

Teaching Autonomous Agents to Move in a Believable Manner within Virtual Institutions

A. Bogdanovych, S. Simoff, M. Esteva, and J. Debenham

Abstract Believability of computerised agents is a growing area of research. This paper is focused on one aspect of believability - believable movements of avatars in normative 3D Virtual Worlds called Virtual Institutions. It presents a method for implicit training of autonomous agents in order to "believably" represent humans in Virtual Institutions. The proposed method does not require any explicit training efforts from human participants. The contribution is limited to the lazy learning methodology based on imitation and algorithms that enable believable movements by a trained autonomous agent within a Virtual Institution.

1 Introduction

With the increase of the range of activities and time humans spend interacting with autonomous agents in various computer-operated environments comes the demand for believability in the behaviour of such agents. These needs span from the booming game industry, where developers invest their efforts in smart and absorbingly believable game characters, to inspiring shopping assistants in the various areas of contemporary electronic commerce.

Existing research in the field of believable agents has been focused on imparting rich interactive personalities [1]. Carnegie-Mellon set of requirements for believable agents include personality, social role awareness, self-motivation, change, social re-

A. Bogdanovych, S. Simoff
University of Western Sydney, Australia, e-mail: {a.bogdanovych, s.simoff}@uws.edu.au

M. Esteva
Artificial Intelligence Research Institute (IIIA), Barcelona, Spain, e-mail: esteva@iiia.csic.es

J. Debenham
University of Technology, Sydney, Australia, e-mail: debenham@it.uts.edu.au

relationships, and "illusion of life". The research in building models of different features that contribute to believability utilises the developments in cognitive modeling and attempts to formalise those models in computational form to implement them in virtual environments [2].

Integrating these believability characteristics into virtual environments (i) is associated with computational and architectural complexity; (ii) is platform and problem dependent, and (iii) is essentially far from achieving a high level of believability [3]. In order to address these drawbacks, rather than identifying, modeling and implementing different characteristics of believability some researchers investigate the automated approach of learning by imitation [4]. Imitation learning is most effective in environments where the actions of a human principal are fully observable and are easy to interpret by the agent [5]. Virtual Worlds where both humans and agents are fully immersed are quite efficient in terms of human observation facilities [5]. Even better means of observation are offered by Virtual Institutions [6] - a new class of normative Virtual Worlds, that combine the strengths of Virtual Worlds and normative multi-agent systems, in particular, electronic institutions [7]. In this "symbiosis" Virtual Worlds provide the visual interaction space and Electronic Institutions enable the rules of interaction. The environment assumes similar embodiment for all participants, so every action that a human performs can be observed and reproduced by an agent, without a need to overcome the embodiment dissimilarities. Moreover, the use of Electronic Institutions provides context and background knowledge for learning, helping to explain the tactical behavior and goals of the humans.

Further in the paper we outline the learning method called "implicit training". The explanation of this method and its role within Virtual Institutions is structured as follows. Section 2 outlines the basics of Virtual Institutions technology. Section 3 presents the principles of the implicit training method, with the implementation details given in Section 4. Section 5 describes the experimental results on learning to move in believable manner. Section 6 concludes the paper.

2 Virtual Institutions

Virtual Institutions are 3D Virtual Worlds with normative regulation of participants' interactions [6]. The development of such Virtual Worlds is separated into two phases: specification of the institutional rules and design of the visualization. The specification defines which actions require institutional verification while the rest of the actions are assumed to be safe and can be instantly performed. Rule specification utilises the "Electronic Institutions" methodology [7], which provides facilities for formalizing the interactions of participants through interaction protocols and runtime infrastructure that ensures the validity of the specified rules and their correct execution. The rules of a Virtual Institution are determined by three types of conventions (for detailed explanation see [7]):

1. *Conventions on language* form the *Dialogical Framework* dimension. It determines language ontology and illocutionary particles that agents should use, roles they can play and the relationships or incompatibilities among the roles.

2. *Conventions on activities* form the *Performative Structure* dimension. It determines in which types of dialogues agents can engage during the activities they perform in a scene, which protocols to use in the dialogues, which sublanguage of the overall institutional language can be used in each scene, and which conventions regulate the in and out flux of agents in scenes. Scenes are interconnected through “transitions” to form a network that represents sequences of activities, concurrency of activities or dependencies among them.

3. *Conventions on behavior* form the *Norms* dimension. Electronic Institutions restrict agent actions within scenes to illocutions and scene movements. Norms determine the commitments that agents acquire while interacting within an institution. These commitments restrict future activities of the agent. They may limit the possible scenes to which agents can go, and the illocutions that can henceforth be uttered.

Virtual Institutions are visualized as 3D Virtual Worlds, where a single Virtual Institution is represented as a building located inside the space labeled as “garden.” The visualization is aligned with the formalised institution rules. The participants are visualized as avatars. Only participants with specified roles can enter the institutional buildings, where they can act according to the rules specification of respective institution. Each institutional building is divided into a set of rooms (every room represents a scene), which are separated by corridors (transitions) and doors. The doors are open or closed for a participant depending on the acceptance of participant’s role by the corresponding scene and the execution state of the institution. Inside each of the rooms only actions that comply with the protocol of the corresponding scene can be executed (for more details see [6]).

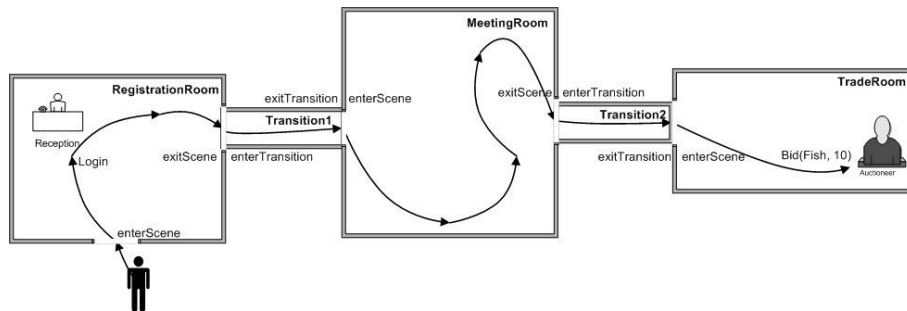


Fig. 1 Outline of a prototypical Virtual Institution containing 3 scenes.

Fig. 1 outlines a prototypical Virtual Institution containing 3 scenes - RegistrationRoom, MeetingRoom and TradeRoom, visualized as rooms connected via corridors. The actions controlled by the institution (institutional level actions) include: enterScene, exitScene, enterTransition, exitTransition and login. The rest of the actions (visual level actions) require no institutional control, these are: moving, jumping, colliding with objects, turning etc. The directed line represents the trajectory of the participant’s movement. The solid figure is the participant, the rest correspond to internal agents (employees of the institution), in this case, a Receptionist and an Auctioneer. The Receptionist verifies the login and password of the participant

in the Registration Room, and unlocks the doors to other rooms if the identity of the participant is proven. The Auctioneer sells different goods in the TradeRoom. It announces the product to be auctioned, waits for incoming bids and sells it to the winner. The Meeting Room is used for social interaction between buyers. In the scenario shown in Fig. 1 the goal of the human is to buy fish in the TradeRoom.

3 Principles of Implicit Training in Virtual Institutions

Existing 3D Virtual Worlds are mostly human centered with very low agent involvement. Virtual institutions, in contrast, is an agent-centered technology, which treats humans as heterogeneous, self-interested agents with unknown internal architecture. Every human participant (principal) is always supplied with a corresponding software agent, that communicates with the institutional infrastructure on human's behalf. The *couple agent/principal* is represented by an avatar. Each avatar is manipulated by either a human or an autonomous agent through an interface that translates all activities into terms of the institution machine understandable language. The autonomous agent is always active, and when the human is driving the avatar the agent observes the avatar actions and learns how to make the decisions on human's behalf. At any time a human may decide to let the agent control the avatar via ordering it to achieve some task. If the agent is trained to do so it will find the right sequence of actions and complete the task in a similar way a human would.

The training of autonomous agents in Virtual Institutions happens on both visual and institutional levels. The actions of the visual are important for capturing human-like movement. The actions of the institutional level, on the one hand, help the autonomous agent to understand when to start and stop recording the actions of the visual level and which context to assign to the recorded sequences. On the other hand, analyzing the sequence of institutional level actions helps, in a long run, to understand how to reach different rooms and separate the sequences of actions there into meaningful logical states of the agent.

Every dimension of the institutional specification contributes to the quality of learning in the following way.

Dialogical Framework: the roles of the agents enable the separation of the actions of the human into different logical patterns. The message types specified in the ontology help to create a connection between the objects present in the Virtual Worlds, their behaviors and the actions executed by the avatars.

Performative Structure: Enables grouping of human behavior patterns into actions relevant for each room.

Scene Protocols: Enable the creation of logical landmarks within human action patterns in every room.

4 Implementation of the Implicit Training Method

The implicit training has been implemented as a lazy learning method, based on graph representation. The Virtual Institution corresponds to the scenario outlined in

Fig. 1. It is visualised as a garden and an institutional building inside the garden. The institutional building consists of 3 rooms connected by corridors. Starting as an avatar in the garden, each participant can enter inside the building and continue moving through the rooms there. In our case, the participants in the institution play two different roles: receptionist and guest. The implicit training method is demonstrated on learning movement styles.

4.1 Constructing the learning graph

When a human operator enters the institution, the corresponding autonomous agent begins recording operator’s actions, storing them inside a learning graph similar to the one outlined in Fig. 2. The nodes of this graph correspond to the institutional messages, executed in response to the actions of the human. Each of the nodes is associated with two variables: the message name together with parameters and the probability $P(Node)$ of executing the message. The probability is continuously updated, and in the current implementation it is calculated as follows:

$$P(Node) = \frac{n_a}{n_o} \tag{1}$$

Here n_o is the number of times a user had a chance to execute this particular message and n_a is the number of times when s/he actually did execute it.

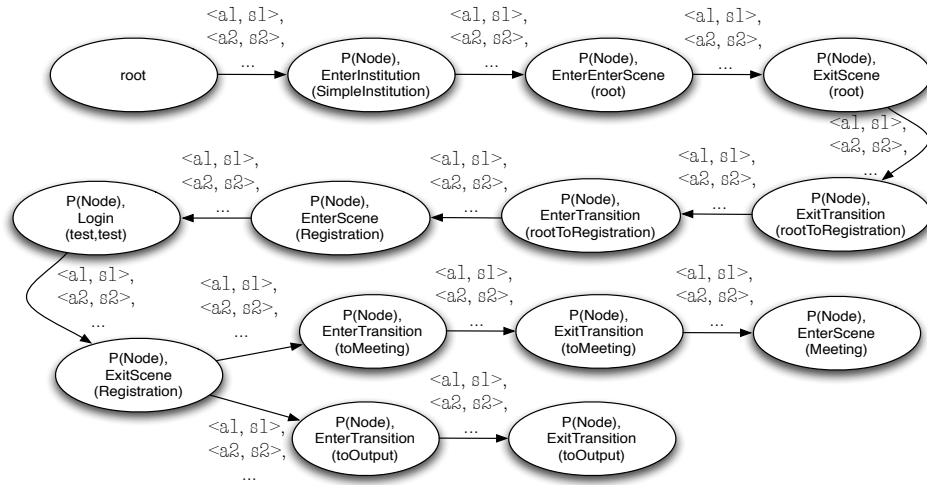


Fig. 2 A fragment of the learning graph.

The arcs connecting the nodes are associated with the prerecorded sequences of the visual level actions (s_1, \dots, s_n) and the attribute vectors that influenced them (a_1, \dots, a_n) . Each pair $\langle a_n, s_n \rangle$ is stored in a hashtable, where a_i is the key of the table and s_i is the value. Each a_i consists of the list of parameters:

$$a_i = \langle p_1, \dots, p_k \rangle \quad (2)$$

A simplifying assumption behind the training is that the behaviour of the principle is only influenced by what is currently visible through the field of view of the avatar. We limit the visible items to the objects located in the environments and other avatars. So, the parameters used for learning are recorded in the following form:

$$p_i = \langle V_o, V_{av} \rangle \quad (3)$$

where V_o is the list of currently visible objects; V_{av} is the list of currently visible avatars. The list of the visible objects is represented by the following set:

$$V_o = \{ \langle O_1, D_1 \rangle, \dots, \langle O_j, D_j \rangle, \dots, \langle O_m, D_m \rangle \} \quad (4)$$

where O_j are the objects that the agent is able to see from its current position in the 3D Virtual World; D_j are the distances from the current location of the agent to the centers of mass of these objects.

The list of visible avatars is specified as follows:

$$V_{av} = \{ \langle N_1, R_1, DAv_1 \rangle, \dots, \langle N_p, R_k, DAv_p \rangle \} \quad (5)$$

Here, N_k correspond to the names of the avatars that are visible to the user, R_k are the roles played by those avatars, and DAv_k are the distances to those avatars. Each of the sequences (s_i) consists of the finite set of visual level actions:

$$s_i = \langle SA_1, SA_2, \dots, SA_q \rangle \quad (6)$$

Each of those actions defines a discrete state of the trajectory of avatar's movement. They are represented as the following vector:

$$SA_l = \langle pos, r, h, b \rangle \quad (7)$$

where pos is the position of the agent, r is the rotation matrix, h is the head pitch matrix, b is the body yaw matrix. Those matrices provide the most typical way to represent a movement of a character in a 3D Virtual World.

Each time an institutional message is executed, the autonomous agent records the parameters it is currently able to observe, creates a new visual level sequence and every 50 ms adds a new visual level message into it. The recording is stopped once a new institutional message is executed.

4.2 Applying the learning graph

Once the learning graph is completed an agent can accept commands from the principal. Each command includes a special keyword "Do:" and a valid institutional level message, e.g. "Do:EnterScene(Meeting)". The nodes of the learning graph are seen as internal states of the agent, the arcs determine the mechanism of switching between states and $P(Node)$ determines the probability of changing the agent's cur-

rent state to the state determined by the next node. Once the agent reaches a state $S(Node_i)$ it considers all the nodes connected to $Node_i$ that lead to the goal node and conducts a probability driven selection of the next node ($Node_k$). If $Node_k$ is found: the agent changes its current state to $S(Node_k)$ by executing the best matching sequence of the visual level actions recorded on the arc that connects $Node_i$ and $Node_k$. If there are no visual level actions recorded on the arc - the agent sends the message associated to $Node_k$ and updates it's internal state accordingly.

For example, let the agent need to reach the state in the learning graph expressed as “ $S(EnterInstitution(SimpleInstitution))$ ”. To achieve this it has to select and execute one of the visual level action sequences stored on the arc between the current node and the desired node of the learning graph. The parameters of this sequence must match the current situation as close as possible. To do so the agent creates the list of parameters it can currently observe and passes this list to a classifier (currently, a nearest neighbor classifier [8]). The later returns the best matching sequence and the agent executes each of its actions. The same procedure continues until the desired node is reached.

5 Experiments on Learning Believable Movement

During 10 sessions we have trained an autonomous agent to believably act in the institution from Fig. 1. We started recording the actions of the human playing the “guest” role in the garden, facing the avatar towards different objects and having the receptionist agent located in various positions. In each training session the trajectory was easily distinguishable given the observed parameters.

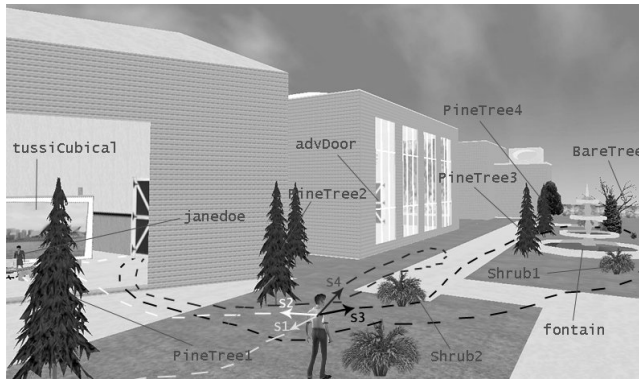


Fig. 3 Training the agent in the garden.

Fig. 3 gives an impression of how the training was conducted. It shows fragments of the 4 different trajectories (displayed as dotted lines - $S_1...S_4$) generated by the “guest” avatar. The arrows marked with $S_1...S_4$ correspond to the direction of view of the avatar at the moment when the recording was initiated. The location of the receptionist agent, it's role and position together with the objects located in the en-

vironment and distances to them were the parameters used for learning. In Fig. 3 the objects include “Pinetree”, “tussiCubical”, “advDoor” and the receptionist, showed in the figure as “janedoe”.

Table 1 Parameters used in the training session.

| parameterID | parameterName | possibleValues |
|-------------|-----------------|--------------------|
| p_1 | janedoeRole | {recep,null,guest} |
| p_2 | DISTadvDoor | numeric |
| p_3 | DISTPineTree4 | numeric |
| p_4 | DISTBareTree | numeric |
| p_5 | SEefountain | {y,n} |
| p_6 | DISTussiCubical | numeric |
| p_7 | DISTPineTree3 | numeric |
| p_8 | DISTPineTree2 | numeric |
| p_9 | DISTPineTree1 | numeric |
| p_{10} | SEEtussiCubical | {y,n} |
| p_{11} | SEEBareTree | {y,n} |
| p_{12} | SEEadvDoor | {y,n} |
| p_{13} | SEEsShrub3 | {y,n} |
| p_{14} | SEEsShrub2 | {y,n} |
| p_{15} | DISTfontain | numeric |
| p_{16} | SEEsShrub1 | {y,n} |
| p_{17} | DISTshrub3 | numeric |
| p_{18} | SEEPineTree4 | {y,n} |
| p_{19} | DISTshrub2 | numeric |
| p_{20} | SEEPineTree3 | {y,n} |
| p_{21} | DISTshrub1 | numeric |
| p_{22} | SEEPineTree2 | {y,n} |
| p_{23} | SEEPineTree1 | {y,n} |
| p_{24} | SEEjanedoe | {y,n} |
| p_{25} | DISTjanedoe | numeric |

The agent has been trained to enter the Meeting Room. The resultant learning graph was similar to the one in Fig. 2. Table 1 presents the list of all parameters stored in the graph on the arc between “root” and “EnterInstitution(SimpleInstitution)” nodes. Parameters, having names beginning with: (i) “SEE” correspond to the objects or avatars that were appearing in the field of view of the user at the moment of recording; (ii) “DIST” correspond to the distance measure between the user and the center of mass of a visible object. The distance to objects not visible is equal to zero. Parameter “janedoeRole” defines the role of the receptionist agent “janedoe”. When the receptionist was not visible in the field of view of the trained guest agent, the values of janedoeRole is “null”. When the “jandoe” was located outside the institution it’s role was “guest” and inside the registration room it was “receptionist”.

Table 2 A fragment of the data used in the training session.

| Nr | p_1 | p_2 | p_3 | p_4 | p_5 | p_6 | p_7 | p_8 | p_9 | p_{10} | p_{11} | p_{12} | p_{13} | p_{14} | p_{15} | p_{16} | p_{17} | p_{18} | p_{19} | p_{20} | p_{21} | p_{22} | p_{23} | p_{24} | p_{25} | S | | |
|----|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-------|-------|
| 1 | null | 0 | 0 | 0 | n | 0 | 0 | 31 | n | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | 0 | n | y | n | 0 | S_1 | |
| 2 | recep | 0 | 0 | 0 | n | 55 | 0 | 0 | 39 | y | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | 0 | n | y | y | 67 | S_2 |
| 3 | null | 0 | 0 | 0 | y | 0 | 61 | 0 | 0 | n | n | n | n | n | 96 | y | 0 | n | 0 | y | 70 | n | n | n | n | 0 | S_3 | |
| 4 | null | 74 | 0 | 0 | y | 0 | 61 | 0 | 0 | n | n | y | n | n | 95 | y | 0 | n | 0 | y | 67 | n | n | n | n | 0 | S_4 | |
| 5 | recep | 0 | 0 | 0 | n | 56 | 0 | 0 | 0 | y | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | 0 | n | n | y | 68 | S_5 |
| 6 | null | 0 | 0 | 0 | n | 0 | 0 | 43 | 77 | n | n | n | n | y | 0 | n | 0 | n | 48 | n | 0 | y | y | n | n | 0 | S_6 | |
| 7 | guest | 0 | 0 | 0 | y | 0 | 0 | 0 | 0 | n | n | n | n | n | 96 | y | 0 | n | 0 | n | 69 | n | n | y | 24 | S_7 | | |
| 8 | null | 0 | 0 | 24 | n | 0 | 0 | 0 | 0 | n | y | n | y | n | 0 | n | 42 | n | 0 | n | 0 | n | n | n | n | 0 | S_8 | |
| 9 | null | 0 | 0 | 0 | n | 0 | 0 | 0 | 0 | n | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | 0 | n | n | n | 0 | S_9 |
| 10 | guest | 0 | 41 | 0 | y | 0 | 96 | 0 | 0 | n | n | n | n | n | 69 | n | 0 | y | 0 | y | 0 | n | n | y | 38 | S_{10} | | |

Table 2 presents the training data stored on the arc between “root” and “EnterInstitution(SimpleInstitution)” during 10 recording sessions along the parameters, listed in Table 1. The “S” column shows the acronyms of the sequences of actions of the visual level of execution. The first four sequences correspond to the trajectories $S_1 \dots S_4$ outlined in Fig. 3.

Each of the tests was conducted as follows. Two operators entered the Virtual World by two different avatars: “janedoe” (driven by one of the researchers in our lab), playing the “receptionist” or “guest” role and “agent0” (controlled by an independent observer), always playing the “guest” role. Both avatars were positioned in various locations and the avatar “agent0” was facing a selected direction (with janedoe either visible or not). On the next step agent0 was instructed to leave the garden, enter the institution, walk into the registration room, exit it and then walk through the next transition to the Meeting Room. The agent then looked for the right sequence of the institutional level actions, which in the given case were: EnterInstitution(SimpleInstitution), EnterScene(root), ExitScene(root), EnterTransition(rootToRegistration), ExitTransition(rootToRegistration), EnterScene(Registration), Login(test, test), ExitScene(Registration), EnterTransition(toMeeting), ExitTransition(toMeeting), EnterScene(Meeting). To execute those actions the agent needed to launch the appropriate sequence of the visual level actions, stored on the arcs of the learning graph. The classifier was given the list of currently observed parameters as the input and as the output it returned the sequence that was supposed to fit best. After completion of recording, we conducted a series of 20 tests to check whether the trained agent would act in a believable manner. Table 3 presents the experiments’ results.

Table 3 Classifier performance: input data and recommendations.

| Nr | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 | P17 | P18 | P19 | P20 | P21 | P22 | P23 | P24 | P25 | S | |
|----|-------|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----------------|----------------|
| 11 | guest | 0 | 0 | 0 | n | 0 | 0 | 0 | 0 | n | n | n | n | y | 0 | y | 0 | n | 40 | n | 91 | n | n | y | 36 | S ₇ | |
| 12 | null | 0 | 0 | 0 | n | 59 | 0 | 0 | 36 | y | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | y | n | 0 | S ₁ | |
| 13 | guest | 0 | 36 | 0 | y | 0 | 91 | 0 | 0 | n | n | n | n | n | 75 | y | 0 | y | 0 | y | 63 | n | n | y | 17 | S ₁₀ | |
| 14 | null | 0 | 0 | 0 | y | 0 | 0 | 0 | 0 | n | n | n | n | n | 36 | y | 0 | n | 0 | n | 82 | n | n | n | 0 | S ₂ | |
| 15 | null | 0 | 0 | 77 | y | 0 | 0 | 0 | 0 | n | y | n | y | n | 30 | n | 94 | n | 0 | n | 0 | n | n | n | 0 | S ₈ | |
| 16 | guest | 0 | 0 | 0 | n | 0 | 0 | 0 | 0 | n | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | n | n | y | 25 | S ₉ |
| 17 | guest | 0 | 0 | 86 | n | 0 | 0 | 0 | 0 | n | y | n | y | n | 0 | n | 77 | n | 0 | n | 0 | n | n | y | 34 | S ₈ | |
| 18 | null | 65 | 0 | 0 | y | 0 | 51 | 0 | 0 | n | n | y | n | n | 87 | n | 0 | n | 0 | y | 0 | n | n | n | 0 | S ₄ | |
| 19 | recep | 0 | 0 | 0 | n | 41 | 0 | 0 | 0 | y | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | n | y | 51 | S ₅ | |
| 20 | recep | 0 | 0 | 0 | n | 72 | 0 | 41 | 40 | y | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | y | y | y | 78 | S ₇ | |
| 21 | guest | 0 | 0 | 0 | n | 0 | 0 | 63 | 95 | n | n | n | n | y | 0 | n | 0 | n | 42 | n | 0 | y | y | y | 18 | S ₆ | |
| 22 | null | 0 | 0 | 0 | y | 0 | 18 | 0 | 0 | n | n | n | n | n | 50 | y | 0 | n | 0 | y | 99 | n | n | n | 0 | S ₃ | |
| 23 | null | 0 | 0 | 0 | n | 0 | 0 | 0 | 0 | n | n | n | n | n | 0 | n | 0 | n | 0 | n | 0 | n | n | n | 0 | S ₉ | |
| 24 | guest | 0 | 95 | 0 | y | 0 | 0 | 0 | 0 | n | n | n | n | n | 71 | y | 0 | y | 0 | n | 41 | n | n | y | 14 | S ₇ | |
| 25 | guest | 0 | 0 | 0 | y | 0 | 26 | 0 | 0 | n | n | n | n | y | 59 | y | 0 | n | 82 | y | 84 | n | n | y | 71 | S ₇ | |
| 26 | null | 0 | 0 | 0 | y | 0 | 26 | 0 | 0 | n | n | n | n | y | 59 | y | 0 | n | 82 | y | 84 | n | n | n | 0 | S ₃ | |
| 27 | null | 0 | 0 | 43 | n | 0 | 0 | 0 | 0 | n | y | n | y | n | 0 | n | 60 | n | 0 | n | 0 | n | n | n | 0 | S ₈ | |
| 28 | null | 0 | 45 | 0 | y | 0 | 96 | 0 | 0 | n | n | n | n | n | 58 | n | 0 | y | 0 | y | 0 | n | n | n | 0 | S ₁₀ | |
| 29 | null | 0 | 0 | 0 | n | 0 | 0 | 0 | 0 | n | n | n | y | n | 0 | n | 63 | n | 0 | n | 0 | n | n | n | 0 | S ₉ | |
| 30 | guest | 0 | 0 | 0 | n | 0 | 0 | 0 | 0 | n | n | n | y | n | 0 | n | 63 | n | 0 | n | 0 | n | n | y | 14 | S ₉ | |

Fig. 4 shows the eye direction of the guest and the positions of both avatars. Solid dots marked with the number of experiment in the figure correspond to the positions of the guest. The arrows represent guest’s eye direction. The female figure marked with the experiment number shows the positions of the receptionist (when it was visible to the guest). The experiment numbers in Fig. 4 correspond to the ones specified in the “Nr” columns in Table 2 and Table 3. The numbers 1-10 are the initial recordings and 11-30 represent the conducted experiments. The “S” column in Table 3 outlines the acronyms of the action sequences (as used in Table 2) executed by the agent as a result of classifier’s recommendation.

In every test the believability of the movement was assessed by an independent observer. In all cases it was evaluated as believable.

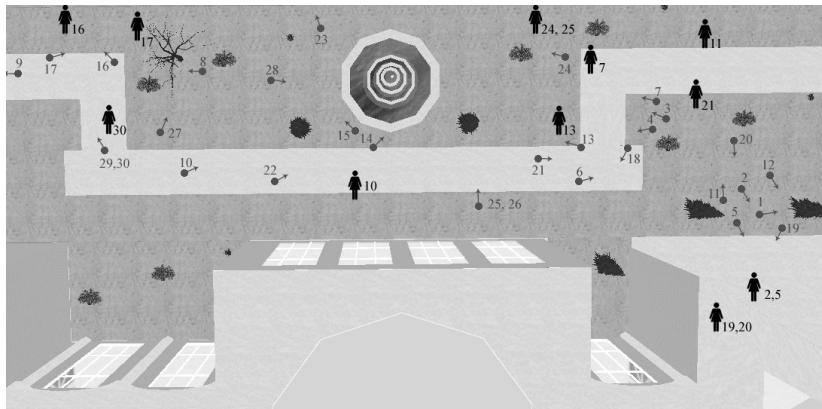


Fig. 4 Design of the experiments in the institution space.

6 Conclusion

We have presented the concept of implicit training used for teaching human behavioral characteristics to autonomous agents in Virtual Institutions. The developed prototype and conducted experiments confirmed the workability of the selected learning method and the validity of the implicit training concept.

Acknowledgements. This research is partially supported by an ARC Discovery Grant DP0879789 and the e-Markets Research Program (<http://e-markets.org.au>).

References

1. Loyall, A.B.: Believable agents: building interactive personalities. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA (1997)
2. Magnenat-Thalmann, N., Kim, H., Egges, A., Garchery, S.: Believability and Interaction in Virtual Worlds. In: Proceedings of the Multimedia Modeling Conference (MMM'05), Washington, DC, USA, IEEE Computer Society (2005) 2–9
3. Livingstone, D.: Turing's test and believable AI in games. *Computers in Entertainment* **4**(1) (2006) 6–18
4. Breazeal, C.: Imitation as social exchange between humans and robots. In: Proceedings of the AISB Symposium. (1999) 96–104
5. Gorman, B., Thureau, C., Bauckhage, C., Humphrys, M.: Believability Testing and Bayesian Imitation in Interactive Computer Games. In: Proceedings of SAB'06 conference. Volume LNAI 4095., Springer (2006) 655–666
6. Bogdanovych, A.: Virtual Institutions. PhD thesis, University of Technology, Sydney, Australia (2007)
7. Esteva, M.: Electronic Institutions: From Specification to Development. PhD thesis, Institut d'Investigació en Intel·ligència Artificial (IIIA), Spain (2003)
8. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification and regression. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems. Volume 8., The MIT Press (1996) 409–415