A High-Level Architecture for Believable Social Agents

A. Guye-Vuillème, D. Thalmann

Computer Graphics Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland

Abstract: The creation of virtual humans capable of behaving and interacting realistically with each other requires the development of autonomous believable social agents. Standard goal-oriented approaches are not well suited to it because they don't take into account important characteristics identified by the social sciences. This paper tackles the issue of a general social reasoning mechanism, discussing its basic functional requirements using a sociological perspective, and proposing a high-level architecture based on Roles, Norms, Values and Types.

Keywords: Socially believable agents; Socially intelligent virtual agent

Introduction

Most agent architectures are dedicated to the fulfillment of precise task/problem-solving, e.g. [1], or to the simulation of purely emotional behaviour, e.g. [2]. However, pure goal-oriented and emotional behaviour only correspond to a small part of our daily activities as human beings. Every day, we engage in many interactions and social activities, adapting our behaviour to the situation, dealing with complex motivations and different cultures, and also simply following routines. Our group belonging, gender or cultural origin are very important criteria to explain our behaviour.

These facts need to be taken into account in the development of virtual humans as socially intelligent virtual agents, capable of adapting to and interacting realistically with other agents in 3D environments. We argue that the next logical step in the development of such agents is to model their behaviour at a higher level, adding social identities, social motivations

and social reasoning to them, on top of their specific competency.

Unfortunately, there is no real agreement about what the word 'social' refers to, and thus on what a social agent should be and do. In the computer graphics field, the implicit definition of social behaviour is often restricted to crowd and group activities, e.g. [3]. Agent researchers usually link the concept with a communication task and define social ability as the capability of interacting with other agents or humans [4]. These definitions refer to important characteristics, but the concept has a broader meaning in sociology where all human behaviours are considered social since they are always the product of a social-cultural context, e.g. [5]. This means that behaviours of temporarily isolated humans are still social and therefore that the concept can be used to model the agent's global behaviour.

'Socially situated', 'socially intelligent' and 'socially adept' are terms used in the agent literature which apply well to our definition, but two other important characteristics of our social agent must be

mentioned: it is 'socially motivated', i.e. it acts according to its identity and position in society, and has to be 'socially believable' for an external observer. In our definition, social reasoning is a mechanism allowing a socially situated agent to analyse the situation with its own specific view, extracting the relevant information for 'right' decision-making in terms of social acceptance and personal values. In our view, the classic opposition in AI approaches between thinking/acting humanly and thinking/acting rationally is replaced by the goal of allowing agents to think/act in a humanly rational way.

The building of believable social agents is useful in applications such as interactive games and production animations to populate the scenes with lifelike synthetic actors, in collaborative virtual environments where autonomous agents would assist the participants in a graceful way, in the visual simulation of social phenomena, emergency situations, fire evacuations, etc.

This paper is organised as follows. The next section discusses the task of developing social agents, the existing work and our starting approach. In the third section we try to identify basic requirements for social agents and introduce useful concepts from the sociological literature. The fourth section presents the specifications of an architecture proposal, the fifth briefly describes the implementation and finally we provide a simple example of the produced behaviour.

Development Approaches

Designing believable social agents is a challenging task because the behaviours to be reproduced are very complex and require a multi-disciplinary approach. Social behaviours make intensive use of many human abilities, such as vision, hearing, form recognition, memory, subtle body control, etc. Related work includes 3D visualisation, human animation with articulated and deformable bodies, and sound propagation modelling, which have shown impressive progress recently. Nevertheless, reasoning is still a major issue and strong theoretical oppositions exist within the AI field. In this section we discuss the different existing approaches and explain our heuristic choices.

Related Work

The topic of artificial social intelligence is gaining importance in the AI and ALIFE fields. A fundamental work of identification of the key issues and clarification of the main concepts has been conducted by Carley and Newell [6], and more recently by Dautenhahn [7]. In the context of social simulation, Castelfranchi and Conte have produced interesting papers about such important topics as norms acceptance, cognitive emergence and agent rationality, e.g. [8].

The existing architectures often propose both psycho-social attitudes and goal-based or emotional behaviours. The OZ project [9] integrates goal directed and reactive behaviour, as well as emotions and social behaviour, in a global architecture. Rousseau and Hayes-Roth [10] propose a social-psychological model based on moods and attitudes. In these systems, which don't use a dedicated architecture, important characteristics of social behaviour (described later) are sometimes missing.

An opposite approach consists in modelling and implementing specific aspects of social interaction separately. Interesting work has been conducted by Sichman [11] on the use of dependance relations for social simulation, Rizzo et al. [12] have built goal-based personalities for social behaviour using a specific taxonomy, and such concepts as 'trust' have been modelled for social agents [13]. Unfortunately, these systems do not all provide a general-purpose social reasoning mechanism.

A connectionist approach has been used by Channon and Damper [14], who argue that evolutionary emergence is the best technique to obtain socially intelligent agents. They describe a multi-agent system using neural networks and genetic algorithms that generates basic social behaviours such as cooperation and competition. However, the resulting behaviours are still very simple, and the general specification of the behaviour is an issue in such an approach which doesn't rely on the experience of human designers (e.g. would the simulation of the behaviour of a Spanish passer-by, which is dependent on the cultural and historical context, require the replication of the evolution of Spanish civilisation?).

The cognitive approach is also used to animate virtual creatures. An interesting system based on cognitive modelling to direct any kind of autonomous characters has been developed by Funge in the field of behavioural animation [15]. Unfortunately, human cognitive mechanisms used in social interaction are extremely complex, as cognitive psychology has shown, and still little understood. Moreover, because of the complexity involved, the cognitive approach is subject to the criticisms that symbolic AI has received in the past

A. Guye-Vuillème, D. Thalmann

years, e.g. [16]. Still, the specification of how knowledge is acquired is a crucial issue that must be addressed in any advanced system.

Approach

The BDI architecture has proved to be a useful abstraction to model autonomous agents. Its components (Belief, Desire, Intention) offer a convenient and intuitive way to structure the agent's action selection. Systems built on this architecture have produced interesting results (e.g. dMARS [1]). Nonetheless, we agree with Balzer [17] when he points out that this architecture is deficient for the simulation of social behaviour. Based on the Bayesian decision theory, it does not take into account the socially situated nature of human behaviour nor allow the inclusion of such important phenomena as social constraint, moral standards, routine behaviours, etc. We feel that the conceptualisation of a high level architecture for social agents could greatly help in their development, in the same way as the BDI architecture does for standard autonomous agents.

Our approach consists of building such a high-level architecture, implemented as a layer on top of the agent's standard apparatus, driving its goals, beliefs and intentions in a socially realistic way. It should allow human designers to add specifications easily, and express them using intuitive concepts. The realistic simulation of contemporary human behaviour by using 3D autonomous agents is, as previously stated, our target application. Considering the hazards involved in developing a full cognitive apparatus, we decided to restrict the cognitive aspect of our work, emulating only the crucial mechanisms.

As will be shown later, our architecture is hybrid in nature: reactive mechanisms process the perceptual input to activate a subset of the agent's behavioural and cognitive data, then standard symbolic reasoning (rules matching) is performed on the lower-level information to complete the action-selection process.

It is also important to underline that sociology plays an important role in our approach. By using the sociological corpus which offers a high-level model of human behaviour, it is possible to implement the logic of the social actor without developing a whole cognitive apparatus, nor having to control the evolution of complex neural networks.

Requirements Analysis and Useful Concepts

Requirements

We will now focus on the functionalities that a social agent architecture should provide for the resulting social behaviour to show a sufficient level of realism. We present and develop our preceding work about this issue using the sociological literature [18]. Here are the functional requirements that we have identified.

Multi-behaviour architecture: Since the everyday activities of humans in society are multiple and diverse, the architecture should allow several distinct global behaviours to be assigned to one agent. These behaviours are not different ways to reach one common final goal as with problem-solving agents, but are self-sufficient sets of actions. They should be organised in a socially coherent way and facilitate reuse.

Behaviour switching mechanism: A general mechanism that is in charge of switching between these global behaviours should be provided. The local social context should be used to determine if a change is required. As pointed out by Sengers footnote, 'behaviours transitions' are important if the agent is interacting with a real user, to explain why it is changing its behaviour.

Acting according to the social environment: When assigned one of these global behaviours, the agent still has to make decisions about what to do. The architecture should allow it to make decisions not only according to its physical environment, but also to the social and cultural environment. The issue for the agent is not only about what to do, but also how to do it: e.g. should it greet this particular agent in a friendly or distant way? Thus, the agent must have specific knowledge about its social environment.

Basic cognitive mechanism: A well-known fact in sociology is that one does not act according to the objective situation, but on the basis of one's

Sengers P. Socially intelligent agent-building. In: Proceedings of AAAI-97 Workshop on Socially Intelligent Agents. Menlo Park CA: AAAI, 1997

'definition of the situation' [19]. Since this has a major impact on the resulting behaviour, a minimal cognitive apparatus should be provided. It should be able to generate different beliefs for different agents in the same situation.

Standard behaviours: In many situations, humans don't have to make decisions about what to do but act in a 'natural' way. This is the case with routine behaviours and norm-based behaviors which need to be handled by the architecture. Routine behaviours are learned and practised day after day, e.g. professional tasks. Social norms are socially and culturally situated standards for behaviour, which specify what must be done and what is forbidden in a given situation. They are important to provide in the form of actions that are automatically triggered/inhibited in a given situation.

Acting according to social values: The architecture should allow the agent's actions to be oriented to social values, and not only to the fulfillment of specific goals. This means that such criteria as 'what's right', 'what's wrong', 'what's cool', 'what's out', etc. are used to determine the behaviour independently of its usefulness. Actions should be rated according to individual values, which can contribute to activate or inhibit them.

These requirements do not cover all social competencies, but their fulfillment should provide a solid base for social behaviour. We are now going to discuss several social sciences concepts that can help provide content to these requirements.

Useful Concepts

The concepts presented here are among the most fundamental concepts of sociology. Before more advanced features are considered, we think that the basic requirements they involve should be addressed.

Role

The concept of Role has been developed by several important authors such as Goffman and Merton to account for the regularities in face-to-face interactions and to explain the relationship between the individuals' behaviours and social status. A role can be considered as a coherent set of standard behaviours, but also includes such elements as world view, which distinguish it from its dramaturgic equivalent. A role is linked to such variables as age or gender, professional status, etc. Moreover, one individual masters several roles and uses them successively, e.g. a woman who alternatively behaves as a secretary, as a mother or as a football fan.

The Role could fulfill the requirement for global behaviours which has been emphasised. It seems to be an ideal way of organising social agents' behaviours: instead of assigning behaviours directly to an agent, it would be very advantageous to build these behaviours into roles and then attribute them to the agent. We propose to use this concept of Role because it is a coherent and documented way to group behaviours, and greatly facilitates reuse. Since the concept is widely used in field studies, it should be possible to easily integrate specific role descriptions into the system. Finally, it is a scientific concept but also has the advantage of being intuitive enough to be easily used by designers who are not social scientists.

Since adequate role-taking is crucial for the realism of social interactions, proper directing criteria should be identified. The ethnomethodologist Cicourel [20] has described how each participant adapts their role to the other participants' during social face-to-face interaction. This could be an important observation for building the switching mechanism in charge of activating one role or another.

Norms

As previously stated, social norms are guides for behaviour. A norm is fulfilled because it has been fully integrated by the individual or to avoid sanctions. According to Balzer, social norms can be translated into rules: people constantly check if the situation fits some precise conditions and, if it does, apply the rule by behaving in a precise way, e.g. opening the door for a woman. Contrary to the Bayesian view, 'this check is not a decision, it is a process of fitting or of recognition' [17]. This feature makes the classic rule-based systems very useful for the development of social agents.

It makes sense to attach norms to roles, even if some of them are widely shared within a culture. One could, for example, build a 'Frenchman role' and an 'American role' containing basic behavioral norms (e.g. rules of courtesy). These roles would be assigned to the agents together with, for instance, a professional role that contains its own norms (e.g. dress properly: wear a tie).

 A. Guye-Vuillème, D. Thalmann

Values

Max Weber's famous typology introduces the orientation of the action to a value, along with the rational orientation to individual ends, the affectional orientation and the traditional orientation [21]. An action selected on the basis of a value is believed to be 'right' independently of its chances of success. Values involve the conscious pursuit of an ideal abstract state (e.g. 'I give it back, because I am honest') and also work as classifying criteria ('I don't like him because he's a crook'). Values constitute a personal moral system.

This value-oriented rationality has to be implemented because it is an important specificity of human beings, and is missing in all systems based on the principles of the Bayesian decision theory since it is independent from any kind of 'utility calculus'. As previously stated, actions can be 'rated' according to individual values, which can contribute to activate or inhibit them. Values can also be very useful to generate the agents' specific world views. Again, it could be very convenient to attach the values to roles: 'discipline' could be a strong value in a 'military role', and 'efficiency' may be an important classifying criterion to an agent with a 'manager role'.

World View

Identifying the 'world view' of groups and how it is created and maintained is an important task of microsociology, because it helps understand the actions of the individuals. Thomas's previously-mentioned idea of the 'definition of the situation' shows well that cognitive mechanisms act on the perception and create potentially very different interpretations of the world. Bourdieu's 'Habitus' concept describes an abstract entity that fulfills this task: it is a 'system of acquired dispositions functioning on the practical level as categories of perception and assessment or as classificatory principles as well as being the organizing principles of action' [22]. In a different approach, ethnomethodologists have also emphasised the role of categories of perception, by describing the 'typification' mechanism through which we organise our perception of others and of the world. The 'types' help give meaning to the world (e.g. 'she's a woman, she's certainly timid') but their definition is also constantly updated according to one's personal experience (eg. 'women may not be that timid after all').

Typification (horizontal) and classification (vertical) are certainly among the most important cognitive

mechanisms for social behaviour, and should be implemented in some way. There is no need to produce a complete world view by designing a complex ontology using the 'is-of-type' relationship, only the most important decision criteria for each role should be provided. Beliefs about other agents are particularly crucial (e.g. for a salesman: Is he a client? A colleague? The boss?). Stocking this information in a synthetic memory is even optional, which allows the use of a reactive architecture. The handling of self-view could be a useful feature for specific situations, such as dilemmas.

These concepts together form a specific view of human and human action known as the 'homo sociologicus' model. We believe that this view is well suited to the development of believable social agents, and this is why it is central to our work. Nevertheless, a goal for future developments may be to integrate more fully this view with the standard goal-oriented approach. Parsons's theory of action [23] could be a valuable source of inspiration, since it explains the individual's action through their own interests but also through integrated values, and combines the orientation of the action toward an end with norm-based behaviours. For instance, the individual can choose to fulfill a norm or not, depending on their own interests and evaluation of the consequences. In our approach, the individual is composed of a set of roles and their 'freedom', defined as their behaviour's unpredictability, emerges from the contradictions between these roles.

Specifications

We present here the specifications of the social agent architecture that we are implementing. The conception is directly inspired by the presented sociological model. The main prerequisite for this architecture is the use of an inference engine capable of forward chaining.

We first present the data structures and the high level organisation which is based on roles. Then we describe the core architecture, with its different modules and tasks.

Role-Based Organisation

The Role has the following features:

- it contains standard behaviours, values and types

- it is part of a role hierarchy and inherits data from its parent roles
- it is assigned to an agent with a specific weight, along with other roles
- it can be associated to other roles
- it is activated or inhibited by a switching mechanism.

Standard behaviours, values and types

Roles have the following structure (Fig. 1):

$Role_i = (Name_i, StandardBehavioursList_i, ValuesList_i, TypesList_i)$

Standard behaviours are competing plans, which specify a sequence of actions. They have the following structure:

$StandardBehaviour_i = (Name_i, PreconditionsList_i, ActionsList_i, PostconditionsList_i)$

Preconditions specify the conditions that must be met for the behaviour to be performed in the 'correct' situation. They are predicates expressed in terms of the agent's categorised beliefs about the world, e.g. 'PerceivedAgent.TypeName = Salesman', 'PerceivedAgent.Honesty<=0.5'. Actions can involve verbal and nonverbal communication, moving to a location, manipulation of objects, etc. Our standard actions catalog is mainly based on keyframe animation. Postconditions induce a change in the 'state of beliefs', e.g. lowering of the agent's honesty rating. By using empty actions lists and postconditions, standard reasoning can be achieved as in classic production rules systems.

Values are classifying criteria used to 'rate' behaviours and perceived entities. A role contains a list of weighted values:

$Value_i = (Name_i, Weight_i)$ where $0 <= Weight_i <= 1$

Additionally, a value score is assigned to standard behaviours:

$Score_{ij} = (Value_i, StandardBehaviour_j)$ where $-1 <= Score_{ij} <= 1$

For instance, 'Honesty' should be an important value for a 'policeman role': it is unlikely that a policeman will break the law and he clearly won't act the same way towards someone he thinks is a criminal and someone he does not. Thus, a value named 'honesty' is added to the role and receives a high weight – behaviours involving criminal activities will be poorly rated on this criterion – and the perceived agent's honesty score is used to write the preconditions

for the policeman actions.

Types are used to generate beliefs about the world, especially about other agents. Each role receives a list of types and a default value score is assigned to each type:

$DefaultScore_{ij} = (Value_i, Type_j)$ where $- <= DefaultScore_{ij} <= 1$

This means that a 'woman role' may use a 'man type' highly rated on 'strength' and lowly rated on 'compassion'. If the system is mainly reactive, the agent will directly react to the type itself or to the type's default scores. If the agent has a synthetic memory, the man's default scores are saved at the first interaction. Then, the type is not used any more for this perceived agent but its scores can be updated by postconditions.

Roles Hierarchy

Since there are specialised roles in real life and in order to make the incremental design of behaviours possible, roles are organised in an hierarchy (Fig. 2). Child roles can have multiple parent roles, and automatically inherit their parent's characteristics, as described later.

For instance, a 'saleswoman role' can be declared as a child of a 'woman role' and of a 'salesperson role'', but also has its own behaviours, values, etc.
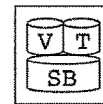

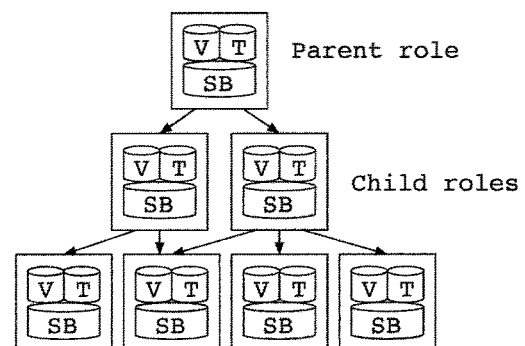**Fig. 1.** A role is represented with its values, types and standard behaviours.
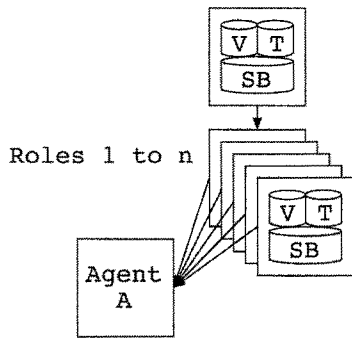

Fig. 2. Hierarchical organisation of roles.

A. Guye-Vuillème, D. Thalmann

Fig. 3. Multiple role assignment to an agent.

Fig. 4. Associated roles.

Multiple Role Assignment to an Agent

Multiple roles can be assigned to an agent (Fig. 3) so that rich behaviours can be produced. Each assigned role has a weight to allow the designers to specify their relative importance to the agent,s personality.

When roles are assigned, all their associated values are integrated in the agent using the roles weights to balance them. Behaviours and types are not directly integrated, but can be used when their corresponding role is active. For parent roles, values are directly integrated in the agent, but behaviours are only activated if the preconditions of no more specialised behaviours are met. This is explained by the fact that values are intimate beliefs that are dependent on the assigned roles but are relatively stable and don't vary when a short-term role switch is required.

Associated Roles

In the same way as roles can be hierarchically connected to parent roles, they can be associated to other roles, typically with a symmetrical function, as in Fig. 4.

Associated roles are, for instance, a 'mother role' inheriting from a 'woman role', and a 'father role' inheriting from a 'man role'. We have seen that the use of a role leads the interlocutor to adapt its own role: e.g. a woman talking as a mother to her husband will certainly expect him to react as a father. Associated roles are used for this task, to specify that a strong interactional link exists between roles.

Role-Switching Mechanism

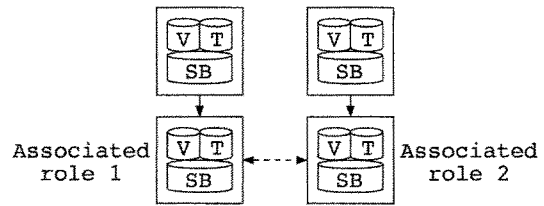Roles are activated and inhibited during execution by a switching mechanism, which allows the agent to adapt its behaviour to the situation. If a low priority is given to the active role, the agent is able to change its behaviour easily when the context requires it. If the active role has a high priority, the switching mechanism does not allow a change, and the agent keeps on focusing on its current activity. Activation energy is used to compute this priority. It comes from four sources:

● the initial role weight given by the designers
● the location: a role can be designed to receive extra activation energy when the agent is at given locations
● the time: a role can be designed to receive extra activation energy at certain times
● the interactional context: a role can receive extra activation energy when interlocutors have active associated roles

Here is the algorithm performed at every timestep:

1. The activation energy corresponding to the role weight is added; if necessary, the activation energies linked to the location, time and interactional context are added.

2. A decay function is executed so that the overall amount of activation energy remains the same.

3. The role with the highest level of activation energy becomes active.

The algorithm works so that the agent slowly recovers its default role priorities after having been able to respond to specific situations. The location and time criteria are useful to simulate roles linked to such scenarios as an occupation at an office or a shop, people going out at night, etc. This switching mechanism should make possible the following scene: a man in a shop, after behaving as a classic client, begins to flirt with the saleswoman. She is able to switch from her sale role temporarily and reacts using her 'woman role' behaviors, before returning to her 'saleswoman role' with other clients.

This role-based organisation allows the emergence of rich and surprising behaviours as the number of assigned roles grows, and multiplies interactions between conflicting values and norms.

Agent Architecture

The goal of the agent architecture presented here allows an agent situated in a virtual environment to perform social reasoning. In order to achieve this goal, the classic stimulus/response scheme has to be enriched and is, in our approach, replaced by a process involving perception, cognition, reasoning and action, as represented in Fig. 5.

Here are the logical modules active at every step of the process:

Perception module: This module is in charge of generating a list of the agents and static objects that are perceived by the agent in the scene. Working as a synthetic vision, it filters the objects using vision angle and range parameters. It is also in charge of decoding the messages sent by other agents using a dedicated ACL, which contains verbal communication or information about the other agent's state.

Cognitive module: The cognitive module obtains data from the perception module and encodes it using the active roles types. Beliefs about other agents are especially important to generate. Ideally, both the perceived entity and its current action are encoded, which allows the designers to create more realistic conversational agents. When an entity is recognised, its type and its type's default scores are added to the beliefs database.

Type identifiers: In order to assist the cognitive module in linking entity characteristics to types, dedicated type identifier modules can be provided. A general type identifier mechanism which simply links a list of entity IDs to a type is necessary, but a sophisticated processing would give more interesting results. To determine an entity type by processing its known characteristics, a set of rules could be run in the inference engine, as in expert systems. Neural networks are also ideally suited to this task, owing to their ability to identify patterns in the data. For instance, perceptrons could be trained to recognise the agents' 'gender' using their sizes and garments.

Social reasoner: This module is mainly composed of the inference engine. It processes the behaviours' preconditions using the beliefs generated by the cognitive module, and calculates a global score for each behaviour using the values and their weight. All the behaviours whose score does not surpass a certain threshold are removed from the list of possible behaviours. If several behaviours meet the necessary conditions, the one with the highest score is chosen. As in real life, conflicting roles may produce surprising behaviours or passivity. Finally the corresponding actions list is sent to the actions manager, the postconditions are computed and the beliefs updated.
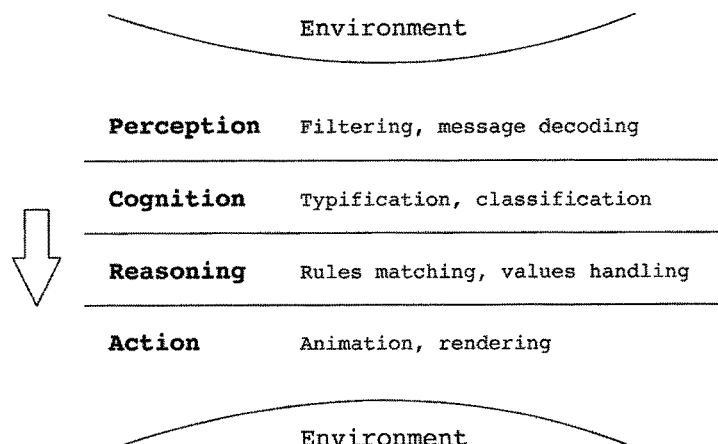
Environment

Perception	Filtering, message decoding
Cognition	Typification, classification
Reasoning	Rules matching, values handling
Action	Animation, rendering

Environment

Fig. 5. Behavioural process.

A. Guye-Vuillème, D. Thalmann

Actions manager: This module collects the adequate action scripts and runs them successively. Tasks such as keyframe animation of deformable bodies, speech synthesis and rendering are performed. It is also in charge of stopping the ongoing actions when a role switch is detected and of queuing the new actions received from the same role.

Similarly to BDI agents, systems based on this architecture require a parametrisation effort. As previously discussed, there is no way to avoid it in the simulation of complex and culturally situated human behaviours. Moreover, it should be compensated for by the incremental nature of the approach which greatly facilitates reuse.

Implementation Overview

Our implementation takes advantage of the following platforms:

- ANSI C++ and the Standard Template Library (STL)
- the Allegro CL 5.0 Lisp implementation
- Python language v1.5.2.

C++ is used for animation and rendering but also for perception, since it is efficient to handle and filter objects pipelines [24]. A class hierarchy is being developed to handle all the necessary data, providing a convenient interface for their manipulation.

The Lisp environment is well suited to rules matching and encapsulates our inference engine footnote.

We are also using a Python interpreter working as a high level interface to our graphics libraries, which allows us to easily run behaviour scripts without compilation. This interpreter has required the development of an AgentLib++ Python

---

Caicedo A, Thalmann D. Intelligent decision-making for virtual humanoids. Workshop of ECAL'99. Lausanne EPEL. 1999
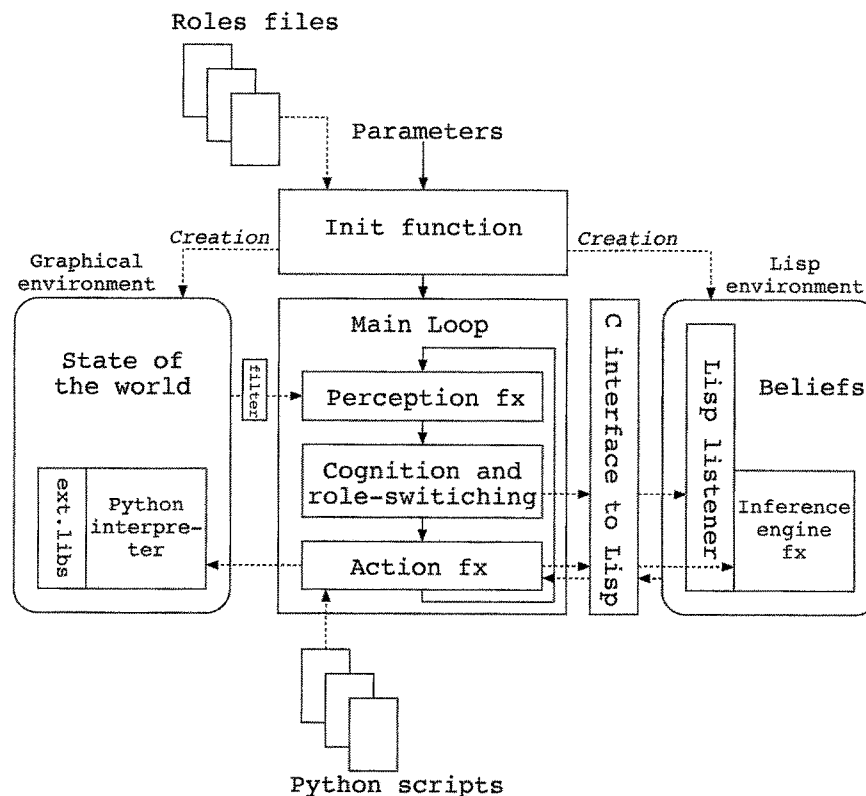

Fig. 6. Software architecture.

wrapper which links Python to C/C++ libraries. Our standard actions catalog is mainly based on keyframe animation, with actions such as walking, performing gestures, adopting postures, etc.

Figure 6 shows the application's general software architecture, which can be described through a sequence of actions of which the application is in charge. As it is run, the application receives parameters for a list of agents to create, along with a list of role files and their corresponding weights. It first initialises the Lisp environment, the graphical environment (including the scene), and creates a window displaying the scene. It also creates the necessary C++ objects (Agents, Roles, etc.) and integrates all the required data, such as values, in the agent objects according to the roles specific weights. Then it enters the C++ main loop and successively calls the perceptive, cognitive and action methods of every declared agent. The perception method filters the environment and returns a list of perceived entities with their known characteristics. When cognition is performed, beliefs generated about these entities are sent to the Lisp environment through C library calls. Role switching may occur during this phase if necessary. Finally, forward chaining is run in the inference engine, which tests all the norms of the active role, and the script corresponding to the best possible option is sent to the Python interpreter. The actions are performed and rendered before a new perception/cognition/reasoning process is run.

Example

Here is a simple scenario illustrating action selection and basic adaptive behaviour in our current implementation: our social agent Elvis wanders around and tries to appropriately greet the other agents he meets in the scene.

In Fig. 7, the presence of nearby agent has been detected by the perception module. The perception loop is suspended and the cognitive module tries to identify the encountered agent's type. A lisp-based gender identifier using the height and garments is dynamically loaded and run.

A new belief stating that the encountered agent is a male has now been generated in Fig. 8. Rules matching is then performed, and a specialised greeting for other men is found in the current active role ('male'). The corresponding Python script is run.
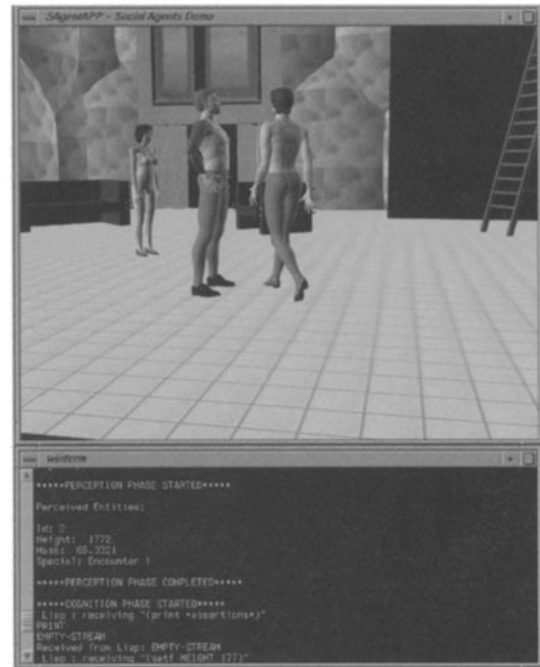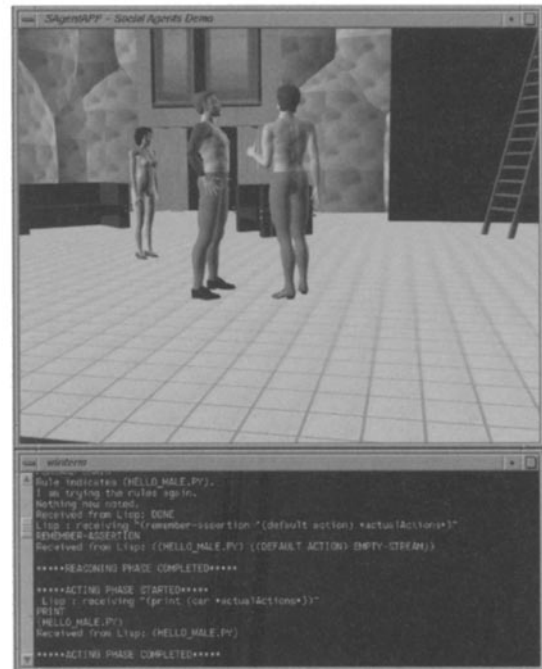

Fig. 7. Another agent is perceived.

Fig. 8. A male agent is greeted.

In Fig. 9, a female agent is met. The 'male' role defines a different way to greet women, which is chosen and performed.

A. Guye-Vuillème, D. Thalmann

Fig. 9. A female agent is greeted.

Fig. 10. A neutral greeting is performed.

Finally, a third agent is encountered (Fig. 10). Owing to her height and garments, her type is not

recognised and preconditions are not fulfilled for any behaviour in the 'male' role. However, since the 'male' role inherits behaviours from a general human role, a standard action can still be chosen and a neutral greeting is performed.

Conclusion

In this paper, we have underlined some of the most important mechanisms of social behaviour that we think are crucial to the creation of realistic virtual humans, and have proposed an architecture allowing their replication in a flexible way. We hope that the introduced concepts can prove useful for the simulation of human behaviour, especially the role hierarchy and the perception type that can make the agents more adaptable to new situations by letting the designers specify their behaviours at a higher level.

The heuristic goal of creating truly believable social agents is very challenging and can be compared to the task of developing agents capable of passing a 3D Turing test, involving both verbal and nonverbal communication. We hope that our work and ideas based on the sociological corpus will help approach this goal.

References

1. Ingrand FF, Georgeff MP, Rao AS. An architecture for real-time reasoning and system control. IEEE Expert/ Intelligent Systems 1992; 7: 34–44

2. Botelho LM, Coelho H. Artificial autonomous agents with artificial emotions. In: Proceedings of the Second International Conference on Autonomous Agents. New York: ACM Press, 1998, 449–450

3. Bouvier E, Guilloteau P. Crowd simulation in immersive space management. In: Proceedings of the Third Eurographics Workshop on Virtual Environments. Wien Springer, 1996, 104–110

4. Wooldridge M, Jennings N. Intelligent agents: theory and practice. Knowledge Engineering Review 1995; 10: 115–152

5. Berger PL, Luckmann T. The social construction of reality: a treatise in the sociology of knowledge. New York: Anchor Books, 1966

6. Carley KM, Newell A. The nature of the social agent. Journal of Mathematical Sociology 1994; 19: 221–262

7. Dautenhahn K. The art of designing socially intelligent agents: science, fiction, and the human in the loop. Applied Artificial Intelligence 1998; 12 : 573–617

8. Conte R, Castelfranchi C, Dignum F. Autonomous norm acceptance. Intelligent Agents V 1999; 1555: 99–112

9. Bates J, Loyall AB, Reilly WS. An architecture for action, emotion, and social behavior. In: Artificial social systems. Fourth European Workshop on Modelling Autonomous Agents in a Multi-agent World. Berlin: Springer, 1992, 55–68

10. Rousseau D, Hayes-Roth B. A social-psychological model for synthetic actors. In: Proceedings of the Second International Conference on Autonomous Agents. New York: ACM Press, 1998, 165–172

11. Sichman JS. Du raisonnement social chez les agents. Institut National Polytechnique de Grenoble: Grenoble 1995

12. Rizzo P, Veloso M, Miceli M, Cesta A. Goal-based personalities and social behaviors in believable agents. Applied Artificial Intelligence 1999; 13: 239–273

13. Marsh S. Formalising trust as a computational concept. University of Stirling: Stirling 1994

14. Channon A, Damper R. The evolutionary emergence of socially intelligent agents. Technical report. Zürich: University of Zürich, 1998

15. Funge J, Tu X, Terzopoulos D. Cognitive modeling: knowledge, reasoning and planning for intelligent characters. In : Proceedings of SIGGRAPH '99, 29–38

16. Brooks RA. Intelligence without representation. Artificial Intelligence 1991; 47: 139–159

17. Balzer W. Multi-agent systems for social simulation and bdi-architecture: a critical discussion. http://www.uni-koblenz.de/~kgt/Dag90717/Balzer.html 1997

18. Guye-Vuillème A, Thalmann D. Requirements for an architecture for believable social agents. In: Proceedings of the Fourth International Conference on Autonomous Agents. New York: ACM Press, 2000, 48–49

19. Thomas WI. The unadjusted girl. Boston: Little, Brown, and Co, 1923

20. Cicourel AV. Cognitive sociology: language and meaning in social interaction. Harmondsworth: Penguin, 1973

21. Weber M. Economy and society. New York: Bedminister Press, 1968

22. Bourdieu P. In other words: essays towards a reflexive sociology. Stanford CA: Stanford University Press, 1990

23. Parsons T, Shils EA (eds). Toward a general theory of action. : New York: Harper and Row, 1951

24. Bordeux C. Boulic R, Thalmann D. An efficient and fexible perception pipeline for autonomous agents. Computer Graphics Forum 1999; 18(3); 23–30

Correspondence and offprint requests to: *A. Guye-Vuillème, Computer Graphics Laboratory, Swiss Federal Institute of Technology, CH1015 Lausanne, Switzerland. Email: {aguye, thalmann}@lig.di.epfl.ch*

A. Guye-Vuillème, D. Thalmann