BEAR 2.0: Enhancing the Environment Model for Animating Environment-Aware BPMN Collaborations

Flavio Corradini[1], Luca Mozzoni[1,*], Jessica Piccioni[1], Barbara Re[1], Lorenzo Rossi[1] and Francesco Tiezzi[2]

[1]School of Science and Technology, University of Camerino, Italy
[2]Dipartimento di Statistica, Informatica, Applicazioni, University of Florence, Italy

Abstract
In most scenarios, to fully understand how organizations work, it is fundamental to study the interplay between the collaboration among process participants and the environment in which they operate. To this aim, we propose BEAR 2.0, a tool for modeling, animating, and debugging environment-aware BPMN collaborations. This version of the tool enhances its predecessor mainly for what concerns (i) the supported environment model, extended with a logical layer to provide abstraction and different levels of granularity, and (ii) the integration of real-world geographic maps, simplifying both the modeling of the environment and the understanding of the process.

Keywords
BPMN collaboration, Semantically Enriched Place Graph, Environment, Modeling, Animation, Debugging.

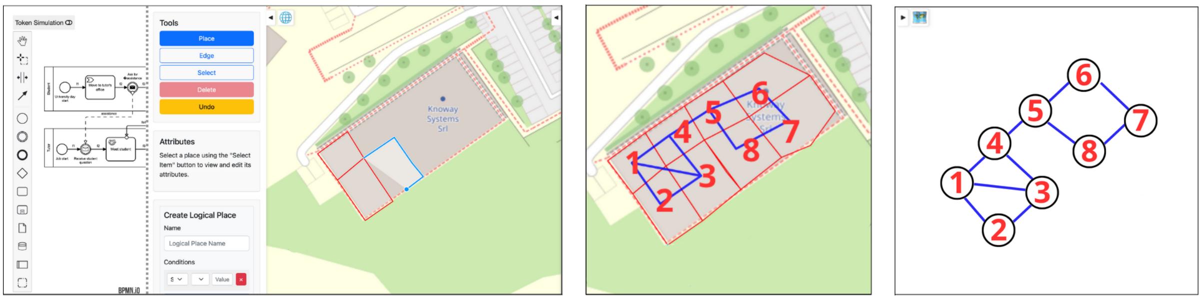
1. Introduction

In modern organizations, the environment in which processes take place plays a central role in shaping how participants behave, interact, and make decisions, e.g., a room's temperature may determine the next activity to perform in a home automation system. In turn, a participant may affect the environment and change its status, e.g., by closing a window. Nevertheless, existing modeling languages, including the de facto standard Business Process Model and Notation (BPMN) [1], overlook the explicit representation of the environment's topology and related attributes, e.g., temperature, purpose, or geometric extent. This limitation highlights the need for a modeling approach that captures the bidirectional interplay between a collaboration and its environment, where each continuously influences the other.

In this regard, in [2], we analyzed such interplay and provided an approach to interconnect BPMN collaborations and the environment. To enforce and support a more precise understanding of these findings, we proposed BEAR (BPMN Environmental AnimatoR) [3]. Then, in [4], we leveraged semantically-enriched place graphs to model an abstract representation of the environment in which collaborations take place. This environment model enables the representation of both a physical layer, which describes places and their topology, and a logical layer, providing high-level abstractions through dynamic place groupings. This enables the representation of both indoor and outdoor environments across different scales, assuming an appropriate level of abstraction. By connecting the concepts of this environment model to the BPMN meta-model, we define environment-aware BPMN collaboration diagrams, and equip them with formal operational semantics.

In this paper, we introduce BEAR 2.0 (referred to simply as BEAR from now on), an enhanced version of the tool that provides a modeling environment for designing environment-aware BPMN collaborations. This version of the tool improves its predecessor mainly in terms of the supported environment model, which integrates a logical layer to provide abstraction and different levels of granularity, and the integration of a geographic map, therefore simplifying the modeling, animation

Proceedings of the Best BPM Dissertation Award, Doctoral Consortium, and Demonstrations & Resources Forum co-located with 23rd International Conference on Business Process Management (BPM 2025), Seville, Spain, August 31st to September 5th, 2025.
*Corresponding author.
flavio.corradini@unicam.it (F. Corradini); luca.mozzoni@unicam.it (L. Mozzoni); jessica.piccioni@unicam.it (J. Piccioni); barbara.re@unicam.it (B. Re); lorenzo.rossi@unicam.it (L. Rossi); francesco.tiezzi@unifi.it (F. Tiezzi)



(a) Environment-aware BPMN modeler. (b) Environment (map). (c) Environment (graph).

Figure 1: The BEAR Modeler GUI.

and debugging of environment-aware BPMN collaborations. These advancements allow users to design process models that are grounded in real-world environments.

It is indeed well known from the literature that animator tools play an important role in easing the understanding of the behavior of business processes [5]. There exist several tools to model and animate BPMN models. Allweyer and Schweitzer [6] and Signavio [7] provide tools to animate the control flow of BPMN processes; Corradini et al. [8] propose the tool MIDA to animate BPMN collaborations with multi-instance participants and data; Abdul et al. [9] present UBBA that creates and then animates a virtual 3D representation of a BPMN collaborations. Within this landscape, BEAR remains the only tool that integrates environmental aspects directly into the BPMN notation.

The rest of the paper is organized as follows. Section 2 describes BEAR functionalities. Section 3 assesses its maturity via case studies. Section 4 concludes the paper by providing links and information about downloading, installing, and using BEAR.

2. BEAR features

This section presents BEAR and its features for supporting the understanding of the interplay between collaboration business processes and the environment. BEAR provides *modeling*, *animation* and *debugging* functionalities. Modeling features allow the designer to create environment-aware BPMN collaboration diagrams, while animation and debugging are used to visualize and inspect the collaboration execution and the evolution of its environment.

Modeling. The designer can use the modeler to create and link the BPMN collaboration diagram and the environment model, as shown in Figure 1a. By clicking on any element in the two models, a property panel opens, which permits adding information related to the selected element and using them to drive the animation. Concerning the environment, BEAR supports a representation of both the physical and logical layer in which collaborations take place. The physical layer includes individual locations where participants operate, called **places**, and their connections, called **edges**. Each physical place and edge can be enriched with a set of key-value attributes, e.g., `available = true`. These attributes can be read and modified by the collaboration model using the element name as a prefix, e.g., `place1.available`. This enables processes to both react to and influence the environment during execution. Attributes can be either spatial, referring to tangible dimensions, e.g., the length of a corridor, or contextual, describing semantic properties, e.g., the purpose of a room. Spatial attributes enable the definition of the area covered by each place, allowing for different levels of granularity in the model. As a result, a place can flexibly represent anything from a single indoor room to an entire outdoor parking area. The physical layer can be directly drawn on a geographic map, but the tool also allows visualizing its underlying place graph, as shown in Figures 1b and 1c, respectively. This dual representation plays a key role in supporting the designer's understanding of the environment, as the place graph highlights the topology of the physical layer. Building on this, the environment model includes a second layer: the logical one. This consists of **logical places** and **views**, which abstract over the physical layer by grouping places dynamically based on their attributes. Logical places are defined by boolean expressions that predicate over environmental attributes. For instance, in the context of a hospital, the expression

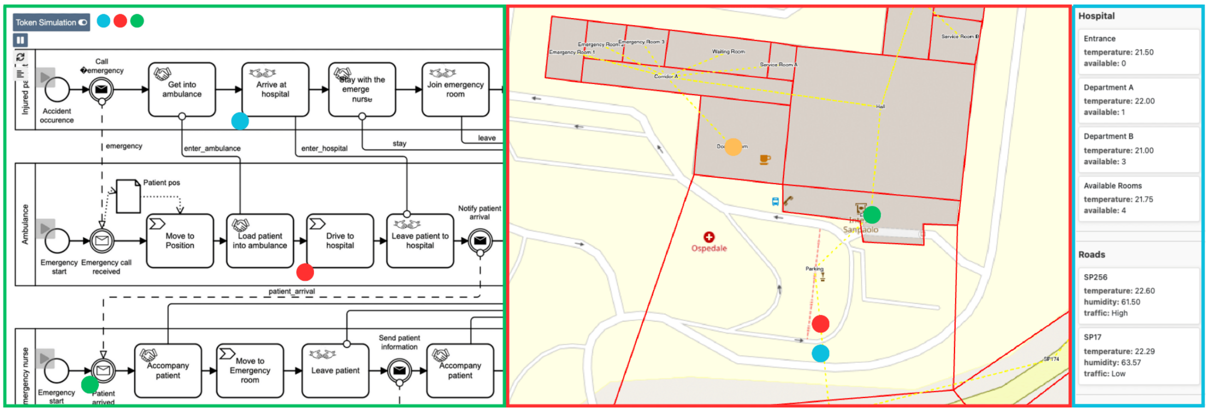


Figure 2: BEAR Animator GUI.

`available == true && department == radiology` may define a logical place called *Available Radiology Rooms*. Since attributes may change over time, logical places are inherently dynamic. Logical places can be directly referenced in the BPMN collaboration model, thus streamlining the modeling of environment-aware BPMN collaborations. Views further enhance the model's expressiveness by enabling interaction at different levels of granularity. A view is defined by specifying a name and a set of logical places it includes. For each attribute of the included places, an aggregation function can be assigned to compute a collective value. For instance, the attribute `availableBeds` in multiple rooms can be aggregated using the `SUM` function to compute the total number of available beds in a building. Concerning the collaboration, the property panel shows for each BPMN element, beyond the already existing properties like element name and identifier, additional information related to the interplay with the environment. In particular, selecting a *pool* element, a **position** attribute can be set, stating the initial place of the participant represented by the pool. This property appears in BEAR as a text box to be filled with a place id. About *BPMN tasks*, the designer can specify (i) **guard** conditions, used to constrain the execution of a task to specific values of environmental attributes, which has to be defined with an expression, e.g., `place1.available == true`, (ii) a list of **assignments**, used to modify environmental attributes values, possibly affecting logical places. In addition, the designer can specify three different types of tasks. *(i) Movement tasks*, characterized by an arrow icon, which can define guards, assignments and also a **destination**, used to define a place to reach, e.g., *Room 123*, from the current participant's position. *(ii) Binding tasks*, graphically represented with a handshake icon, which can be linked to a binding task in the pool of another participant, enabling one to follow the movements of the other. *(iii) Unbinding tasks*, graphically represented with a two-separated hand icon, which breaks the binding between two participants, enabling them to resume independent movements. For what concerns events, the designer can define *intermediate message events* with value passing. Specifically, send events can exhibit a payload in the form of values, e.g., `"Payload"` or `3`; or attribute names, e.g., `place1.attr`. Instead, receive events can exhibit an attribute name where to store the received payload. Finally, boolean expressions can be assigned to the outgoing sequence flows of XOR gateways to drive decisions dynamically, which can also depend on the value of environmental attributes.

Animation. The interface of the BEAR animator, shown in Figure 2, is divided into three main parts, i.e., the **BPMN diagram** on the left (highlighted in green), the **environment model** in the center (highlighted in red), and the **data panel** on the right (highlighted in light blue) displaying elements from both the physical layer and the logical layer, with their associated environmental attributes. The key characteristic of the animator is its dynamic visual animation. It shows, in a step-by-step fashion, the evolution of the BPMN collaboration with its environment. By selecting the *Token Simulation* toggle in the interface, the environment model will be frozen in its current state and a play button will appear over each fireable start event. When a button is clicked, the respective process is activated. This creates a new token, represented as a small colored circle, at the start event of the BPMN process. Another token of the same color is placed in the corresponding location of the environment model, based on the position property set on the pool. Then, the tokens start to cross the BPMN collaboration model and the environment model according to the operational rules induced by our formal semantics [4]. The right-side panel allows users to inspect attributes of places, both physical and logical, and edges, and track their evolution over time. It lists contextual and spatial attributes, with the latter shown on the map by highlighting a place's extent when hovered.

Debugging. During the animation, warning and error messages support effective debugging of environment-aware BPMN collaborations by enabling the designer to identify undesired executions. Warnings, depicted in Figure 3 in yellow, indicate potential deadlock situations, i.e., two participants attempting a binding task from different positions, a participant unable to reach a specific destination, or a violated guard. Animation continues, as these conditions may eventually resolve. Instead, errors, depicted in Figure 3 in red, occur when the collaboration cannot proceed, i.e., a participant's initial position is missing or two bound participants are moving in opposite directions, and cause the animation to stop. If no errors or deadlocks occur, the animation ends when no tokens can move forward. Nevertheless, it can be paused at any time, allowing users to inspect the token distribution across the two models and assess the environment's state through the right-side panel.

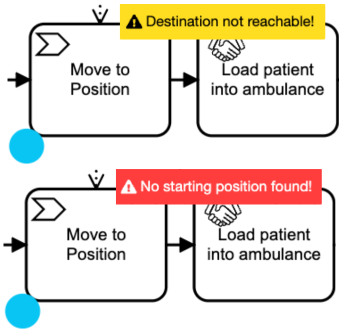


Figure 3: Debugging features.

3. Maturity of the tool

BEAR is a web application written in JavaScript, which makes it accessible from any browser without a server backend, and hence, runnable in any operative system. It extends the bpmn-js[1] toolkit and its token simulation plug-in [10] and it leverages the OpenLayers library[2] to embed real-world coordinates into the model. BEAR can be installed as a Node application or accessed online without the need to install any software. By its nature, it does not require significant computational power. Indeed, an animator executes only a single run of the model using the input configuration, differently from simulators and model checkers that execute many runs of the same model, even all possible runs if necessary. For this reason, the tool performances are not affected by the dimension of the input environment-aware BPMN collaboration diagram, and thus, it does not suffer from scalability issues.

To assess the maturity of BEAR, we rely on a set of 21 environment-aware BPMN collaborations. The 10 case studies originally introduced in BEAR 1.0 have been successfully adapted to the current version of the tool. Moreover, we have added 11 new case studies spanning a variety of domains, e.g., agriculture, transportation, restaurants, universities, healthcare, product sales, and logistics, three of which are inspired by examples from the literature [11, 12, 2]. We chose environment-aware BPMN collaborations covering various dimensions, topology, and elements. The BPMN collaborations range from 2 to 5 participants and from 11 to 60 elements. They are both structured and unstructured and contain all the BPMN elements formalized in [4]. The space models' dimensions range from 17 places and 22 edges to 38 places and 82 edges in the physical layer, and up to 12 places in the logical layer. Each case study comes with a short description and two environment-aware BPMN diagrams, one is correct, i.e., the animation terminates without errors and each process terminates successfully; the second contains an intentional modeling error to showcase the debugging features of the tool.

The evaluation of these case studies has shown how environmental conditions can change the outcome of the collaboration process. For the sake of presentation, we briefly discuss one of the case studies concerning an *Emergency Response* collaboration within a hospital and its surroundings. The collaboration concerns four participants: *Injured Patient*, *Ambulance*, *Emergency Nurse*, and *Emergency Doctor*. It starts when the *Injured Patient* is involved in a car accident and calls an ambulance, providing its

[1] https://bpmn.io/toolkit/bpmn-js/
[2] https://openlayers.org

position. Upon receiving the call, the *Ambulance* moves to the patient's position. During this operation, the ambulance's travel can be seen in both models: its process token waits near the movement task while its environment token moves across the map. However, if the accident occurs in an unreachable location, such as due to a closed road, the tool displays a warning message. Similarly, failing to specify the ambulance's initial position displays an error message and halts the animation. The collaboration continues with the ambulance picking up and transporting the patient to the hospital. From a process perspective, the tokens wait for each other at their respective binding tasks. When bound, the patient's token waits at the next unbinding task, while the ambulance executes the movement task to return to the hospital. About the environment, both tokens move across the map, with the ambulance dictating the patient's movements. Once at the hospital, the ambulance's token reaches an unbinding task, detaching from the patient, who will be rescued by the *Emergency Nurse* and the *Emergency Doctor* participants. By relying on both place graphs and geographic maps, this new version of BEAR simplifies both the modeling of the environment and the understanding of the process.

4. Resources

We provide BEAR under the MIT license in source code form and as an online service. A demonstration video shows the modeling, animation and debugging of the *Emergency Response* collaboration presented in Section 3. Additionally, a comprehensive user guide is available in the repository, detailing installation and usage, along with a set of environment-aware BPMN collaboration models, each with a short description of the scenario. All resources are accessible at https://pros.unicam.it/environmental-bpmn.

Declaration on Generative AI

No generative AI tools were used in the preparation of this work.

References

[1] OMG, Business process model and notation, BPMN V2.0 (2011).

[2] F. Corradini, et al., On the Interplay Between BPMN Collaborations and the Physical Environment, in: BPM, volume 14940 of *LNCS*, Springer, 2024, p. 93–110.

[3] F. Corradini, L. Mozzoni, J. Piccioni, B. Re, L. Rossi, F. Tiezzi, BEAR: BPMN and Environment AnimatoR, in: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM, volume 3758, 2024, pp. 121–125.

[4] F. Corradini, L. Mozzoni, J. Piccioni, B. Re, L. Rossi, F. Tiezzi, Modeling, Formalizing, and Animating Environment-Aware BPMN Collaborations, in: BPM2025, Springer, 2025. To Appear.

[5] I. Maslov, S. Poelmans, Facilitating the comprehension of business process models for unexperienced modelers using token-based animations, Information and Management 61 (2024) 103967.

[6] T. Allweyer, S. Schweitzer, A Tool for Animating BPMN Token Flow, in: Int. Workshop on BPMN, volume 125 of *LNBIP*, Springer, 2012, pp. 98–106.

[7] S. GmbH, Signavio, 2018. URL: http://www.signavio.com/.

[8] F. Corradini, C. Muzi, B. Re, L. Rossi, F. Tiezzi, Formalising and animating multiple instances in BPMN collaborations, Infomation Systems 103 (2022).

[9] B. M. Abdul, F. Corradini, B. Re, L. Rossi, F. Tiezzi, UBBA: unity based BPMN animator, in: CAiSE Forum, volume 350 of *LNBIP*, Springer, 2019, pp. 1–9.

[10] P. Fromme, S. Warnke, P. Dehn, bpmn-js token simulation, 2017. URL: https://github.com/bpmn-io/bpmn-js-token-simulation.

[11] R. Saddem-Yagoubi, P. Poizat, S. Houhou, Business processes meet spatial concerns: the sbpmn verification framework, in: Formal Methods, Springer, 2021, pp. 218–234.

[12] F. Corradini, S. Pettinari, B. Re, L. Rossi, F. Tiezzi, A technique for discovering BPMN collaboration diagrams, Software and Systems Modeling (2024) 1–21.