**Using BPMN for structured PostgreSQL programming**

Apoorva Patrikar

Otto-von-Guericke University, Magdeburg, Germany

[apoorva.patrikar@st.ovgu.de](mailto:apoorva.patrikar@st.ovgu.de)

Prathamesh Pradhan

Otto-von-Guericke University, Magdeburg, Germany

[prathamesh.pradhan@st.ovgu.de](mailto:prathamesh.pradhan@st.ovgu.de)

**Abstract:**  Currently there are several business methods available which produce and provide the services or product to the end user. Such business process comprises of several activities and decision points. Tools for development of such a business structure is a necessity in today’s environment. This scientific project involves developing a business process tool which has a potential of understanding and processing business tasks and procedures represented in a graph, known as Business Process Model and Notation (BPMN) diagrams.

**Keywords:** BPMN, Tool, Python, Business process, Modelling, XML, PostgreSQL

**1 Introduction:**

This project makes use of the concept of BPMN diagram and provides a tool which processes these diagrams. The tool is developed in Python language, which initially parses through XML file developed primarily from BPMN diagrams.

**BPMN Introduction**

A standard Business Process Model and Notation (BPMN) is kind of a blueprint of the business process flow. It provides businesses with the perception of their internal business procedures using symbols and notations. It gives organizations the ability to communicate these procedures in a standard manner. These diagrams depict the steps in a business process by executing the underlying tasks and decisions.

BPMN depicts the end to end flow of a business process. The notation has been specifically designed to coordinate the sequence of processes and the messages that flow between different process participants in a related set of activities. BPMN diagrams improves the understanding with the participants in their business and ensure a quicker way to adjust to new internal and B2B environments. [1]

**Importance of BPMN diagrams**

The world of business processes has changed dramatically over the past few years. A business process now spans multiple participants and coordination can be complex. Until BPMN, there has not been a standard modelling technique developed that addresses these issues. BPMN has been developed to provide users with a royalty free notation. This will benefit users in a similar manner in which UML standardised the world of software engineering. [1]

BPMN is targeted at a high level for business users and at a lower level for process implementers. The business users should be able to easily read and understand a BPMN business process diagram. The process implementer should be able to adorn a business process diagram with further detail in order to represent the process in a physical implementation. BPMN is targeted at users, vendors and service providers that need to communicate business processes in a standard manner.

* 1. **Motivation:**

An end user or customer generates many such business process diagrams. Managing and executing such diagrams efficiently is important. Development of such a tool is helpful which fills the gap between executing the general BPMN diagrams effectively. Execution of fundamental task successfully is to be taken into consideration. This tool acts as a bridge which effectually processes the message flows, tasks and decision points and represent the output in standardized way.

**2. State of the art:**

There are some tools available for handling BPMN, like yEd Editor, Visual Paradigm, Gliffy’s, Aris, Camunda and Bizagi Modeler etc which have their own use cases suitable for business processing. Here, we are using yEd Editor as a tool to get familiar with BPMN diagrams. This application possess several advantages and features suitable to use to create diagram with ease and automatically arrange an underlying diagram elements. It imports our own data from an XML or Excel spreadsheet(.xls) yEd is an open source and runs on all major platforms : Windows, Unix/Linux, and Mac OS X.

Python has been used as a programming language because of its simplicity and rich built in libraries to deal with XML data. Here are some of the advantages which makes Python as a selection factor for development [2]

* Easy Syntax
* Readability
* High-Level Language
* Object oriented programming
* It's Free
* Cross-platform
* Widely Supported
* It's Safe
* Batteries Included
* Extensible

PostgreSQL is a powerful, open source object-relational database system. Functioning on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows, PostgreSQL provides reliability, data integrity and correctness. It is fully ACID compliant, has full support for concepts such as foreign keys, joins, views, triggers, and stored procedures. Immunity to over-deployment, stability and extensibility are some of the advantages which makes the PostgreSQL a de facto in business application [3]

## 

## Fig. 1 Simple BPMN Diagram

## This is an example of a lane of BPMN where the circle with the green circumference represents the start. ‘System Boundary’ is a task or an activity. ‘Transformation Validation’ is a property of this task. To execute this task, data object ‘VG 250 State Municipality’ is needed and in order to process this data object, another data object ‘Administrative Boundaries’ is needed. The output of the activity is sent to another set of data objects. And the execution ends with the last circle. The project can efficiently deal with these kind of diagrams.

## 

## Fig. 2 Complex BPMN Diagram

## 

## Fig. 3 Sub-graph of complex BPMN diagram

## These two diagrams shown above are a special kind of diagram. They have graphs embedded in nodes. These nodes are referred to as ‘Generic Group Nodes’ in the graphml file. The processing of such structures is done as follows. We start with the main start event. It eventually leads to a group node ‘Municipality Categories’. Then this node is processed completely. After it is done, we move on to the next group node ‘Grid Districts’. And then the main parent node ends.

Separate scripts and diagram:

Advantage:

• Modular structure.

Disadvantage:

• No direct connection between diagram and scripts.

• The flow is difficult to understand.

• No automation possible.

Annotate diagram with scripts:

Advantage:

• Fast as compare to other options because we don’t need to jump to a file; we simply execute the scripts as and when it comes in the diagram.

• Flow is easy to understand.

• Automation possible.

Disadvantage:

• Lengthy scripts may make the diagram ugly or simply might not fit.

Annotate diagram with script’s file name:

Advantage:

• A connection exists between the scripts and diagram.

• They are easy on the eyes.

• Automation possible.

• Advantages are intermediate between previous methods.

Disadvantage:

• Fetching scripts is required and this could be more time consuming than ‘Annotate diagram with scripts approach’.

• Activities or scripts with the same name could cause some ambiguity problems.



Fig. 4 XML file with Namespace declaration



Fig. 5 XML file structure

**Issues Faced:**

1. Understanding BPMN diagrams

The project began with the team members understanding the concepts of BPMN, as it is the foundation of its execution. We learnt about BPMN mostly from its official websites and some other online examples. It helped us gain an insight of how essential these diagrams can be for business processes.

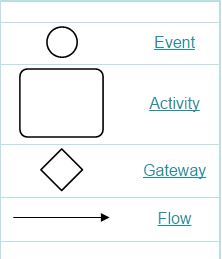
 

Fig. 6 BPMN basic notations Fig. 7 BPMN Events

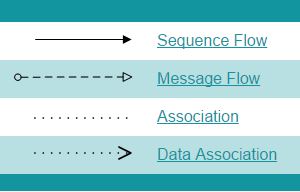
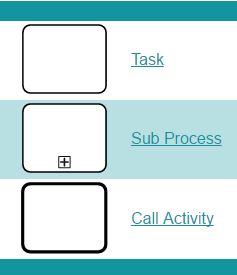
 

Fig. 8 BPMN Flow Notation Fig. 9 BPMN Task and Activity



Fig. 10 BPMN Gateway



Fig. 11 BPMN Pool and Lane

1. Transform yEd diagrams to BPMN

Once we had a clear understanding of BPMN diagrams, our first task was to convert non-standard linked diagrams to proper BPMN diagrams. This removed heterogeneity from the diagrams and they were all then in compliance with BPMN norms.

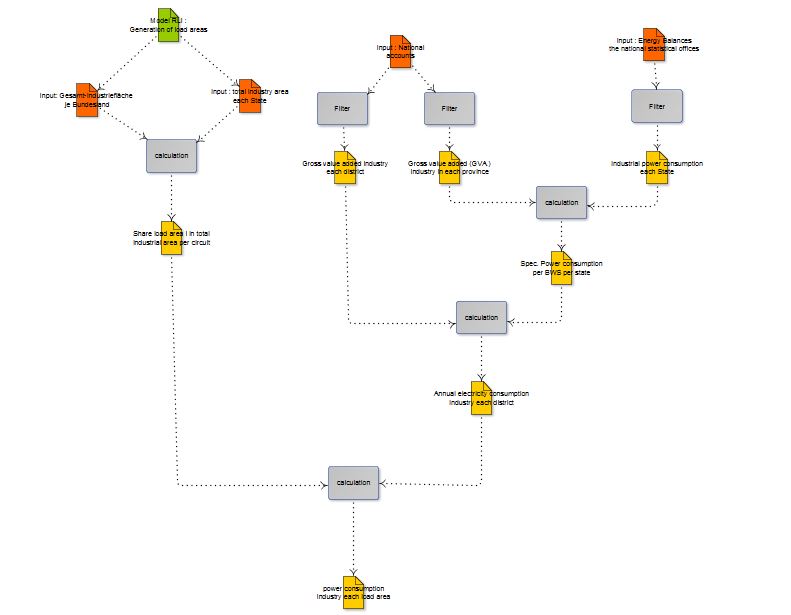


Fig. 11 Transformed from non BPMN diagram to BPMN

1. Object creation

This is where the coding for the execution starts. The first step to achieve the goal was to make objects for every type of object. For instance, a start node is an object of ‘Event’ or a data object is an object of type ‘Data Object’, denoted by DataObj in the project. The main elements however, which contain other elements, were graphs and nodes.

Objects were made for lanes, pools, graphs, nodes, events, edges, annotations, gateways, connection types and data objects.

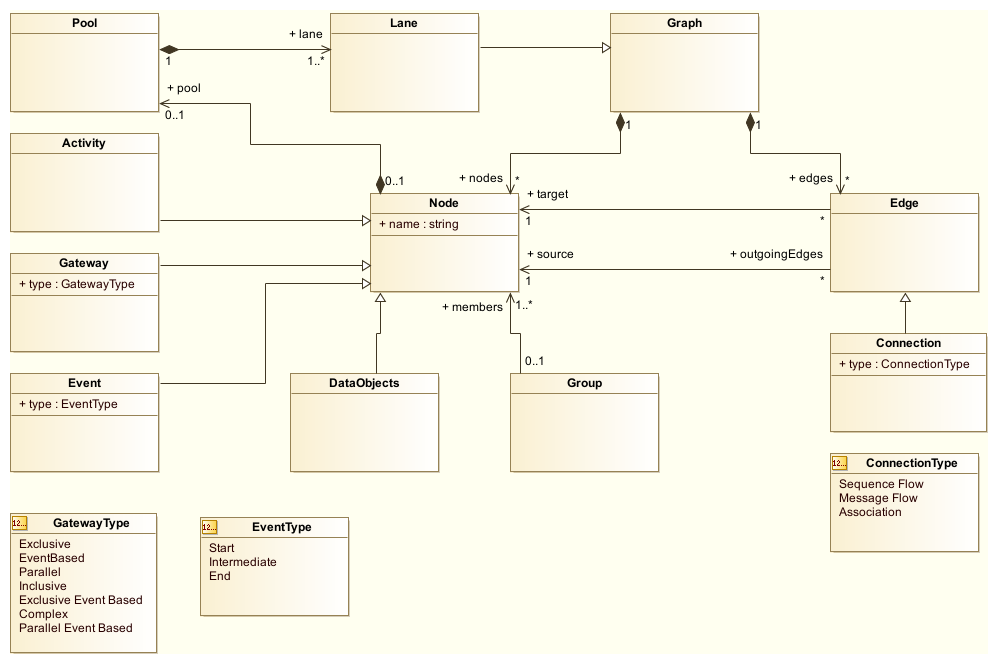


Fig. 12 UML representation of BPMN Objects

Using UML diagram helped us to understand the interrelation between objects. It simplified the further coding and development phase. Inheritance, associations and compositions made way simpler to create objects of respective classes. Object oriented approach helped us to go smoothly on our next task for development.

An output for Object creation task is as follows:

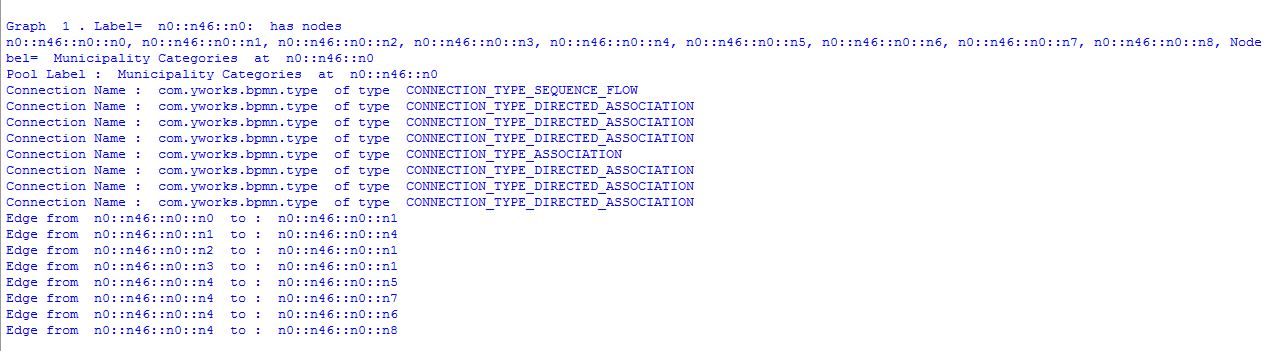


Fig. 13 Representation of BPMN Objects in Python

Here in the output, we can see the objects of nodes and their respective source and target edges with any connection object under any pool object.

1. Represent the graph of a yEd graphml file in Python

A recursive function was written in order to parse through them because the main graph contains all nodes, and also some nodes contain graphs. The recursion enables this type of mutual calling up to any desired depth



Fig. 14 Mutual Recursion in XML



Fig. 15 processing of given BPMN diagram

1. Given BPMN diagram execute it

All parsing starts with the start event and ends until there are no further elements or an end event is encountered. Execution here means only printing the nodes in the order of the required parsing.

**Future Work:**

The BPMN diagram is currently being executed. There can be significant amount of future features to be worked on. As automating the process of BPMN diagrams, several benefits could emerge for several business functional units. Process management, capturing data and processing it for further analysis are some of the work can be carried out in the future. One of the crucial thing with respective to our tool is to execute the underlying scripts which are invoked on a particular task or activities. The SQL scripts are annotated in annotations and are the point of centre to execute it.

**Conclusion:**

An alternative to traditional development techniques prototype is the Business process execution. In order to understand and communicate business processes across out technique, modelling with BPMN is an essential point to be taken into consideration. BPMN provides a powerful magnification to several other modelling techniques such as relational data modelling, application, and system design with UML, XML schema design, and network architecture design. For an individual or business firms to apprehend its business flow and to design its enterprise architecture, modelling with BPMN can be add on advantage, not only in standardized manner but also in the safe way.

BPMN is an easy to use and understand, thanks to its built in established method for process discovery and documentation, thus provides the potentiality to capture and document our processes.

**Acknowledgement:**

We would like to thank Prof. Dr.-Ing. habil. Till Mossakowski, Mr. Martin Glauer from the Otto von Guericke University of Magdeburg, to guide us through the project. They provided a starting point which helped to make this project possible.

**References:**

[1] <http://www.bpmn.org/>

[2] <https://en.wikiversity.org/wiki/Python/Why_learn_Python>

[3] <https://www.postgresql.org>