

BPMN-Redrawer: From Images to BPMN Models

Alessandro Antinori^{1,†}, Riccardo Coltrinari^{1,†}, Flavio Corradini^{1,†}, Fabrizio Fornari^{1,*,†}, Barbara Re^{1,†} and Marco Scarpetta^{1,†}

¹University of Camerino, School of Science and Technology, Computer Science Department, Via Madonna delle Carceri, 7, Camerino, Italy

Abstract

BPMN models are often used by researchers to illustrate and validate new approaches that operate over such models. However, not always models are distributed in their source format but they are distributed as images within a document (e.g., a scientific contribution). To actually reuse those models, one has to manually redraw them. Manually redrawing a BPMN model is a time-consuming and error-prone activity. In this work, we present BPMN-Redrawer a tool that makes use of machine learning techniques for supporting the redrawing of BPMN models from images to actual models in .bpmn format. The tool is made available open source and it is open to contribution from the community.

Keywords

Process Images, Machine Learning, BPMN, Process Model

1. Introduction

BPMN is the de facto standard for modeling business processes¹ and models designed with such a notation are often used to conduct research activities in the BPM field. From the most common scenario where a BPMN model is used to illustrate the proposal of a new approach, to more systematic and complex activities such as studies on modeling practices [1, 2], and validation of techniques and tools related to the various phases of the BPM life cycle [3]. Those BPMN models are then reported, as images, in scientific works (e.g., BPM conference proceedings). However, models reported in scientific works, are rarely made available in their source format (i.e., .bpmn). This requires those who would like to experiment on those specific models, to actually “redraw” them from scratch. This is the procedure that the authors of [4] conducted to harvest BPMN models from BPM conference proceedings. Redrawing a BPMN model from an image is a manual and error-prone activity that requires a considerable amount of time and effort.

Proceedings of the Demonstration Resources Track, Best BPM Dissertation Award, and Doctoral Consortium at BPM 2022, BPM 2022, Münster, Germany, September 11-16, 2022

*Corresponding author.

†These authors contributed equally.

✉ alessandro.antinori@unicam.it (A. Antinori); riccardo.antinori@unicam.it (R. Coltrinari); flavio.corradini@unicam.it (F. Corradini); fabrizio.fornari@unicam.it (F. Fornari); barbara.re@unicam.it (B. Re); marco.scarpetta@unicam.it (M. Scarpetta)

🆔 0000-0003-3670-0415 (A. Antinori); 0000-0002-2137-6731 (R. Coltrinari); 0000-0001-6767-2184 (F. Corradini); 0000-0002-3620-1723 (F. Fornari); 0000-0001-5374-2364 (B. Re); 0000-0002-9659-7264 (M. Scarpetta)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

¹BPMN official page: <https://www.bpmn.org/>

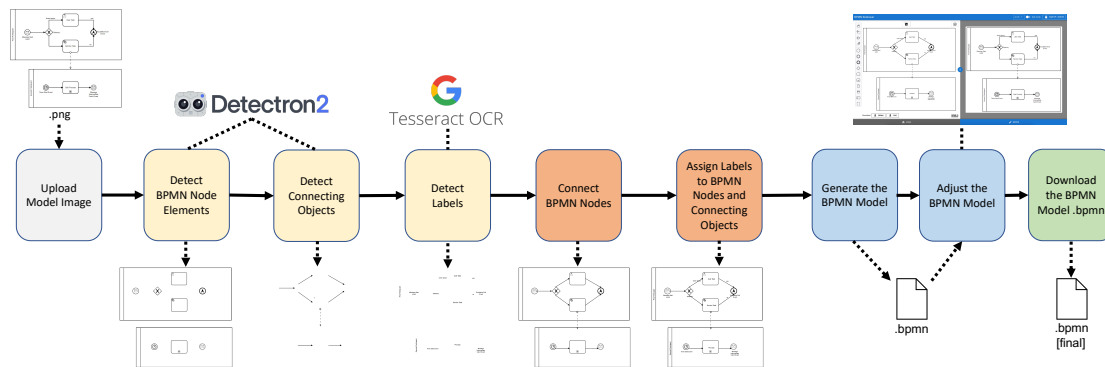


Figure 1: BPMN-Redrawer Approach Schematization.

In this paper we present BPMN-Redrawer, an approach, and the corresponding open source tool, to support the activity of “redrawing” BPMN models from images. BPMN-Redrawer comes in the form of a web application that provides a user interface to request the automatic redrawing of a BPMN model. The conversion of images in actual BPMN models can be seen as an object detection problem in which the to-be-detected objects are BPMN elements. Our approach makes usage of supervised machine learning algorithms and tools for the training of models capable of detecting BPMN elements. This approach could foster the reuse of BPMN models reported within documents, converting them from images to .bpmn files that can be used for conducting further activities.

Recently, a tool that shares our objective of speeding up the redrawing of BPMN models has been developed [5]. The tool focuses on the digitalization of hand-drawn BPMN models. Despite the source code is not made available, which hinders the possibility for the community to contribute to its improvement, the tool provides great results over models manually designed by university students. In our case we focus instead on images of BPMN models originally designed with BPMN editors. We believe that in a realistic scenario no one would choose to manually design a BPMN model when there are more than 70 BPMN modeling tools² that could be used. In addition, we open BPMN-Redrawer to the research community so that anyone can access it, take inspiration, apply changes and enhancements.

2. BPMN-Redrawer Main Functionalities

The functionalities of *BPMN-Redrawer* are made available through a web application that allows users to upload .png images and request their conversion into actual BPMN models stored in .bpmn format. We refer to such a process with the term “model redrawing”. A schematization of such a process together with the technology involved, is reported in Fig. 1. The process of redrawing a BPMN model could be divided into three main phases: *BPMN element detection*, *BPMN element linking*, and *BPMN model generation*. In the following we report a detailed description of each phase and step.

²BPMN Modeling Tools: <https://bpmnmatrix.github.io/>

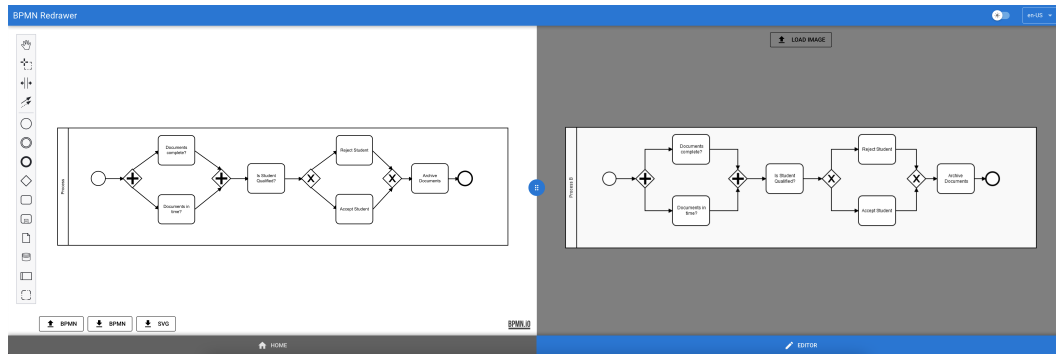


Figure 2: BPMN-Redrawer User Interface After Model Redrawing.

Detection Phase. It is the main phase of the approach, that starts after a user uploads an image. It is composed of three steps: BPMN nodes detection, BPMN connecting object detection, and BPMN label detection. For detecting BPMN nodes and connecting objects we use the well-established framework Detectron2³ which provides state-of-the-art detection algorithms with already trained baselines. *BPMN nodes* can be detected by means of their bounding boxes. To do so, we fine-tuned a pre-trained Faster Region Based Convolutional Neural Networks (R-CNN) using Stochastic Gradient Descent with a batch size of 4 and a decreasing learning rate starting from 0.0025. As a Convolutional Neural Network (CNN) backbone, we chose the ResNet-50 with Feature Pyramid Network (FPN). The detection of *BPMN connecting objects* implies not only the discovery of its bounding box but also the prediction of two keypoints: one for the head and one for the tail. For this reason, we fine tuned a pre-trained Keypoint R-CNN network using Stochastic Gradient Descent with batch size at 4 and a learning rate fixed at 0.00025. In this case, as a backbone, we use a ResNet-101 with FPN. For detecting *BPMN labels*, we used the Optical Character Recognition (OCR) engine *Tesseract* [6] which analyses the image from left to right and from top to bottom. BPMN elements and labels are stored in dedicated data structures that will be input for the next steps.

Linking Phase. It is the second phase of the redrawing process and it comprehends two steps: *Connect BPMN Nodes* and *Assign Labels to BPMN elements*. For such a connection we relied on the euclidean distance between elements: the target and the source of a connecting object are associated with the elements that are respectively closer to its head and tail, while each label is assigned to the closest element.

BPMN Model Generation Phase. It is the third and last phase of the approach. It is composed of two steps: *BPMN model generation* and *BPMN model adjustment*. The first step consists in parsing the results of the previous phases into a BPMN model (in .bpmn format). The resulting file is obtained by populating a BPMN template, that we created with the Jinja templating engine.⁴ Once the BPMN model is obtained, the user can either download it or adjust it using the integrated *bpmn-js* editor. This operation is facilitated by the display of the original image beside the editor as reported in Fig. 2.

³Detectron2: <https://github.com/facebookresearch/detectron2>

⁴Jinja: <https://jinja.palletsprojects.com/en/3.1.x/>

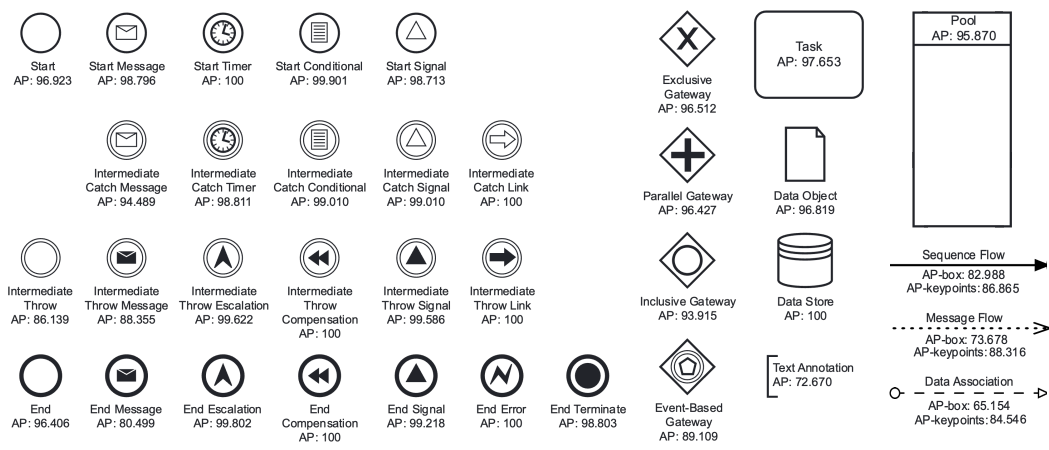


Figure 3: BPMN elements recognized by BPMN-Redrawer and Average Precision (AP).

3. Maturity of the Tool

BPMN-Redrawer is able to recognise up to thirty-two BPMN nodes and three BPMN connecting objects (e.g., Sequence Flow, Message Flow, Data Association). All the elements are reported in Fig. 3 together with their name and the average precision (AP) with which BPMN-Redrawer recognises them. For conducting the training activities we started from a dataset of 663 BPMN models images derived from models stored on the RePROsitory platform.⁵ However, considering that the usage of the BPMN elements varies between models [1], not all the BPMN elements were present in a sufficient number in such a dataset. Therefore, we extended the dataset with the addition of 165 images of models designed ad-hoc for increasing the amount of instances of those BPMN elements. The resulting dataset is composed of 828 BPMN models images; the images and the dataset in COCO format are available online.⁶ With a larger dataset we could reach a higher precision and we could also enable the redrawing of elements that have not been considered in this version of the tool such as lanes and complex gateway.

Some BPMN elements may present graphical markers (e.g., send/receive tasks, manual tasks, script tasks etc.). In the actual version of the tool, elements are redrawn without markers leaving to the user the possibility to adjust the model by means of the available editor and original image (see Fig. 2). Some BPMN models may also report customised elements due to BPMN extensions [7]; those elements to be recognised must be included in the training dataset.

Different BPMN editors can have different ways of graphically representing the same BPMN element (e.g., adding a coloured background, having a different size of the element, etc.). In this first version of BPMN-Redrawer we started from images of BPMN models designed with the bpmn-js editor therefore more accurate results are get when we ask BPMN-Redrawer to redraw a model originally designed with bpmn-js. We plan to improve our tool by extending the dataset we used for the training activities by also including images of BPMN models designed with

⁵Models Collection used: https://pros.unicam.it:4200/guest/collection/bpmn_redrawer
⁶BPMN-Redrawer Dataset: <https://huggingface.co/PROSLab/BPMN-Redrawer-Dataset>

other editors such as: Signavio, Eclipse BPMN Modeler, etc.

Since BPMN-Redrawer operates starting from images of BPMN models, the quality of the image affects the capability of properly recognising and redrawing the elements. We are working to define possible indicators of the image quality that can impact the redrawing of the model. As future work we also plan to add functionalities to the platform in such a way to provide, together with the redrawn model also information about the quality of the obtained result, reporting whether the model is a valid BPMN model, and the amount of elements that have been redrawn. In addition, we plan to improve the results of the connecting objects and labels recognition steps by investigating additional techniques to be used.

BPMN-Redrawer is open source, this enables anyone to access the code, apply changes, train new machine learning models starting from different datasets of BPMN images, and easily deploy them. The same approach can be used for automatising the redrawing of other types of models such as: Petri Nets, Event Process Chain, UML Diagrams, etc.

4. Screencast and Website

BPMN-Redrawer tool is accessible at <http://pros.unicam.it/bpmn-redrawer-tool>. The screencast available at <https://youtu.be/0e2qnbSp9XY> shows a typical user experience. The source code is available at <https://github.com/PROSLab/BPMN-Redrawer>. A Docker image to easily deploy the tool is also made available at <https://hub.docker.com/repository/docker/proslab/bpmn-redrawer>.

References

- [1] I. Compagnucci, F. Corradini, F. Fornari, B. Re, Trends on the Usage of BPMN 2.0 from Publicly Available Repositories, in: BIR 2021, Vienna, Austria, September 22-24, 2021, Proceedings, volume 430 of *LNBIP*, Springer, 2021, pp. 84–99.
- [2] F. Corradini, A. Ferrari, F. Fornari, S. Gnesi, A. Polini, B. Re, G. O. Spagnolo, A Guidelines framework for understandable BPMN models, *Data Knowl. Eng.* 113 (2018) 129–154.
- [3] F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi, A. Vandin, A formal approach for the analysis of BPMN collaboration models, *J. Syst. Softw.* 180 (2021) 111007.
- [4] F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi, RePROSitory: a Repository Platform for Sharing Business PROcess modelS, volume 2420 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 149–153.
- [5] B. Schäfer, H. van der Aa, H. Leopold, H. Stuckenschmidt, Sketch2BPMN: Automatic Recognition of Hand-Drawn BPMN Models, volume 12751 of *LNCS*, Springer, 2021, pp. 344–360.
- [6] R. Smith, An overview of the tesseract OCR engine, in: 9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil, IEEE Computer Society, 2007, pp. 629–633.
- [7] I. Compagnucci, F. Corradini, F. Fornari, A. Polini, B. Re, F. Tiezzi, Modelling Notations for IoT-Aware Business Processes: A Systematic Literature Review, in: BPM 2020 International Workshops, Seville, Spain, September 13-18, 2020, volume 397 of *LNBIP*, Springer, 2020, pp. 108–121.