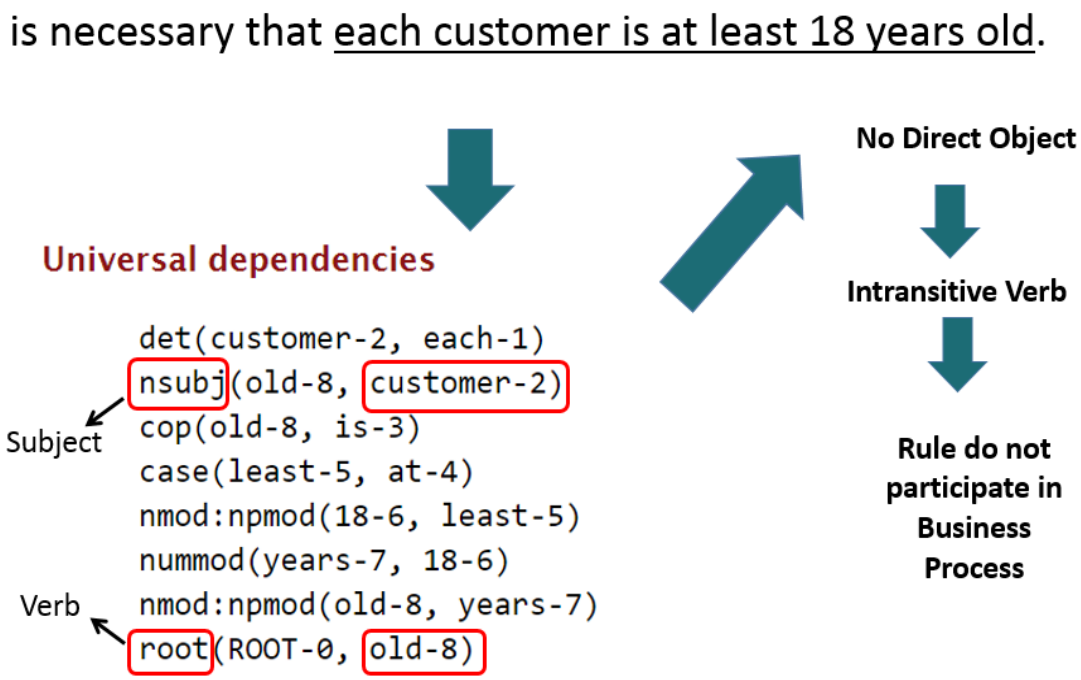


Figure 3.10: Classification of Rule (Structure-Oriented)

Third sub-phase of second phase involves the generation of a graph considering action-oriented rules. Actor and Object are considered as nodes of the graph. Action represent the edge from the actor to the object. We are making a list of edges in the form of A,B where A and B represents Actor or Object. Thus, we obtained adjacency list of SBVR rule-set which is the representation of the graph.

Table 3.1 represents the triplet extracted for the action-oriented rules. We then generate graph for action-oriented rules considering the actor and object as nodes of the graph and action represents an edge from actor to object as shown in Figure 3.11.

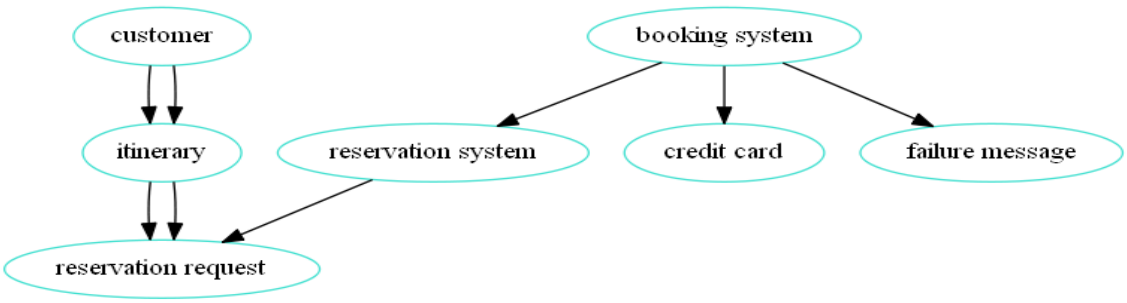


Figure 3.11: Graph Representation of SBVR Rule Model

Table 3.1: List of Triplet Extracted from Business Rules for Cab Booking

Actor	**Action**	**Object**
Customer	Enters	Itinerary
Itinerary	Has	Reservation request
Booking system	Contacts	Reservation System
Itinerary	Has	Reservation request
Reservation system	Confirms	Reservation request
Booking system	Debits	Credit card
Booking system	Receives	Failure message
Customer	Revises	Itinerary

3.3 Subgraph Graph Algorithm: VF2 Algorithm

Third and final phase of our solution approach focuses on the use of graph isomorphism algorithm to compare two graphs generated by the initial two phases. VF2 Algorithm is one of the popular and most widely used sub graph-graph isomorphism algorithm that is used to compare sub graph of a graph with the another graph to identify whether they are isomorphic or sub isomorphic to each other. We are generating all possible sub graph of both the graphs and then comparing them to identify if the subset of business process diagram is isomorphic or sub isomorphic to the subset of business rule. We then identify which part of BPMN diagram is isomorphic to which part of SBVR rule.

The pseudo-code of VF2 Matching Algorithm [25, 26] is shown in Algorithm 4. The input to the algorithm is the initial state and intermediate state. Algorithm returns the mapping between two graphs.

Lets assume we have two graphs V and V′. Graph V and V′ is shown in figure 3.12 and figure 3.13 respectively. We want to check if graph V and graph V′ are isomorphic. We will use the notation as XV is the node X in V. We will try to match each node in V with a node in V′. Following steps shows the working of Vf2 Algorithm: In Step 1, we will match empty V with empty V′. It will always work. In Step 2, we will try to match 1V with any one of the node, 1V′, 2V′, or 3V′. We will match 1V with 1V′. It will always work. In step 3, we will match 2V with any one of the node, 2V′, or 3V′. We will match 2V with 2V′. It will work because 1V 2V and 1V′ 2V′ are isomorphic. In step 4, we will try to match 3V with 3V′. We can't match it because there is no

Algorithm 4: PROCEDURE Match(s)

Data: an intermediate state s; the initial state s_0 has $M(s_0) = \phi$

Result: the mappings between the two graphs

```
1  if M(s) covers all nodes of G2 then
2      OUTPUT M(s)
3  else
4      Compute the set P(s) of the pairs candidate for inclusion in M(s)
5      foreach p in P(s) do
6          if the feasibility rules succeed for the inclusion of p in M(s) then
7              Compute the state s' obtained by adding p to M(s)
8              CALL Match(s')
9      Restore Data Strucutres
10 END PROCEDURE Match
```

edge in Graph V′ between node 2 and node 3. We will traverse back to step 3 where there was mapping between two nodes. In step 5, we will match 2V with 3V′. In step 6, when we try to match further 3V with any node in V′, it is not possible. So we will go back to step 2, but there is again no solution., so we will go back to step 1. In step 7, we will try to match 1V with 2V′. It will work. In step 8, we will try to match 2V with 1V′. It will work. In step 9, we will try to match 3V with 3V′. It will match (1V 2V 3V) with (2V′ 1V′ 3V′), so the graphs V and V′ are isomorphic.

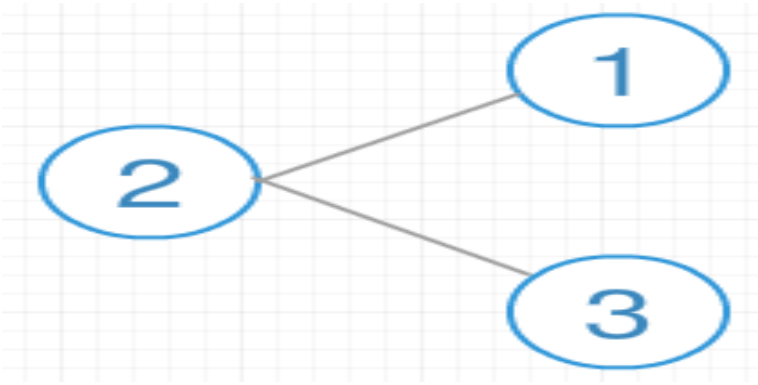


Figure 3.12: Graph V

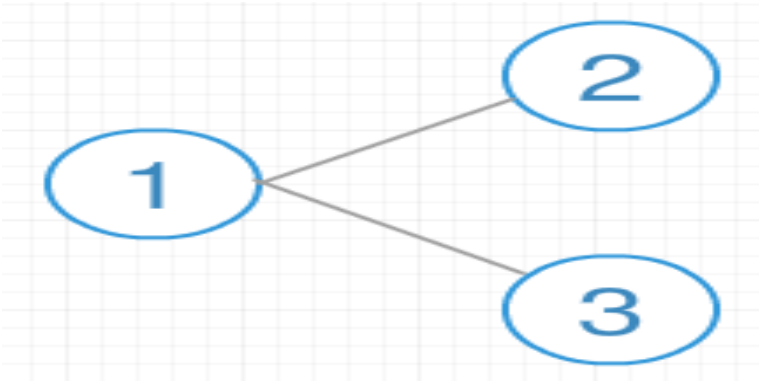


Figure 3.13: Graph V′

The pseudo-code for finding sub isomorphic pairs of mapping of nodes of one sub graph to another sub graph is shown in Algorithm 5. The input to the algorithm is the two adjacency lists generated by the intial two phases. Algorithm returns the isomorphic mapping pairs of sub graphs.

Algorithm 5: Subgraph-Subgraph Mapping Algorithm

Data: Two Adjacency Lists
Result: mapping between two graph which are isomorphic

```
1 Create a Graph G1 by reading file containing adjacency list of first graph
2 Create a Graph G2 by reading file containing adjacency list of second graph
3 Find node-induced subgraph of all possible combination of length of nodes of G1
4 foreach subgraph G′ of G1 do
5     Find subgraph of all possible combination of length of nodes of G2
6     foreach subgraph G″ of G2 do
7         Perform Matching between subgraph G′ and G″
8         if (G′,G″) are isomorphic then
9             Print Mapping between (G′,G″)
```

4

Experimental Analysis and Results

4.1 Experimental Dataset

We have created experimental test bed which constitutes small synthetic dataset depicting multiple scenarios covering various aspect of daily life. We have covered almost all sectors ranging from business to education, financial to health care and many more. We have designed dataset for both the consistent and inconsistent scenarios. A brief introduction about the scenarios involved in conducting experiment as follows:

4.1.1 Travel Reservation

We conducted experiment on a small synthetic dataset to identify the inconsistency between the BPMN Process Model and SBVR Rule Model. Travel booking process illustrates about how customer perform vehicle, train and hotel reservation. A customer must be at least 15 years of old to make a reservation. He or she enters itinerary details into the booking system. Booking system checks itinerary details for the availability of the entered details. If itinerary is available and booking is successful, then debit credit card of the customer. Generate the reservation notification and send the reservation notification to the customer. If the booking is unsuccessful, then notify the customer and check if the customer wants to change the itinerary details.

4.1.2 Order Fulfillment

It represents an end to end fulfillment process from order taking to product shipment. System takes an order and check if the entire ordered article is available. If the entire order is available and customer has decent past record, it ships the order to the customer, otherwise the order is cancelled. System also generates the invoice for the order and sends it to the customer along with the order. Order is completed when the customer pays the invoice. In either cases, customer is notified about the order.

4.1.3 Employee Reimbursement Request

A small synthetic dataset is designed to represent the activities required to accept, analyze, approve and pay an expense statement submitted by an employee of the organization. Depending on the amount mentioned in the expense statement, it is reimbursed to the employee or transferred to the higher authority to see in detail about the matter. Higher authority advises the employee if the reimbursement is not possible. Else amount the transferred to the employee account.

4.1.4 Cab Booking

This scenario depicts the sequences of activities that take place during the cab bookin gprocess. It shows the communication between customer, travel agent and cab driver. Booking is initiated by the customer by requesting for a cab. Travel agent checks the availability of the cab and inform the availability status to the customer. If the cab is available and customer confirms the cab booking, then travel agent assigns the cab driver for the customer. Cab driver picks up the customer and drops him or her at scheduled destination.

Table 4.1 shows the synthetic dataset for the research problem in integration of BPMN Process Model and SBVR Rule Model. Table 4.1 lists all the different types of scenario depicting consistent or inconsistent ones.

Table 4.1: Collection of Synthetic Data set of Consistent and Inconsistent Scenarios

SNo.	Scenario	Consistent Scenario
1	Travel Reservation	No
2	Cab Booking	Yes
3	Employee Reimbursement Request	Yes
4	Order Fulfillment	No

4.2 Experimental Results

4.2.1 Consistent Scenario : Cab Booking Result

We perform experiment on cab booking scenario to detect whether it is possible to identify consistency or inconsistency between BPMN and SBVR. We designed BPMN diagram considering defined modeling practices using popular BPMN Editor[1]. Figure 4.1 depicts the BPMN process model for the cab booking scenario.

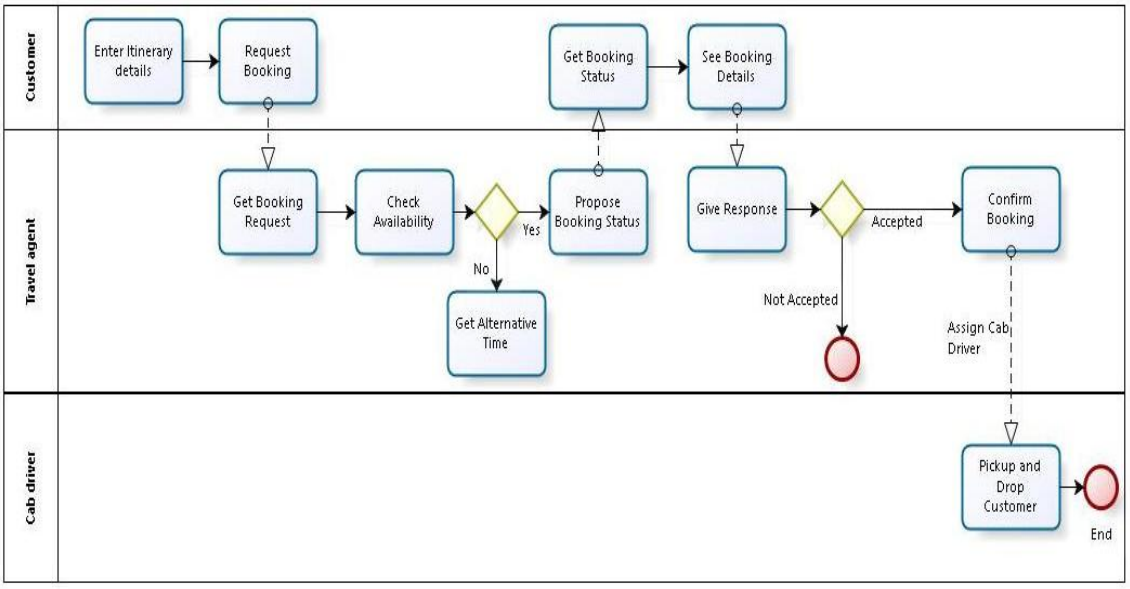


Figure 4.1: BPMN Diagram for Cab Booking

Further, we used Bizagi Process Modeler to generate XPDL of the BPMN diagram. We extracted tags namely Activity, Message Flow and Transition. We assign numbering

[1]http://www.bizagi.com/

Table 4.2: List of Activity (Nodes) of Cab Booking Scenario

Activity ID	Number
839b6582-dfb0-4c74-9115-999630225964	1
1f142692-4b9c-42bd-9507-6b0f1b2a245c	2
587a2fb7-58be-49f9-8fdc-04d28eb9b161	3
9ca0e033-0d04-467e-a3cc-41f652981cb3	4
6e92b4d3-a87a-41d6-82a8-48eb107dba58	5
8f71ea67-d2d8-45d4-9082-c4bed32e3cfc	6
d1625337-5235-4fa3-8a31-22bbcc16ce9e	7
5d116fee-2c01-4616-8f24-64b51352be7a	8
733b1c38-a17d-4349-b814-1cab5c914336	9
568469a9-3e2c-4a86-8dde-7c678369c3c2	10
139271c9-9b7c-4728-806f-60fdd21e2232	11
52fc9249-c95c-4171-bc07-d57d812b05f3	12
52d843ba-623f-4d5f-8c7f-d2bc79431911	13
e0c4df79-0d6a-40df-8489-531939ae4f26	14
a8ae54ee-d6eb-4ccb-9219-908127c7fdba	15

Table 4.3: List of Message Flow and Transition (Edges) of Cab Booking Scenario

Activity From	Activity To
52fc9249-c95c-4171-bc07-d57d812b05f3	e0c4df79-0d6a-40df-8489-531939ae4f26
587a2fb7-58be-49f9-8fdc-04d28eb9b161	9ca0e033-0d04-467e-a3cc-41f652981cb3
9ca0e033-0d04-467e-a3cc-41f652981cb3	568469a9-3e2c-4a86-8dde-7c678369c3c2
139271c9-9b7c-4728-806f-60fdd21e2232	52d843ba-623f-4d5f-8c7f-d2bc79431911
6e92b4d3-a87a-41d6-82a8-48eb107dba58	8f71ea67-d2d8-45d4-9082-c4bed32e3cfc
1f142692-4b9c-42bd-9507-6b0f1b2a245c	6e92b4d3-a87a-41d6-82a8-48eb107dba58
e0c4df79-0d6a-40df-8489-531939ae4f26	a8ae54ee-d6eb-4ccb-9219-908127c7fdba
5d116fee-2c01-4616-8f24-64b51352be7a	587a2fb7-58be-49f9-8fdc-04d28eb9b161
8f71ea67-d2d8-45d4-9082-c4bed32e3cfc	d1625337-5235-4fa3-8a31-22bbcc16ce9e
d1625337-5235-4fa3-8a31-22bbcc16ce9e	733b1c38-a17d-4349-b814-1cab5c914336
568469a9-3e2c-4a86-8dde-7c678369c3c2	139271c9-9b7c-4728-806f-60fdd21e2232
839b6582-dfb0-4c74-9115-999630225964	1f142692-4b9c-42bd-9507-6b0f1b2a245c

Table 4.4: List of Edges of Cab Booking Scenario

Activity From	Activity To
12	14
3	4
4	10
11	13
5	6
2	5
14	15
8	3
6	7
7	9
10	11
1	2

to Activities as shown in Table 4.2. Table 4.3 shows the message flow and transition between activities. Again, we assigned numbering to the list of message flow and transition as shown in Table 4.4. We construct graph for the BPMN diagram using a graph visualization software, Graphviz[1]. The input to the Graphviz is in the form of adjacency list. Figure 4.2 shows the graph generated for BPMN diagram.

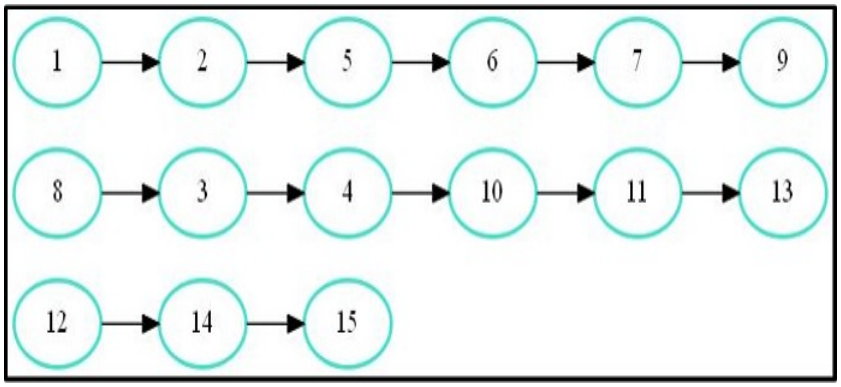


Figure 4.2: BPMN Graph for Cab Booking

We developed SBVR Business Vocabulary and Business Rules for the cab booking

[1]http://www.graphviz.org/

Table 4.5: List of Triplet Extracted from Business Rules for Cab Booking

Actor	Action	Object
Customer	Enters	Itinerary details
Customer	Request	Cab booking
Travel agent	Checks	Availability
Travel agent	Provides	Information
Customer	Confirms	Cab booking
Customer	Cancels	Cab booking
Travel agent	Assigns	Cab driver
Customer	Confirms	Cab booking
Cab driver	Picks	Customer

scenario using SBVR Lab 2.0[1]. We extracted triplet (Actor-Action-Object) from a English Sentence using Stanford NLP Parser. We considered only those triplets which participate in the business process. Extracted triplets are shown in the Table 4.5. We construct graph for the SBVR rules using a graph visualization software, Graphviz. The input to the Graphviz is the extracted triplet. Actor and Object form the nodes of the graph. Action represent the relationship between actor and object of the English sentence. Figure 4.3 depicts the graph generated for the SBVR rules.

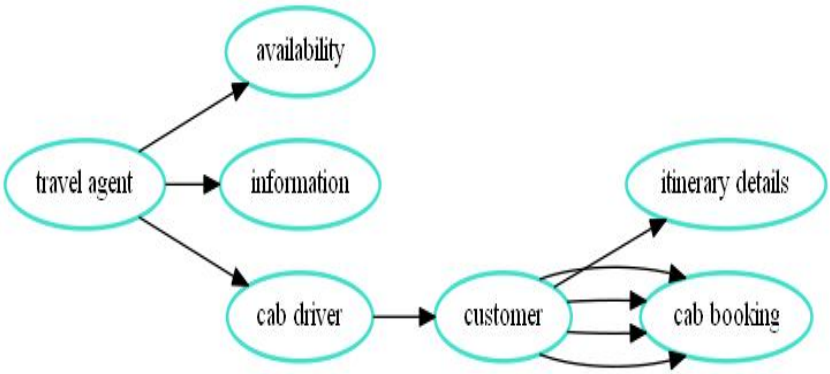


Figure 4.3: SBVR Graph for Cab Booking

We applied VF2 sub graph isomorphism on the two graphs generated in the Figure 4.2 and Figure 4.3. We used Python NetworkX Library[2] for the subgraph isomorphism

[1]http://www.sbvr.co/
[2]https://networkx.github.io/

algorithm. We identified subgraphs of BPMN graph of size of length of SBVR graph or less than that. We identified that if we take length of sub graph as of 2 less than number of nodes of SBVR Graph then we get two sub isomorphic graphs.

4.2.2 Inconsistent Scenario : Order Fulfillment

We perform another experiment on an inconsistent scenario of order fulfillment. We designed BPMN diagram for the process of taking an order to the product delivery in the organization. Figure 4.4 represents the BPMN process model for the order fulfillment process.

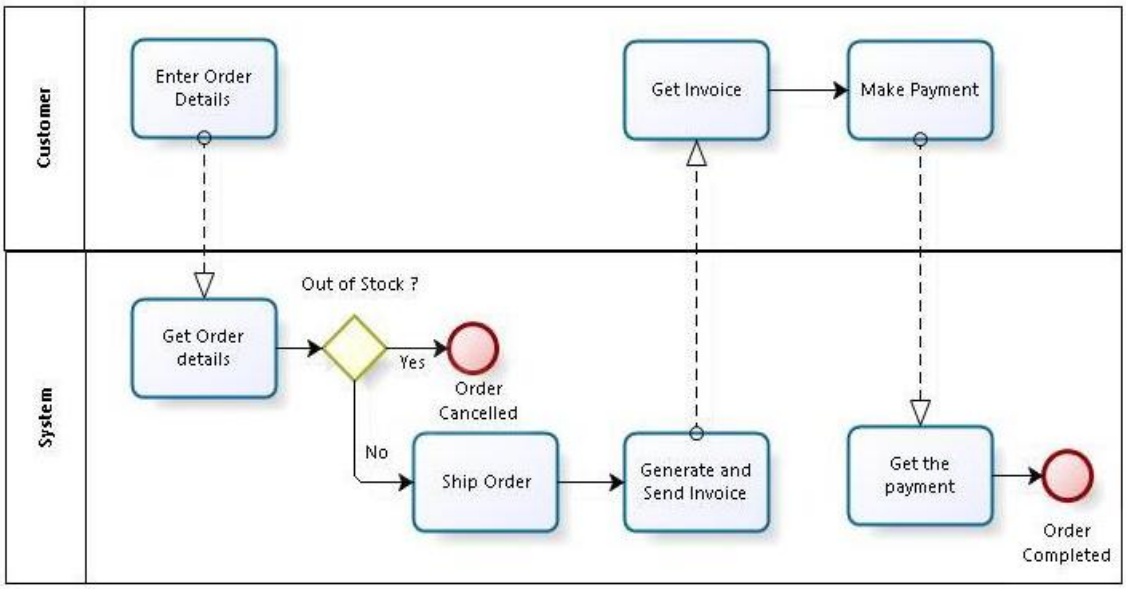


Figure 4.4: BPMN Diagram for Order Fulfillment

We designed this BPMN diagram and generated XPDL of the BPMN diagram using one of the popular process modeler, Bizagi Process Modeler. We extracted 'id' attribute of the activity tag and they will be nodes of the BPMN graph. Also, we extracted 'from' and 'to' attribute of the MessageFlow and Transition tags and these will be edges of the BPMN Graph. As these edges will be directional from 'from' attribute to 'to' attribute of Message Flow and Transition tags. Activity will be numbered and accordingly message flow and transition will be numbered. Table 4.6 list the activities involved in the BPMN process model of Order Fulfillment process. Table 4.7 list the

Table 4.6: List of Activity (Nodes) of Order Fulfillment Process

Activity ID	Number
d1122f19-d24f-4c5a-90d9-cd5e41373dda	1
c50b21c8-5fbc-4b5d-939c-250cfe053b4f	2
09c714a3-4115-44c9-ac21-128d7cf41488	3
74362d4f-2695-4429-8241-3ce994afa628	4
fe24142d-c73d-4ce4-9d2f-7d1ca8f9c986	5
e45ac891-ca14-44cc-ab02-a485fbc6284b	6
cd78f057-fe0e-4acd-9146-0694c342f4be	7
984f7edd-83b0-4b88-a9eb-a5d80b1c2d38	8
f146b1b9-d199-4fd8-a989-5833e6567700	9
9ae27e1b-1bae-43f4-8166-f723861071c1	10

Table 4.7: List of Message Flow and Transition (Edges) of Order Fulfillment Process

Activity From	Activity To
f146b1b9-d199-4fd8-a989-5833e6567700	9ae27e1b-1bae-43f4-8166-f723861071c1
c50b21c8-5fbc-4b5d-939c-250cfe053b4f	09c714a3-4115-44c9-ac21-128d7cf41488
d1122f19-d24f-4c5a-90d9-cd5e41373dda	74362d4f-2695-4429-8241-3ce994afa628
09c714a3-4115-44c9-ac21-128d7cf41488	f146b1b9-d199-4fd8-a989-5833e6567700
fe24142d-c73d-4ce4-9d2f-7d1ca8f9c986	cd78f057-fe0e-4acd-9146-0694c342f4be
984f7edd-83b0-4b88-a9eb-a5d80b1c2d38	c50b21c8-5fbc-4b5d-939c-250cfe053b4f
74362d4f-2695-4429-8241-3ce994afa628	fe24142d-c73d-4ce4-9d2f-7d1ca8f9c986
e45ac891-ca14-44cc-ab02-a485fbc6284b	984f7edd-83b0-4b88-a9eb-a5d80b1c2d38

message flow and transition between activities. Table 4.8 shows the numbered edges of the Order Fulfillment process. We then construct graph for the BPMN diagram using the Graph Visualization Software, Graphviz as shown in figure 4.5.

Table 4.8: List of Edges of Order Fulfillment Process

Activity From	Activity To
9	10
2	3
1	4
3	9
5	7
8	2
4	5
6	8

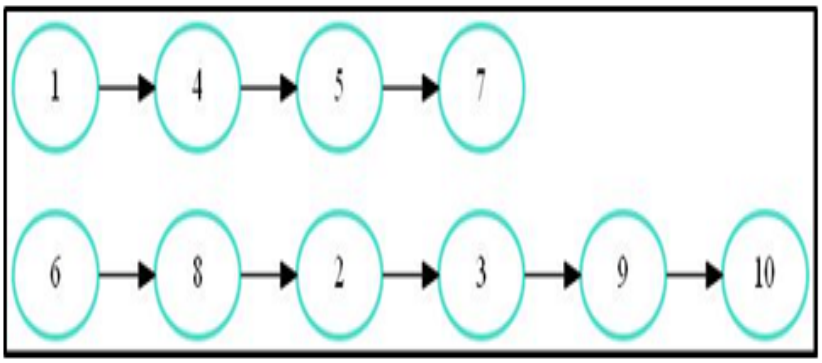


Figure 4.5: BPMN Graph for Order Fulfillment Process

We designed SBVR Business Vocabulary and Business Rules for the Order Fulfillment process using SBVR Lab 2.0 [1]. Business Vocabulary and Rules are written in Natural Language. We then parsed business rules using Stanford NLP parser. We extracted triplet (Subject-Verb-Object) of the business rules and categorized rules as action-oriented or structure-oriented. We are concerned only with action-oriented rules for further processing. Table 4.9 shows the list of triplet extracted for the action-oriented rules. We then constructed SBVR graph using the triplet extracted for action-oriented rules as shown in figure 4.6.

[1] http://www.sbvr.co/

Table 4.9: List of Triplet Extracted from Business Rules for Order Fulfillment Process

Actor	Action	Object
Customer	Enters	Order details
System	Receives	Order
System	Has	Order
Customer	Has	Feedback
System	Sends	Invoice
System	Cancels	Customer
System	Receives	Payment

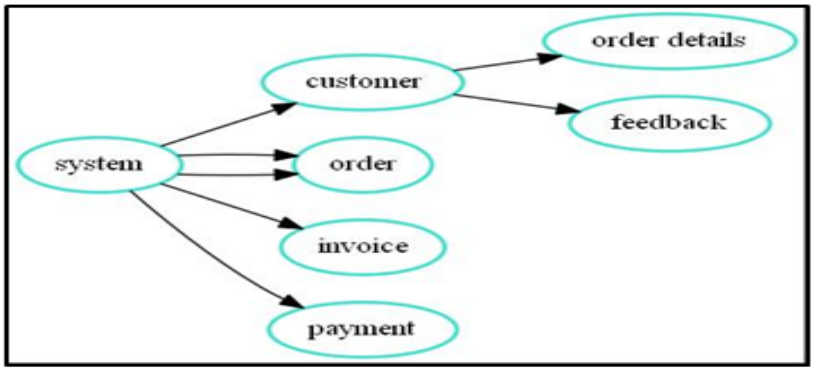


Figure 4.6: SBVR Graph for Order Fulfillment Process

We pruned multiple edges of the BPMN and SBVR graph. We also considered both the graphs as undirected graph. We then applied VF2 subgraph isomorphism algorithm on the resultant BPMN graph and SBVR graph. We obtained two isomorphic graphs of size two less than length of SBVR graph.

4.2.3 Experimental test-bed result

In this subsection, we show the analysis observed from the experimental results that we obtained. Table 4.10 depicts the complete information of the multiple scenarios. Our hypothesis is we can detect instances of inconsistency between process model and rule model but the experimental results clearly indicates that our hypothesis needs to be modified.

Table 4.10: Collection of Synthetic Data set of Consistent and Inconsistent Scenarios

Scenario	**Expected**	**BPMN Nodes(G1)**	**SBVR Nodes(G2)**	**Subgraph of BPMN(G3)**	**subisomorphic (G2,G3)**
Travel Reservation	Inconsistent	17	10	7	True
Cab Booking	Consistent	15	7	5	True
Employee Reimbursement	Consistent	13	5	3	True
Order Fulfillment	Inconsistent	10	7	4	True

5

Limitations and Future Work

The experiments that we have performed are conducted on a very small synthetic dataset comprises of some scenarios. Future work involves performing experiments with the real world dataset.

There is no feedback or involvement of industry people in the approach. Future work involves exposing our work to industrial environment.

The Triplet (Subject-Verb-Object) Extraction is limited to simple sentences and thus more work need to be done on improving the proposed algorithm. Improving algorithm to handle complex sentences and compound sentences will be used in future extension. We use Typed Dependency of parse tree generated using Stanford NLP Parser. In future, we can combine this approach with other machine learning techniques to increase the confidence in results.

As experimental results are indicating that it is not possible to detect instances of inconsistency between BPMN process model and SBVR rule model. In future work, we will try to improve results by conducting experiments on real world dataset and working on the proposed framework.

6

Conclusion

We present a novel approach for detection of instances of semantic inconsistency between BPMN Process Model and SBVR Rule Model developed for an IT system in an organization. The key components of the proposed framework are: extraction of tags from generated XML of BPMN diagram, triplet (actor-action-object) extraction of action-oriented rules from SBVR rule-set using grammatical relations between words represented by Typed Dependency of parse tree using Stanford NLP parser, and find node-induced sub graph of all possible length of nodes of a graph and apply VF2 Algorithm to detect instances of semantic inconsistency between sub graphs.

We have created a synthetic dataset comprises of multiple number of consistent and inconsistent scenarios. We also calculated an error metric (Precision and Recall) for the triplet (actor-action-object) extraction of SBVR rule-set to determine the accuracy of results. We conducted a series of experiments on synthetic dataset to demonstrate that the proposed approach is effective. Experimental results reveal that it is not possible to detect such type of inconsistency between process model and rule model.

References

[1] Jie Meng, Stanley YW Su, Herman Lam, and Abdelsalam Helal. **Achieving dynamic inter-organizational workflow management by integrating business processes, events and rules**. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2002. 1

[2] Wikipedia. **Business process modeling — Wikipedia, The Free Encyclopedia**. 1

[3] Object Management Group. **Business Process Model and Notation (BPMN), Version 2.0.2**. 2, 4

[4] Wikipedia. **Business Process Model and Notation — Wikipedia, The Free Encyclopedia**, 2015. 2, 4, 5

[5] Camunda. **BPMN 2.0 Best Practices**. 5

[6] Olfa Chourabi TANTAN and Jacky AKOKA. **Automated transformation of Business Rules into Business Processes**. 6

[7] Tomas Skersys, Kestutis Kapocius, Rimantas Butleris, and Tomas Danikauskas. **Extracting business vocabularies from business process models: SBVR and BPNM standards-based approach**. *Computer Science and Information Systems*, **11**(4):1515–1535, 2014. 6, 14

[8] Object Management Group. **Semantics of Business Vocabulary and Business Rules (SBVR), Version 1.3**. 6

[9] Wikipedia. **Semantics of Business Vocabulary and Business Rules — Wikipedia, The Free Encyclopedia**, 2015. 6

[10] John Krogstie, Peter McBrien, Richard Owens, and Anne Helga Seltveit. **Information systems development using a combination of process and rule based approaches**. In *Advanced Information Systems Engineering*, pages 319–335. Springer, 1991. 13

[11] Michael Zur Muehlen, Marta Indulska, and Kai Kittel. **Towards integrated modeling of business processes and business rules**. *ACIS 2008 Proceedings*, page 108, 2008. 13

[12] Bas Steen, Luís Ferreira Pires, and Maria-Eugenia Iacob. **Automatic generation of optimal business processes from business rules**. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International*, pages 117–126. IEEE, 2010. 13

[13] Dirk Habich, Sebastian Richly, Birgit Demuth, Franziska Gietl, Joachim Spilke, Wolfgang Lehner, and Uwe Assmann. **Joining Business Rules and Business Processes**. *Proc. of IT*, 2010. 13

[14] Ran Cheng, Shazia Sadiq, and Marta Indulska. **Framework for business process and rule integration: a case of BPMN and SBVR**. In *Business Information Systems*, pages 13–24. Springer, 2011. 13

[15] Tomas Skersys, Lina Tutkute, Rimantas Butleris, and Rita Butkiene. **Extending BPMN business process model with SBVR business vocabulary and rules**. *Information technology and control*, **41**(4):356–367, 2012. 13

[16] Tomas Skersys, Lina Tutkute, and Rimantas Butleris. **The enrichment of BPMN business process model with SBVR business vocabulary and rules**. *CIT. Journal of Computing and Information Technology*, **20**(3):143–150, 2012. 13

[17] Deepak Kumar Sharma, Naveen Prakash, Himani Sharma, and Dheerendra Singh. **Automatic construction of process template from business rule**. In *Contemporary Computing (IC3), 2014 Seventh International Conference on*, pages 419–424. IEEE, 2014. 14

[18] Egle Mickeviciute, Lina Nemuraite, and Rimantas Butleris. **Applying SBVR Business Vocabulary and Business Rules for Creating BPMN Process Models**. In *Business Information Systems Workshops*, pages 105–116. Springer, 2014. 14

[19] Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. **Triplet extraction from sentences**. In *Proceedings of the 10th International Multiconference" Information Society-IS*, pages 8–12, 2007. 14

[20] Lorand Dali and Blaz Fortuna. **Triplet extraction from sentences using svm**. *Proceedings of SiKDD*, 2008. 14

[21] Zin Thu Thu Myint and Kay Khaing Win. **Triple Patterns Extraction for Accessing Data on Ontology**. *International Journal of Future Computer and Communication*, **3**(1):40, 2014. 14

[22] J. R. Ullmann. **An Algorithm for Subgraph Isomorphism**. *J. ACM*, 1976. 14

[23] Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. *Subgraph transformations for the inexact matching of attributed relational graphs*. Springer, 1998. 14, 15

[24] Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. **Performance evaluation of the VF graph matching algorithm**. In *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 1172–1177. IEEE, 1999. 15

[25] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. **An improved algorithm for matching large graphs**. In *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen*, pages 149–159, 2001. 15, 29

[26] L.P. Cordella, P. Foggia, C. Sansone, and M. Vento. **A (sub)graph isomorphism algorithm for matching large graphs**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(10):1367–1372, 2004. 15, 29

[27] Bernhard G Humm and Janina Fengel. **Assessing Semantic Consistency of Business Process Models**. *language*, **3**:4, 2012.