UBBA: Unity Based BPMN Animator

Basit Mubeen Abdul, Flavio Corradini, Barbara Re, Lorenzo Rossi(✉), and Francesco Tiezzi

School of Science and Technology, University of Camerino, Camerino, Italy
basitmubeen.abdul@studenti.unicam.it,
{flavio.corradini,barbara.re,lorenzo.rossi,
francesco.tiezzi}@unicam.it

Abstract. In the last years BPMN became the most prominent notation for representing business processes, thanks to its wide usage in academic and industrial contexts. Despite BPMN is very intuitive, it's way of representing activities with static flow charts may result effective just for the BPM experts. Stakeholders who are not too much aware of the BPMN notation could misread the behavior of the business process. To this aim, BPMN animation tools can help model comprehension. However they are mainly based on 2D diagrams, just few works investigate the use of a 3D world as an environment for closely portray the reality of the business process. In this paper, we propose our tool UBBA, which creates a custom 3D virtual world from an input *.bpmn* file. Besides this 3-dimensional view of the diagram, we also integrate into UBBA the semantics of the BPMN elements in order to enable the animation.

Keywords: BPMN · Collaboration · 3D visualization · Animation

1 Introduction

Business Process Model and Notation (BPMN) [14] is a well-established standard for describing organization activities in a very intuitive way. BPMN allows to easily represent single organization workflows and their compositions by means of *process* and *collaboration* diagrams. Therefore, process stakeholders can communicate and reason about their processes in a standard manner. Business processes in the BPMN graphical notation are represented by means of static 2D flow charts, where the graphics of the elements embeds their semantic categories (e.g., rectangles for activities, diamonds for choices, circles for events, etc.). This graphical representation is very straightforward for experts, but it results difficult to understand for stakeholders [6] who have to know the syntax and semantics of BPMN. Furthermore, when models become very large it is difficult to follow their execution semantics [7] and the use of 2D instead of 3D representations limits the amount of information the user can perceive [16].

In this regard, recent works foster new techniques for modeling and visualizing organization processes capable to bridge the gap between business people

C. Cappiello and M. Ruiz (Eds.): CAiSE Forum 2019, LNBIP 350, pp. 1–9, 2019.
https://doi.org/10.1007/978-3-030-21297-1_1

and BPMN. On the one hand, a virtual world representing a business process can enhance the communication activities, thus facilitating interactions between businessmen and stakeholders [11]. On the other hand, the animation of business processes can increase their understanding [4,10,13] and also the possibility to debug them [8].

Table 1. Literature comparison

	[6]	[5]	[12]	[1]	[15]	[3]	[2]	[9]	UBBA
Collaboration	✗	✗	✗	✗	✗	✗	✓	✓	✓
Visualization	3D	3D	3D	▲	✗	2D	2D	2D	3D
Animation	✗	✗	✗	▲	3D	2D	2D	2D	3D
Custom graphics	✗	✗	✗	✗	✗	✗	✗	▲	✓
Standard input	✗	✓	✓	✗	✗	✓	✓	✓	✓

✓: fully, ▲: partially, ✗: not supported

The literature proposes several tool prototypes that follow such principles. Indeed, in [6] an implementation of a 3D virtual BPMN editor that embeds process models into a 3D world is presented. Similarly, in [5] and [17] the representation of Petri net-based business process models is enriched with the third dimension. Virtual worlds have also been used in the context of Workflow Management Systems (WfMSs). In [12] the authors have implemented an agent-based simulation architecture that can be used as a simulation component for any WfMS. It is also worth mentioning the BPM simulation game *Innov8* by IBM [1]. It gives to both IT and business players an excellent introduction to BPM, useful for learning the anatomy of a model. Another example is *Be a Token* [15], a javascript tool based on the A-Frame framework. This tool represents sequence flows as hallways to cross, and tasks as rooms with a door in the wall for each incoming/outgoing sequence flow. For what concerns the business process animation there are works attempting to show the processes execution, which however just provide a 2D visualization. These contributions use token flow or elements highlighting to indicate the current execution state of the process models. In [3], business processes are animated by means of a token game within the Signavio modeler, where users can step through the process element-by-element. Visual Paradigm [2] provides an animator that supports also collaboration diagrams. Finally, [9] provides an animator of BPMN collaborations enriched with data and multiple instances, which is based on token flow animation. However, the above solutions suffer from three main limitations. Firstly, works recreating a 3D world do not provide any animation of the business process, but just visualization. This means that they statically show a 3D version of the model without supporting a representation of its execution. Moreover, these projects are not very customizable, but instead are limited to describing a particular setting without the possibility of using custom 3D models. On the other hand, works providing animation of business processes use only 2D environments. Table 1 summarizes the features provided by the tools available in the literature and by our work. We compare the tools on their capability to: deal with BPMN *collaboration* models, *visualize* and *animate* in 2D or 3D the models execution, insert *custom graphical elements*, and parse model files compliant with the *standard format*.

To overcome the above limitations, in this paper we propose the prototype tool UBBA. Taken in input a BPMN collaboration diagram with standard XML format, UBBA recreates a custom virtual world where to visualize and to animate

the input diagram decorated with 3D graphics chosen by the user. The user can follow the execution of its model looking at the animation on the newly created virtual world.As far as we know, UBBA is the first tool that can be exploited for representing and animating any kind of business scenario (e.g., order fulfillments, retrieval of healthcare data, bureaucratic procedures), thanks to its capability of loading standard BPMN XML files and custom 3D graphics. UBBA is conceived to support its users, i.e. business process designers, in *(i)* validating their diagrams and, possibly in collaboration with domain experts who help in choosing effective 3D graphics, *(ii)* creating appealing 3D animations for stakeholders, who are the final audience for the products of the tool.

The rest of the paper is organized as follows. Section 2 provides an overview of the UBBA development and features. Section 3 introduces the case study we used to test our solution and describe the tool functioning. Finally, Sect. 4 closes the paper with UBBA links and information.

2 UBBA

In this section we present our tool UBBA (Unity Based BPMN Animator). UBBA is a cross-platform and stand-alone tool that has been realized in Unity (unity3d.com), a game engine for creating 2D/3D video-games or other interactive contents, such as architectural visualizations or animations in real time. More in detail, UBBA aims at reproducing the setting described in a BPMN collaboration diagram and animating its execution, by means of token flow [14, p. 27]. Indeed, the BPMN elements are transformed into 3D graphics and visualized in a virtual space. Then, one or more tokens cross the diagram following the semantics of the BPMN elements they met. Figure 1 depicts the UBBA workflow functioning. In the following, we introduce the tool focusing on the visualization and then on the animation features.

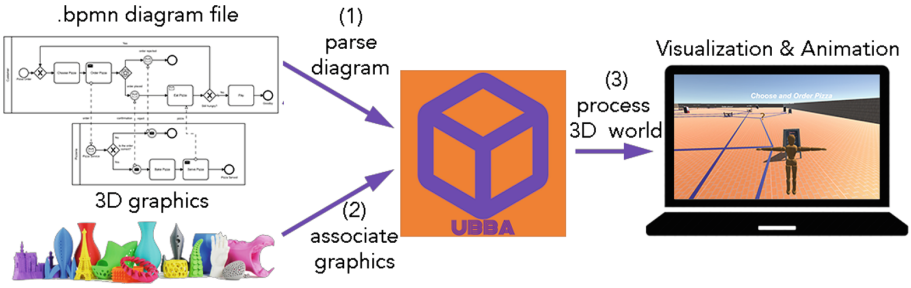

Fig. 1. UBBA functioning

3D Process Visualization. The first characteristic of UBBA is the rendering of a BPMN model into a 3D space. To this aim, UBBA provides three main

features: (i) read a standard *.bpmn* file; (ii) associate a graphic to each diagram element; (iii) show the resulting 3D scene.

To read a BPMN file we resort to our parser, which takes in input a *.bpmn* file and produces a custom data structure. From the parsed model, UBBA collects the information regarding the elements contained in the diagram, such as the id, the name, the type (i.e., activity, gateway, event, etc.), the position in the 2D plane, and the list of nodes reachable from it. The tool exploits this information to set the 3D space. For each discovered element the tool allows to associate a 3D graphic; the designer can choose whether to load them from his/her PC or choose the ones already present in UBBA. External graphics have to be *.fbx* files[1], available online for free or to pay. Even the tokens can be customized: the user can specify a personal graphic representing tokens generated by each pool in the diagram. The chosen graphics for elements and tokens are then embedded by the tool in a 3D space. Those graphics are positioned following the spacial information of the *.bpmn* file and connected by lines that represent sequence flows. Then a view from above of the model is shown, and the user can start to navigate it (see Fig. 2).

Fig. 2. View from the top of the 3D world

3D Animation. The already created custom 3D world let us introduce the animation feature of UBBA. It supports the diagram execution by means of token movements so that the user can continuously check their distribution. This facility results particularly useful for understanding not only the role of the token as process actor but also the semantics of the BPMN elements and the meaning of the diagram.

The animation can be triggered by the user by clicking the *Play* button depicted on the top right side of the interface. In turn, from each start event the chosen 3D graphic, representing the token, starts to cross the diagram following the path indicated by sequence flows. The animation terminates once

[1] Filmbox (*.fbx*) is a file format for geometry definition widely adopted by 3D graphical application vendors, storing 2D, 3D, motion, audio, and video data.

no more token moves are allowed. A token can behave differently depending on the type of the node it is going to cross. UBBA implements the semantics of a core set of BPMN elements. Clearly, this collection is not the full list of BPMN elements, but it is enough to captures several semantic concepts. More in detail, UBBA considers BPMN collaboration or process models with the following elements: *Task*, *Send Task*, *Receive Task*, *Exclusive Gateway*, *Parallel Gateway*, *Event-Based Gateway*, *Start Event*, *Start Message Event*, *End Event*, *Intermediate Catch Event*, and *Intermediate Throw Event*. The semantics of such elements is implemented in UBBA leaving the possibility of adding other element behaviors just extending a project class.

During the animation, different points of views on the 3D environment are available for the user, who is free to switch from a camera to another one using the buttons on the right side of the interface. UBBA has a point of view for each active token (see Fig. 3), plus another that covers the whole collaboration.

Fig. 3. Token point of view

3 UBBA in Action

This section provides the validation of our tool throughout the implementation of a case study. Firstly, we present the BPMN collaboration diagram we used as an example, and then we show how to use it in UBBA.

Pizza Order Collaboration. To better show the tool functionalities we rely on the BPMN collaboration diagram depicted in Fig. 4. The diagram concerns an order placement in a pizzeria that involves two participants: a *Customer* and the *Pizzeria* itself. It is a collaboration diagram, composed of two pools communicating with each other. The whole collaboration aims at illustrating the procedure the customer has to follow in order to get a pizza. The customer plays the first move choosing the desired pizza and placing the order through a message. The message arrival in the pizzeria pool triggers the start of its internal process that immediately retrieves the *order* message. Then, the pizzeria decides to accept or reject the order advising the customer that, in turn, reacts on the acceptance/rejection via an event-based gateway. If the order is rejected the collaboration ends, otherwise the pizzeria prepares the order and sends it to the customer that eats the pizza. Lastly, if the customer is still hungry, he can decide to place another order, otherwise he pays, thus ending the collaboration.

Using UBBA**.** A double click on the tool executable file starts UBBA, which provides the first interaction interface where to load the collaboration diagram,

(see Fig. 5(a) and (b)). For the sake of presentation, the tool provides a preloaded diagram (i.e. the case study in Fig. 4), but of course a *.bpmn* file can be chosen from the file system. Subsequently, UBBA asks for the element graphics to be selected from the tool assets or directly from the file system (see Fig. 5(c)). To recreate the setting described in the *Pizza Order* diagram the designer needs to use graphics capable to embody together the element type and the specific meaning it has in the diagram. For instance, a mailbox is suitable for symbolizing a receive task in general, but in our case study it may results less effective if used in place of the *Eat Pizza* receive task. The user has to consider the specific meaning of the element (the customer receives and eats the requested pizza), hence a 3D table with a pizza on it better clarify this setting.

The choice of the 3D graphics for the tokens is crucial, as well as a meaningful 3D graphic is essential to carry additional information on the diagram meaning. Tokens should represent the characters who perform the activities in the pool (see Fig. 5(d)). Alternatively, tokens describing more abstract actors, for instance, processes representing software components, can be depicted with a sphere. In our case study, we associate a *man* and a *pizza chef* respectively to the tokens of *Customer* and *Pizzeria* pools.

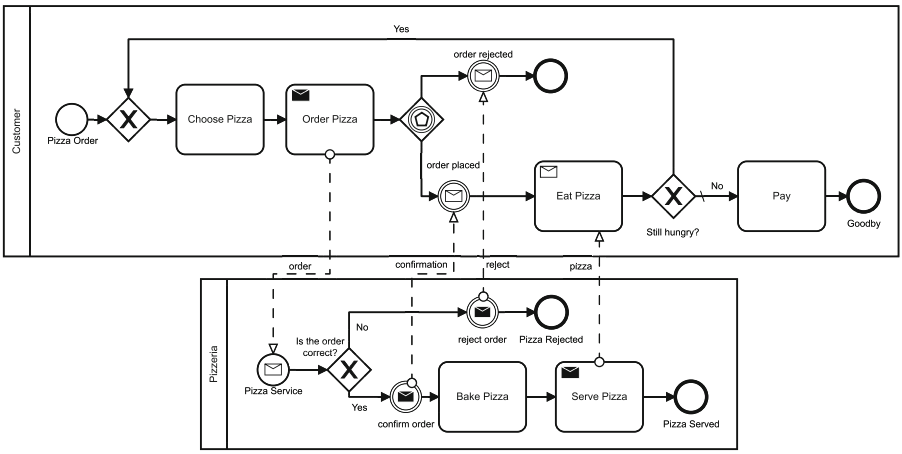

Fig. 4. Pizza order collaboration

Once the association of graphics is finished, UBBA generates the 3D model ready to be animated. A token, depicted with the *man*, is placed on the *Pizza Order* start event, which is represented by a three dimensional model of a pizzeria building. Starting the animation, it is possible to appreciate the UBBA capability to represent the reality. The *man* (the token of *Customer*) crosses the building graphic, meaning that the customer is entering in the pizzeria. In this fashion, the *man* goes ahead following the sequence flows until it reaches a 3D model of a menu that stands for the *Choose Pizza* task. Then, the *man* performs the

Order Pizza send task, played by a pizzeria cash desk graphic, which produces a message token instantiating the *Pizzeria* process and routing the *man* to the event-based gateway.

Having a new token in the collaboration, UBBA adds in the interface a new button identifying the *Pizzeria* instance just created (see Fig. 5(e)). The button switches from the current view (i.e. whole collaboration or man perspective) to that of the *pizza chef* token. It starts its execution crossing the start event until it reaches the XOR split gateway. The choice of the path to follow is made by the user; indeed he/she has to click on one of the 3D arrows depicted over the outgoing sequence flows (see Fig. 5(f)). Following the *No* path the *pizza chef* reaches a *red cross* graphic symbolizing the order rejection. Consequently, a message is sent to the *man* and the collaboration terminates. Otherwise, the *pizza chef* reaches a *green tick* symbolizing the confirmation of the order.

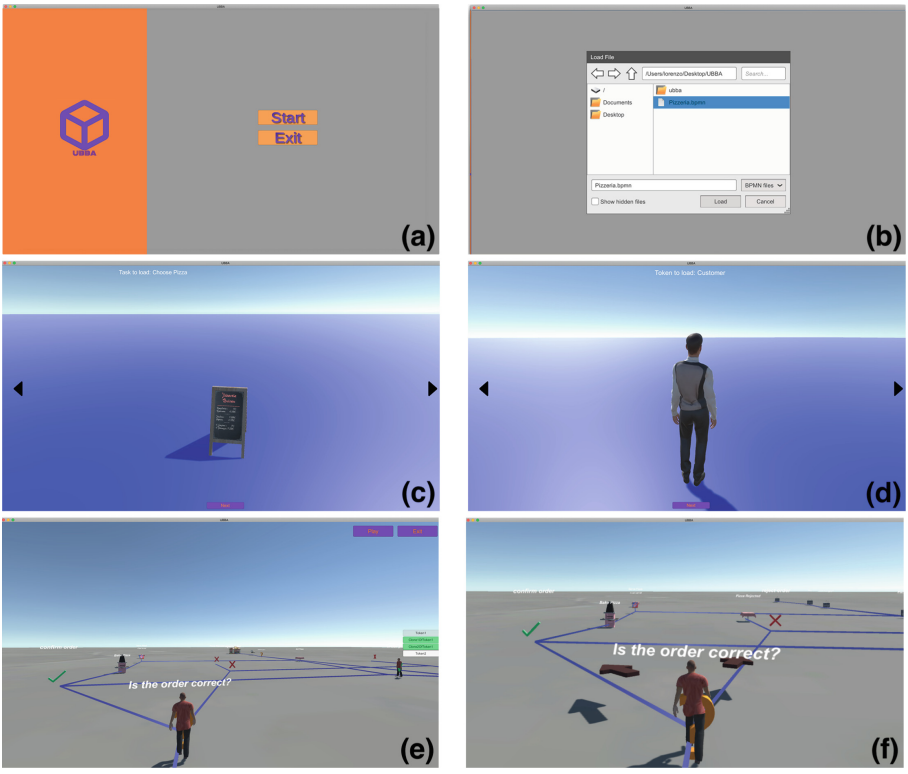

Fig. 5. UBBA in action

Therefore, the *pizza chef* goes ahead to the *Bake Pizza* task, rendered as a pizza oven, and then to the *Serve Pizza* send task, which shows a pizza into its box. The *man* instead waits for the pizza at the *Eat Pizza* task depicted as

a restaurant table with a pizza on it. Once the pizza arrives, the *man* decides to either loop back and ask for another pizza or pay the bill. The *Pay* task is indeed represented as a cash register. The collaboration ends with the *Customer* crossing the *Pizzeria* building.

4 Conclusions

Visualization and animation of BPMN collaborations in a 3D virtual world can improve the communication between business analysts and stakeholders. In this regard, UBBA can effectively help people in sharing business knowledge. UBBA, as well as its source code, binaries, guide, and a short demonstration video are available at http://pros.unicam.it/ubba/. UBBA can be redistributed and/or modified under the terms of the MIT License.

Our tool is still a prototype, several improvements and further developments can be carried. By now, we plan to increase the number of possible customizations, allowing the user to associate 3D graphics also to sequence/message flows, message tokens, pools. Moreover, even if we already support a significative set of BPMN elements and characteristics, some interesting features of the modelling notation have not been considered yet, due to their intricate semantics (e.g., OR-join, sub-processes, cancellation/escalation, boundary events) and/or due to the effort required for the 3D rendering (e.g., data and multiple instances). Finally, in order to assess the potential and the scalability of the approach, we plan to conduct a validation with groups of business process designers, composed by: students with an academic knowledge of BPMN; designers from industry that have more practical skills; and stakeholders that are domain expert. The validation should provide feedback both on the usability of UBBA and on the quality and the benefits of the produced 3D animation.

References

1. IBM Innov8 2.0. http://www-01.ibm.com/software/solutions/soa/innov8/full.html
2. Visual Paradigm. https://www.visual-paradigm.com/
3. Allweyer, T., Schweitzer, S.: A tool for animating BPMN token flow. In: Mendling, J., Weidlich, M. (eds.) BPMN 2012. LNBIP, vol. 125, pp. 98–106. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33155-8_8
4. Becker, J., Kugeler, M., Rosemann, M.: Process Management: A Guide for the Design of Business Processes. Springer, Heidelberg (2013)
5. Betz, S., et al.: 3D representation of business process models. MobIS **8**, 73–87 (2008)
6. Brown, R.A.: Conceptual modelling in 3D virtual worlds for process communication. In: APCCM, pp. 25–32. Australian Computer Society, Inc. (2010)
7. Corradini, F., et al.: A guidelines framework for understandable BPMN models. Data Knowl. Eng. **113**, 129–154 (2017)

8. Corradini, F., Muzi, C., Re, B., Rossi, L., Tiezzi, F.: Animating multiple instances in BPMN collaborations: from formal semantics to tool support. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 83–101. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_6
9. Corradini, F., Muzi, C., Re, B., Tiezzi, F., Rossi, L.: MIDA: multiple instances and data animator. In: Dissertation Award, Demonstration, and Industrial Track at BPM 2018. CEUR Workshop Proceedings, vol. 2196, pp. 86–90 (2018)
10. Desel, J.: Teaching system modeling, simulation and validation. In: WSC, pp. 1669–1675 (2000)
11. Guo, H., Brown, R., Rasmussen, R.: Virtual worlds as a model-view approach to the communication of business processes models. In: CAiSE Forum 2012. Citeseer (2012)
12. Guo, H., Brown, R., Rasmussen, R.: Human resource behaviour simulation in business processes. In: Pooley, R., Coady, J., Schneider, C., Linger, H., Barry, C., Lang, M. (eds.) Information Systems Development. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4951-5_14
13. Hermann, A., et al.: Collaborative business process management - a literature-based analysis of methods for supporting model understandability. In: WI (2017)
14. OMG: Business Process Model and Notation (BPMN V 2.0) (2011)
15. Stamm, S.: Creating a 3D Renderer for BPMN (2018). https://blog.camunda.com/post/2018/02/creating-a-3d-renderer
16. Ware, C., Franck, G.: Viewing a graph in a virtual reality display is three times as good as 2D diagram. In: VL (1994)
17. Wynn, M.T., et al.: ProcessProfiler3D: a visualisation framework for log-based process performance comparison. Decis. Support Syst. **100**, 93–108 (2017)