A dynamic scenario by remote supervision: a serious game in the museum with a Nao robot

Damien Mondou, Armelle Prigent, Arnaud Revel

A dynamic scenario by remote supervision: a serious game in the museum with a Nao robot.

Damien Mondou, Armelle Prigent, and Arnaud Revel

L3i - University of La Rochelle - France,
damien.mondou@univ-lr.fr,
Home page: http://l3i.univ-larochelle.fr

Abstract. This paper presents a new approach to designing and supervising an interactive experience. The approach is implemented by creating a serious game with a Nao robot. This game allows youth to discover the ethnographic artifacts of La Rochelle's natural history museum in a playful manner. The design phase of the game is divided into two steps. The first step defined the atomic behaviors grouped within the pattern. In the second step, the agents implementing these patterns were created; they specified the contents and behaviors to be executed on the controlled process, in our case the Nao robot. The first objective is to externalize the contents of the game (e.g. robot speech) and the real behaviors of the process (e.g. the different postures and gestures of the Nao) in a database. The second objective is to be able to define a serious game without constraints on the process piloted.

Keywords: Robot, serious game, supervision, game design

1 Introduction

Museums continually search for new ways to both increase their visitor count and allow visitors to discover the museum artifacts while gaining new knowledge. For several years, the serious game has demonstrated that it can playfully help users increase their understanding of a subject. The benefits of learning through play have been demonstrated [24, 9]. Thus, many cultural sites have been equipped with fun features that allow visitors to find out about the collections or works of art in another manner.

For it, museum's curators may use on-screen games [7], augmented reality [17], transmedia and virtual immersion solutions [5] to allow visitors to navigate the artifacts and to learn about the cultural and artistic items at the cultural center. A possible continuation of these serious games is playing with a robot, which has been proposed for several years.

There are many advantages to interactive games with a robot. Robot-based computer vision systems can be integrated into the museum space and challenge a visitor to participate. Thus, a simple visitor can become a player for part of their visit. In addition, people can visit the museum and play without having

to be equipped with a tablet, phone or others equipments which might interfere with their visit.

In collaboration with the Museum of Natural History of La Rochelle in France, we have developed an interactive experience for young visitors. This serious game allows the youth to discover, in a interesting manner, the ethnography section of the museum. This serious game is embedded in the Nao robot, that was initially developed by the French company Aldebaran (now Softbank Robotics). Nao encompasses a set of sensors (e.g. a camera, sonar, and tactile sensors) that allow it to perceive the environment. This robot is also able to interact with a person through its microphones and speakers. Thanks to a serious game, Nao is in charge of the playful discovery of the museum's artifacts.

In parallel with this experience and in the framework of a transdisciplinary study conducted with a laboratory specialized in marketing research, we studied the reactions of visitors to the presence of a robot in a cultural place. We found that, depending on the age of the visitor, the reactions differ. Children responded quicker to the robot and could be impatient waiting for the robot's response; adults were more hesitant. Thus, it is important to plan for different visitor scenarios (e.g. accounting for the individual's age).

It is important to note that scripting a game through a robot creates new challenges in design, especially when several scenarios are desired. For instance, the user can interact with the robot in various manners, which means these different interactions must be precisely orchestrated. Thus, the robot can move and gesture; see, recognize, and talk to the visitor; and wait for the visitor's responses. Another issue is that the current programming interfaces do not allow for content outsourcing, which is a challenge for the replicability of the interactive experience (e.g. in another museum).

In this paper, we offer an approach for the supervision of interactions with the robot through a high-level model to represent the different execution contexts.

We propose a formal model for the scenario, a generic supervision tool and the connection from Nao to the supervision tool (wrapper). After describing the interactive experience with Nao, developed for the museum, we will present the method for the game's modeling and dynamic supervision.

2 Robots in Everyday Life

Robots become more and more present in our everyday lives and allow us to simplify our daily activities. For instance, at King's College London, the receptionist is a robot named Inkha (Figure 1 (b)). Robots can also be found in supermarkets to help shoppers find different products and provide nutritional advice, such as LoweBot [19], Aiko Chihira [23], and Pepper [3]. Robots can also be major actors in artistic works, such as Nao [2] and Poppy [13]. Artists and scientists have proposed a human-robot dance performance at the School of Moon in 2016 in order to question perceptions and representations of the body.

In this article, we are particularly interested in the presence of robots in cultural places. We have found several projects that introduce robots in museums

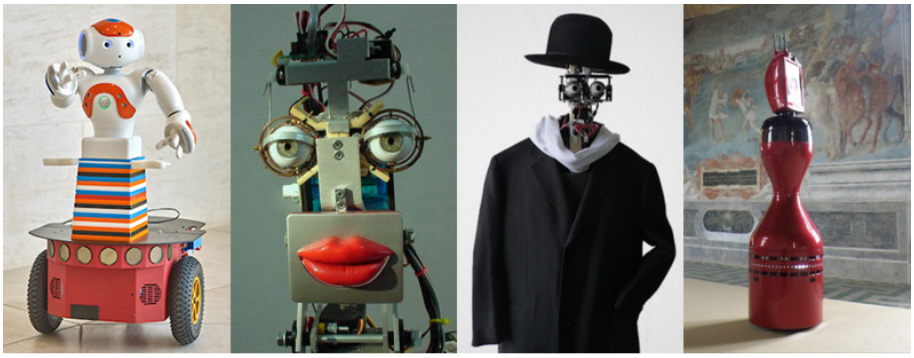

Fig. 1. Experiences with robots in museums

for the reception and orientation of the visitors. Since 2014 at the National Museum of Emerging Science and Innovation in Tokyo, two robots have been in charge of welcoming visitors and can also read the news to them. At the Eppur Si Muove exhibition at the Museum of Modern Art in Luxembourg, a Nao robot with the base of a mobile robot (Figure 1(a)) is itself an interactive work and is responsible for guiding and presenting the works to the public [10]. In France, as part of a project on the artificial esthetism of machines, the robot Berenson (Figure 1(c)) was deployed at the Quai Branly Museum. This robot behaves like an art critic, as he expresses an emotion when he sees a work. His opinion will evolve thanks to the reactions of the people around him [4]. Furthermore, some museums, because of their architecture, are not accessible to people with a motor handicap. The Oiron castle museum in France observed this issue and, on the first floor of the museum, introduced a robot that is able to be remotely controlled by a joystick from the ground floor, which allows individuals with a handicap to visit the works that were previously inaccessible to them. This robot is called Norio (Figure 1(d)) and has been used in the museum since November 2014 [11].

3 Playing with Nao in the Museum

The scenario proposed in our experiment aims to allow the visitor (or player) to interact with the Nao robot to discover the Museum's Kanak masks collections. The robot asks the player a number of questions. This first experiment of dynamic supervision is based on an interactive quiz. With each new question, the player is required to move around the museum to find the answer and give it to the robot. Players can start a session with Nao at any time since a visual

recognition algorithm allows the robot to continue the question sequence with the visitor.

Fig. 2. Nao in the La Rochelle museum, France

The experiment, shown in Figure 3, proceeds as follows. At the beginning of the session, Nao is in a waiting position, the robot remains so until it detects the presence of a human in its field of vision. Then, Nao attracts the visitor's attention by calling and inviting the individual. Once the visitor is close enough to Nao, the robot asks the individual to place themselves in front of Nao in order to trigger the visual recognition algorithm, which will allow the robot to identify the player during the different phases of the session. If the face is recognized, Nao resumes the question sequence with this identified visitor. If the robot does not recognize the face, Nao asks the visitor if they want to play; if the visitor refuses, Nao goes back into the waiting situation. If the user accepts, then Nao records the visitor's face with an identifier and starts the quiz.

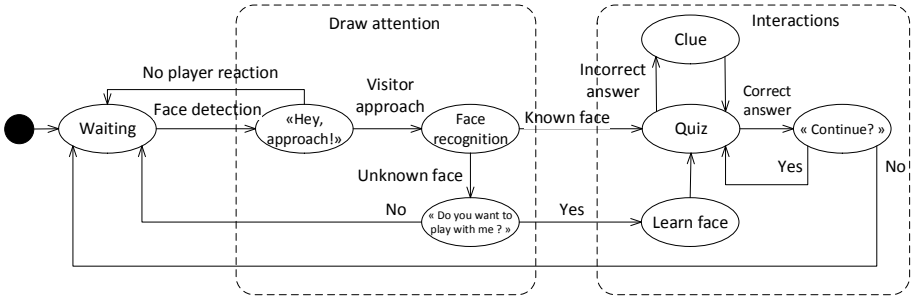

Fig. 3. The interactive experience scenario

3.1 First Approach : the Linear Programming of Nao

The first game session was completed by a purely linear development through the graphical programming tool called Choregraphe, which was delivered with Nao - and supported by Softbank Robotics - in order to pilot Nao. Choregraphe is a simple tool for combining predefined behaviors and descriptions of new behaviors, with Python language support. The pre-existing behaviors provided to Nao can be modified, since the Python code is accessible from Choregraphe. This tool offers some visual abstraction, which is then interpreted by the robot's NAOqi, the software that runs on the robot and controls it. The visual language of Choregraphe is translated through behaviors. Each behavior is defined by a name, input(s), output(s), parameter(s), and typically a Python script that contain the commands to be executed. When a signal arrives at the input, the behavior is loaded and performed. Once the output is stimulated, the box is onloaded.

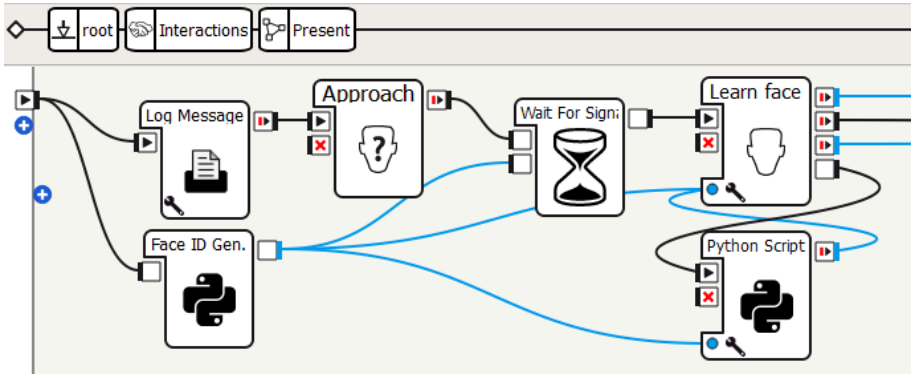

Fig. 4. Choregraphe programming for the game

The entire interactive experiment in the museum was modeled and implemented through Choregraphe. The model obtained can be compared to a Petri net [20] where the places are behaviors (modeled by boxes) and the transitions are the links between these behaviors. While this tool has proven to be effective, it does not meet all of the researchers' expectations. For instance, Choregraphe does not make it possible to define a scenario, accounting for the **space** for which it was designed. The goal is to be able to deploy this type of experience in different museums with minimal processing effort. In order for this experience to be adaptable to various museums, the **contents** of the experiment must also be modified. Though the **interactions** are relatively simple and the sequence with a player contains only three questions, the possible modifications of the dialogues and, the content or the re-scheduling of the interactions proved to be laborious and hazardous when they intervened shortly before a session with the

public. The final limitation concerns the **temporality** of the experience. We wish to be able to apply a duration period to a particular event in the scenario, such as a time limit on how long the visitor has to answer the question posed by the robot.

3.2 Modeling and Dynamic Supervision of the Game

These different limitations led the researchers to propose a new architecture allowing to take into account the activity's temporality, to manage external contents and to facilitate the deployment of the same scenario in another museum. Softbank Robotics proposed a software development kit (SDK) in C++ and Python to control Nao. Thus, we used the Python SDK in our approach. The first step consisted of developing a wrapper which would allow dialogue to occur between the robot and the supervision engine. Therefore, the robot's sequence consisted of waiting for a command that was supplied through a server (i.e. an order corresponding to a module programmed in Nao that was triggered by the supervision engine). The set of these orders (programming module) found their match in the model we created. This modeling method then considered different entities corresponding to the various possible states of Nao based on the sequence of the quiz, including waiting, "hey, approach", facial recognition, and learning the face. These entities were then grouped to define the execution contexts. The architecture of our solution, which allows for the design ans supervision of an interactive experiment (here, the Nao robot) is shown in the figure 5.

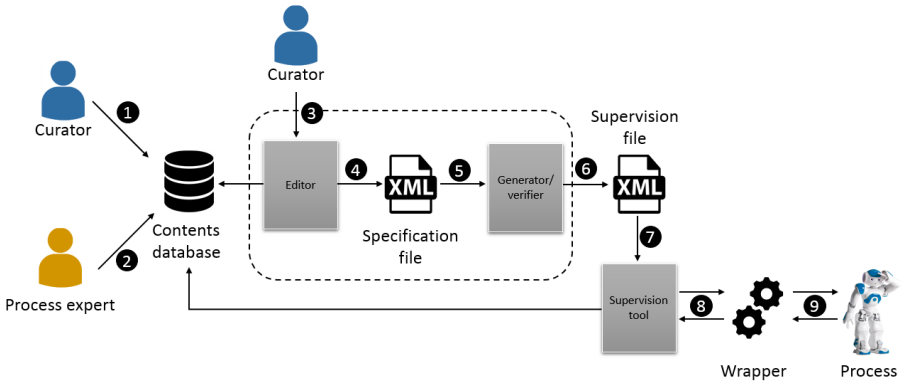

Fig. 5. Architecture of the design and supervision of an interactive experience

As shown in figure 5, in the first step, the museum's curator and process' expert define the contents (e.g. the text of presentation of the works and quiz) and the executable behaviors (e.g. postures and gestures) respectively and then insert them into the database. The museum's curator can then design the interactive experience through the model's editor. The curator uses the contents

present in the database to construct its scenario, associating the robot's behavior and speech and obtaining a specification file in XML format (Figure 7). This file is then used to generate the supervision file which contains the scenario modeled by timed automata as the input of the supervision tool. This tool, described in the following section, will then pilot the robot thanks to the scenario designed by the curator.

4 A Dynamic Supervision Approach

The final objective of our work is proposing a platform that allows for dynamic adaptation to the user and that permits the supervision of the activity, taking into account the content, interactivity, time, and space. Indeed, designing an interactive experience that fully corresponds to a user is tricky. A way of managing the quality of the experience is adjusting the content to the user's context. In this case, the management of the activity is based on an *a priori* modeling approach called *top-down*, as management have been used in web adaptation [26, 8] and in a video game adaptation framework [14, 16, 15].

We propose a formal model of the game's execution based on two layers of modeling. The behaviors of the entities are modeled by input/output finite state machines, and the objective is to leave a significant possibility of choice to the user while guaranteeing the quality of the narrative framework. A first approach of dynamic supervision has been proposed in [22, 21], and having given rise to a supervision framework called #Telling. We propose an extension of this work that facilitate the control allows content outsourcing and time constraints representation. Thus, #Telling, whose modeling is divided into three layers, can supervise the activity based on the execution scenes. Although this framework's effectiveness has been proven, it does not allow for the temporal dimension to be taken into account, in either the definition of the behavior or in the transition from one situation to another. Finally, content management is not supported. We propose a new model that is able to address these issues and simplify the modeling phase of the experiment using the two-layer model, which is described in the next section.

4.1 Two-Layers Model

Formal models have been proven to be effective for controlling interactive experiments. Petri nets are often used to model and verify asynchronous activities, particularly serious games [1, 18]. Linear logic approaches have been used to represent ressources consumption [6]. In our approach, we use timed I/O finite state machines networks, which are more suited to represent synchronous systems with time constraints. Modeling an interactive experience is divided into two parts:

– We first describe the experience in an abstract manner, by defining the reusable entities (e.g. behaviors and patterns), modeled by timed automata. This step creates the *declarative layer*;

– By instantiating patterns into agents, which are grouped into scheduled execution contexts, we define the *implementation layer*.

The process of creating an interactive system is illustrated in Figure 6.

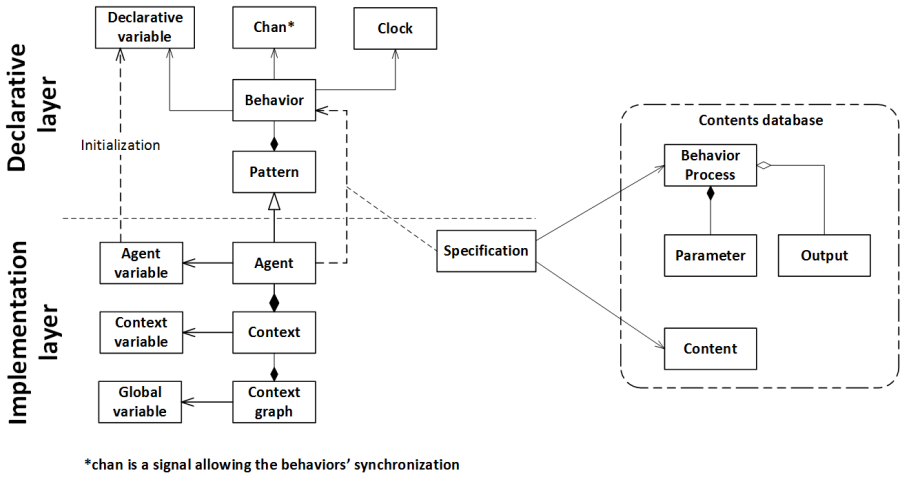

Fig. 6. Two-layer approach for interactive modeling with content management

Declarative layer The declarative layer is the layer in which we define the generic entities of the modeled system, namely the atomic behaviors that can be executed. These behaviors are associated with declarative variables (e.g. integers, strings or Boolean) and channels, which are signals that allow several behaviors to be synchronized. The behaviors are then grouped into patterns that represent a set of entity specific behaviors that can be reused by the multiple inheritance principle.

Implementation layer The implementation layer is the layer where the designer defines the agents, that can instantiate the behaviors of several patterns and aggregate atomic behaviors not present in the instantiated patterns. This principle of multiple inheritance allows the designer to reuse the patterns defined in the declarative layer as many times as necessary without needing to redefine the patterns. When the designer is defining the agents, for each behavior they specify the content or action to be performed on the process. These agents are then linked in **execution contexts** representing the specific context in which agents interact with each other.

Fig. 7. Scenario specification in XML format

4.2 Example of the Model for the Robot in Museums

We used our two-layer model to pilot the Nao robot in the natural history museum. We now present examples of behaviors for the implementation of the serious game embedded in the robot. In the declarative layer, we defined two behaviors, including "say" and "animate". Our scenario contains several phases in which the robot had to make a speech while being animated. We combined these two behaviors in an "animatedSay" pattern. This pattern can be reused as many times as necessary in the implementation layer.

For example, we wanted to define a "speechAwait" agent who would be responsible for attracting visitors. This agent would implement the previously defined "animatedSay" pattern. When defining the agent, the designer would perform a specification of the "say" and "animate" behaviors.

Thus, according to Figure 7, the module with ID number 673 in the database is associated with the "say" behavior and corresponds to the speech module programmed in Nao. With this behavior, we associated the content with ID number 9, which represents the speech uttered by the robot. Speed speech and language parameters had to be entered.

The interactive experience thus scripted was modeled by timed automata. We then obtained a supervisory file, as the input of our supervision tool.

4.3 Dynamic Supervision

This section introduces the dynamic supervision of the interactive experience. Since the interactive experience has been modeled, our supervision tool was responsible for running the experience and invoking it remotely in the Nao robot. The aim of our platform was to dissociate the modeling and the supervision of the activity. It was then possible to use the same model to control different processes, provided one develops a specific client for each process. A sequence diagram of the supervision is detailled in Figure 8.

This supervision tool uses the specification file previously generated as an input and loads the defined context graph (Figure 9). The transition from one context to another is completed by receiving signals (e.g. errorDetection, wait, detectUser, noGame and confirmGame) from a source context. The transition can also be done via global variables to which we can apply guards and updates, like an automaton. Thus, we define four contexts:

- **Waiting**: The robot goes into the rest position and waits until it detects the presence of a visitor. If a detection problem occurs, the context re-executes with the signal *errorDetection*. Otherwise, the supervision tool changes to the *Explain* context;
- **Explain**: At this step, the serious game is presented and the robot asks the visitor if they want to play;
- **Disappoint**: The visitor left or does not want to play. Nao expresses disappointment;

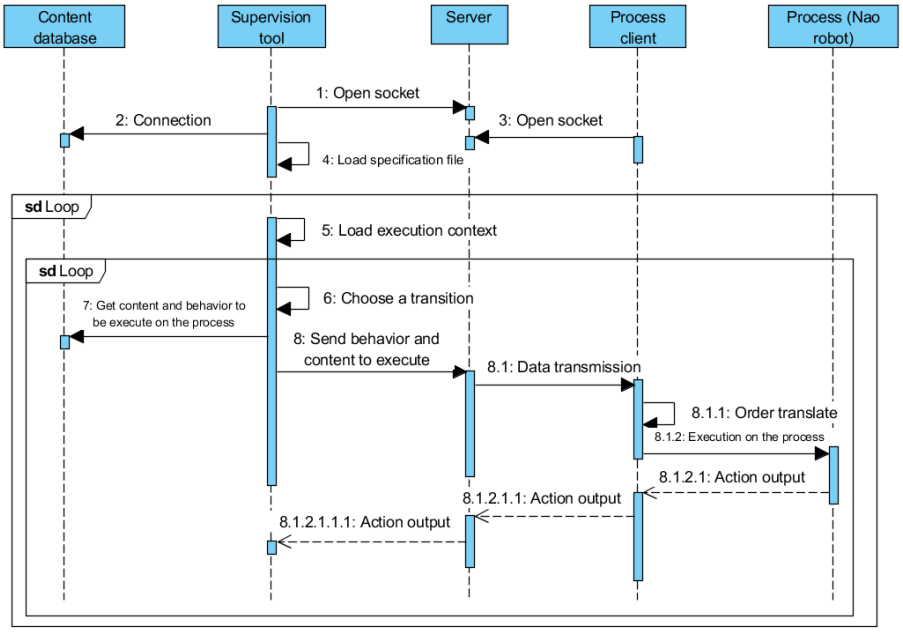

Fig. 8. Supervision sequence diagram

– **Quiz**: Here, the robot utilizes a facial recognition to determine if it knows the player. If Nao knows the individual, the robot waits for the answer to the question previously asked; otherwise, Nao learns the player's face and ask them a question.

Each context is defined as a timed automaton, and each transition resembles a command to be executed on the process. An example of the context "Waiting" is shown in Figure 10.

This serious game is executed on the Nao robot. We use the Python SDK provided by Softbank Robotics to control the robot through a client developed for this purpose. Each controlled process must possess its own client in order to interpret the data received from the server. The client links the supervision tool and the controlled process. Nao receives orders from the supervision tool and translates them into a language that is understandable by the robot.

5 Conclusion and Future Work

This paper presented a supervision model for interactive experiences. The modeling was divided into two steps. The first step defined the atomic behaviors grouped within the pattern. In the second step, the agents implementing these patterns were created; they specified the contents and behaviors to be executed

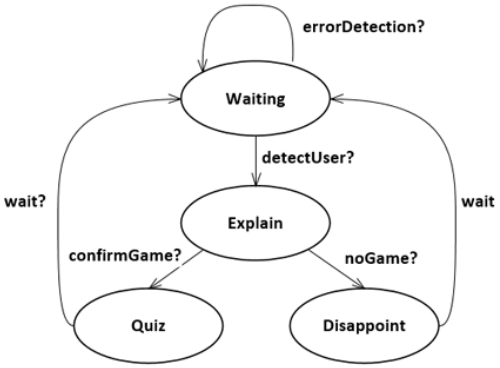

Fig. 9. Context graph of the serious game

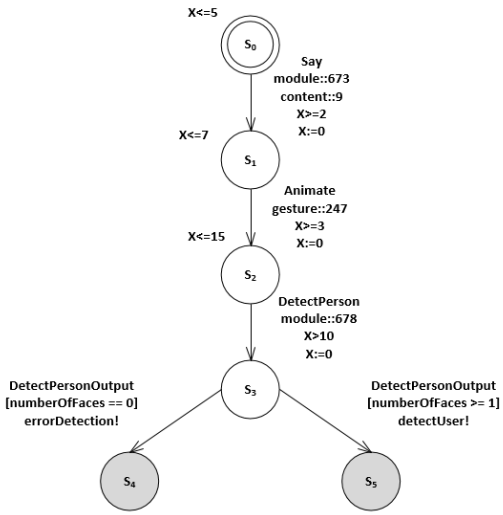

Fig. 10. Context "Waiting" modeled by time automaton

on the controlled process (the Nao robot). Our contribution lies in two main points:

- **A simplified construction of the scenario** based on a two-layer model that helped with the modularity, reuse and automatic construction of the entities;
- **The external content management**. Conventional design tools (like Choregraphe) do not allow this management. The modeling and adaptation of an experiment from one museum to another would have been laborious tasks. Thanks to our approach, the contents were managed externally. Our supervision tool was then in charge of recovering the contents in the database to control the process.

With the exception of the editor for designing the supervision's input file (which was thus designed manually), which is under development, all of these features were implemented. To test this approach, we modeled a simple serious game with the Nao robot. The validation of our model allows us today to consider the design of a more complex game. In the long run, we aim to combine this top-down approach with a bottom up approach [25, 12] in order to integrate a relevance loop. Thus, due to the observations made during the execution of the activity and the machine learning process, we aim to be able to modify the *a priori* model by adding or removing behaviors. The machine learning process will allow the experience to be adaptated to the actual behavior of the end user, which is nearly impossible to predict during the design. Finally, we want to extend our model to the management of the locations where the different contexts of the

scenario are executed; this outreach will allow us to take advantage of the visitors' real-time geolocation thanks to the e-beacon technology. It will also be possible to choose the dialogues and behaviors of the robot according to the observed displacements of the visitor.

References

1. Araújo, M., Roque, L.: Modeling Games with Petri Nets. Digital Games Research Association (DiGRA) (2009)
2. Aspord, E., Becker, J., Grangier, E.: Link human/robot. Van Dieren eds (2014)
3. Boom, D.V.: Pepper the humanoid robot debuts in france, https://www.cnet.com/news/pepper-the-popular-humanoid-robot-debuts-in-france/, year=2015
4. Boucenna, S., Gaussier, P., Andry, P., Hafemeister, L.: A robot learns the facial expressions recognition and face/non-face discrimination through an imitation game. International Journal of Social Robotics 6(4), 633–652 (Nov 2014)
5. Carrozzino, M., Bergamasco, M.: Beyond virtual museums: Experiencing immersive virtual reality in real museums. Journal of Cultural Heritage 11(4), 452 – 458 (2010)
6. Champagnat, R., Estraillier, P., Prigent, A.: Adaptative execution of game: Unfolding a correct story. In: Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology. ACE '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1178823.1178941
7. Coenen, T., Mostmans, L., Naessens, K.: Museus: Case study of a pervasive cultural heritage serious game. J. Comput. Cult. Herit. 6(2), 8:1–8:19 (May 2013)
8. De Virgilio, R.: AML: A modeling language for designing adaptive web applications. Personal and Ubiquitous Computing 16(5), 527–541 (2012)
9. Dietze, D.: Playing and learning in early childhood education. Wadsworth Publishing Company (2011)
10. Henaff, P.: Entre art et science: Guido, un robot guide espiègle au musée d'art moderne de luxembourg. session vidéo, Journées Nationales de la Recherche en Robotique (JNRR) (october 2015)
11. Khlat, M.: Norio, the robot guide of the oiron castle (2014), *http* : *//www.tourmag.com/Norio − the − robot − guide − of − the − Oiron − Castle*$_a$71190.*html*
12. Kop, R., Toubman, A., Hoogendoorn, M., Roessingh, J.J.: Evolutionary Dynamic Scripting : Adaptation of Expert Rule Bases for Serious Games 2, 53–62 (2015)
13. Lapeyre, M., Rouanet, P., Oudeyer, P.Y.: Poppy: a New Bio-Inspired Humanoid Robot Platform for Biped Locomotion and Physical Human-Robot Interaction. In: Proceedings of the 6th International Symposium on Adaptive Motion in Animals and Machines (AMAM). Darmstadt, Germany (Mar 2013)
14. Liapis, A., Martínez, H.P., Togelius, J., Yannakakis, G.N.: Adaptive game level creation through rank-based interactive evolution pp. 1–8 (Aug 2013)
15. Louchart, Y., Aylett, R.: Emergent narrative, requirements and high-level architecture p. 308 (2004)
16. Magerko, B.: Building an interactive drama architecture (2003)
17. Miyashita, T., Meier, P., Tachikawa, T., Orlic, S., Eble, T., Scholz, V., Gapel, A., Gerl, O., Arnaudov, S., Lieberknecht, S.: An augmented reality museum guide. In:

Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 103–106. ISMAR '08, IEEE Computer Society, Washington, DC, USA (2008)
18. Natkin, S., Vega, L.: A Petri Net Model for the Analysis of The Ordering of Actions in Computer Games. In: GAME ON 2003. France (Jan 2003), london, October 2003
19. Prnewswire: Lowe's introduces lowebot - the next generation robot to enhance the home improvement shopping experience in the bay area (2016), http://www.prnewswire.com/news-releases/lowes-introduces-lowebot—the-next-generation-robot-to-enhance-the-home-improvement-shopping-experience-in-the-bay-area-300319497.html
20. Reisig, W.: A Primer in Petri Net Design. Springer Compass International, Springer Berlin Heidelberg (2011)
21. Rempulski, N.: Synthèse dynamique de superviseur pour l'exécution adaptative d'applications interactives. Ph.D. thesis, Université de La Rochelle (2013)
22. Rempulski, N., Prigent, A., Courboulay, V., Perreira Da Silva, M., Estraillier, P.: Adaptive Storytelling Based On Model-Checking Approaches. International Journal of Intelligent Games & Simulation (IJIGS) 5(2), 33–42 (Nov 2009)
23. reuters: Humanoid robot starts work at japanese department store, http://www.reuters.com/article/us-japan-robot-store-idUSKBN0NB1OZ20150420
24. Samuelsson, I.P., Fleer, M.: Play and learning in early childhood settings : international perspectives. Springer Dordrecht ; London (2008)
25. Spronck, P., Ponsen, M., Postma, I.S.k.E.: Adaptive game AI with dynamic scripting pp. 217–248 (2006)
26. Wang, C., Wang, D.Z., Lin, J.L.: ADAM: An adaptive multimedia content description mechanism and its application in web-based learning. Expert Systems with Applications 37(12), 8639–8649 (2010)