Learning to Interact: Connecting Perception with Action in Virtual Environments

Pedro Daniel Barbosa Sequeira
pedro.sequeira@ist.utl.pt

ABSTRACT
Modeling synthetic characters which interact with objects in dynamic virtual worlds is important when we want agents to act in an autonomous and non-preplanned way. Such interactions with objects would allow the synthetic characters to behave and be perceived more believably. Once objects offer innumerable uses, it is essential that the agent is able to acquire the necessary knowledge to identify such action possibilities while interacting with its own environment. We propose a conceptual framework that allows the agents to identify possible interactions with objects based in past experiences with other objects. Starting from sensory patterns collected during interactions with objects, the agent is able to acquire conceptual knowledge about regularities of the world, its internal states and its own actions. The presented work also proposes that such acquired knowledge may be used by the agent in order to satisfy its needs and current objectives by interacting with objects. Preliminary tests were made and it is possible to state that our agents are able to acquire valid conceptual knowledge about the regularities in its environment objects, its own actions and causal relations between them.

Keywords
Synthetic agents: human-like, lifelike, and believable qualities, learning agents.

1. INTRODUCTION
In the past few years 3D virtual worlds have had an enormous growth mainly due to the success of computer games. These worlds are often populated by autonomous synthetic agents capable of acting and interacting with their environment, with other agents and also with humans. The virtual environments are composed of several objects that may have different types, uses and effects in the world. In the context of the current work, the notion of object comprises all the things that exist within an environment which directly or indirectly affect an agent. Objects are also entities that provide the agent with certain possibilities of action.

As such, in order for the characters to fully take advantage of the objects they perceive in the world, they should “know” them and their consequences beforehand. Of course this is possible when the number of objects is limited and the characters are partially scripted to know all kinds of objects and their uses in the environment. However, as the complexity of the virtual worlds increases and the types of objects become larger and perhaps dynamic, other approaches need to be followed.

The support-mechanisms for a system that deals with agent-object interactions in virtual worlds grow to be very complex especially when the number of both objects and their available interactions becomes bigger. Such system has to allow the characters not only to interact with their environment in a non-planned, non-predictable way, but also to learn from their experiences and actions, thus creating unique personalities within the whole set of interacting agents. As such, the main questions we try to answer in the research here presented are:

How can an agent identify the possibilities of interaction with an object and the consequences of that interaction, based on previous past experiences with other objects?

How can these previous experiences guide the agent’s behavior by making him choose the best interactions with objects in order to satisfy its current goals and needs?

In a previous work [10], we’ve presented a framework and implemented a test suite that allowed an agent to interact with its world and to learn concepts about objects and causal relations between actions execution and changes on the agent’s state. This framework allowed us to validate some learning methods in order to allow agents to acquire conceptual knowledge. With the objective of allowing agents to improve their acting performance by knowing the usual consequences of its interactions in the environment and to be more autonomous and believable, new questions need to be answered. In this paper we propose a conceptual framework including the agent’s architecture in order to answer those questions before-mentioned.

The solution is based in the qualitative and symbolic description of the objects’ perceptual features that affect an agent during an interaction. These features are perceived by the agent throughout time by interacting with its world objects and taking into account its capabilities, current internal state and previous experiences. It is also hypothesized that the inclusion of external knowledge (from other agents, from the user, etc.) has interference over the agent’s decisions when interacting with the environment. The suggested solution allows pertinent actions performed by the agent to be memorized and action sequences to be created. We believe this endows an agent with real-time planning capabilities...
and to be able to predict the usual results of its actions, as it is inspired in the notion of object affordances [6] (possible uses). The solution also proposes the inclusion within the objects’ description of all the necessary information in order to allow agent-object interactions in 3D virtual worlds, inspired by works in the area of Smart-Objects [7][2]. It is also hypothesized the existence of a perceptual system that is able to transform physical and quantitative object features into qualitative perceptual features to characterize its possible interactions.

This paper is organized as follows. First we will present some related work that inspired our current work. Then we will describe the proposed framework to solve the presented problem, including the conceptual model and framework’s architecture. We’ll then present a small case study and show some preliminary results attained with it. Finally we draw some conclusions and provide the ideas of the future planned work.

2. RELATED WORK
Agent-object interactions’ research topic can be seen at two different levels: one relating to several perspectives over object use and object meaning acquisition; the other about current systems that deal with agent-object interaction both in virtual worlds and in the robotics area. We’ll now present the most relevant works in each of the before-mentioned areas which inspired the current work.

2.1 Perspectives over subject-object interactions
The term affordance was first used by the psychologist James J. Gibson [6]. For Gibson, each individual lives in a particular space of its environment, composed by the set of the affordances that he acquires throughout time. Thus affordances are defined by the set of things that the environment offers an individual taking into account his physical capabilities, i.e., what the individual perceives.

Gaver tried to bring some ideas about Gibson’s affordances to the HCI community [8]. In his perspective, we must separate affordances and the available information about them from their perception. As such, he also introduces the concept of sequential (nested) affordances, which explain how the discovery of an affordance can lead to the detection of other affordances and the creation of sequence of actions emerging from each discovered affordance.

Another important perspective is given by Jakob von Uexküll, who introduced the term functional tone. For Uexküll, each individual ascribes a unique meaning to the objects with which it interacts daily, thus creating a subjective universe, his own vision of the world [11]. The objects start by being neutral to a subject. By acting upon them the individual creates effector images of the objects, thus associating perceptions to actions, giving meanings (functional tones) to the objects that were found. It is also important to say that for Uexküll the meanings that are invested upon the objects are influenced by the subject’s current mental and emotional state (prevailing mood) and don’t rely on their physical properties since they are immutable and belong to each individual’s perceptual world.

Finally, in the work developed by Zhang et al. [13] the main idea is that the generation of much of a human’s intelligence results from processing information distributed across an individual’s mind and external data derived from the world. The authors show that internal and external information about an individual can be interconnected in such a way to describe the environment’s affordances, through the actions that are allowed to an individual to execute taking into account its intrinsic acting constraints. This way we can model different behaviors for different individuals.

It is also important to state that one aspect that unites all the perspectives relates to the fact that the acquisition of action possibilities from objects relies in the subject’s social and cultural background. Actions performed over objects make also part of the individual’s experience. They tell an agent what it may expect to happen when interacting again with some other object which perceptual features are somehow similar to those sensed from previous ones.

2.2 Systems dealing with agent-object interactions
By looking at current systems that deal with agent-object interactions we can divide them in different areas: object and world representation, the use of plans to solve problems and the use of affordances to guide agents’ behavior.

In the object and world representation area, Kallman et al. [7] introduced the widely used paradigm of Smart-Objects. Their aim was to solve problems related to the graphical aspect of interactions between synthetic characters and objects. The main idea is to encapsulate within the object descriptions of its characteristics, properties, behaviors and all the necessary scripts associated with each possible interaction with it. Recently, Badawi and Donikian [2] propose a similar approach to the Smart-Object paradigm that allows the generation of real-time interaction animations by storing in the objects a general description of the agent-object interaction process by means of complex actions formed from a set of 7 basic actions.

Another way of dealing with the use of objects by agents is by attaching to the agent’s mind the adequate knowledge and use planning for dealing with the problem of objects. In the area of action planning, the work of Abaci et al. [1] extends Smart-Object architecture by associating plans represented in a formal language to each available action in an object, thus providing the agent with semantic information about their consequences. In the SODAJACK system [5], Geib et al. present a planning module that makes the bridge between action goals that must be achieved by agents and the movements they have to perform, transforming task-actions in action directives. The motivation for this work comes from the idea that a generic task can generate several expansions according the agents’ capacities and intentions, the characteristics of an object and the current state of the world.

Some other systems have made efforts in trying to use the concept of Gibson’s affordances to endow agents of learning facts about their environment by interacting with objects, which will allow them to satisfy its needs by performing specific actions. Within this area, one of the works that greatly inspired the research here presented was done by Viezzer et al. [12]. It is proposed a way to internally represent affordances by joining
external characteristics and possible uses for an object. Such representation is used to index stored tables that include the expected changes in the world and agent when it executes certain action. This information can later be used to select an object in the world that satisfies a certain agent’s need. It is also proposed that a combination of external and internal sensor values describe the agent’s current state, and that a combination of movement sensors describes an action.

Another inspiring work is presented by Cohen et al. [3]. The proposed system characterizes the environment and the objects by statistically analyzing sensory information perceived by the agent when interacting with them. This allows an agent to learn concepts of things and activities through the interaction with its world.

In another work [4], Cos and Hayes allow an agent to associate affordances with expectations of changing its internal state. This estimation can be further refined as the agent interacts again with the respective object, allowing him to learn from past experiences. This work shows how a simple agent can fulfill its needs by interacting with objects which offered functionalities satisfy them.

Finally, another contribution for the work here presented comes from a recent seminar [9]. Some architectural and functional issues were discussed in order to create an affordance-based system which according to the authors “will provide a systematic way to detect agent-specific possibilities and alternatives for action based on function-oriented perception”. A working implementation would enable a robot to find more action alternatives than pure appearance-based perception approaches. We believe that such advantages will also outcome from building synthetic characters inspired in affordance-based paradigms.

3. FRAMEWORK

According to the systems and perspectives presented in the related work section, we now present the proposed solution to the problem being discussed. For that we give an example scenario to better understand the problem being solved.

Imagine Bob, an autonomous agent that comes to a new place in which he never lived before. This new environment is populated by objects and some local inhabitants. As our agent is far away from home, all of the objects in this new environment are totally unknown to him as he never interacted with them before. As time goes by, Bob becomes more and more starving and the only thing that the boy’s experience tells him is that he needs to eat in order to survive. In this desperate moment, Bob comes to a place where his eyes reach objects like the ones present in Figure 1. A nearby native (another agent) sees Bob in trouble and decides to help by offering him a fruit box. This stranger also tells him that must not eat the mushrooms as they are poisonous. According to Bob’s current situation and his interaction experience, what object should he try to eat among the available ones?

The solution is a framework inspired in the notion of affordances presented in the previous section. We propose these possibilities of action to be perceived by the agent by sensing its environment and taking into account its current needs, goals, mood, emotions, etc., i.e., it’s internal state. For this to happen we propose two phases controlling the agent’s behavior: conceptual knowledge acquisition and internal state satisfaction. Throughout time, the agent is able to perceive some patterns in its sensory data and will create concepts relating facts that change or maintain often together. The proposed framework also considers the acquisition of the notion of causality by recording facts that happen before or after other facts thus creating relations between concepts. This conceptual knowledge and relations can later be used to guide the agent’s behavior and help him fulfill its current needs and goals through the use of objects in its environment.

In the framework, the modeling of the agent’s perceptions is based in Cohen’s [3] work and the knowledge acquisition process and behavior control relies in Viezzer’s [12] system and theories. Moreover, the proposed framework’s architecture is based in some ideas taken from the Dagstuhl seminar [9].

Next, the terms associated with the conceptual framework will be presented. As the framework is thought to be implemented in virtual-world simulations, an instant of time is considered to be an update step of such simulation application.

3.1 Conceptual Model

The proposed framework’s conceptual model can be divided in two main elements: world and agent representations, and representations of the conceptual knowledge an agent learns during the interaction with its environment.

3.1.1 World and agent representations

As mentioned before, [10] presents a limited framework that allows an agent to interact with its world and learn concepts about objects and causal relations between actions and changes on the agent’s state. Some conceptual modifications were made in the present framework in order to better represent the agent’s sensors, actions and acquired conceptual knowledge.
As stated before, streams are updated through tokens. A Token is a qualitative symbol of external or internal stimuli, so it comes from the environment or from the agent, respectively. Examples of tokens can be “sweet”, which will be placed in the agent’s “taste” stream, or “happy”, which will be placed in the “mood” sensory channel. Perceived tokens give origin to sensations. A Sensation makes the connection between a certain stimulus and the sensory channel where it must be placed whenever the agent perceives something. It consists in a pair stream-token. As there are 3 types of streams there are also 3 types of sensations.

An Action is the result of an internal process of the agent that can result in a variation of its internal or external state. Action execution provides information to the agent at two levels of perception: perceiving the action itself and perceiving the consequences of the action. Returning to Figure 2, by executing action “eat” over the object “orange”, action-streams “mouth” and “throat” will be activated because the agent is sensing that it is “mouthing” and “swallowing”. As a result of that, and taking into account the object’s properties, the external-streams “taste” and “tactile-tongue” are updated, saying that the agent senses something “sweet” and “soft”.

Actions then define the agents’ experience of activity, forcing both him and its environment to evolve by a sequence of state transitions. In Figure 3 we can see that at a certain instant of time, the agent’s state is defined by its external and internal sensations. Transitions between states are described by action-sensations activated during the execution of certain action.

3.1.2 Conceptual knowledge representation

Conceptual knowledge acquired by the agent can be divided into concepts that represent things in the world and usual agent actions, and activities that create causal relations between concepts and represent the agent’s pertinent activities.

As such, a fundamental term in our framework is the Concept. Concepts are pieces of knowledge that the agent learns throughout time by interacting with its environment. It consists in a set of two or more sensations of the same source type, i.e., a set of internal, external or action sensations. Concepts have associated a certain probability that represents the confidence of the agent in relation to them, i.e., the likelihood of the frequent and simultaneous appearance of such sensations in the agent’s streams. Concepts can have different representations according to their type, as we can see from the examples in Figure 4:
In Figure 4: A describes an action concept representing a body movement; B shows an external concept representing orange and spherical things; C is an internal concept that can represent a hunger state of the agent.

Another kind of knowledge the agent learns throughout time are activities. An Activity relates to the way the agent usually interacts with its environment and what changes normally happen by performing actions over certain objects. An activity represents patterns of actions overtime comprising the agent’s experience. Conceptually, an activity corresponds to the transformation from certain internal and/or external concepts (the initial state) to other internal and/or external concepts (the final state) by means of action concepts (an action), as we can see in Figure 5:

![Figure 5](image URL)

**Figure 5 – Example of an agent’s pertinent activity.**

Besides the initial and final states and an action-concept, an activity has a certain probability associated. It represents the chance of, having the agent a current state matching the initial state, by executing the action, ending up with a state which stream configuration will include the sensations present in the final state.

As such, an activity is created when the transformation between states occurs frequently in the agent’s experience, i.e., when the same actions usually lead to the same results. Activities can also be seen as affordance concepts because they allow the agent to “predict” the typical results of performing actions over the objects he senses.

Finally, the result of the agent’s experience stored within its memory is called an Activity-network. It represents chains of activities the agent usually experiences, sequences of pertinent state transitions that will enable the framework’s system to plan some behaviors to the agent.

### 3.2 Architecture

Based on the framework’s conceptual model we can create the agents’ architecture and respective modules which use and manipulate such concepts. Our architecture is a 3-layered agent architecture to manage agent-objects interactions in order to solve the presented problem. The architecture’s modules, connections and relations can be seen in Figure 6:

![Figure 6](image URL)

**Figure 6 – Proposed framework’s agent architecture.**

As we can see, the framework’s objective is to ultimately produce conceptual knowledge and guide the agent’s behavior by improving its acting performance. This is achieved by learning concepts relating the objects in the environment and the relations that the agent’s actions have on those properties and on the agent’s internal state. As such, the levels in the architecture are responsible to create, process and manipulate the information that will endow an agent of, starting from its perceptions, knowing what actions must it execute to fulfill its current goals and needs.

The architecture’s **Perception-level** is responsible for the communication between the agent and himself and its environment. It filters perceptual features from objects and external knowledge from other agents. This is the place where the agent’s sensory channels reside and are constantly being updated according to the agent’s interactions.

Another level in the framework’s architecture is the **Learning-level**. It is responsible for the acquisition of knowledge by the agent. It includes learning mechanisms to monitor the agent’s streams and extract some patterns to store in the agent’s memory. It is also accountable of changing the agent’s streams at each time, depending on its current configuration.

The **Behavior-level**. As the name suggests, is responsible for managing the agent’s actions throughout time. The behavior selection mechanism, as later will be explained in more detail, is able to select the next action to perform by taking into account the agent’s current state and its current needs and goals. The result of this architectural level is an action to be performed over an object if the agent and the object can enter in such relationship.

Outside the agent’s architecture there is a system which is relevant for the framework. The **action-validation system** is an external component responsible to verify if the action that was selected by the behavior-level can actually be performed taking into account both the agent and the environment current
configuration, i.e., if agent and object have the necessary requirements in order to certain relation to exist between them. If that is the case, the action can be executed by the agent by means of its effectors.

3.3 Dynamics of the framework

After the summary made over the architecture’s levels, it is important to show how the conceptual knowledge structures are created and evolve overtime, and how the agent’s behavior can be controlled and guided after such knowledge has been generated.

Updating is the most important phase in the description of its dynamics. The system is updated in time steps controlled by the simulation application. At each iteration, the system executes the agent’s pending actions by placing the correspondent action-tokens in the correct action-streams. Both external-knowledge and object perception systems check if new external knowledge has been acquired.

The system also updates some of the agent’s internal streams. This is done by the internal updating-rules system, based in rules that tell what should be the token placed in a certain stream if some token / value is present in another stream / internal variable. For example, we can imagine a rule that places the token “sleepy” when the agent is “awake” for some time, or that makes the association that “energizing” things improve the agent’s “health”.

As suggested by the framework’s architecture and as it was stated in the introductory chapter, the agents’ capabilities evolution is done according to two different phases or levels: the first level consists in gathering information from the environment and from the agent’s usual activities; the second level consists in shaping the agent’s behavior by allowing him to perform actions that, according to the agent’s mind, lead him to good states and situations. These two control phases of the framework will now be described in more detail.

3.3.1 Conceptual knowledge learning

The objective of this phase is to create and update activity-networks stored in the agent’s memory components. The agent’s learning evolution consists in monitoring the occurrence of events related with the presence of stimuli (tokens) in the agent’s sensory channels (streams). If some events occur simultaneously and frequently, then some knowledge structures that relate them are created. Frequently means that the events take place a significant number of times, more frequently than one would expect if they were independent [3]. As we have seen, these events can occur as a result of certain agent-object interactions or originated by the internal updating-rules system. This monitoring process allows conceptual knowledge structures to be created and updated overtime by the pattern-learning system. As we can see from Figure 7, the agent starts by discovering concepts about the world, itself and its activities, and ends up building a set of regular activities in an activity-network, growing overtime both in size and complexity.

![Figure 7 – Conceptual knowledge structures growth overtime.](image)

Let’s now see the concept-learning control evolution by following with the example problem presented earlier. The first kind of knowledge the agent learns is concepts. To see if such sensations are associated we use statistical mechanisms [10] to determine if the set of sensations both start and stop a significant number of times, and also determine their association strength. This strength represents the agent’s confidence in relation to the existence of such concept in the world.

Returning to our example problem, if agent Bob eats a lot of red-apples and oranges, it would be expectable that the system creates some concepts like the following ones (the order is irrelevant):

- Color – red
- Shape – spherical
- Mouth – mouthing
- Throat – swallowing
- Taste – sweet
- Tactile tongue – soft

![Figure 8 – Example of possible concepts acquired by the agent.](image)

These concepts would then describe the agents’ usual activities and also the perceptual features he usually senses from its environment, i.e., from the objects it interacts with. As the system evolves the agent’s confidence on the concepts will change. For example if the agent starts eating things that are sweet but rough instead of soft, the probability of this concept would decrease. We must notice that it is the responsibility of the application to set a threshold confidence in which the agent accepts / rejects the concepts it acquires. For the agent to be able to modify its conceptions about its world and activities, a reinforced-learning mechanism is incorporated in order to update the agent’s confidence about the existence of concepts each time it interacts with its environment. Furthermore, some previously created concepts will be “destroyed”, if the associated probability falls below the established threshold. The same will happen with the creation of bigger conceptual structures. Now that our agent has learnt the facts that rule its activities, the system is ready to find relations between them.

Relations are associations that occur between two types of concepts. Relations between internal and external concepts are created by the system when such concepts seem to be frequently and simultaneously active allowing the characterization of pertinent agent’s states. In this way, when the agent experiences certain action-sensations after perceiving some other sensations of an internal or external-concept, the system can create the relation that tells that such agent’s state potentiates performing that action. Similarly, relations between action-sensations and internal or external-concepts tell us that certain action usually enables the appearance of some agent’s state.
Going back to our example, by eating the fruits present in the environment, the agent would likely apprehend the concept relations present in Figure 9:

![Figure 9 – Example of possible relations between concepts acquired by the agent.](image)

In this case the agent was able to capture that when he is weak and hungry he usually eats something, the same when it perceives things that are orange and spherical. Another rule that the agent can learn from the example is that when he normally eats, later it feels sweet and soft things in its mouth, and that this kind of things often turn him healthy and full.

Furthermore, these relations constitute the base for the construction of larger structures of knowledge: activities. The step that connects relations to the creation of activities is almost direct. Activities are created when the pattern-learning system detects that certain transition between an internal or external concept to other internal or external concept, by means of certain action concepts, occur frequent and simultaneously during the agent’s activities. One should notice that it is not necessary to create an activity having both internal and external concepts present in initial and final state. For example, continuing with our scenario problem, Bob may acquire the activity in Figure 10, meaning that when it perceives something that is burning (like the candle object) and then stretches its arm towards it, he usually becomes alert and feels in pain.

![Figure 10 – Example of an activity acquired by the agent.](image)

As complex activities are gradually created, the system can begin to connect them with each other, storing activities in memory and associating them in activity-networks. An activity-network is created to join two or more existent activities. Two activities can be joined if one’s final state (internal or external concepts) matches the other’s start state, as we can see in Figure 11.

![Figure 11 – Example of an activity-network acquired by the agent.](image)

It’s important to notice that this link between actions that the agent performs in the environment and the learning of concepts about objects allows the agent to be able to experiment in the environment in a proactive manner, leading to goal-directed behavior once the relations are established.

3.3.2 Guiding the agent’s behavior

Activity networks are the elements stored in memory that allow the planning mechanism to create plans of action that the agent can follow to achieve some desired state. This module takes advantage of the representation of activities stored in memory. Activities can be seen as “IF state n AND action j THEN state n+1” rules. This case applies for example if the agent wants to achieve a desired internal and/or external state but its current stream configuration doesn’t match any of the stored activities’ initial states. As such, the system is able to look within the agent’s activity-networks memory for a chain of actions that, taking into account its current state, will lead it with some probability to the desired final state, as we can see from Figure 12:

![Figure 12 – Creation of plans of actions by chaining sequences of activities.](image)

As an example, if Bob is hungry and his wish is to feel full, a plan can be created for him to see “red” and “spherical” objects and then try to eat them. In a similar manner, the adaptive-behavior mechanism can search for the activity that most likely solves the agent’s current needs and select the associated action to be performed next.

This kind of behavior allows the improvement of the agent’s action performance and autonomy. Returning to our example problem, the likelihood of the agent being able to solve its current problem (being hungry) in a short period of time increases as he doesn’t have to try and eat all of the objects he encounters. In fact, some of the objects he perceives in his world can be ignored because they would probably lead to undesirable states. Activity-networks combined with a planned behavior allow the agent to somehow look a few steps further into its future and guess what the results of performing some actions probably are.

4. A CASE STUDY

With the objective of testing the statistical learning mechanisms proposed in the previous chapter, an implementation case study was executed, which description we can find in [10]. To do that, part of the early proposed conceptual framework was implemented. SOTAI framework corresponds to a first step towards the full implementation of the proposed solution framework. Recalling the framework’s control, it occurs at three different levels: a perception level, a learning level and a behavioral level. SOTAI framework implements the first two levels as the objective is to gather pertinent information in the agent’s world, discarding the agent’s activities throughout time.
4.1 Test conditions
We developed a test suite called SOTATester which models an autonomous agent and a virtual environment containing 8 objects described by some perceptual features. The objects’ positions in the environment are randomly generated at the beginning of each test. The implemented agent has a reactive behavior. It explores his world and interacts with the objects. At each time step the agent smells, listens and looks at the objects according to its current position and distance to those objects. When close enough to an object, he touches and tastes that object, unless it’s too big or causes too much pain to be eaten. The agent’s object selection behavior is completely random. Initially the agent doesn’t have any conceptual information about his environment. The agent has internal streams such as hunger, mood or health. External streams include sound, color, smell and pain among others. The application runs 30000 simulation steps. At each step the system is updated and current actions (such as see, touch or mouth) are performed.

4.2 Preliminary results
A series of tests were made using the application according to the conditions described earlier. The preliminary results consist in the largest concepts learned by the agent. The most relevant ones were:

- hunger: hungry; sound: frying; smell: burnt
- taste: metallic; health: ill; taste: bad; pain: painful; power: hazardous
- taste: sweet; health: healthy; taste: pleasant; pain: pleasant; power: energizing

This concept can represent the concept of an internal state of satisfaction for the agent.

5. CONCLUSIONS AND FUTURE WORK
We have presented a conceptual framework that proposes the connection between perceptions and the execution of actions as the basis to improve agent’s acting performance and autonomy in virtual worlds. We believe that such capacity is possible by analyzing sensory data gathered from the environment while the agent interacts with it. Patterns discovered in this data throughout time allow the agent of learning conceptual knowledge about the world objects, its internal states, its actions, and causal relations between them. In this manner, the agent is able to adapt itself to new situations and unknown worlds having into account its past experiences.

The conceptual model of the framework was described, including the agent’s architecture and the dynamics involving not only the learning processes necessary to generate the required conceptual knowledge, but also the planning schemes in order to allow the agents of achieving their goals and fulfill their needs through the use of world objects.

Part of the framework was implemented and some tests were taken. Preliminary results allow us to say that the framework enables an agent of acquiring valid conceptual knowledge about its world and common activities. In the future, more should be implemented according to the architecture here described. Furthermore, performance tests must be taken in order to test not only the framework’s capacity in generating new conceptual knowledge but also if it is able to do it when a large number of objects and possible actions are considered.

The importance of modeling agent-object interactions in virtual worlds resides in the fact that they demonstrate the effect they have on both agent and its environment throughout time. Interactions are of most importance in the agent’s learning process by allowing him to adapt to its world and act accordingly to the verified facts. By continuously interacting and perceiving its environment, the agent is able to understand it, predict its behavior and its own action consequences. The conceptual framework proposed in this paper was born to model this behavior. It allows an agent to evolve through experimentation, discovering, acting, learning and living. This approach will hopefully allow synthetic characters in virtual worlds to become more and more autonomous and thus, more believable.

Like humans, synthetic characters don’t inhabit virtual environments by themselves. They live in societies, exchanging information and creating relationships with other agents. Currently our framework is designed to directly receive external knowledge representing the agents’ social and cultural beliefs. As each agent’s knowledge is also based in its own experiences, we believe that such exchange of information will allow us to build more believable agents as unique personalities emerge. In the future, we would also like to test the possibility of this cultural and social knowledge to be acquired by an agent by observing other agents in their daily activities just like it learns from its own actions.

6. REFERENCES


