

Visualization Libraries for Process Analytics

Nour Assy, Céline Souchet, Thomas Bouffard and Olan Anesini

Bonitasoft, France

Email: firstname.lastname@bonitasoft.com

Abstract—Visualization plays a crucial role in communicating process analytics results. However, the visualization component is often coupled with the analysis technique and thus is treated as a second class-citizen. Therefore, without dedicated and well-defined visualization APIs, customizing the visualization to meet specific needs often requires ad-hoc changes to the source code, which is a time-consuming and cumbersome task. In this paper, we introduce *bpmn-visualization*, a library for visualizing model-based process analytics results using BPMN. The library is being developed as part of the open source (Apache v2) project *Process Analytics*, initiated by Bonitasoft and aiming at developing an ecosystem of visualization libraries for process analytics. These libraries offer significant value to those who are involved in process mining, BPM and RPA development projects. By supporting better visualization control and customization flexibility, these visualization libraries can greatly benefit the community working on combining process and visual analytics.

Index Terms—Process Analytics, visualization API, BPMN

I. INTRODUCTION

The development of process mining techniques [1] has experienced tremendous growth leading to increasing adoption by organizations [3]. The majority of current implementations couple the visualization with the mining technique resulting in ad-hoc, technique-or-tool specific visualization components. On the other hand, there is a growing interest to combine process and visual analytics [2], [4], [5]. This requires developing visualization artifacts for process analytics. However, with the absence of appropriate visualization libraries and tools, creating synergy between visualization and process analytics requires additional development efforts that might otherwise be dedicated to the visualization components.

To fill this gap, the open source *Process Analytics* project¹ initiated by Bonitasoft aims to develop an ecosystem of visualization techniques that can be used in end-to-end process analytics projects. Two main libraries are currently being developed: i) *bpmn-visualization* for visualizing model-based process analytics results using the BPMN standard (developed in Typescript and R) and ii) *bpmn-layout-generators* for generating the graphical layout of BPMN diagrams (developed in Java). The project repository on GitHub² is developed in a collaborative environment where the community can contribute by testing, giving feedback, reporting bugs, adding or requesting desired features and enhancing the documentation.

In this extended abstract, we present the overall project scope. Then, we focus on the *bpmn-visualization* library and

show examples of the TypeScript implementation. A video presenting the library has also been made available.³

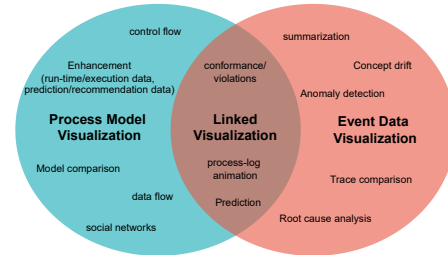


Fig. 1. Visualization categories for process analytics

II. PROJECT OVERVIEW AND CURRENT SCOPE

To clearly define the project scope, we identified three main categories of visualization libraries that need to be developed according to the input data and the information to be visualized⁴. The input data can be process models and/or event data. The information to be visualized can be determined based on the analysis technique. Fig. 1 shows a Venn diagram of the three categories: i) process model visualization, ii) event data visualization and iii) linked visualization between process models and event data. For each category, the information to be visualized is defined in terms of the process analysis task. For instance, for process models, we need to be able to visualize the model and to enrich it with additional data derived from event data. For event data, visualization artifacts that allow summarizing data, detecting anomalies, concept drifts, and the like, need to be developed. Finally, linked visualization artifacts are required for process analytics tasks (e.g. process-log animation) that require a linked visualization between process models and event data.

The current scope of the project is focused on developing the *bpmn-visualization* library that covers use cases of the process model visualization category.

III. ARCHITECTURE AND FEATURES

The general architecture of the *bpmn-visualization* library consists of three key components: *BPMN parser*, *BPMN renderer and interactions*. The *BPMN parser* loads XML BPMN files, and converts them into an internal model. The internal BPMN model is then passed to the *BPMN renderer*

³https://youtu.be/dtE6l2S0j_g

⁴For a more fine grained classification of visualization techniques for event data, the reader is referred to [6]

¹<https://process-analytics.dev/>

²<https://github.com/process-analytics>

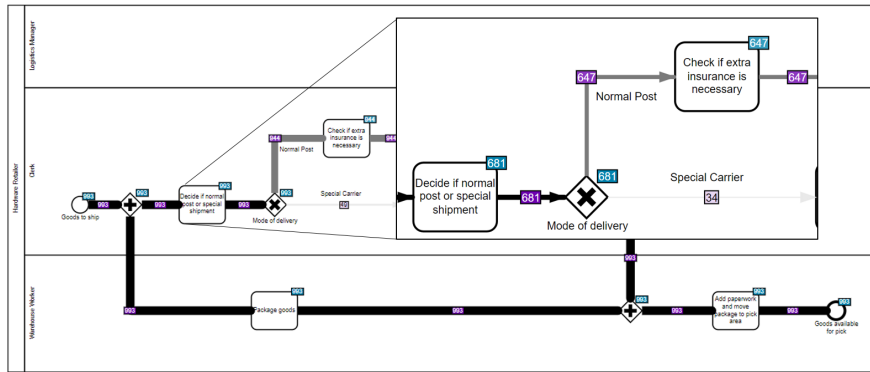


Fig. 2. An example of a BPMN process enriched with process execution statistics using the BPMN visualization API

component, which is in charge of rendering the BPMN elements. The API offers the following features: i) BPMN graph rendering and fitting (horizontal/vertical/center); ii) Styling and customizing BPMN shapes including colors, sizes, fonts and icons; and iii) Enriching the diagram with customized overlays. These features cover a wide range of model-based process analytics results. For instance, overlays can be added to visualize process data statistics (e.g. performance and frequency data); the colors and width of activities and edges can be customized to visualize conformance checking data; and animations can be added to BPMN elements to highlight specific paths (e.g. to show happy/exceptional flows).

The *interactions component* empowers developers and end users with flexibility in interacting with the BPMN elements. The following features are currently supported: i) Explore and navigate through BPMN diagrams and subprocesses by zooming, fitting and panning; and ii) Add custom listeners to dynamically enrich and/or modify the model style. For instance, tooltips can be added on mouseover; BPMN element style can be modified at runtime allowing to mark a task as in error or warn that an activity seems blocked.

Figure 2 shows an example of a BPMN process enriched using overlays and customized edge styles. More details can be found on the documentation page.⁵

IV. MATURITY AND USAGE

The bpmn-visualization library has been recently integrated into the Bonita 2022.2 process automation platform Enterprise edition. In addition, the library has been used in the development of a BPMN-based process discovery tool.⁶

Listing 1 shows an example of using the library API. The user of bpmn-visualization is responsible for retrieving the BPMN diagram (local file, fetch via an API, etc.). In the example, we assume that the variable bpmnContent contains the BPMN xml content. A new instance of the BpmnVisualization class is created and activates the support of the BPMN diagram navigation. The diagram is loaded and centered in the viewport by calling the function

⁵<https://process-analytics.github.io/bpmn-visualization-js/>
⁶<https://process-analytics.dev/model-generation-application/>

load. In this example, we show how an overlay can be added to the start event of the process in Fig. 2. The interaction with the diagram is achieved via bpmnElementsRegistry which offers, among others, the addOverlays method. This method takes as input the id of a BPMN element and one or several overlays to be added. An overlay is defined in terms of its position, label and the style.

```
import {BpmnVisualization} from 'bpmn-visualization';
let bpmnContent; // your BPMN 2.0 XML content
const bpmnVisualization = new BpmnVisualization({
  container: "bpmn-container",
  navigation: { enabled: true }
});
bpmnVisualization.load(bpmnContent, fit: {type: 'center'});
bpmnVisualization.bpmnElementsRegistry.addOverlays(
  'startEventId', {
    position: 'top-right',
    label: '993',
    style: {
      font: { color: 'white', size: 8 },
      fill: { color: 'cyan', opacity: 50 },
      stroke: { color: 'DarkSeaGreen', width: 2 } } });
```

Listing 1. A JavaScript example of the bpmn-visualization API usage

V. CONCLUSION

The *Process Analytics* project aims is to make visualization comes into reach of all organizations involved in process analytics development projects with a minimal effort. This Apache v2 project allows developers to implement many use cases, such as projecting process execution data on top of process models, highlighting predicted activities and paths at run-time and enriching the process with recommendation data.

REFERENCES

- [1] van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
- [2] Gschwandtner, T.: Visual analytics meets process mining: Challenges and opportunities. In: SIMPDA (Revised Selected Papers). Lecture Notes in Business Information Processing, vol. 244, pp. 142–154. Springer (2015)
- [3] Kerremans, M., Srivastava, T., Choudhary, F.: 2021 Gartner Market Guide for Process Mining (2021)
- [4] Kriglstein, S., Pohl, M., Rinderle-Ma, S., Stallinger, M.: Visual analytics in process mining: Classification of process mining techniques. In: EuroVA@EuroVis. pp. 43–47. Eurographics Association (2016)
- [5] Mendling, J.: Past, present, and future of visual process analytics. In: FEMIB. p. 5. SCITEPRESS (2022)
- [6] Yeshchenko, A., Mendling, J.: A survey of approaches for event sequence analysis and visualization using the esevis framework. CoRR abs/2202.07941 (2022)