Multimedia Tools and Applications (2024) 83:61261–61286
https://doi.org/10.1007/s11042-023-18051-5

Temporal automata for robotic scenario modeling with CIT framework

Damien Mondou[1]

Received: 16 March 2023 / Revised: 15 September 2023 / Accepted: 26 December 2023 /
Published online: 6 January 2024

Abstract
Many approaches for designing robotic applications have been proposed in recent years. If they are relevant to the problem they solve, most of them are platform- or language-dependent. To solve this problem, the use of formal models for scenario or activity generation is essential to ensure a high-quality experience. In this context, the use of Petri nets or automata is widespread. Based on this observation, we propose the formal CIT (content interaction time) model [37], which is dedicated to the development of interactive robotic applications and based on networks of input/output timed automata. In this paper, we propose the use of the CELTIC (Common Editor for Location Time Interaction and Content) and EDAIN (Execution Driver based on Artificial INtelligence) software platforms to allow for a simplified and generic modelling of robotic applications and their supervision based on the CIT model. We used this approach to design a serious game with the humanoid robot Nao. This game allows young people to playfully discover the ethnography exhibition at the Natural History Museum of La Rochelle, France, and the archaeology exhibition at the Sainte Croix Museum of Poitiers, France. After these experiments, we identified problems regarding dynamic adaptations of the scenario (robot listening time, speech recognition threshold...). The end of this article presents the beginning of our reflections on the integration of a reinforcement learning algorithm that optimises the parameters of a scenario's execution.

Keyword Formal modelling · Automaton · Robotics · Human-robot interaction

1 Introduction

Museums continually seek new ways to enhance the visitor experience. The use of serious gaming has demonstrated the potential to create engagement and promote players' understanding of a subject [22]. The benefits of game-based learning have been demonstrated [18, 48], and many institutions have equipped themselves with game-based features as a result [9, 12, 14, 54].

✉ Damien Mondou
damien.mondou@univ-lr.fr

1 La Rochelle University - L3i Laboratory, 23 avenue Albert Einstein, La Rochelle 17042, France

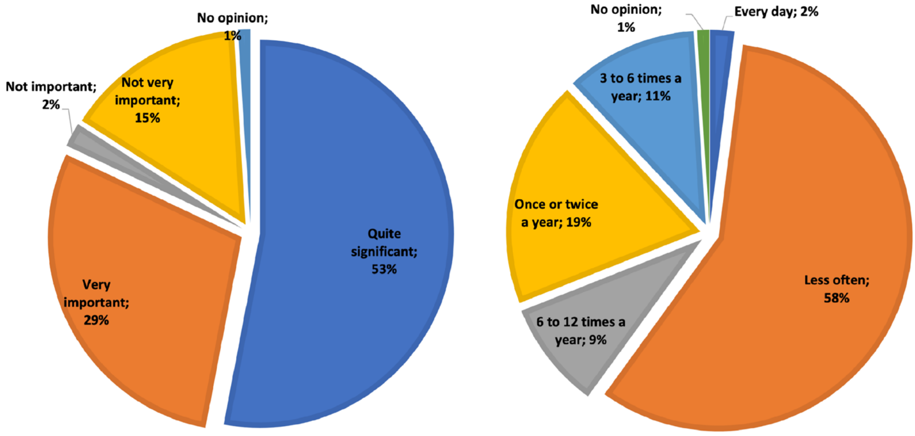

Fig. 1 Left: Importance of places of culture to young people. Right: Frequency of museum visits among young people

The use of serious games can therefore be a solution for young people, who may show a lack of interest in museums. The cultural practices of young people are far removed from those of their elders due to the widespread use of digital technologies that make it easier to access information and culture at any time of day, without any schedule or programming constraints. While 82% of people under 30 declare that culture is an important part of their lives [1](Fig. 1, left side), paradoxically, 77% of this group report going to museums only one to two times or less per year 1 (Fig. 1, right side). These figures are in line with those of the French Ministry of Culture [6] which as early as 2007 recorded a drop in museum and exhibition visits among young people aged 15 24 versus an increase in visits for those aged 25 34.

As such, places of culture, particularly museums, are increasingly taking advantage of new technologies to transmit their knowledge in a participative and playful rather than a unilateral and vertical way. Philippe Chantepie emphasis [41], "the instances of cultural transmission [...] are called upon to revisit their model of mediation to adapt it to the younger generations".

To this end, museums have introduced robots as an alternative to traditional information dissemination devices. The benefits of this new approach are manifold. Firstly, it allows for more natural interaction with visitors because it is easier to visit a museum without having to be equipped with a tablet, a phone or other devices that could interfere with the discovery of the works. Secondly, it facilitates the social dynamics between visitors, who can exchange with each other even if they are mere spectators of the interaction. Finally, the computer vision systems integrated into the robot allow it to detect the presence of visitors, and the robot can then call out to invite them to the game. Thus, even a visitor who has not taken the step of playing on a tablet or equipping themselves with a virtual reality device can become a player for part of their visit.

[1] According to the 2016 opinionway study for Agefa PME, carried out on 807 young people under 30

1.1 Technology in the service of museum mediation

Museums are therefore integrating new devices to attract young visitors. This section gives some examples of new technologies integrated in museums.

Interactive and personalied audio guides are widely used in museums so that visitors can obtain more information on what they are visiting. Additionally, immersive mediation equipment using augmented reality [3, 34], 3D [47] and holograms provide visitors with a rich and innovative experience (e.g., the British Museum or the Museum of Modern Art in New York).

From a decade, robots have integrated artistic works [2, 29, 53] and cultural venues as an alternative to digital devices for disseminating information or creating engagement with visitors. For example, since 2014, two robots have been in charge of welcoming visitors to the National Museum of Emerging Science and Innovation in Tokyo and can also give them the latest news. In the same vein, the Inkha robot (Fig. 2[b])) at King's College London is in charge of receiving and guiding visitors. Several experiments have been carried out to analyse the feasibility and impact of introducing robots in cultural venues.

During the Eppur Si Muove exhibition at the Musée d'Art Moderne in Luxembourg, a Nao robot coupled to a mobile base (Fig. 2[a])) is a work of art in itself but also guides the public by introducing them to the works on display [23]. In France, the Berenson robot (Fig. 2[c])) has been deployed at the Musée du quai Branly as part of a project on artificial aesthetics. This robot behaves like an art critic by expressing an emotion when it is in front of a work. Its opinion evolves in response to the reactions of the visitors around it [10].

Moreover, some museums are not accessible to people with reduced mobility because of their architecture. The Château d'Oiron Museum in France has recognised this problem and has therefore introduced the Norio robot (Fig. 2[d])) on the first floor of the museum. This robot can be remotely controlled via joystick from the ground floor, allowing people with disabilities to discover works otherwise inaccessible to them [25].

Animated conversational agents have also emerged in various applications. For example, Campano et al. [11] propose a virtual character who visits a museum to interact with the visitors, provide information and ask questions. Other applications also use this type of agent to, for example, rehabilitate brain-damaged patients [38] or recommend films [42].

Through our Museums 3.0 project and in collaboration with the Natural History Museum of La Rochelle and the Sainte Croix Museum of Poitiers, we have developed an interactive experience for young visitors through a serious game. This game allows them to discover

(a) (b) (c) (d)

Fig. 2 Experiments with robots in museums a part of the collections in an interesting and playful way through an oral quiz proposed by a Nao robot. The Nao robot, initially developed by the French company Aldebaran (now Softbank Robotics), has a set of sensors (notably a camera as well as sonar and tactile sensors) that enable it to perceive the environment in which it is moving. The robot is also able to interact verbally with humans through its microphones and speakers (the robot integrates speech recognition and synthesis APIs). The game consists of a treasure hunt directed by the robot, which asks the young players questions. The players then set off in search of the answers and return to Nao when they have found them.

1.2 Scientific issues

A first prototype of a serious game was developed in June 2016. Developed in collaboration with the Natural History Museum, the scenario consisted of a set of interactions aimed at leading visitors on a progressive journey through the museum. These interactions were produced using Aldebaran's proprietary tool Choregraphe. Although the results of these three days of experimentation were positive, the modelling method posed limits. The complexity of modifying the scheduling of actions and the modification of the robot's dialogues revealed the need for a higher-level scripting tool.

A: Issues related to scriptwriting

❶ The first reflections concerned the **extensibility** of such a game. Indeed, adding or modifying content proposed in the game, as well as deploying it to other museums, can be difficult. Having a modelling tool that simplifies the scenario's representation and allows for externalised management of content is therefore essential to facilitate game extension.

❷ The second objective of such scripting tool is to facilitate the **reproducibility** of the scenario for other places. The objective is to have a set of reusable elements at the time of future modelling in order to facilitate the work of the designer. This is therefore a question of having a modular vision of the design.

❸ Finally, **time management** in interactions, especially human-robot interactions, is important to guarantee a high-quality of experience. Time management is necessary for managing the dynamics of the execution by constraining the time allocated to the behaviours or expectations of the robot. The modelling platform must therefore be able to manage time and have *model-checking* and *time-checking* algorithms to verify the temporal coherence of the scenario.

These three problems lead us to question the management of contents (how to represent them, store them, adapt them, etc.), the interaction itself (how to represent an interaction between man and machine) and the time required to carry a single interaction or a set of interactions out.

B: Adaptation issues

❹ After designing and testing the scenario, an analysis phase of the experience is necessary to determine whether the session took place under suitable conditions. This analysis is possible thanks to the data collected during the execution of the scenario. The use of graph traversal algorithms is then necessary to determine typical and error paths. Based on these, the initial scenario must be optimised to reduce the number of paths leading to failure. In a second step, we are searching for an architecture that allows the scenario to be modified dynamically through relevance looping, by using data of previous sessions and analysis of the behaviour of the current user.

1.3 Organisation of the paper

The remainder of this article is organised as follows. Chapter 2 outlines the state of the art which regarding methods of modelling scenarisation and their supervision. We highlight the stakes and problems in scripting interactive experiences. Chapter 3 is dedicated to the presentation of the CIT model (content interaction time) and the CELTIC (Common Editor for Location Time Interaction and Content) platform, which is the editor and scenario generator we use to implement the CIT model. The supervision platform, EDAIN (Execution Driver based on Artificial INtelligence), is presented in Chapter 4. It allows for execution of the scenario designed through CELTIC. The proposed methodology has been the subject of several experiments in partner museums and is described in Chapter 5. These experiments have allowed us to reveal new problems linked to the adaptation of a scenario during its execution. Initial thoughts on this subject are detailed in the conclusion in Chapter 6.

2 State of the art

This article deals with the problem of designing and supervising interactive experiences through models to enable control of their execution. Indeed, many works deal with architectures thereby providing a high-quality experience for users while still allowing the designer to maintain control of the main stages of the narration that they wish to propose [13, 40, 45]. Improving players' experiences through interaction therefore comes down to granting players greater independence, or freedom of control over the course of the story. However, designers want to be able to control the unfolding of the narrative and, in particular, to be guaranteed that the player will go through specific narrative points in the order the designer decides. Thus, the challenge lies in proposing a model that allows a compromise between ensuring that the story experienced by the player is the one the designer wishes to tell and the player's desire to control the universe they are interacting with and to influence the story through their choices.

Beyond the issue of interaction in the context of robotics, past research propose models for interaction in the broad sense in other contexts as well, such as games. Issues related to interaction, content, time and space are common to the field of robotics. Indeed, a game designer may want a level of the game to be performed in a given time and place as it may be necessary to design a robotic scenario based on the speed and movements of the robot.

As McGanne points out [33], a monolithic approach is inappropriate when designing a complex robotic system. The need to divide the model is then felt. Reductionist (also known as component-based) approaches are relevant to solving this problem. Szyperski et al. [51] propose the following definition :

A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be developed independently and is subject to composition by third parties.

The major advantage of component-based approaches is reusability. The execution of the same component can occur at different stages of software in particular contexts. Furthermore, the division by components can have positive repercussions on production costs and time-to-market.

Our review of game and robotic platform modelling has therefore focused on component-based approaches and more specifically on two common types of modelling [27, 31]:

- semi-formal methods and in particularly UML (Unified Modelling Language) which has become a standard in the modelling of complex systems. Some approaches are based on UML for the design of robotic activities.
- formal methods, which use a mathematical model for the representation, analysis, verification and proof of certain properties. These aspects are becoming indispensable in the design of robotic activities and in the world of games.

2.1 Semi-formal methods - UML

Due to its standardisation, UML (Unified Modeling Language) has been used for the design of robotic activities. Its use is particularly notable on two platforms, *RobotML: Robotic Modeling Language* and *RobotChart*.

Within the framework of the French research project *PROTEUS* (Plate forme pour la Robotique Organisant les Transferts Entre Utilisateurs et Scientifiques), the RobotML modelling language has been designed to facilitate the design, simulation and deployment of robotic applications [17]. As some approaches are dependent on the robotic platform, the main objective of RobotML is to separate the model from the target robotic architecture. Thus, the same model can be deployed on several target platforms.

RobotML is based on UML profiles and state machines and is structured around four packages (trajectory management, environment observation, localisation and control). The creation phase is carried out via the dedicated Eclipse plugin.

Like RobotML, RoboChart [35, 36] uses semi-formal semantics via a UML representation (state machines). The temporal primitives used are based on the semantics proposed in the timed automata [1] and on the timed CSPs [49] (communicating sequential processes). The specification of a robotic system in RoboChart is based on a set of modules connected to one or more controllers, the semi-formal representation of wich is provided by a UML state machine. RoboChart has a modelling tool: *RoboTool*. *RoboTool* is a set of plugins for the Eclipse editor that allows the design and visualisation of the robotic model, both in textual and graphical form, and wich allows the model to be translated into CSP. Model verification is then possible using the FDR (failures divergences refinement) tool [19].

Global analysis of semi-formal approaches RobotChart and RobotML use state machines to model robotic platforms. The use of UML permits quick understanding of the model for designers. Moreover, the dissociation of the model from the robotic architecture is a major advantage. However, UML does not propose formal semantics to allow for the verification of properties, which limits its use in our case.

2.2 Formal methods

The use of formal methods is justified by the fact that, unlike semi-formal methods, formal methods use a mathematical model for the analysis, verification and proofing of certain properties. This analysis is necessary in order to guarantee a high-quality activity through fundamental properties. The same problems emerge in game design.

All the proposed approaches highlight the need for efficient tools, particularly to control certain safety properties. Indeed, humain-robot interaction can generate undesirable and potentially dangerous events. To prevent these, we focus on two particular categories of approaches:

- resource-oriented approaches based on Petri net;
- action-oriented approaches based on automata networks.

2.2.1 Approaches based on Petri nets

The notion of components, as described in the introduction, is relevant in the field of game design. It is perfectly integrated through certain concepts specific to the field. Unlike the modelling of processes in any software, the modelling of a game involves more specific activities, such as levels or quests. Constraints are also added because of the obligation to carry out certain tasks to achieve a goal. Thus, in the literature, the modelling of a game is divided into levels or quests, which can be compared to reductionist (or component-based) approaches to designing robotic activities.

For example, De Oliveira et al. [16] propose a game structure in the form of quests modelled by Workflow nets [55]. Each of the quests is obligatory or optional and represents a series of activities which can be carried out in a sequential or parallel way. These activities can also be optional. A linear logic [20] is then used to check the robustness of the modelled scenario. Barreto et al. [4] extend the previous approach by adding a topological modelling of the game space to the modelling of player activities by net workflow. This representation uses a Petri net whereby the squares represent the different regions where the action must take place and the token represents the position of the player. Additional squares can be added to model the conditions for moving from one game area to another (e.g., by obtaining a key). While this representation is effective, it cannot be used alone - a link between the representations of the player's activities and the game space is necessary and permits management of interactions between the player and the virtual universe. This double modelling also makes it possible to require players to be in a particular zone to carry out actions. The authors therefore propose a final model to parallel the formal modelling of player activities and the game space. This parallelism is implemented by a coloured Petri net using the concept of place mergers. The verification of robustness properties, notably the verification of the accessibility of each game area as well as the execution of all player activities, is implemented through the algorithms contained in CPN Tools [24]. However, apart from the management of resources, no other content is proposed. The levels and activities are also not timed.

Petri nets and their temporal extensions have been used for discrete event system modelling and particularly in the field of robotics [26, 30]. Control architectures based on multi-layer modelling and simplifying the design stage are widespread and take into account the planning, coordination and execution of tasks.

Thus, Costelha et al. [15] proposes a simplified model divided into three layers, each of which describes the elements at different levels of abstraction :

- **Environment**: each state of the environment is represented by a labelled Petri net;
- **Action Executor**: the set of actions is performed on the robot and modelled by a Petri net. An action has an impact on the environment and can be conditioned (pre-conditions, running-conditions and success-conditions);
- **Action coordinator**: the action coordinator plans the robot's actions through a higher-level Petri net.

The final model is then obtained by composing the Petri nets from the three previous layers. The time required for composition is proportional to the number of states in the different networks. However, the analysis time increases exponentially with the number of states. This analysis (the probability that a condition is respected, time necessary to carry out an action...) is carried out by various algorithms [5, 39, 56].

Global analysis of Petri net based approaches The presented approaches to game and robotic activities design indicate that Petri nets are suitable within a resource-oriented vision. They discussion has highlighted the effectiveness of reductionist approaches in which activities are designed independently of one another. In contrast to semi-formal approaches, tools such as CPN Tools or the linear logic conversion allow an analysis to be carried out on the models.

2.2.2 Approaches based on automata networks

After having presented works based on a Petri net representation, we are interested in this second part in action-oriented representations by the use of automata networks.

The use of automata is also common in the design of robotic platforms. For example, Wang et al. [58] propose a multi-layer planning system for robotics in which the behaviours of the robot and its environment are modelled by timed automata networks.

The logic layer is responsible for building the representation of the environment and the robot's behaviours using timed automata. The verification properties are expressed in CTL (computation tree logic) and are verified with the UPPAAL tool [21]. Two types of properties are verified:

- **reachability properties:** all robots must be able to reach their destination;
- **safety properties:** obstacle avoidance and runtime.

The verification of these properties by UPPAAL generates a trace that satisfies all the constraints. These traces are then used in the second layer, namely the physical layer, which generates a trajectory for each robot based on dynamic equations. A counterexample trace can also be generated when a constraint is not respected.

This work has been extended by Wang et al. [59]. Indeed, the previous proposed approach remained theoretical and had not been tested. A new four-layer approach has therefore emerged.

- the model level which can be compared to the logical layer of the previous approach;
- the verification level as a CTL query under UPPAAL;
- the code synthesis level. In this level, the generator automatically produces the C++ code for the ROS platform;
- the implementation level, wich allows the execution of the code on the Gazebo simulator.

The principle of multi-layer design has also been used in [44, 45]. To reduce the complexity linked to the modelling of the scenario, this approach proposes an automatic scenario design by defining a set of generic behaviours and execution scenes. The process of creating an interactive scenario is divided into three layers (Fig. 3).

A major issue in game design is the ease of extension or modification, whether at the level of the structure or the narrative entities. The three-layer structure of this approach addresses this issue. The concepts of extensibility and reuse of generic entities through multiple inheritance are a major asset. Although this approach has proven its efficiency, the scenario design requires long steps which can be a hindrance for the designer. Moreover, the temporal dimension is not considered, either at the level of behaviours or at the level of scene transitions. Finally, content management is not supported here.

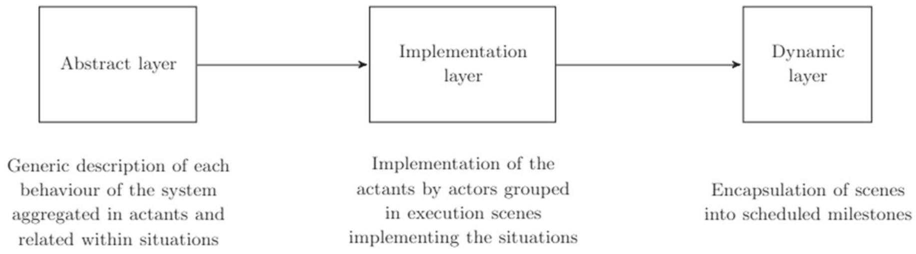

Fig. 3 Three-layer model of [44]

Global analysis of automata networks based approaches In the presented approaches, automata are adapted into the modular modelling of activities via the different mechanisms of composition and synchronisation. Automata also allow the application of model checking to prove their robustness. Finally, the possibility to temporally constrain the behaviours is a major asset given the importance of time in the field of human-robot interactions.

2.3 Positioning

Although the approaches introduced are effective, the models on which they are based do not address all the criteria set out in the research problem.

Existing approaches are mainly based on models specific to the issues addressed (planning, narrative, etc.) that permit the verification of properties. The state of the art indicates the need for efficient modelling tools that allow for a modular design of the system or scenario. Although UML modelling facilitates the understanding and reading of a system, the verification of certain properties such as safety or accessibility is sometimes difficult or even impossible. Approaches based on Petri nets are resource-oriented. Our scientific problem requires a representation of robotic behaviour by actions. We therefore opt for the use of automata networks.

Formally designing a robotic activity guarantees that the designer obtains a model that meets his expectations. Using mathematics, a formal representation of the activity can also ensure that the model meets safety criteria through model-checking techniques. It is then possible to pilot any process, including a robotics platform.

3 CELTIC: Interactive experience modelling editor

This chapter proposes a generic model for all the dimensions of the interaction, including representation of the contents, the interaction itself (through the behaviours) and the time. A first model, CITE, as proposed by Prigent et al. [43] aimed to also consider space in scenario conception. This problem constitutes a potential future integration for our model, but it does not fall within the scope of the work presented in this paper.

CELTIC is a generic model production editor for interactive experiences based on the CIT model [37]. It can be a game, an application or - as in our case study - a game experience with a robot. The objective of this study is to propose a generic and modular model that allows

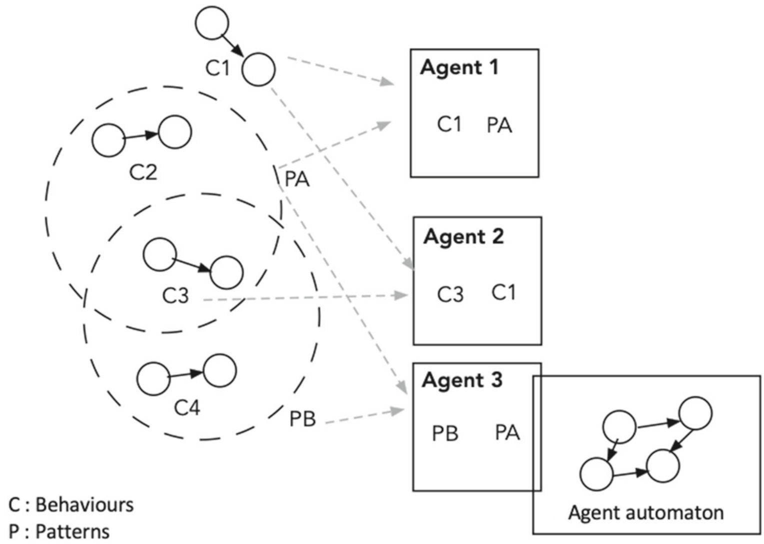

Fig. 4 Principle of behaviours, patterns and agents creation

supervision and an analysis of the users' traces. This model also aims to consider the time and content components of the interaction.

3.1 Principles of the CIT model

Our approach is also based on timed automata networks, which are very suitable for representing synchronous systems with time constraints. We simplify the modelling task into a two-step method:

- Firstly, the description of the experiment is done in an abstract way, by defining the reusable entities (behaviours and patterns) modelled by timed automata. This step creates the *declarative layer*.
- Secondly, the *implementation layer* is defined by instantiating generic entities in agents, which are grouped into ordered execution contexts.

The process of creating an interactive system is illustrated in Figs. 4 and 5 and is detailed in Sections 3.2 and 3.3.

3.2 Declarative layer

The declarative layer is where we define the generic entities of the modelled system, namely the atomic behaviours that can be executed. These behaviours are associated with declarative variables (integers, strings or booleans) and messages (channels), which are the signals that allow several behaviours to be synchronised for dynamic composition. The behaviours are then grouped into patterns which represent a set of behaviours that are specific to the entity and which can be reused according to the principle of multiple inheritance.

The major challenge of this declarative layer is to allow the representation of atomic behaviours and behaviour patterns. These can be reused and composed dynamically within agents in the implementation layer.

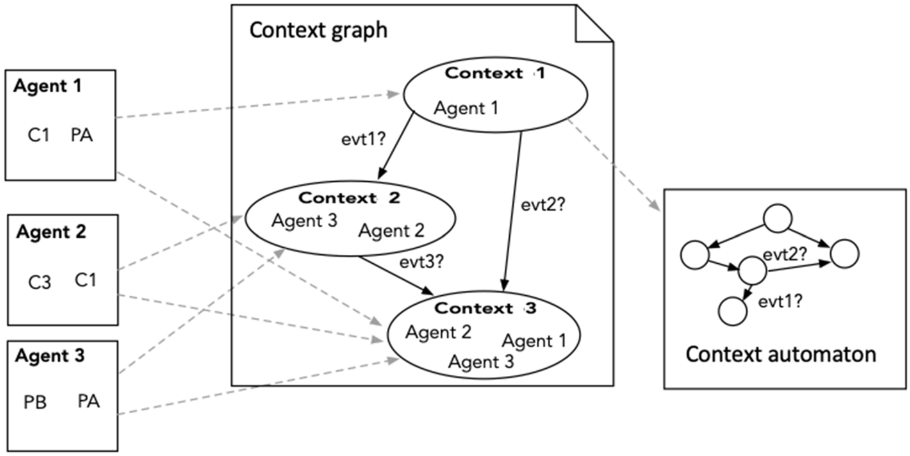

Fig. 5 Principle of context and context graph creation

3.2.1 Behaviours

In our approach, an atomic behaviour is a timed automaton [21] with one transition which will aggregate with all other possible behaviours of the agent. For example, we wish to implement a first behaviour in wich the robot stands up (in a time between 2 and 5 time units) and a second in wich it walks for a duration of 3 to 10 time units. The models of these behaviours are represented in Fig. 6. In this example, x represents time.

3.2.2 Patterns

Patterns describe a set of behaviours that can be executed jointly within the application and which are implemented by agents. A timed automaton of a pattern is obtained through the synchronised product of the set of automata which compose it (according to the synchronisation algorithm described in [7]). This composition allows the system to evolve through the

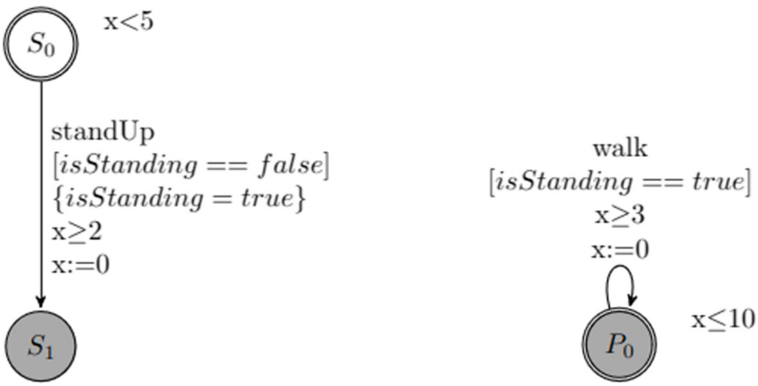

Fig. 6 Timed automaton of behaviours *standUp* and *walk*

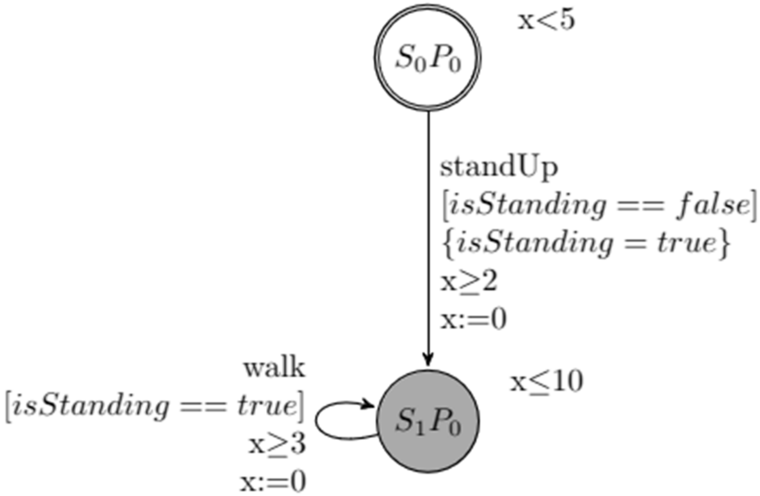

Fig. 7 Timed automaton of the pattern *stroll*

synchronisation of two entities or by their individual evolution. An example of an automaton obtained from the previously defined behaviours is given in Fig. 7 (pattern *Stroll*).

3.3 Implementation layer

The implementation layer uses the described behaviours and patterns to design the entities involved in the interactive experience, namely the agents and the execution contexts. The first step is to build agents that implement behaviours and patterns. These agents are then linked in execution contexts representing a specific situation in which the agents interact with each other.

3.3.1 Agents

An agent can implement the role of one or more patterns thanks to the multiple inheritance mechanism. It can also implement atomic behaviours defined in the declarative layer. Thus, the designer can specialise an agent via the different behaviours it implements. The agent's behaviour automaton is obtained by synchronising all the behaviours. The major challenge of this layer is to guarantee the reusability of behaviours. An agent specific to a given interactive experience can use atomic behaviours designed for another experience.

The behaviours aggregated by an agent propose default values when they are defined in the declarative layer. However, these may not correspond to the designer's expectations for a particular agent. In this case, the designer can modify the values. Finally, the agent makes it possible to specify the actual content that will be executed on the controlled process for each of the aggregated behaviours.

Specification of behaviours
The specification of behaviours is achieved by using an external database (Fig. 8) integrating all the executable actions in the controlled process as well as the various contents, especially the textual contents.

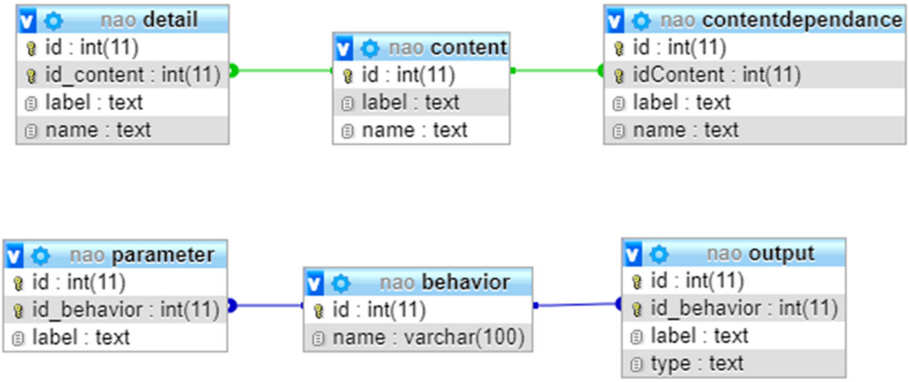

Fig. 8 Contents database

Example We define an agent `walker` implementing the pattern `walk` and the atomic behaviour `sitDown`, which will trigger the `sit` signal (Fig. 9). The designer must therefore specify the three behaviours that the agent implements. To do this, the designer uses the database and in particular the `Behavior` table containing all the actions that can be executed in the process. For each behaviour, the table specifies the action to be executed. In order to be executed, some actions have parameters (a dictionary and a listening time for speech recognition, for example) and outputs (the recognised word and a confidence rate for speech recognition, for example). These parameters must also be specified by the designer. Finally, if the designer wants to use content, such as textual content when using speech synthesis, the designer uses the `Content` table (Fig. 8).

3.3.2 Contexts and context graph

The context describes a situation in which a number of agents will be involved. The context automaton is constructed by synchronising all the automata of the agents present. The global execution scenario is produced using the context graph. This is a high-level timed automaton that represents the passages between execution contexts (an example is given in Fig. 10).

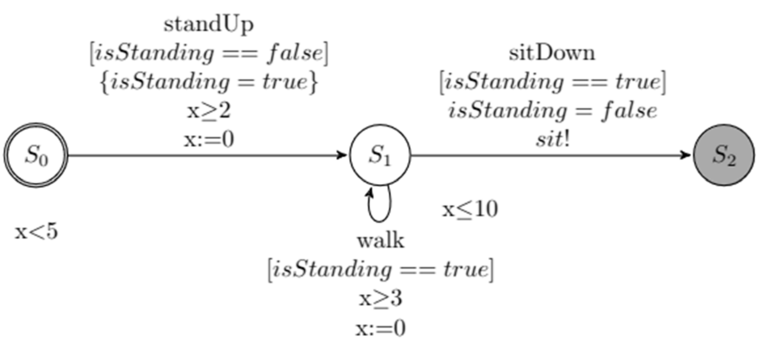

Fig. 9 Timed automaton of the agent `walker`

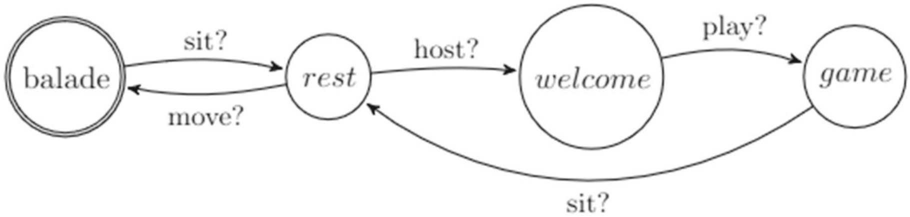

Fig. 10 Context graph

This implementation layer first aggregates the behaviours in the agents and then the agents in the contexts, and it dynamically produces the behaviour automata of the latter via synchronisation.

3.4 Automaton generator

Based on the model described by the editor and serialised in XML format, the CELTIC platform's automaton generator is responsible for generating the communicating timed automaton for each entity in the system. These automata are then combined to meet the designer's specifications thanks to the various composition algorithms it integrates. Once each entity has been generated, the generator serialises the scenario again in XML format and produces the supervision file. This file integrates all the communication channels, the variables, the timed automaton of each context and the high-level context graph.

4 EDAIN: The supervisory architecture

Supervision of a task in robotics consists of making the right decisions according to the different events that may occur. When robots interact with a human, the tasks become collaborative and require consideration of certain specificities. In fact, the successful execution of the task is no longer due only to the robot but also to the realisation of the activity expected from the human. For example, during a vocal exchange between a robot and human, the task can only be successful both if the robot is listening and the human decides to interact with the robot by talking to it. It is therefore necessary to account for all possible outcomes to supervise a robotic activity. In this section, we detail how the activity is supervised and controlled in our approach using the automata-based model.

EDAIN is the supervision tool associated with the CIT model. It allows a connection to be established in the Nao robot. The communication mechanism that allows dialogue between the robot and EDAIN is based on a client-server communication. A server developed in Python has been embedded in the robot. A JAVA client is in charge of establishing the connection from the EDAIN tool.

After having filled in the addresses of the database and the robot and compiling a file to save all the logs, the EDAIN supervisor interface prompts the user to upload the supervision file. The different automata are then represented graphically, and the tool loads the initial context . All the variables and clocks are then displayed as well as the automaton of the current context. Next, EDAIN chooses, among the possible transitions, the transition which meets the time and variable constraints.

When an action must be executed by the robot, EDAIN retrieves the name of the module to be executed from the database and possibly the values of the contents to be proposed to the user, and it connects to the robot's server to transmit the control parameters. The exchange of information takes place through a socket and uses the JSon format. For example, when the voice recognition module is executed, EDAIN sends the following information to the robot:

{"speechReco":{"parameters":{"time":"15",
"dictionnary":"yes,no"}}}

On the robot's side, upon receiving this information, the system will execute the script *speechReco.py* by passing it through the associated parameters. After executing this script, the robot will send back to EDAIN the recognised word as well as an associated confidence rate in the same format:

{"out":{"wordRecognized":"yes",
"confidence":"0.3216"}}

Using the information received, EDAIN updates the state vector of the model variables and can thus choose the next transition to execute. When context synchronisation is performed, the target context is loaded on the interface, and the same process is reproduced.

5 Public experiments and analyses of typical situations

In this section, we present the different experiments that we carried out at the Natural History Museum of La Rochelle and at the Sainte-Croix Museum of Poitiers. These experiments allowed us to test the proposed modelling method and to verify the genericity of the representation of an interactive experience, the reusability of the elements of the declarative layer from one experience to another, and the ease with which the contents are unbundled from the model (through the database-based architecture presented in Fig. 8, Chapter 3). Another validation concerns the accuracy of our dynamic behaviour composition algorithm for building agents and system behaviours within contexts and the quality of the supervision process for executing the context graph.

The first experiment took place during the Science Festival at the Natural History Museum of La Rochelle. The scenario created for the occasion, in collaboration with the museum's teams, aimed to encourage people to discover some of the exhibited works through quizzes proposed by Nao robots (Fig. 11).

The second experiment took place during the Gamers Assembly at the Sainte Croix Museum of Poitiers. On this occasion, we proposed a new interactive scenario involving Nao robots and games developed on tablets.

5.1 Natural history museum of la rochelle

The first experiment took place over three afternoons during the Science Festival at the Natural History Museum of La Rochelle. The objective was to present three works of art to the public in the form of a quiz delivered orally through the robot. Each question invited the player to search the museum for the answer and return to the robot. Players could interact with Nao at any time simply by visually recognising the badge they were wearing. Nao would then continue the game sequence by interrogating its internal memory.

We proposed two experiments based on the same model (only the textual contents in the database were modified):

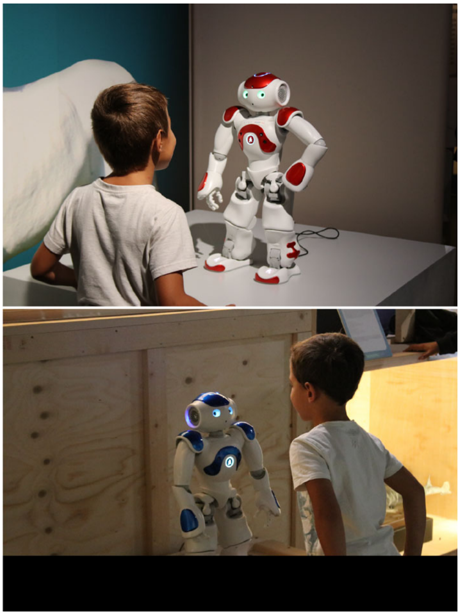

Fig. 11 Experiments at the Natural History Museum of La Rochelle

- The first was intended for children and allowed them to discover the exhibition devoted to monsters, imaginary or real;
- the second was intended for teenagers and adults and prompt them to discover the permanent exhibition devoted to archaeology.

5.1.1 Presentation of the scenario

The experiment began by putting Nao on standby. When a player wanted to interact with him, a touch on the sensor located on Nao's head woke him up. Nao is able to detect the presence of people thanks to its cameras, but we did not want to use this feature because of the large number of visitors that could pass in front of the robot. After triggering the execution of the scenario, the visitor was invited to present their badge to Nao to identify themselves. This procedure had to be repeated in each phase of the game. At the beginning of the game, Nao did not know the player and therefore presented the game to him. Nao then recorded the badge identifier and the identifier of the question he had to answer (in this case, the first one). When Nao recognised the player, it repeated the quiz question and waited for the answer (an example of content is given in Table 1). A hint was given to the player in the case of a

Table 1 Example of content proposed during the experiment at the Museum of La Rochelle

Question	In the wooden boxes, find a large object that could have belonged to a unicorn. In reality, which animal does it belong to?
Expected response	Narval
Explanation of the artwork by Nao	Its large fingers allow it to search for insects that burrow into the wood of trees.

wrong answer. If the question answered correctly, details of the work were offered and the next question was asked. At the end of the quiz, the player was given their total time they took to complete the quiz and provisional ranking.

5.1.2 Limitations of the experiment

During these three days of experimentation, we encountered problems both on the technical level and in the design of the scenario itself. Due to the configuration of the museum, it was impossible to isolate Nao in a room. This disadvantage resulted in difficulties with voice recognition. Indeed, Nao was placed in the middle of the works of art in a location with high ambient noise. The quality of the interaction between the player and the robot was thus negatively impacted. Moreover, the execution of the scenario is dependent on certain parameters controlling the activity. The valuation of these parameters (time or speech recognition thresholds) is done by the designer and may, in some cases, not be adapted to the environment or the player.

Our supervision tool **EDAIN**, presented in Chapter 4, integrates a system for observing the behaviour of the user and the scenario in order to generate execution traces. The logs obtained during these experiments reveal the problems presented previously. Each event produced by the supervision tool was saved in a flat file along with the feedback from Nao. For the experiment carried out at the Museum of La Rochelle, we obtained 6,500 lines of logs.

These logs are then used to construct the timed automaton representing each player's game, a posteriori. Without being expert, the automaton shown in Fig. 12 shows a large number of loops, which suggests that something has gone wrong. The ambient noise in the room caused the robot to make speech recognition errors.

- Presentation of the game & Question 1 asked
- Answer to the first question & Question 2 asked

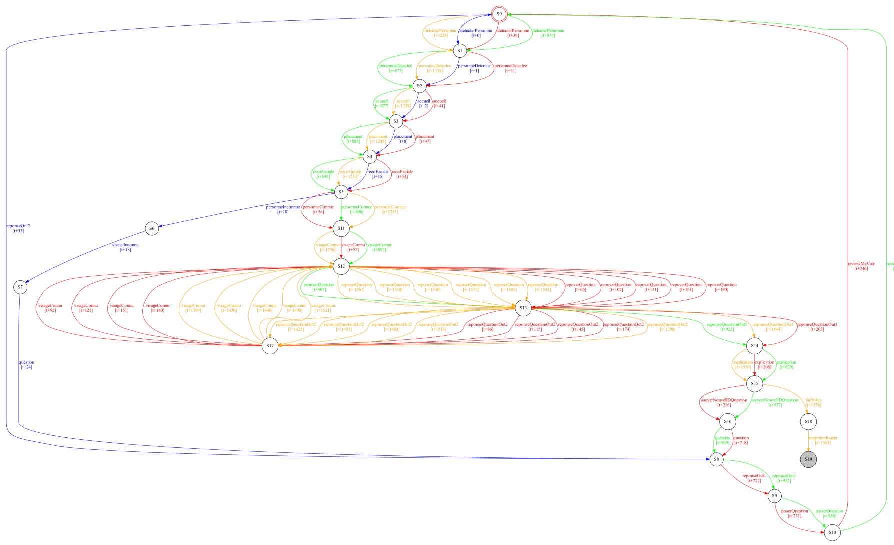

Fig. 12 Graph of a player who had voice recognition problems with Nao

Table 2 Ground truth of the experiment carried out at the Museum of La Rochelle

Game finished	Speech recognition problem	Mis-identification	Abandonment
18	13	3	25

- Answer to the second question & Question 3 asked
- Answer to the third question & end of the game

5.1.3 Analysis

A victim of its own success, the Nao robot aroused great interest, causing unwanted use cases. The details are given in Table 2.

Although the first afternoon, dedicated to a class of schoolchildren, went well (7 groups of 2 or 3 students), the game phases open to the general public did not result in an optimal game experience for the players. Indeed, 36 players played with Nao on the last two afternoons, but only 42% of them could finish their game. During the different games, the graphs of each player indicate that many voice recognition errors occurred. This high number may have led some players to give up during the game. It is also possible that the number of simultaneous players increased the waiting time to play with Nao, and some parents did not want to stay longer to finish the game. This phenomenon was particularly noted during the Sunday sessions. The voice recognition errors are due in part to the ambient noise in the museum and the acoustics of the room. The recognition threshold was 40% for the Friday and Saturday experiments, but this threshold was lowered to 30% for the Sunday sessions and did not significantly increase the quality of the game.

5.2 Sainte croix museum of poitiers

The second experiment took place at the Sainte Croix Museum of Poitiers during the *offs* of the Gamers Assembly Festival. In this context, we organised a game session to allow the public to discover the museum in a playful way.

The game took place in four stages, as shown in Fig. 13.

5.2.1 Visite patrimoine

The first part of the game took place outside the museum using the application *Visite patrimoine Poitiers*. This application, developed by students from La Rochelle University, allows users to discover some of the city's monuments. For our experiment, a personalised version of the application was developed.

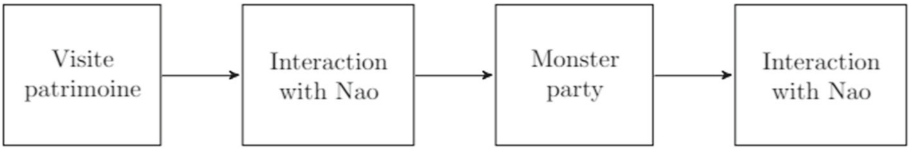

Fig. 13 Steps in the game at the Sainte Croix Museum of Poitiers

The players followed a route outside the museum to discover monuments such as Saint Peter's Cathedral, the Baptistery and the exterior of the museum. At each stage of the tour, they were given a riddle, the answer to which served as a clue. At the end of the outdoor tour, the players had to reformulate a code word from the various clues they had collected to move on to the second stage of the game. Once the code word was obtained, the players headed for the museum, where Nao was waiting for them.

5.2.2 Launch of the game: First interactions with Nao

Nao was responsible for welcoming the players to the museum. The objective of these first interactions was to check the players' code word and to explain the second part of the game. The player identification mechanism was identical to the one used during the experiments at the Natural History Museum of La Rochelle. When no player was present in front of Nao, Nao went into standby and sat down. The first action to perform was to press on its head to wake him up and allow him to welcome the visitor. Nao then asked the player to present their badge. It was possible to determine which player was interacting with Nao and which stage of the game they were at as each badge had a unique *NaoMark*.

In contrast to the experiment carried out at the La Rochelle Museum, a flow management was set up to prevent too many players from reducing the quality of the experience. As part of this, prior registration with the town hall was necessary. On the basis of the list of players thus obtained, we added an interaction during which Nao asked for the player's first name. The dictionary of first names allowed a symbolic reference to the first name with the highest confidence rate.

At the end of this interaction, the badge identifier/first name pair was saved in Nao's memory. This information was used in the rest of the scenario.

After this backup, Nao checked the code word proposed by the player. If the code word was incorrect, the robot gave a new clue until it got the right answer. When the answer was correct, the second part of the game was explained to the player. The *Monster Party* game could then begin.

5.2.3 Monster Party

The game `Monster Party` aimed to make people discover the floor dedicated to archaeology. To this end, the game proposed to collect weapons to capture monsters associated with works of art. Via augmented reality technology, players had to find three weapons and capture three monsters. For each monster captured, a clue was offered to determine the final code word. At each event produced in the game (discovery of a tool, capture of a monster or recovery of a clue), a save was made on a server.

As part of an other research project on the analysis of players' movements [8], each badge worn by the players recorded their journey through the museum using iBeacons technology. Raspberry computers were deployed on the floor where the game took place, and the iBeacons connected to these when they were nearby. In this way, a timestamp and the Raspberry's ID were recorded on a server to identify the player's journey. At the end of the game, the player was invited to go in front of Nao to check the final code word they obtained through the different clues they collected.

Table 3 How would you describe this experience?

	Simple	Boring	Original	Positive
Totally agree	9%	0	82%	64%
Agree	28%	0	9%	36%
Not agree	45%	18%	0	0
Strongly disagree	0	82%	0	0
No opinion	18%	0	9%	0
Total	100%	100%	100%	100%

5.2.4 End of the game: Last interactions with Nao

To finish the game, the player had to return to Nao to give him the final code word. After identifying themselves with their badge, Nao retrieved all the clues the player obtained via their login from the server. These clues were given back to the player. The player then proposed the final code word to the robot.

If the answer was incorrect, the robot gave an additional clue until the player offered the right answer. Finally, when the correct answer was given, Nao invited the players to go to the reception desk to receive a prize.

5.3 Results

To overcome the difficulties encountered with the large flow of players during the experiment at the Museum of La Rochelle, registrations were set up by the reception of the city hall of

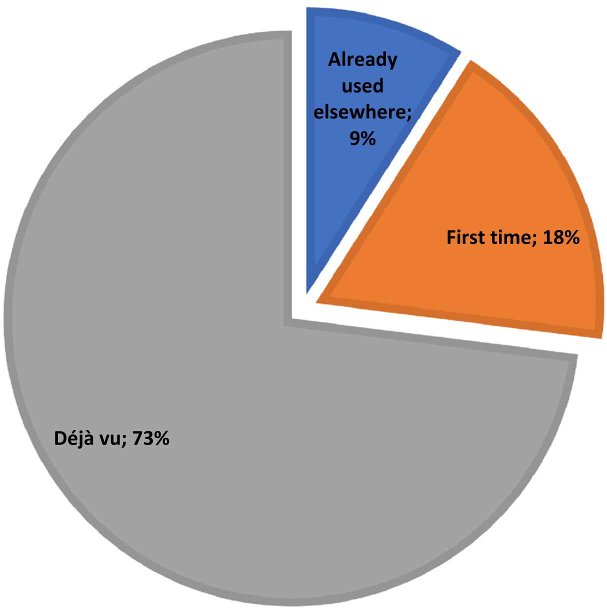

Fig. 14 Did you know Nao before the game started?

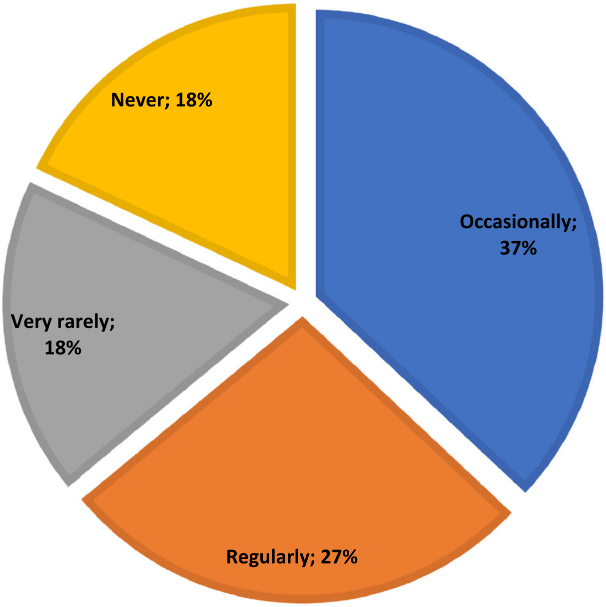

Fig. 15 Have you ever been to the Holy Cross Museum?

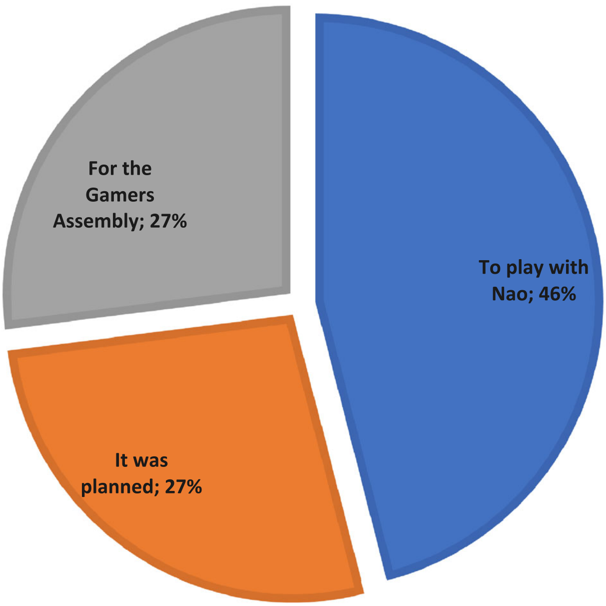

Fig. 16 Why did you come?

Poitiers. Thus, we had control over the flow of players wanting to interact with Nao, who was isolated from the rest of the exhibition.

In total, we counted 13 players or teams. Although the number of participants was low, this experiment allowed us to extend and propose a more successful serious game than the one at the Museum of La Rochelle. Indeed, in this experiment, we use three games, on tablets and on the Nao robot. Unlike the experiment in La Rochelle, no player abandoned the game. Voice recognition errors persisted, but their number was reduced compared to the La Rochelle experiment. Isolating Nao and controlling the flow of players limited the ambient noise, thus improving interaction quality. A questionnaire evaluating the quality of the game was delivered to each player at the end of their game. The results are presented in Tables 3 and Figs. 14, 15 and 16.

This experience was considered unique and positive by a large majority of players, but it remained difficult to grasp for the youngest players, especially with the augmented reality game on the tablet. Nonetheless, these results indicate that players were satisfied with their experience at the Sainte Croix Museum. More than a third of the players were not used to visiting museums, and almost half came to play with Nao. Although these results only concern 13 players or teams, they nevertheless imply that the use of new technologies can encourage young people to discover the artifacts on display in cultural venues.

6 Conclusion

This article presents our model for designing interactive experiences and the supervision mechanics to drive the system. We conducted experiments with the public to validate our modelling and supervision approach.

These experiments have underlined how important it is to consider time to obtain a quality interaction. In particular, even more than in classical computer systems, the parameterisation of the robot's speaking time and the time during which it listens to a verbalised response from the audience is essential. For example, we observed that the speed of a user's response is most often dependent on their age. A child will respond very quickly (sometimes even before the robot's recording sequence is triggered), whereas an older person takes more time before verbalising their response (and often after the recording sequence is over). Moreover, the choice of the robot's waiting time before reacting to the presence of a visitor and its speed of speech or movement must be parameterised with care because these elements have a strong impact on the public's perception [46]. With this in mind, we have integrated temporal constraints for the behaviour of agents and for the management of transitions between contexts into the framework.

Beyond this, we are beginning adaptation work by using user traces to determine the optimal settings for certain parameters by reinforcement learning and thus to set up a dynamic refinement of the model via a relevance loop.

Adapting the behaviour of a system during its execution is a difficult task that has been widely developed in recent years [28, 32]. The main difficulty is due to the lack of precise methods for the description and modelling of systems. In the field of robotics, many approaches use reinforcement learning for this problem, particularly for assistant and domestic robots [52, 57].

The objective of reinforcement learning is to find, by trial and error, the optimal action to perform. Each action of the system is associated with a reward (positive, negative or null).

This reward is used to evaluate the quality of a particular action. The objective for an agent is to maximise the accumulated reward during the execution of the system.

We are especially interested in time difference methods [50], which are the most widely used methods in robotics. They combine two advantages:

- learning from experiments without prior knowledge of the environment (Monte Carlo methods);
- use estimates of successor states to estimate the value of a state (dynamic programming).

The main method is `Q-Learning` [60]. We wish to use this method on the traces collected during the experiment carried out at the Natural History Museum of La Rochelle. Our interest in using this dataset lies mainly in the fact that it contains many examples of error cases that could be avoided by using our reinforcement technique. Based on these logs, we will train our model to determine the different Q-values associated with each pair (locality, transition). We will then use the valuation obtained for each pair (locality, transition) to dynamically adapt the parameters controlling the execution of the scenario.

Funding The author declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent to publish The author affirm that human research participants provided informed consent for publication of the images in Fig. 11.

Competing Interests The author have no relevant financial or non-financial interests to disclose.

References

1. Alur R, Dill D (1992) The theory of timed automata, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 45–73. https://doi.org/10.1007/BFb0031987
2. Aspord E, Becker J, Grangier E (2014) Link human/robot. Van Dieren eds
3. Banterle F, Cardillo FA, Malomo L, et al (2015) Leccear: An augmented reality app. In: Digital Presentation and Preservation of Cultural and Scientific Heritage – DiPP2015. Institute of Mathematics and Informatics - Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Block 8 Sofia (1113), Bulgaria, pp 99–108, http://vcg.isti.cnr.it/Publications/2015/BCMPGA15
4. Barreto FM, Julia S (2014) Modeling and analysis of video games based on workflow nets and state graphs. In: Proceedings of 24th annual international conference on computer science and software engineering. IBM Corp., Riverton, NJ, USA, CASCON '14, pp 106–119
5. Bause F, Kritzinger P (2013) Stochastic Petri Nets -An Introduction to the Theory
6. Pj Benghozi (2007) Aproche générationnelle des pratiques culturelles et médiatiques. Culture Prospective 3:1–32. https://doi.org/10.3917/culp.073.0001
7. Bengtsson J, Yi W (2004) Timed automata: Semantics, algorithms and tools. In: Lectures on concurrency and petri nets. Springer Berlin Heidelberg
8. Bernard G, Faucher C, Bertet K (2018) Towards reconstruction of human trajectories in indoor environments. In: Proceedings of the EKAW 2018 posters and demonstrations session co-located with 21st international conference on knowledge engineering and knowledge management (EKAW 2018), Nancy, France, pp 37–40

9. Bolognesi C, Aiello D (2020) Learning through serious games: A digital design museum for education. ISPRS - International Archives of the Photogrammetry. Remote Sens Spat Inf Sci, XLIII-B5-2020:83–90. https://doi.org/10.5194/isprs-archives-XLIII-B5-2020-83-2020
10. Boucenna S, Gaussier P, Andry P et al (2014) A robot learns the facial expressions recognition and face/non-face discrimination through an imitation game. Int J Soc Robot 6(4):633–652. https://doi.org/10.1007/s12369-014-0245-z
11. Campano S, Clavel C, Pelachaud C (2015) I like this painting too: when an ECA shares appreciations to engage users. In: 14th International conference on autonomous agents and multiagent systems AAMAS'15, Istanbul, Turkey, https://hal.telecom-paris.fr/hal-02412126
12. Carrozzino M, Bergamasco M (2010) Beyond virtual museums: Experiencing immersive virtual reality in real museums. J Cult Herit 11(4):452–458. https://doi.org/10.1016/j.culher.2010.04.001
13. Champagnat R, Estraillier P, Prigent A (2006) Adaptative execution of game: Unfolding a correct story. In: International Conference on Advances in Computer Entertainment Technology. ACM, New York, ACE 06. https://doi.org/10.1145/1178823.1178941
14. Coenen T, Mostmans L, Naessens K (2013) Museus: Case study of a pervasive cultural heritage serious game. J Comput Cult Herit 6(2):8:1–8:19. https://doi.org/10.1145/2460376.2460379
15. Costelha H, Lima P (2012) Robot task plan representation by petri nets: modelling, identification, analysis and execution. Auton Robot 33(4):337–360
16. De Oliveira GW, Julia S, Soares Passos LM (2011) Game modeling using workflow nets. In: 2011 IEEE International conference on systems, man, and cybernetics, pp 838–843. https://doi.org/10.1109/ICSMC.2011.6083757
17. Dhouib S, Kchir S, Stinckwich S et al (2012) Robotml, a domain-specific language to design, simulate and deploy robotic applications. In: Noda I, Ando N, Brugali D et al (eds) Simulation, Modeling, and Programming for Autonomous Robots. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 149–160
18. Dietze B, Dietze D, Kashin D (2011) Playing and Learning in Early Childhood Education. Pearson Education Canada
19. Gibson-Robinson T, Armstrong P, Boulgakov A et al (2014) Fdr3 - a modern refinement checker for csp. In: Ábrahám E, Havelund K (eds) Tools and Algorithms for the Construction and Analysis of Systems. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 187–201
20. Girard JY (1987) Linear logic. Theoret Comput Sci 50(1):1–101. https://doi.org/10.1016/0304-3975(87)90045-4
21. Guldstrand Larsen K, Pettersson P, Yi W (1997) Uppaal in a nutshell. Int J Softw Tools Technol Transfer 1:134–152
22. Hamari J, Shernoff DJ, Rowe E et al (2016) Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning. Comput Hum Behav 54:170–179
23. Henaff P (2015) Entre art et science: Guido, un robot guide espiégle au musée d'art moderne de luxembourg. session vidéo, Journées Nationales de la Recherche en Robotique (JNRR)
24. Jensen K (1998) An introduction to the practical use of coloured Petri Nets, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 237–292. https://doi.org/10.1007/3-540-65307-4_50
25. Khlat M (2014) Norio, the robot guide of the oiron castle, published on Tourmag. http://www.tourmag.com/Norio-the-robot-guide-of-the-Oiron-Castle_a71190.html
26. Kobayashi K, Nakatani A, Takahashi H, et al (2002) Motion planning for humanoid robots using timed petri net and modular state net. In: IEEE International conference on systems, man and cybernetics, vol 6, pp 6. https://doi.org/10.1109/ICSMC.2002.1175609
27. Kortenkamp D, Simmons R, Brugali D (2016) Robotic Systems Architectures and Programming, Springer International Publishing, Cham, pp 283–306. https://doi.org/10.1007/978-3-319-32552-1_12
28. Krupitzer C, Roth FM, VanSyckel S et al (2015) A survey on engineering approaches for self-adaptive systems. Pervasive Mob Comput 17:184–206. https://doi.org/10.1016/j.pmcj.2014.09.009 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian
29. Lapeyre M, Rouanet P, Oudeyer PY (2013) 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
30. Lehmann A, Mikut R, Asfour T (2006) Petri nets for task supervision in humanoid robots. In: in Proc., 37th International symposium on robotics (ISR 2006, pp 71–73
31. Luckcuck M, Farrell M, Dennis LA, et al (2018) Formal specification and verification of autonomous robotic systems: A survey. arXiv:1807.00048
32. Martins GS, Santos L, Dias J (2019) User-adaptive interaction in social robots: A survey focusing on non-physical interaction. Int J Soc Robot 11(1):185–205. https://doi.org/10.1007/s12369-018-0485-4
33. Mcgann C, Rajan K, Olaya AG (2009) Integrated planning and execution for robotic exploration. In: in International workshop on hybrid control of autonomous systems

34. Miyashita T, Meier P, Tachikawa T et al (2008) An augmented reality museum guide. Proceedings - 7th IEEE International Symposium on Mixed and Augmented Reality 2008. ISMAR 2008:103–106. https://doi.org/10.1109/ISMAR.2008.4637334
35. Miyazawa A, Ribeiro P, Li W, et al (2016) Robochart: a state-machine notation for modelling and verification of mobile and autonomous robots. Tech Rep
36. Miyazawa A, Ribeiro P, Li W et al (2019) Robochart: modelling and verification of the functional behaviour of robotic applications. Softw & Syst Model. https://doi.org/10.1007/s10270-018-00710-z
37. Mondou D, Prigent A, Revel A (2018) A dynamic scenario by remote supervision: A serious game in the museum with a nao robot. In: Advances in Computer Entertainment Technology. Springer International Publishing, p 103–116. https://doi.org/10.1007/978-3-319-76270-8_8
38. Morales-Rodríguez ML (2007) Modèle d'interaction sociale pour des agents conversationnels animés : Application à la rééducation de patients cérébro-lésés. Theses, Université Paul Sabatier - Toulouse III. https://tel.archives-ouvertes.fr/tel-00660114
39. Murata T (1989) Petri nets: Properties, analysis and applications. Proc IEEE 77(4):541–580. https://doi.org/10.1109/5.24143
40. Natkin S, Vega L (2003) A Petri Net Model for the Analysis of The Ordering of Actions in Computer Games. In: GAME ON 2003, France, london, October 2003
41. Octobre S (2009) Pratiques culturelles chez les jeunes et institutions de transmission: un choc de cultures? Culture prospective 1(1):1. https://doi.org/10.3917/culp.091.0001
42. Pecune F, Murali S, Tsai V, et al (2019) A model of social explanations for a conversational movie recommendation system. In: Proceedings of the 7th International Conference on Human-Agent Interaction. Association for Computing Machinery, New York, NY, USA, HAI '19, p 135–143. https://doi.org/10.1145/3349537.3351899
43. Prigent A, Revel A (2017) Cite – content interaction time and space: a hybrid approach to model man-robot interaction for deployment in museums. EAI Endorsed Transactions on Creative Technologies 4(13). https://doi.org/10.4108/eai.8-11-2017.153336
44. Rempulski N (2013) Synthése dynamique de superviseur pour l'xécution adaptative d'applications interactives. PhD thesis, Université de La Rochelle
45. Rempulski N, Prigent A, Courboulay V et al (2009) Adaptive Storytelling Based On Model-Checking Approaches. International Journal of Intelligent Games & Simulation (IJIGS) 5(2):33–42
46. Revel A, Andry P (2009) Emergence of structured interactions: from a theoretical model to pragmatic robotics. Neural Networks 22(2):116–125. https://hal.archives-ouvertes.fr/hal-00417847
47. Ridel B, Reuter P, Laviole J et al (2014) The revealing flashlight: Interactive spatial augmented reality for detail exploration of cultural heritage artifacts. Journal on Computing and Cultural Heritage 7:1–18. https://doi.org/10.1145/2611376
48. Samuelsson IP, Fleer M (2008) Play and learning in early childhood settings: International perspectives, vol 1 Springer Science & Business Media
49. Schneider S (2000) Concurrent and Real-time systems. Wiley
50. Sutton RS (1988) Learning to Predict by the Methods of Temporal Differences. Mach Learn 3(1):9–44. https://doi.org/10.1023/A:1022633531479
51. Szyperski C, Gruntz D, Murer S (2002) Component Software: Beyond Object-Oriented Programming, 2nd edn. ACM Press and Addison-Wesley
52. Tapus A, Ţăpuş C, Matarić MJ (2008) User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy. Intel Serv Robot 1(2):169. https://doi.org/10.1007/s11370-008-0017-4
53. Thörn O, Knudsen P, Saffiotti A (2020) Human-robot artistic co-creation: a study in improvised robot dance. In: 2020 29th IEEE International conference on robot and human interactive communication (RO-MAN), pp 845–850. https://doi.org/10.1109/RO-MAN47096.2020.9223446
54. Torsi S, Ardito C, Rebek C (2020) An interactive narrative to improve cultural heritage experience in elementary school children. J Comput Cult Herit 13. https://doi.org/10.1145/3382771
55. Van Der Aalst W, Van Hee K (2004) Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge, MA, USA
56. Viswanadham N, Narahari Y (1992) Performance Modeling of Automated Manufacturing Systems. Prentice-Hall Inc, Upper Saddle River, NJ, USA
57. Wada K, Shibata T (2006) Robot therapy in a care house - its sociopsychological and physiological effects on the residents. In: Proceedings 2006 IEEE international conference on robotics and automation, pp 3966–3971. https://doi.org/10.1109/ROBOT.2006.1642310
58. Wang R, Luo P, Guan Y, et al (2014) Timed automata based motion planning for a self-assembly robot system. In: 2014 IEEE International conference on robotics and automation (ICRA), pp 5624–5629. https://doi.org/10.1109/ICRA.2014.6907686

59. Wang R, Guan Y, Song H et al (2018) A formal model-based design method for robotic systems. IEEE Syst J 13:1096–1107. https://doi.org/10.1109/JSYST.2018.2867285
60. Watkins CJCH, Dayan P (1992) Q-learning. Machine Learning 8(3):279–292. https://doi.org/10.1007/BF00992698