HAL
open science

Format-Store: a Multi-Agent Based Approach to Experiential Learning

Philippe Mathieu, David Panzoli, Sébastien Picault

To cite this version:

Philippe Mathieu, David Panzoli, Sébastien Picault. Format-Store: a Multi-Agent Based Approach to Experiential Learning. 3rd International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES 2011), May 2011, Athens, Greece. pp.120-127. hal-00826444

HAL Id: hal-00826444

https://hal.science/hal-00826444v1

Submitted on 29 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

2011 Third International Conference on Games and Virtual Worlds for Serious Applications

FORMAT-STORE: a Multi-Agent Based Approach to Experiential Learning

Philippe Mathieu, David Panzoli and Sébastien Picault
Laboratoire d'Informatique Fondamentale (LIFL)
University of Lille I,
59655 Villeneuve d'Ascq,
France
Email: *firstname.surname*@lifl.fr

***Abstract*—FORMAT-STORE is a serious game application designed for training salesmen and managers in the context of a retail store or a larger supermarket. In this paper, we argue that a relevant way to train a salesperson to their daily activities (e.g. customer relationship management, store management and stock control) consists in immersing them in a 3d environment populated with realistic virtual customers. The first part of this paper presents the multi-agent approach we apply to the design of the intelligent customers. Specifically, we analyse the contribution of the interaction-oriented methodology IODA in facilitating the conception of a game for non computer-scientists by means of a user-friendly design tool and the automated implementation of the conceptual model. The second part includes a case study where we present the organisation of the game around scenarios modelled with respect to the pedagogical requirements. We discuss how we intend to engage the learner by carefully adjusting the difficulty of these scenarios.**

I. INTRODUCTION

FORMAT-STORE is a learning platform designed to train business school undergraduate students to retail or wholesale trade. FORMAT-STORE is composed of a traditional learning content management system (LCMS) and a serious game. The LCMS contains approximately 25 thematic lectures dealing for instance with customer welcome, information or argumentation, illustrated with practical examples of dialogues and case studies. The serious game comes as a complement to the learning platform and addresses two problematics.

- Contextualising the knowledge from the LCMS by offering a complementary tool where the learner can apply their newly learnt skills *in situ* in a virtual store and experiment with different ways of dealing with a customer.
- Training the learner to new skills like task prioritisation or personnel allocation by means of the realistic simulation of a store.

FORMAT-STORE is designed for learners with different ages and therefore adapts to several training stages. Undergraduate or 2-years university graduates assume the role of employees in an organic convenience store. 5-years university graduates play the game as managers in a large supermarket. Different skills must be acquired by the learner depending on the training level. Basically, all of them can be classified in one of the following categories.

- Store management skills help maintaining the store functional and safe to the customers. Unpleasant to hazardous incidents can happen like a neon light flickering or a slippery stain in an aisle. A salesperson or a manager must be able to notice such an event and make a decision quickly.
- Stock control is related to making sure the store is supplied with products at any time. A junior salesperson must check the expiration date of the goods every morning and take an appropriate action (clear or trash the item) if necessary. A senior salesperson has in addition the responsibility to order supplies from the wholesaler. Finally, the manager acts at a higher level by analysing the sales or introducing new references.
- Customer relationship management (CRM) is the most important aspect of a trade. It consists of looking after the customers' satisfaction, solving their problems, giving them information, etc. CRM involves a good knowledge of the products but more importantly a good practice in dealing and arguing with customers of different profiles.

The main objective of the FORMAT-STORE serious game is to answer the two problematics defined earlier in this section by immersing the learner in a virtual store populated with realistic customers. Assuming the role of a salesperson or a manager, depending on the training objectives, the learner is confronted to the daily tasks of managing a store and dealing with the customers.

II. RELATED WORK

Training future personnel to CRM alone is already a serious challenge for a company and for this reason the serious game approach has been investigated in different contexts.

Knowledge Drive is a serious game developed by Caspian Learning [1] for Volvo Car UK and aimed at replicating the experience of an actual showroom. The main objective of the game is to train salesmen to the products sold by Volvo but also raise their awareness regarding the legislation. In a 3d environment (see figure 1.a), the learner meets virtual customers and builds profiles on the basis of clues they give during dialogues. As the learner makes assumptions regarding the profile and the expectations of the customers, he/she is expected to identify the right car for them and build an

978-0-7695-4419-9/11 $26.00 © 2011 IEEE
DOI 10.1109/VS-GAMES.2011.23

IEEE computer society

(a) (b) (c)

Fig. 1. (a) "Knowledge Drive" from Caspian Learning replicates the experience of an actual car showroom where the learner builds a line of argument to eventually close the deal. (b) The "Sales Game" from PIXELearning broadens the activity of the learner by enabling them to manage a professional network or a customer database. (c) The "BCV bank" project trains advisors to argue with scripted customers about financial products and bank services.

appropriate presentation by ruling out the irrelevant arguments from an initial argumentation. Law breaking scenarios are introduced in the argumentation; they must be identified and discarded by the learner.

The Sales Game by PIXELearning [2] spans a broader range of missions related to business. In addition to sales training (depicted in figure 1.b), the learner is expected to attend virtual meetings, meet virtual colleagues, manage a professional network and build a customer database. Gaining in experience, the learner increases their knowledge and their skills and competes for the salesperson of the year election.

Another game of interest has been developed by Daesign [3] for the Cantonale Vaudoise bank (BCV) to train customer advisors selling financial products and services. The game reenacts an interview with a virtual customer. Although the development of the dialogue is mostly scripted (greeting the customer, analysing the needs, arguing with the customer and closing the deal), the player must select at each significant step of the interview one option among several attitudes: analyse, elaborate, carry on (figure 1.c).

Although all the games cited above focus on the CRM only, several reasons make them particularly interesting for the scope of this paper. Firstly, they point out the many advantages of teaching the relationship with the customer – which is a critical aspect of a sale – using an interactive simulation over relying on traditional teaching methods. Besides, the serious game approach is cost effective. Also, they demonstrate the usefulness of a game in complement of a knowledge base – all the games cited above come in addition of a traditional learning platform – for the knowledge to be contextualised and translated into skills by the learner. Indeed, although a LCMS enables the content to be personalised to the learner, the knowledge is neither personally constructed nor applied. A game offers this opportunity as the learner uses it as a playground where new skills can be tested and old ones can be rehearsed.

In the context of FORMAT-STORE however, some aspects of the role of a salesperson are little or not represented in those games. Particularly, FORMAT-STORE considers as essential the unpredictable expectations and needs of the customers, the necessity for prioritising the tasks when several customers are in the store and the ability to identify a customer who needs information or help from another customer who does not.

Addressing all these aspects, not to mention the dynamic nature of the products and the store itself, involves the immersion of the learner in a virtual store. Customers behaving in a realistic and autonomous way train the learner to solve their problems and look after their satisfaction. The paper is organised as follows. Sections IV and V introduce the multi-agent approach. We argue the adequacy of such an approach to modelling a crowd of intelligent characters in a virtual environment. Section VI illustrates in a case-study how such a virtual environment can be used as a training tool provided the adequate integration of the pedagogical scenarios. finally, section VII sums up the main contributions of our methodology to the FORMAT-STORE project.

III. A SERIOUS GAME FOR IMMERSIVE TRAINING

A. Presentation

The sales argumentation training games presented in section II are presented as Flash applications integrated in a web page. The FORMAT-STORE serious game maintains this online availability requirement since the game must be included in the LCMS. Yet, a 3d environment requires high-end graphics, which are provided here by X3d (formerly VRML) technologies. The game requires a – freely available – plugin at the user's end but displays in return compelling 3d graphics, yet computationally efficient enough to allow for the game to run on an Internet browser.

Using the arrows on the keyboard or a graphic user interface (GUI), the player is enabled to move an avatar freely in a virtual store (figure 2) or a larger supermarket (figure 4).

Using the mouse, the player can also interact with many elements of the store including the product items and the autonomous characters – customers and other employees. In that latter case, a specific GUI is loaded as the player

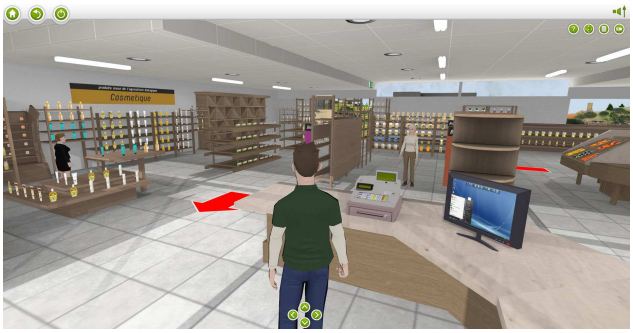

Fig. 2. FORMAT-STORE features a virtual environment populated with autonomous customers. The player can control the avatar anywhere in the store using the arrows on the keyboard or graphic controls on the screen. The environment in this figure represents a convenient grocery store. Other environments are being tested at the moment, like a larger supermarket (see figure 4).

enters a conversational mode (figure 3) where they can select propositions.

Fig. 3. Conversational interactions in FORMAT-STORE offer a specific GUI and allow the player to select a proposition among several possible. Although customers are autonomous adaptive agents, their behaviour are scripted during a dialogue.

B. Game design and scenarios

Traditionally in games where the player faces virtual characters, unitary scenarios defined in accordance with the educational requirements are represented by the specific behaviour of virtual characters. For instance, in the salesmen training games presented in section II, each character represents a pedagogical situation to explore (advise the right product in accordance with the client's profile, cope with a customer difficult to argue with, etc). In practice, each character embeds a script describing either a specific dialogue or a predefined sequence of actions/interactions.

In FORMAT-STORE, we put an emphasis on the game's scalability, namely to what extent the client will be able to mend or remove existing cases or add new pedagogical elements. We explore an original approach taking advantage of the behaviour's adaptiveness in a multi-agent system. The customers wandering in the store at any time are merely going to their business – shopping for goods – trying to fulfil internal goals – purchasing items on a shopping list or querying for information – instead of following a scripted behaviour. In this context a scenario is not attached to a specific character but rather consists in attributing goals to the customers or introducing disturbances in the environment. For instance, an item a customer is looking for can be removed, a customer can be introduced in the store with a missing information, an information sign can be misplaced, an aisle can be made impassable by an oil stain, etc. As a result of the agents' adaptive behaviour, one or several customers will be affected by the trouble introduced by the scenario and inspire a specific reaction, ranging from being upset to complaining to the employee depending on their profile.

The next section presents how multi-agent systems, and specifically the IODA methodology developed by the SMAC team in Lille, are suitable to implement the adaptive behaviour of a crowd of customers. Section VI then illustrates how educational skills can be translated into game dynamics, keeping in sight the fundamental elements of a gaming experience.

IV. MULTI-AGENT SYSTEMS

A multi-agent system [21] is an organised set of entities called agents interacting in an environment. The term was coined early in the 1990s and encompasses every simulation of a complex phenomenon where interacting particles can be identified.

Multi-agent systems (MAS) have been gathering an increasing interest lately as an alternative approach to mathematical modelling which more traditionally aims at modelling a phenomenon with equations. The main interest in MAS is the ability to consider a complex phenomenon as the – often emergent – result of simple agents interacting with each other. In this context, MAS offer the ability to apply a bottom-up methodology by locally defining the role and the behaviour of each agent participating in the global phenomenon. Besides, MAS help understanding the contribution of each agent whereas a mathematical model can only describe the global mechanisms of the phenomenon. Application areas of MAS include simulating natural phenomena like molecular biology [9] or ecosystems [7], animating artificial worlds [6], video games or computer-generated imagery [4], social behaviour [10], economy [5] or disaster management [14]

A. Individual-centred simulations

Whereas classical simulations aim at modelling a phenomenon with one or several mathematical equations, MAS focus on the individuals participating in the phenomenon. The question MAS address is: knowing the entities participating in a given phenomenon, what must be the behaviour of each individual for the whole phenomenon to demonstrate a desired property?

To solve Artificial Life (AL) problems, MAS have proved well suited for modelling emerging collective phenomenons composed of simple interacting individuals. Reynolds [17] has shown that a visually realistic flock of virtual birds can be obtained by applying a local behaviour composed of three simple rules to each flockmate. Another famous illustration of emergent complexity is provided by Resnick [16] with his simulation of ants and termites foraging behaviour. The common point of these simulations is the proof that self-organisation can emerge without the need for a supervising body provided the agents are locally endowed with an appropriate behaviour and means to communicate with each other.

As a consequence, simulating a phenomenon using a MAS does not require one to understand the phenomenon but merely observe the agents participating in the phenomenon. An individual-centred methodology thus offers a great advantage over a more complex approach where the global description of the phenomenon is necessary. Besides, the principle of emergence remains valid regardless of the reactive or cognitive nature of the agents.

B. Reactive and cognitive agents

All the agents in a MAS support common characteristics although some properties are inherent to different kinds of agents.

An agent is always situated in an environment according to one or several metrics attached to the environment. Possible metrics can be a distance in an Euclidean space or relationships in a social network for instance. Relying on the metric, an agent has a neighbourhood and is itself located in the neighbourhood of other agents. Each agent has a local perception of the environment, usually restrained to its neighborhood. Similarly, an agent is allowed to act or interact locally with the other agents in the environment. Interactions can be direct between two agents or indirect when a media (usually the environment itself) is necessary for a message to be conveyed from one agent to another. Finally, the behaviour of an agent is defined by an internal perception-decision-action loop. The agent is said autonomous as the decision is locally and internally taken by the agent itself.

Traditionally, a dichotomy is made between reactive and cognitive agents. Reactive agents use a trivial decision process that mostly consists in triggering an action according to a perception. Cognitive agents are proactive and plan actions in the long term to achieve internal goals or objectives. The Belief-Desire-Intention model [15] is a good illustration of how such an agent works.

The MAS design methodology involves modelling the behaviour of the agents and their interactions. Depending on where the priority is set, two methodologies exist: agent-oriented programming (AOP, [18]) and interaction-oriented programming (IOP, [19], [13]).

V. IODA: INTERACTION-ORIENTED DESIGN OF AGENTS.

IODA [13] is a multi-agent simulation methodology whose originality is to focus primarily on how the agents interact instead of how they behave. This methodology is grounded on a simple observation coming from experimental experience. The design process a simulation always involves making assumptions at some point, irrespective of the phenomenon to simulate. In the context of a multi-agent-based simulation – e.g. when the actual phenomenon is the obvious outcome of multiple interacting entities – the description of these interactions is the only objective assumption one can formulate. The set of processes occurring "inside" each entity and leading to the interaction can only be guessed. In order to avoid introducing a bias too early in the process, a safe approach would consider setting the foundations of every model on observable characteristics, then building on this model with the constant concern of delaying the introduction of hypothetical assumptions as long as possible. IODA follows this principle by discarding the first hypothesis concerning the selection of which entity is an agent and which is a mere passive object, by providing a user-friendly tool (JEDI) for domain-experts to participate in the design of the interactions and by providing a tool for translating the model (JEDI-Builder) into a set of classes where the interactions can be implemented by a programmer.

A. Everything is agent

The starting point of designing a multi-agent simulation consists in identifying the agents participating in the simulation. What defines an agent in a MAS is a minimum degree of behavioural autonomy and the subsequent ability to trigger autonomously an action or an interaction. Historically, "living" characters in a virtual simulation are considered as agents, "inanimate" objects like trees, furniture or items are not. Unlike other approaches, the first simplifying hypothesis of the IODA methodology is to consider every objet in the environment as an agent, including the environment itself.

Families of agents are listed in such a way that every agent in the environment strictly belongs to one family. Figure 4 illustrates the virtual supermarket environment where every character – employee, customer – and almost every object – item, shelf, information sign, etc. – is an agent in the simulation.

Considering every entity as an agent simplifies the first step of the MAS design, but also provides a convenient way to describe the interactions, as detailed in the next sections.

B. Interactions made concrete

In a similar way every entity in the simulation is represented by an agent family, any behaviour is described by an interaction in IODA.

Unlike other MAS approaches, where an interaction is virtually expressed in the behaviour of two agents interacting together, each interaction in IODA has a software tangibility and a central position in the design. An interaction is a rule involving two agents: the source agent performs the interaction while the target agent undergoes it. The condition and action parts of the rule rely on generic perception and action primitives so that interactions are independent from

Fig. 4. Almost every object in the environment is an agent on the same account as the characters. In addition to simplifying the design process, considering every object as an agent also allows spreading the "intelligence" of the characters across the objects they interact with.

the concrete implementation of the agents. As a consequence, IODA exhibits two unique features. Firstly, the interactions can be represented independently from the agents, as libraries of interactions for instance. Interactions are reusable from one agent family to another and from one simulation to another. They can be allocated to agents in a plug and play fashion. The other advantage of interactions reification is the ability for all the agents to be processed by a generic engine through a single iteration loop, irrespective of the nature of each agent.

C. Interaction matrix

Having defined the agent families participating in the simulation, the next step of the IODA methodology consists in describing their interactions. This is achieved by allocating the interactions in a matrix named the interaction matrix.

Figure 5 presents a simplified version of the interaction matrix used in FORMAT-STORE. All the agents participating in the simulation are listed in the matrix along with their mutual interactions. Note the avatar of the player is included in the matrix like any other agent.

In addition to a name, an interaction in the matrix has several parameters (for clarity reasons, only the priority value of each interaction is visible in figure 5).

- the distance defines the minimum distance between the source and the target for the interaction to be allowed. For instance, seizing an item on a shelf requires the character to be standing in front of the item; a character can read an information sign from a certain distance; etc.
- preconditions must be verified before the interaction can effectively be triggered. Seizing an item is only possible when the quantity of this item is not null.
- the priority is used for sorting between several interactions whose preconditions are verified.

The interaction matrix in figure 5 is a simplified version of the matrix currently in use. Yet, the behaviour of each agent is entirely described (this point is elaborated in the next section). We argue that this representation is functionally equivalent to a more complex algorithm, put aside the difficulty for a non computer scientist to read the latter.

In the current version of the matrix, more agents are represented (a security guard, a cleaner and other employees) and the behaviour of some agents is more complex. For instance, customers are now endowed with an opportunistic behaviour allowing them to change their plans if they happen to notice a product in their visual range which is not their current goal. Customers are also able to make an impulsive purchase.

D. Computer-aided design for non experts

We have mentioned earlier in this paper the importance of involving domain-experts as far as possible in the design process. IODA follows this principle by proposing a simplified multi-agent methodology made accessible to non computer scientists. The implementation stage has received the same attention with two additional elements of the IODA methodology: JEDI and JEDI-Builder. JEDI is an application programming interface (API) providing a set of Java classes for a user to ensure the rigorous implementation of their IODA conceptual model. JEDI-Builder is a Java applet assuming two roles. Firstly, JEDI-Builder provides a computer-aided design tool for the user to model the interaction matrix using a user-friendly graphical interface where agents and interactions can be added in a drag-and-drop fashion. Secondly, JEDI-Builder is also able to translate such a IODA-compliant matrix into a JEDI application that can thereafter be used as a simulation or integrated in a larger project. Using these tools enables a user to implement the major part of a MAS without any particular knowledge of computer programming. At the end of the process though, a computer-scientist is required to implement the core of each interaction, if those interactions are not part of an already existing library – which is often the case as JEDI natively includes a set of predefined generic interactions.

The implementation of an interaction can take many shapes, from the most trivial to more complex tasks. For instance, in FORMAT-STORE, the player as an employee can restock an item on a shelf or help a customer. The interaction "restock" simply consists in increasing the quantity value of a target item. In contrast, the interaction "help" involves more complex operations. The camera position is changed for a closer look on the customer. A dialogue window is opened and a script is started. This script displays a narrative text and several possible answers among which the player has to choose one. Depending on the answer, the facial expression of the customer changes (satisfied, lukewarm, upset) and the player is rewarded (positively or negatively).

Although complex interactions like the dialogue situation seem at first glance the most useful in terms of behaviour, the very interest of multi-agent simulations lies in using multiple simple interactions, as explained in the next section.

source \ target	∅	Employee	Customer	Sign	Item	Checkout	Store	Exit
Employee **(Player)**	+move(-)		+help(-)	+restock(-)				
Customer	+exit(8) +findCheckout(5) +chooseNextItem(4) +moveTowardsTarget(2)				+takeItem(3) +askForHelp(1)	+queue(6)		+find(7)
Sign		+inform(2)	+inform(1)					
Item		+inform(2)	+inform(1)					
Checkout	+cashFirstCustomer Queuing(1)		+inform(2)					
Store	+spawnCustomer(-)							
Exit			+inform(1)					

Fig. 5. The interaction matrix presents the interactions allowed for any agent family as a source towards any other agent family as a target (including self) or the environment (degenerate interaction, column ∅). How to read this matrix? For the Customer agent family for instance, the column labelled Customer lists all the interactions of which a Customer is a target. In this example, the Customer is basically informed by almost any other agent (of the location of Items by a Sign, of the price and quantity of an Item by this Item, etc). The row labelled Customer lists all the interactions of which a Customer can be the source. Priority values (n) are used when several interactions can be applied at the same time. The higher the value, the higher the priority.

E. Adaptive behaviours

This interaction-oriented vision offers an original implementation of the concept of affordances. In his influential book [11], Gibson states that the interaction capabilities – the functions – of any object in a real environment are mainly suggested by the object itself – its shape, position, etc. One interpretation of the concept of affordances has been used many times since in character animation, where computer graphics scientists have found more intuitive to attach the animations and the algorithms necessary for an interaction to the target of this interaction. For instance, a virtual character wanting to open a door would be acknowledged by the door itself of the position it should stand at and the animation it should play. Affordances are implemented in most simulations or games featuring at least one virtual character interacting with objects in the environment. That way, every passive object describes to the character how it should be handled, under which circumstances and what is the outcome of the interaction. The character is therefore freed from that knowledge, putting a focus on managing internal goals and searching the environment for objects likely to solve these goals. Another benefit of using affordances is to avoid any glitch in the animation by carefully adjusting the positions of the interacting entities and synchronising their respective animations. In such simulations though, the character is always at the origin of every interaction. Whichever interaction is triggered is the result of a complex and often time-consuming decision process selected by a cognitive controller.

The idea that the cognitive abilities of an agent may be spread as well among the interactions offered to the agent is part of the multi-agent approach, and particularly of IODA which considers that an intelligent behaviour can be expressed by a reactive agent. This argument is close to earlier research in artificial intelligence postulating that intelligence is not the result of planning and reasoning – like a cognitive agent using scripts and complex algorithms – but on the contrary that the mere appearance of such cognitive processes is the result of an agent's reactive behaviour in a dynamic environment [8].

To illustrate this principle, let us consider the behaviour of an intelligent customer shopping in FORMAT-STORE. In FORMAT-STORE, a customer is endowed with a profile – 8 different profiles were provided initially by the client, which basically corresponds to 4 age ranges times 2 genders – and a shopping list containing items to be purchased. The number of items on the list depends on the customer's profile.

A virtual customer entering the store will consider every interaction in decreasing priority until one is realisable (e.g. the preconditions are verified). Getting out of the store is only realisable when all the items have been paid. Paying requires first to queue at the checkout (actually checking out is managed by the checkout agent). Queuing can be started when all the items in the shopping list have been retrieved. An item can be taken when the customer is actually standing at reach of hand of the item. Moving towards an item relies on standard pathfinding and navigation algorithms but requires knowing the location of the item, which is acknowledged after an interaction with one of the information signs located throughout the store. While shopping in the store, the customer seems to follow a plan. Yet, every action is independent from the following and their sequence has merely been established by selecting priorities.

The adaptive behaviour of the customers is illustrated by the way they select the items. An agent does not embed a map of the product items. The location of every item is provided by information signs located throughout the store, while the customer is wandering or shopping for already known items. That way, selecting the next item – in practice, the closest item whose location is known) depends on the customer's current knowledge, which depends in turn on the signs the customer has already interacted with. Customers will therefore always take different paths, even when their shopping lists are similar. Disturbances can also be created in the game by adding/modifying/removing signs. The customers can be placed in a different store without affecting their ability to shop for goods.

Without getting into the details, other illustrations of the

adaptive behaviour of a customer include the way customers select the emptiest checkout among the opened positions, how they react when a product is missing in store and how they deal with unexpected events in the store.

VI. Learning new skills in Format-Store

The first part of this paper has shown how the simulation environment could be populated with a crowd of intelligent adaptive customers. The next step towards achieving a game is to introduce scenarios for the player to actually learn in this realistic context.

A. Educational requirements

Despite the novelty of the technique, learning in a populated virtual environment can be supported by classical theories. In particular, two models readily apply to Format-Store.

Experiential or exploratory learning models [12] promote the free exploration and the autonomous and personalised construction of cognitive associations and understandings. The idea underlying Kolb's experiential learning is that a realistic virtual environment like Format-Store allows a contextualised learning, as opposed to the declarative and decontextualised learning provided by the LCMS. Format-Store provides means to learn from real life situations: elaborate a routine, learn to prioritise different assignments.

Socio-cultural models of learning [20] point out the fundamental role of social interactions in the development of cognition. Vygotsky's model replaces the social interaction with the teacher, and by extension with the virtual representation of a tutor, at the centre of any learning activity. In practice, this involves the constant delivery to the user of a feedback on his performance. A first requirement for the game designer is therefore to provide means to assess the player's performance and to guide them toward increasing their skills. The feedback can be delivered continuously in a game whereas it is hardly conceivable in real life. Another strong concept, which is a direct consequence of the initial idea, is to scaffold the learning by building new knowledge on top of the existing, whilst consolidating the latter. Vygotsky claims the utter importance of maintaining the learner in what he names the zone of proximal development (ZPD), situated beyond what the learner already knows – in which case the benefit is null – and below what he cannot achieve on his own – in which case no learning can happen but frustration, no matter how much help is provided. In the context of a game, adjusting the difficulty is a relevant way to control the position of the challenge in the learner's ZPD. Besides, a proportionate challenge provides a leverage on the engagement and the motivation of the player. Therefore, the second requirement commands the game designer to ensure that the level of difficulty is adaptive.

B. Modelling scenarios

The construction of a game session is inspired by the typical week of a salesperson in a supermarket, as presented in figure 6.

day	schedule
Monday	6:30am to 12:30pm
Tuesday	6:30am to 12:30pm
Wednesday	6:30am to 12:30pm
Thursday	day off
Friday	2:00pm to 8:00pm
Saturday	6:30am to 12:30pm 6:30am to 12:30pm

Fig. 6. A typical week for a salesperson in a grocery store.

The duration of a game session has been fixed arbitrarily to 20 minutes for practical reasons. The week in figure 6 therefore needs to be abstracted with respect to the requirements of the following activities:

- Stock control: must be done before the store opening; actions have an implication from one day to another.
- Store management: unpredictable events can happen anytime but have a different repercussion depending on the time.
- Customer care: deals with repetitive tasks, yet every case is different from the others.

As a consequence, a session must include several days, not necessarily consecutive. Each day must be split between store management before the opening (if applicable) and customer care after opening. An example organisation is presented in figure 7.

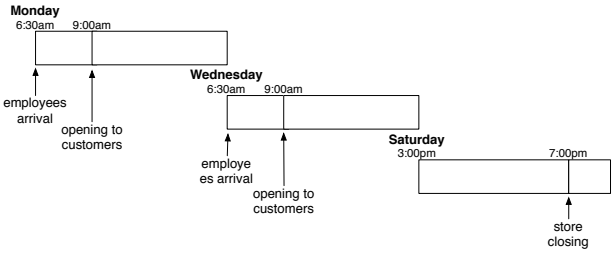

Fig. 7. The weekly schedule is abstracted into a game session following the pedagogical requirements. The game session features three days.

The final step of building the game session consists in populating each virtual day with scenarios randomly extracted from the several activities and with respect to some obvious constraints, as presented in figure 8.

Scenarios are straightforward interpretations of the pedagogical skills provided by the client. The detailed process is too long to appear in this paper. Briefly, it consists in turning an actor's script-like description into a representation more adapted to the requirements defined in section III-B. Basically, each scenario considers a customer with a profile and a set of goals, and possibly an associated event occurring in the store. Examples of scenarios:

- A 25-year-old male customer wants to know why organic food is more expensive.
- A 45-year-old female customer cannot find the product she wants in the shelf.

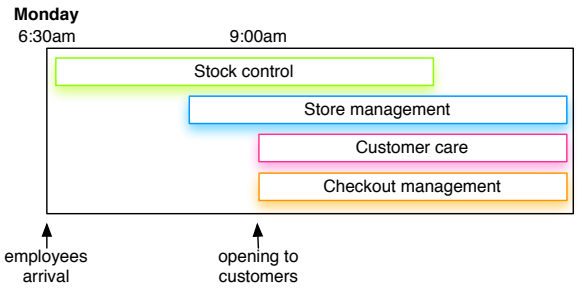

Fig. 8. An average day contains scenarios randomly extracted from all the activities. Activities are active at specific times during the game session. The figure illustrates how they populate the first day of the simulated game session. The following days (see figure 7) are populated following the same principle.

C. Scoring and difficulty setup

Each scenario contains their own condition(s) of success or failure. Every time a scenario comes to an end, the learner is acknowledged of the outcome by an on-screen notification. The success of several scenarios within an activity raises the level of difficulty for this activity which leads in turn to selecting scenarios with a higher difficulty. Conversely, consecutive failures lead to returning to a more adapted level of difficulty. The more difficult the scenario, the higher the score. At the end of the game, an overall score is presented to the player, which basically consists of all the scores of the succeeded scenarios summed up.

VII. Conclusion and Future Work

The Format-Store project is a serious game aimed at training salesmen by immersing them in a dynamic virtual store populated with intelligent customers. In addition to its intrinsic originality, Format-Store brings two original features. Instead of scripting the agents, the educational scenarios attribute goals to the adaptive customers or define disturbances in the environment affecting their behaviour. The scalability of the game is therefore increased as changing the educational content can be handled by the content providers themselves. Another aspect concerns the uncompromising implementation of the affordances concept, allowing to simplify the visual animation of the agents, but above all to obtain intelligent behaviours in spite of the reactive nature of the agents.

Two other aspects are worth mentioning. Owing to a user-friendly design tool, domain-experts are maintained in the conception process farther than any other methodology, thus avoiding the premature introduction of biases and the translation of the formal model into a programming language is partially automated.

To date, Format-Store features a first playable demonstrator including a crowd of characters and their shopping list and a sample of a few scenarios. In the next few months, the whole set of skills provided by the client will be integrated in the game.

In the long term, the ability for the game to be scaled to other contexts will be investigated. We are also considering the side development of a decision support tool for supermarket marketing strategy units. We intend to take advantage of our realistically-profiled crowd of intelligent customers in the context of product placement or shelves layout in a large supermarket.

Acknowledgment

The FormatStore Project is supported by a funding from the French Ministry of Research. In-game pictures of Format-Store belong to Ides-3com. The authors also wish to thank Jean-Baptiste Leroy for his involvement in the Format-Store project.

References

[1] http://www.caspianlearning.co.uk/.
[2] http://www.pixelearning.com/services-the_sales_game.htm.
[3] http://www.daesign.com.
[4] http://www.massivesoftware.com/.
[5] W. J. Arthur, B. Holland, R. LeBaron, Palmer, and P. Tyaler, "Asset pricing under endougenous expectations in an artificial stock market," *Economic Notes*, vol. 26, pp. 297–330, 1997.
[6] E. Bonabeau, "Agent-based modeling: methods and techniques for simulating human systems," in *Proc. National Academy of Sciences*, vol. 99, no. 3, 2001, pp. 7280–7287.
[7] F. Bousquet and C. Le Page, "multi-agent simulations and ecosystem management: a review," *Ecological Modelling*, vol. 176, no. 3-4, pp. 313–332, 2004.
[8] R. A. Brooks, "Intelligence without reason," in *proceedings of the 1991 International Joint Conference on Artificial Intelligence*, 1991, pp. 569–595.
[9] G. Desmeulles, S. Bonneaud, P. Redou, V. Rodin, and J. Tisseau, "In-virtuo experiments based on the multi-interaction system framework: the RéISCOP meta-model." *CMES, Computer Modeling in Engineering & Sciences*, 2009.
[10] J. M. Epstein and R. Axtell, *GrowingArtificial Societies: Social Science from the Bottom Up*. Washington: Brookings Institution Press, 1996.
[11] J. J. Gibson, *The ecological approach to visual perception*. Hillsdale ; New Jersey ; London, 1979.
[12] D. A. Kolb, *Experiential Learning: experience as the source of learning and development*. New Jersey: Prentice-Hall, 1984.
[13] Y. Kubera, P. Mathieu, and S. Picault, "Interaction-oriented agent simulations : From theory to implementation," in *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI'08)*, M. Ghallab, C. Spyropoulos, N. Fakotakis, and N. Avouris, Eds. IOS Press, 2008, pp. 383–387.
[14] R. Querrec, P. Reignier, and P. Chevaillier, "Humans and autonomous agents interactions in a virtual environment for fire figthing training." *Virtual Reality International Conference*, pp. 57–63, 2001.
[15] A. S. Rao and M. P. Georgeff, "Modeling rational agents within a bdi-architecture." in *In Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, 1991, pp. 473–484.
[16] M. Resnick, *Turtles, Termites and Traffic Jams. Explorations in Massively Parallel Microworlds*. Cambridge, Massachusetts, MIT Press, 1994.
[17] C. W. Reynolds, "Flock, herds ans schools: a distributed behavioural model," in *SIGGRAPH'87 vol. 21(4) of Computer Graphics*. Anaheim (USA): ACM Press, 1987, pp. 25–34.
[18] Y. Shoham, "Agent oriented programming," *Journal of Artificial Intelligence*, vol. 60, no. 1, pp. 51–92, 1993.
[19] M. Singh, "Conceptual modeling for multiagent systems: Applying interaction-oriented programming?" in *Conceptual Modeling*, ser. Lecture Notes in Computer Science, G. Goos, J. Hartmanis, J. van Leeuwen, P. Chen, J. Akoka, H. Kangassalu, and B. Thalheim, Eds. Springer Berlin / Heidelberg, 1999, vol. 1565, pp. 195–210.
[20] L. Vygotsky, *Mind in Society*. Cambridge, MA: Harvard University Press, 1978.
[21] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review*, 1994.