Intelligent Virtual Environments - A State-of-the-art Report

Ruth Aylett and Marc Cavazza
University of Salford, CVE, Salford; University of Teesside, School of Computing and Mathematics, Middlesbrough,
M5 4WT UK; TS1 3BA; United Kingdom
r.s.aylett@salford.ac.uk, m.o.cavazza@tees.ac.uk

Abstract
The paper reviews the intersection of AI and VEs. It considers the use of AI as a component of a VE and Intelligent Virtual Agents as a major application area, covering movement, sensing, behaviour and control architectures. It surveys work on emotion and natural language interaction, and considers interactive narrative as a case-study. It concludes by assessing the strengths and weaknesses of the current state-of-the-art and what might take it forward.

Keywords: Artificial intelligence, ALife, virtual agents

1. Why and how VEs and AI/Alife technologies are converging

In this report we consider a new field we call Intelligent Virtual Environments, formed by the increasing overlap of the technologies involved in 3D real-time interactive graphics environments and Artificial Intelligence/Artificial Life technologies.

The impetus for this field comes from a number of directions. Firstly, as the amount of processing power available for rendering has increased, it has now become feasible to devote a little of it to aspects of a VE beyond visual realism. However visually appealing a VE is, if it is static and empty of change and behaviour, the immersive experience is of limited interest. Cities without people cannot make the user feel a real sense of presence. While animation can create dynamic interest, its pre-scripted nature runs against the interactional freedom a VE gives users and can only hold their interest for a limited time.

Alongside the availability of processing power, and the desire to make VEs more dynamic and interesting, has come their application to industrial domains in which the direct interaction metaphor of immersion runs out of steam. For example, a model of the RAF Tornado, created in the late 1990s by the company Virtual Presence, has thousands of components, far too many for the realistic visualisation that has been developed to be all a user needs in understanding what each does or how particular tasks need to be carried out.

The growth of e-commerce on the web is producing a similar requirement for user support. While 3D content is still rare, it is clear that users require intelligent assistance in many cases, and 2D interface agents are beginning to give way to 3D 'Talking Heads', which are also being extended to new media such as interactive digital television. In the same way, large-scale distributed chat environments are moving to 3D graphics and finding the need for intelligence to support their user avatar populations. In turn this technology is being applied to long-standing research in Computer-Supported Cooperative Work.

The growth of processing power has also affected the computer games industry, making it much more difficult to compete on visual realism alone. As a result the incorporation of AI technology into computer games – albeit in ad hoc and piecemeal fashion – is now widespread, with Creatures and The Sims examples of new genres entirely dependent on the use of AI and ALife. Given the impetus that computer games have given to the development of 3D graphics in general, it will be interesting to see if the industry has the same accelerating effect on the combination of AI and ALife with 3D graphics.

The incorporation of virtual humans in particular also opens the way to the use of VEs for applications which up to now have used 2D simulations. Evacuation of buildings in emergencies and crowd control in urban spaces are obvious examples, while traffic simulation is also moving towards 3D and intelligent virtual cars. The concept of Digital Biota – 3D graphically based ecology – is also under active investigation. ALife technology is also being applied to VEs by artists, from the online ecology of TechnoSphere to the genetic algorithm- generated Feeping Creatures.

On the side of AI and ALife, groups of researchers are coming to recognise VEs as a powerful testbed for their technologies. Real-time interaction in a visually compelling virtual world is both more motivating than the text-based interaction of earlier research and – by engaging the user's senses – also more supportive, grounding natural language interaction, for example, in a shared visual context. Intelligent tuition systems can move into multi-modal interaction and incorporate an embodied tutor. Work on narrative and story-telling is also able to extend from the novelistic and purely text-based into drama and virtual theatre.

VEs also allow experimentation with complex embodied agents without all the problems of dry joints, sticking wheels and limited battery time that often frustrate researchers in robotics. With less effort required to develop basic control architectures – often adapted directly from robotics – it becomes possible with intelligent virtual agents, or synthetic characters, to investigate the modelling of emotion and the basis for agent social behaviour. This work can be carried out at both the behavioural level, incorporating body language and gesture, and at the cognitive level, using planning and emotionally-based inferencing.

AI technologies are also being used in some cases to deal with complexity or processing overhead. Once physics is incorporated into a VE, the overhead of solving large sets of equations by analytical methods may severely impact the rendering cycle. The use of neural nets can allow the functional mappings required to be learned off-line and effectively held in a dynamic lookup table. In the same way, synthetic characters with complex skeletal structures can use AI learning technology to acquire relevant movement abilities where explicitly programming them would be a difficult task.

In discussing these areas in the report that follows, one should note that though the convergence we have just discussed is real, nevertheless researchers from each of the main communities involved had different starting positions and, still, rather different perspectives. Thus the state-of-the-art is far from homogeneous or universally agreed.

2. Intelligent Virtual Reality Systems

The term of Intelligent Virtual Environments, as this review illustrates, encompasses a broad range of topics and research areas. However, systems that aim at integrating Artificial Intelligence (AI) techniques into the virtual environment itself constitute a specific kind of application, that we propose to name Intelligent Virtual Reality Systems (IVRS).

In these systems, intelligence is embedded in the system architecture itself, by incorporating AI algorithms into the virtual reality system.

The inclusion of an AI layer in a virtual environment can be justified from several perspectives:

- adding a problem-solving component to the virtual environment, for instance in configuration, scheduling or interactive design applications
- building a knowledge level supporting conceptual scene representation, which can support high-level processing of the graphic scene itself, or interface

with natural language processing systems

- describing causal behaviours in the virtual environment, as an alternative to physical simulation
- enhancing interactivity, i.e. by recognising user interaction in terms of high-level actions to determine adaptive behaviour from the system

Though IVRS have been in development for the past few years, they are still to be considered an emerging technology and many research problems remain to be solved.

One key aspect from the user-centred perspective is that the current AI techniques are often much less interactive than would be required for a complete integration into virtual environments, due to the difficulty of developing real-time (and/or reactive) AI techniques. In other words, the Holy Grail of IVRS research would be to have anytime AI techniques whose result production granularity would be compatible with the sampling rate, not so much of the visualisation, but of the user interaction with the virtual world objects, which itself depends on the application.

To date, very few IVRS systems have been described. The first one is a part of the Oz programming system into the DIVE VR software [1]. Though Oz supports constraint programming, the implementation reported is used as a generic high-level programming language for system behaviour rather than for embedding problem-solving abilities into the virtual environment. More recently, Codognet [20] has developed a generic constraint package, VRCC, which is fully integrated into VRML and used to define high-level behaviours for agents through the specification of constraints. Both systems rely on Constraint Logic Programming (CLP).

Current research has put forward CLP as a good candidate technique to support IVRS, essentially for two reasons: it computes solution fast enough to fit the interaction loop and it can accommodate incremental solution. Even though recent implementations of CLP can be used in dynamic environments, CLP is not, strictly speaking, a reactive technique. However, they permit fast solutions even with large-scale problems. This makes it possible in many cases to have a response time which matches the user interaction loop. As a knowledge representation formalism, CLP systems are only moderately expressive, being based on finite domains, though they retains the declarative aspect of other formalisms. As a result, knowledge acquisition may be more tedious than with rule-based systems. Fortunately, the type of knowledge to be represented in IVRS often includes a strong spatial component, which can be captured by the constraint formalism. Knowledge representation using CLP in IVRS is however only just emerging as a research topic, and significant advances in the near future should not be ruled out.

Other AI approaches are possible, such as heuristic repair and local search, which are alternatives to the basic mechanisms used in constraint programming, such as arc-consistency. For instance, heuristic repair works on non-consistent allocation of variables by “repairing” the variable that causes the greatest conflict. This could be exploited in IVRS in the following fashion: when a pre-existing solution is disrupted by acting on a single object, heuristic repair is a good candidate solution and can provide a quick solution. This new direction has been suggested recently by Codognet [21].

IVRS applications integrate real-time problem solving algorithms into VE. They rely on the close integration between the natural interactivity of the VE in terms of user centered visualisation and object manipulation, and the interactive aspects of the problem solving AI algorithms. For those applications in which there is an isomorphism between the spatial layout and the problem space, like configuration problems, the VE can actually be considered as a visual interface to the AI system. This is most often the case when the constraints to be satisfied by object configuration have a major spatial component.

We have ourselves developed an IVRS using GNU Prolog (which includes finite domain constraints) and the game engine Unreal Tournament [13]. The system is an interactive configuration system to be used in interactive building design. Constraints on the placement of building elements are described using finite domain constraints and a constraint solver written in GNU

Prolog produces a solution in terms of object positions. This solution is immediately displayed in the virtual environments, i.e. 3D objects are created matching the solutions. As the objects are part of an interactive environment, they can be manipulated by the user who can displace them to explore visually new configurations and design solutions. User interaction will change the nature of the data and disrupt some of the constraints. The new configuration created by the user should directly be analysed by the system to produce a new solution. The new position information is passed in real-time to the constraint solving programme via a TCP socket and the programme computes a new solution satisfying the new set of constraints.

As a result, as seen from the user perspective the interaction with some objects part of the solution configuration results in a new solution being computed and other objects being re-configured by the system to maintain a coherent solution.

This form of interactivity naturally relies on the interactivity of the solution itself. However, constraint programming in itself is not a reactive technique: it emulates reactivity because it can produce a solution quickly enough. The interaction cycle is determined by the speed at which new solutions are computed. In other words, the sampling rate of object manipulation in the virtual environment must be compatible with the result production granularity of the problem solving algorithm. We are currently experimenting with more complex configuration problems. As constraint programming has proven able to solve efficiently problems involving a large number of variable, it should be able to support an appropriate level of interactivity.

3. Virtual humans and synthetic characters

The biggest single area in which AI and VEs overlap is undoubtedly that of virtual humans and synthetic characters. Such characters need not be human - they could be abstract, [59], mechanical [52] or fictional like Creatures [27], Woggles [37] or Teletubbies [2]; they could be animals such as birds [53], fish [6], dolphins [40] or dogs [8]. As virtual humans, they might vary from the highly naturalistic Virtual Marilyn [63] to the physically stylistic but cognitively endowed pedagogical agent [54, 55].

From an AI perspective, these are embodied agents, and research in robotics is therefore highly relevant. However the graphics perspective starts from animated figures and often results in a slightly different emphasis. In this report we adopt the term *intelligent virtual agent* (IVA) to cover these entities. Note that the term *avatar*, which originally referred to the user's representation in a VE and therefore to a graphical representation with no autonomy and limited actuation capabilities, is currently sometimes used to refer to any humanoid agent in a VE. We do not extend the term like this: here an avatar and an IVA are used as separate terms. A basic yardstick for IVAs is often taken to be *believability* [7], an ill-defined term for evaluation purposes which might be considered the degree to which an IVA supports or undermines the user's overall sense of presence. It is important to understand that this is not the same thing as naturalism, and indeed naturalism and believability may conflict under some circumstances.

If we consider the different aspects of an IVA, a wide range of issues emerge, which we will consider in succeeding sections. First, an IVA has a body which must move in a physically convincing manner in order to support believability. Thus it must have both a body surface, and a body geometry, equivalent to a skeleton (though usually very much simpler than in the real-world case), and some behaviour using that body.

Secondly, in order for an IVA to appear responsive to its environment, there must be some kind of coupling between the IVA's behaviour and the state of the VE, whether through virtual sensing or some other means. The IVA's behavioural repertoire may vary depending on the application: in some cases focussing very much on physical interaction with the environment and in other cases more on cognitive behaviour, usually expressed through speech or at least natural language. Some IVAs can be characterised as 'talking heads', that is their body, nearly always humanoid, stops at the shoulders: and their behaviour is therefore almost entirely language-based. Finally, the IVA's behaviour must be directed by some means, requiring some control architecture within which the degree of autonomy displayed by the IVA is an issue.

Conceptually these aspects of an IVA are brought together in personality. As Martinho [40] points out, one may view the problem of creating an IVA personality from the perspective of ALife, as one of creating the conditions under which personality will emerge bottom-up. Alternatively one may view it from a more artistic point of view as an authoring or design issue, where the IVA is to be designed with respect to a pre-defined personality. In the current state-of-the-art both approaches are tried by different researchers.

Pragmatically, the different aspects would be united in an authoring tool which would incorporate them all if such a thing existed. But one measure of the relative immaturity of this area is that no such tool does exist, and researchers and developers are forced to integrate a number of disparate systems anew for each application.

3.1 Moving the body.

Creating a visually compelling body for an IVA still in general requires the designer to work with a 3D modelling package such as 3D Studio Max or Maya. Specialist packages support the creation of humanoids, ranging from Poser at the low end to the Boston Dynamics DI Guy or Transom Jack at the top end. These packages support editing at the low level of size and shape; there is no package that might modify a body according to high-level semantic categories ('menacing' 'shy'), though one can conceive of such a thing.

IVA movement comes in a variety of forms and though it can be viewed from a purely functional perspective as determining what an IVA can do in a VE, it may be at least as significant as a contribution towards personality and expressiveness. This of course has been known to animators since the origin of cartoons, and animation technology, entirely prescribed and therefore requiring no intelligence, is still widely used, not least because it is widely available through standard 3D modelling packages as well as through specialist packages such as Poser. However in a VE, as distinct from a film, animation has two major disadvantages.

Firstly it is extremely time-consuming and therefore expensive, even using key-frame techniques with some level of intelligence for interpolation. It has been said for example that the character Woody in the film Toy Story, which had a geometry with 700 degrees of freedom, 200 of these for the face and 50 alone for the mouth, required the effort of 150 people at Pixar in a week to generate 3 minutes of animation. A film seen by millions of paying customers can recoup this cost, a Virtual Environment cannot under current market conditions. Secondly, the fixed nature of animation means that its novelty is limited, again, unimportant in a film viewed as a linear spectacle, but a major problem in a Virtual Environment where the user wanders and interacts at will.

The issue of time and effort has been tackled initially using motion capture (mocap), especially as mocap studios have dropped in price and become more widely available .partly due to their extensive use in computer games. The ability to construct novel sequences from a mocap library as discussed below in 3.4 may then be incorporated, Aside from the issues raised by scripting however, mocap also has obvious limitations in terms of coverage - mainly humanoids, certain ranges of movement and almost always flat surfaces.

It usually concentrates on movement for mobility - walking, crawling, running - where generic motion is likely to succeed, and avoids agent-agent (embracing, shaking hands) and agent-object interaction (grasping, lifting, throwing) which vary much more according to the actual situation. For obvious reasons it does not extend to socially coordinated movement in groups of agents. Finally, while in principle such a library could also contain expressive movement (stumbling along wearily, striding along confidently) the combinatorial and indexing issues make this impractical.

AI and ALife technologies become more applicable where *self-animation* or *behavioural animation* [53] is applied. In self-animation, movement is driven directly by the agent's control system much as the movement of a robot would be in the real world. This normally involves the creation of a physical model of the virtual mechanism and the use of inverse kinematics to produce the correct movement in the mechanism for the position desired at each rendering cycle. The physical model can be driven by input from the VE, giving movement which is appropriate to the given moment rather than pre-scripted.

However producing fluid movement in a mechanism with a large number of degrees of freedom is correspondingly a very complex task, while the standard method of calculating inverse kinematics involves inverting a matrix, which is both computationally demanding and is affected by singularities in the solution space where motion becomes undefined. Where dynamics are also important, as in the artificial fish of Terzopolous and Tu [61] - see Figure 1 - the model may be as elaborate as a linked spring-mass system and thus even more computationally demanding, as well as usually non-linear.

Good results have been obtained in reducing the computational demand by applying learning algorithms. The computational demands of the physical model can be reduced by learning its behaviour, especially given that approximation is normally quite adequate. Neural net technology is an appropriate choice [29] since it is able to learn an arbitrary functional mapping from a training set generated by the physical model off-line. A two-step process can be applied [30] in which one neural net learns the forward model, from agent actions to the state transitions these produce in the physical model, and then this neural net is used to train the backwards system which maps state transitions back onto actions. More radically, the agent's control system itself can be learned as discussed below.

Behavioural Animation was a term coined by Reynolds in his seminal work on flocking and herding [53]. He demonstrated with his boids system that complex social movement could be modelled by equipping each agent with a set of simple rules, such as keeping a minimum distance from neighbours and obstacles, matching velocity (speed and direction) with neighbours and flying towards the centre of mass. Note that if one wishes to be truly agent-centred, *centre of mass* ought to be defined as *perceived centre of mass* rather than being calculated globally as in some applications of this work. In the current state-of-the-art, this approach has been applied successfully to stampeding animals on film, as in The Lion King and Jurassic Park; as well as to schooling of fish [61] and other animal social movement. Taking it further might involve extending the rules to include formations, internal state and a wider range of environmental stimuli.

Work has also taken place on social

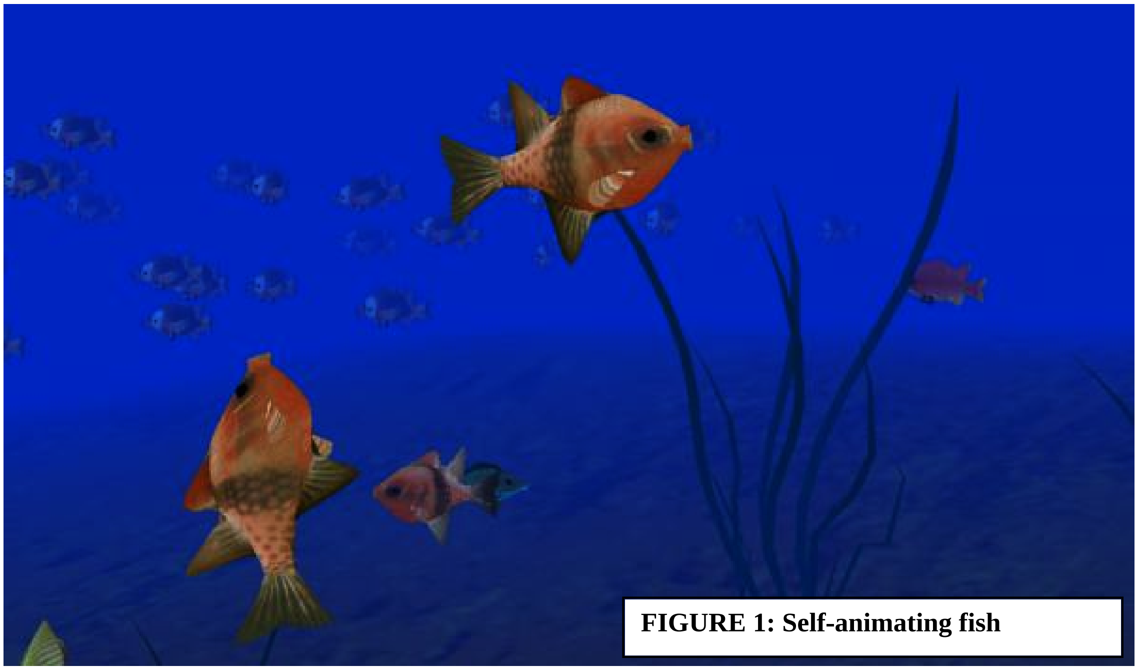


FIGURE 1: Self-animating fish

movement among humans, although here internal state is clearly of much greater importance unless very large crowds are being modelled as for example in egress design for stadia, when particle systems have sometimes been used [10]. rather than drawing on ALife or AI. In dealing with crowds in urban landscapes, rule-based systems have been used to generate goal trees influencing the movement of individuals [57] - for example an agent with a goal to catch a train must have a subgoal of moving to the station and buying a ticket in the ticket office. This is not incompatible with flocking but can be used to motivate it. Note however that the application of a global rule-set to individual agents is an alternative to an integrated architecture for an agent that includes motivations, of the type discussed below.

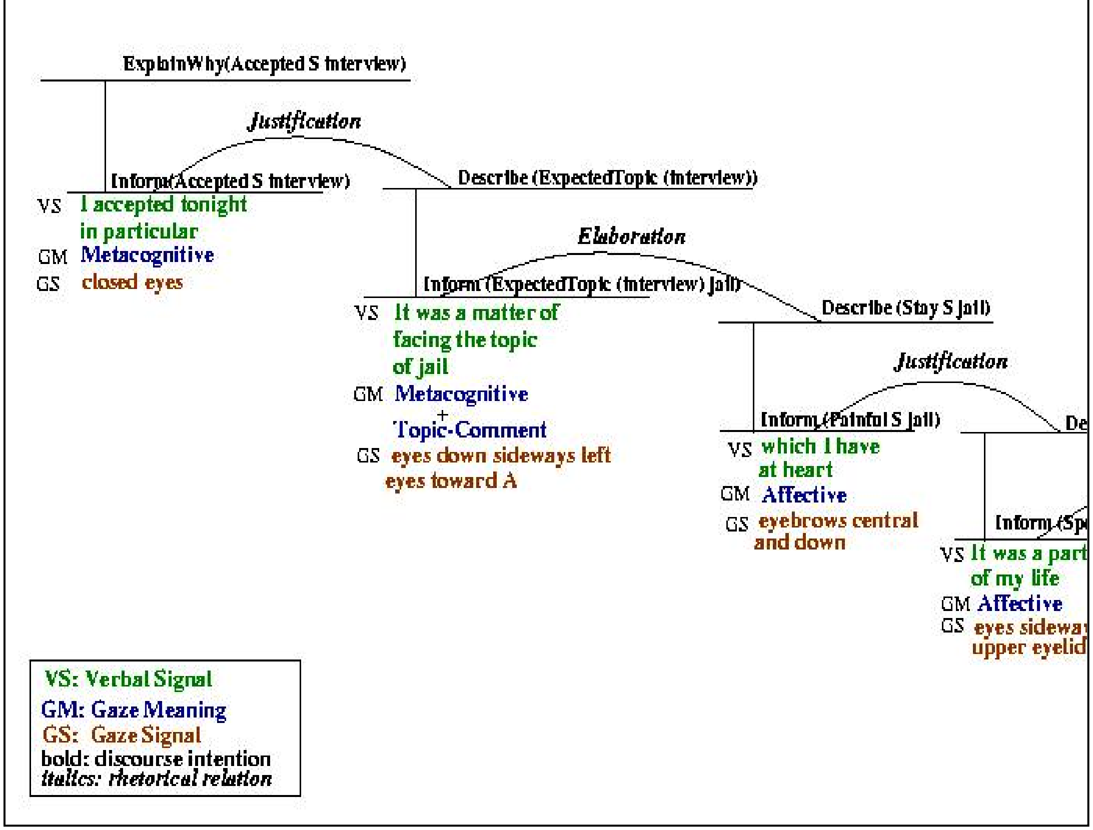
Once humanoid IVAs are involved, movement clearly requires more than low-level calculation on the geometry or physical modelling, since human movement merges the voluntary and goal-directed [33], with the involuntary movement that is driven by physiology or biology. If movement is to be generated from goals - sometimes described as *task-based aniimation* - a representation that mediates between the level of symbolic goals and natural language and the executed movement is needed.,

Badler's Parameterized Action Representation or PAR [3] is one such representation. A PAR holds twelve fields which mix the temporal and spatial control of a movement with specifications of action decomposition, applicability conditions and expected effects familiar from AI Planning. A PAR also references any objects involved and the agent executing the movement and allows the *agent manne*r to be defined, with adverbs such as *carefully* and *quickly* being translated into low-level motion. A PAR can be thought of as an interface to a high-level scripting system, and has in fact been used like that, but is not incompatible with an autonomous agent. Note however that its very flexibility is likely to put a substantial burden upon the designer who must specify all these parameters.

Work in conversational agents [Cassell 00] identified a class of human movements related specifically to communicative acts, drawing on observational work in psychology [43] in which the gestures used by real subjects during conversation or story-telling are noted. This work identified categories of gesture such as *iconics*, which represent features of topics such as the size and shape of objects, *beats*, which occur with accented words and turn-taking behaviour, and *emblems*, gestures carrying (culturally variable) known meanings, such as thumbs-up or obscenity. Categorising movement like this with a coarser granularity allows it to be linked directly into natural language via an annotation system, though this is still usually hand-coded. At least one talking head [48] has taken this approach, and has tried to automate the annotation step through the use of Rhetorical Structure Theory as shown in Figure 2.

However the expressive results of synthetic systems using this approach are still less convincing than the output of skilled animators [Chi et al 00], and this has led to an alternative approach based on Laban Movement Analysis [34]. Here. movement is characterised at a much lower level by *effort* and *shape* components, thus avoiding the issue of what counts as a gesture and what does not. This approach has been taken up by Badler and his group and is neatly incorporated into the PAR format discussed above [18].

Once agent-object and agent-agent interaction is included, the problems of generating movement from first principles become very much more complex. The approach that has been very widely taken to this problems is that of smart objects [49], in which the interaction knowledge is part of the specification of an object in the VE rather than a demand made upon the IVA. Thus an IVA which passes through a door does not have to perform complex reasoning in order to accommodate its hand to the shape of the doorknob - a predefined handshape can be stored with the door object.


ExplainWhy(Accepted S interview)
Justification
Inform(Accepted S interview)
Describe (ExpectedTopic (interview))
VS I accepted tonight in particular
GM Metacognitive
GS closed eyes
Elaboration
Inform (ExpectedTopic (interview) jail)
Describe (Stay S jail)
VS It was a matter of facing the topic of jail
GM Metacognitive + Topic-Comment
GS eyes down sideways left eyes toward A
Justification
Inform (Painful S jail)
De
VS which I have at heart
GM Affective
GS eyebrows central and down
Inform (Sp
VS It was a part of my life
GM Affective
GS eyes sidewa upper eyelid
VS: Verbal Signal
GM: Gaze Meaning
GS: Gaze Signal
bold: discourse intention
italics: rhetorical relation

This means that objects must be defined by more than a collection of polygons in the scene graph, and are for example annotated with interaction features which are associated with the object's functionality [31]. This technique is effective but has clear limitations since in its basic form it requires every agent to interact in the same way - not a problem with doorknobs, but difficult to generalise. Consider agents with hands of different sizes, and different skin tones, or some wearing gloves. Then the interaction either requires a transformation to be passed from the interaction feature to the agent geometry or the geometry of the hand to be passed to the interactional feature.

3.2 Responding to the environment

It is noticeable that there is far more work in IVA movement than in virtual sensing. While movement is an indispensable attribute, the nature of a VE, in which the designer can produce agent movement from the top, with a god-like perspective, and use the scene graph as an omniscient source on events and attributes, means that virtual sensing need not be included at all. In computer games for example, monsters can usually detect a player through walls by accessing his or her position from the data structures and the application of some AI algorithms such as A* to path planning in such environments typically depends on a global map. This can adversely impact gameplay by making opponents unrealistically persistent or difficult to combat.

However, once the requirement for *believable* sensing is established, it seems that it is actually more difficult to work out what an agent *should* be able to perceive from an omniscient position than it is to equip the agent with virtual sensors which as a direct result deliver the appropriate information [50]. Gaze is also a potent communicator of attentional focus, and an IVA with virtual eyes will use gaze in order to direct its virtual sensing in a natural and realistic manner.

The extent to which virtual sensing is actually applied varies. At one extreme, the artificial fish discussed above, were equipped with a model of the primate visual system [62]. This involved a binocular projection of the 3D world onto the 2D virtual retinas of the fish, which were modelled with high resolution foveas and lower resolution peripheries and moved by a set of virtual motor lenses, implemented as four virtual coaxial cameras. The system applies an active vision principle: (in the sense of the term in the AI Vision community); using incoming colour data, the fish control system sends control signals to stabilise the visual system during movement. Interesting objects in the peripheral vision are identified by their colour histogram and the eyes saccade to locate the object first in the fovea and then in the centre of the eye. Stabilisation was carried out for small displacements by computing the overall translational displacement (u,v) of light patterns between the current foveal image and that from the previous time instant, and updating the gaze angles to compensate. Large displacements produced re-foveation.

The virtual dog, Silas [9] incorporated a less complex artificial vision system that used optical flow to control navigation, again very much an active vision idea. Again, the 3D virtual world was mapped onto 2D as if through an RGB camera, with false colouring used to render the current-object-of-interest. A motion energy calculation was separately carried out on left and right halves of the 2D field, using pixel movement between frames and a pseudo-mass weighting. The difference in total flow energy between left and right fields was then input into the control system for steering, with an absolute threshold to stop Silas crashing head-on into walls.

At the other end of the spectrum, the STEVE pedagogical agent [54,55] has a sensorimotor system which monitors the communication bus in a distributed system linking the agent to the virtual world for messages describing changes in terms of objects and attributes. These messages are used to update a symbolic world model on which STEVE's reasoning operates. Gaze is then unnecessary for perception since the agent has 'eyes in the back of his head', which is however useful in a pedagogical process (as many real teachers would agree). However gaze still turns out to be important for believability and is incorporated as a planned action within the reasoning system so that STEVE looks at objects that his bodily representation is interacting with and also at the user when interacting using speech.

Somewhere between these two extremes are virtual sensors inspired by robotics rather than by living things, modelling active sensing systems such as infra-red and ultra-sound rather than passive ones like vision. This is an obvious step where a VE is being used for virtual robotics, but is easy to carry over into other IVAs [2]. An infra-red sensor can be modelled very simply by a line or cone attached to the geometry of the agent which gives the distance of any object with which it intersects within its range. This can be used for obstacle avoidance in the same way as for a real robot, but can also be used to give the agent information about the identity of objects which would not normally be available in the real world without a great deal of processing. In the same way, the bounding boxes available in most VE toolkits can be used so that when two agent bounding boxes intersect they exchange information about identity and possibly internal state.

The state-of-the-art covers the spectrum discussed above for virtual vision and active robot-like sensors, but very much less work has been carried out on any other kind of virtual sensing. Though sound models exist within various VE toolkits, there is little evidence of virtual hearing being modelled, though one exception here occurs in the computer game Thief! where the user tries to steal goods from locations guarded by virtual characters who will only hear noises generated by the player within a certain range. Smell has been investigated by a few groups in relation to the human user of a VE, but there seems to be little work as yet (but see [23]) using virtual smell or virtual noses, though this would fit well into some of the more ALife-oriented IVAs.

Outside the scope of this particular report is the use of vision technology in particular to provide a link between IVAs and the human user. Since a VE is being rendered in relation to a user's position and orientation, that information is available to IVAs, but the non-intrusive capturing of gesture and facial expression is an active field where much remains to be done.

3.3 Intelligent Behaviours

For an IVA, intelligence comprises several components, such as sensing, learning, natural language communication and reasoning, all to be integrated. In the remainder of this section, we will essentially discuss reasoning abilities and its integration with physical action. The integration principles we introduce are also valid for virtual sensing, just described. Natural language communication in conjunction with high-level reasoning will be discussed in the next section.

From a generic perspective, Intelligent behaviour consists in determining the best sequence of actions to be executed in the virtual environment, taking into consideration the agent’s goals and the environment’s resources. For these reasons, AI planning techniques are a good formalism to analyse the problem as well as a generic technique to implement agent behaviour. Planning as a hierarchical approach is also in a good position to co-ordinate lower levels of control and animation of virtual humans. Advanced research in the field is indeed mostly based on planning techniques, though in some cases the actual implementation might resort to simpler formalisms that appear as compiled plans.

Planning techniques support a "cognitive" level that should control lower-level animation. This is a generalisation of techniques developed in the area of virtual human animation. For instance, Perlin & Goldberg [49] explicitly introduce a distinction between low-level animation control and high-level behaviour modules in their Improv system architecture. Similarly, Magnenat-Thalmann & Thalmann [39] introduce a distinction between general motion control and more complex behavioural patterns.

They resort to "Displacement Local Automata" (DLA) for the high-level control of motion form a goal-oriented perspective. Because of their direct relations to Scripts [56], DLA can also be considered as some form of precompiled plans, without the flexibility that plans normally support. A sophisticated architecture for animation control has been developed by Badler et al. [4], which will be described in the next sections.

To state the problem in the simplest possible terms, we suggest three levels of description for an agent's behaviour, which deal with motion, action and intentions. Motion corresponds to the actual physical simulation in the virtual world, action to the patterns of movements required to execute specific actions and intentions to the high-level goals that the agent is pursuing.

Integrating the various levels of description, starting with the AI level, is thus a major endeavour in the development of intelligent virtual humans. We will first illustrate the integration problem with a "historical" example, before pointing at the most recent research in the field.

Early forms of integration used to take place directly between an AI algorithm and the animation primitives, as can be illustrated with the case of path planning. Path planning consists in finding an optimal path (generally the shortest one) between a starting point and a destination point in a virtual environment, avoiding obstacles.

Traditionally, path planning has been solved using a heuristic search algorithm such as A* [3,6] directly coupled with the low-level animation of the agent. The use of A* for path planning is based on a two-step process. The virtual environment is first discretised to produce a grid of cells. This grid is formally equivalent to a connectivity tree of branching factor eight, as each cell in the discretised environment has eight neighbours. Searching this connectivity tree with A* using a distance-based heuristic (Euclidean distance or Manhattan distance) produces the shortest path to the destination point (Figure 3).

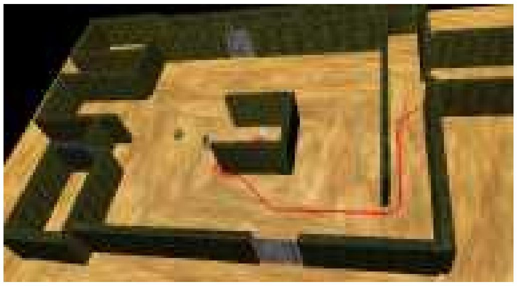


FIGURE 3 Path Planning

This path is calculated offline, as A* is not a real-time algorithm, and the agent is subsequently animated along this path. As a consequence, this method cannot be applied to dynamic environments. This direct integration of A* with low-level animation primitives is faced with a number of limitations. For instance, it is not connected to high-level decision making, nor to the agent perception. Monsieurs et al. [45] have proposed to use A*-based path planning in conjunction with synthetic vision for more human-like behaviour, while Badler et al. [4] have incorporated path planning into generic high-level behaviour planning.

Though the example of path planning is based on an AI algorithm, the integration achieved is rather weak and it clearly does not cover properly the levels of description we have introduced.

3.4 IVA control architectures

Typically there is still a gap between the methods adopted by researchers from graphics backgrounds for controlling an

IVA and those favoured by researchers from AI and ALife backgrounds. The dividing issue is often one of artistic or directorial control versus agent autonomy, with many researchers who have moved from animation still favouring various kinds of scripting where AI and ALife researchers often think in terms of sensor-driven behavioural control or of goal-driven action supported by symbolic reasoning. There are many definitions of autonomy in different fields, from psychology and philosophy to animation. Here we adopt the robotic definition that an autonomous agent has a sense-reflect-act cycle of its own operating in real-time in interaction with its environment. The amount of autonomy possessed by an IVA is therefore related to its control architecture.

Scripting has been a very popular method of allowing developers and authors to control IVAs at a higher level than that of conventional animation. One of the best known of such scripting systems was IMPROV [49] which was intended to support the directorial level control of virtual actors. Though not a 3D system, the scripting language of the Microsoft Agent toolkit has also been widely used, and the humanoid toolkits of Transom JACK and Boston Dynamics' DI Guy both supply scripting languages.

Scripting languages act upon libraries of *behaviours*, but it is important for the sake of clarity to understand that *behaviour* here does not correspond to the use of the term in AI and ALife. In this context it usually refers to a pre-scripted animation or sometimes chunk of motion capture, such as *walk, run, look-left.*

In contrast, the term *behaviour* in AI refers to a sensor-driven control system [11,12] where incoming stimulus is tightly coupled to outgoing reaction. An emergent behaviour such as *run* would be produced by the reaction of the physical system to the strength and type of the control signal generated at any particular moment by the sensory stimulus. Scripting supposes an author or director who determines the behaviour of agents, and not an agent-centred mechanism where the agent autonomously controls itself

The important issues in scripting languages are then those of making smooth transitions from one animation to another and deciding which animations can be combined and which cannot, allowing an agent to 'walk and chew gum' but not crawl and run simultaneously.. One model for controlling graphical representations in this flexible way is that of Parallel Transition Networks or PaT-Nets [3]; other human animation systems have adopted similar approaches. In a PaT-Net, network nodes represent processes while arcs contain predicates, conditions, rules, or other functions that cause transitions to other process nodes. Synchronisation across processes or networks is effected through message-passing or global variable blackboards.

The conditional structure used in PaT-Nets gives them more power than just the parallel organisation and execution of low level motor skills. It provides a non-linear animation model, since movements can be triggered, modified, or stopped by transition to other nodes rather than being laid out linearly along the timeline as in traditional animation. This is a step toward autonomous behaviour since it enables an agent to react to the environment and could support autonomous decision-making capabilities.. However it is a very low-level representation which is encoded in a programming language and is therefore not very accessible to the non-expert. It was for this reason that the PAR formalism discussed in 3.1 above was developed at a higher level.

Scripting is one method of dealing with the problem known in AI as *action selection*: which of the many things an agent can do at any moment is the *right* thing to do? A solution at the level of symbolic representation might involve AI planning, the use of reasoning mechanisms such as the situation calculus, or lower-level mechanisms such as the rule-based production system, in which rules which match a current world condition may fire subject to conflict resolution among competing alternatives. The important issue, as discussed in 3.3, is how these levels can be smoothly integrated with the rest of an agent system right down to the animation level.

Recent research at the University of Pennsylvania has produced the most comprehensive framework for intelligent virtual agents to date. This framework integrates all the components for high-level reasoning, action patterns and physical simulation of motion. Most importantly, it is strongly based on up-to-date AI planning techniques. The starting point for this research was the observation reported above that realistic behaviour could not be based on procedural animation only, as an embodied intelligent agent has to deal with a changing environment. Simultaneously, another motivation for the development of high-level behaviour techniques was the investigation of natural language control of character animation.

Some of the components of this integrated approach have been discussed already as we will see.. For a more detailed presentation, we refer the reader to the original literature from the project [4,66].

The objective of the agent's architecture is to support the integration of sensing, planning and acting. It comprises three control levels:

- An AI planner based on the agent's intentions, using incremental symbolic reasoning
- The Parallel transition networks (PaT-Nets) discussed above as an action formalism.
- A Sense-Control-Act (SCA) loop performs low-level, reactive control involving sensor feedback and motor control

The SCA or behavioural loop is analogous to a complete behaviour in a subsumption architecture [11,12]: it is mainly used for locomotion reasoning.

Pat-Nets are used to ground the action into parametrised motor commands for the embodied agents, which constitute the lowest level of description (the "motion" level). They are invoked or generated by the planner.

The planner is in charge of high-level reasoning and decision making. It is based on the "intentional" planner ItPlans [26]. ItPlans is a hierarchical planner in which expansion takes place incrementally, only to the degree necessary to determine the next action to be carried out. This makes it possible to interleave planning with execution taking into account the dynamic nature of the environment. The theoretical analysis behind the planner is based on the notions of intentions and expectations [67]. Geib and Webber [25] have demonstrated that the use of intentions in planning for embodied agents was not compatible with the traditional definition of pre-conditions in planning. They suggested replacing traditional preconditions with situated reasoning, i.e., reasoning about the effects of performing an action in the agent's environment. This apparently theoretical point actually has important consequences for the effective coupling of planning and action performance, and situated reasoning enables the planner to make predictions about the results of executing an action, without having to maintain a complex model of the world. An example using this approach is discussed below [Funge 98].

Several demonstrators have been implemented with this framework, such as a "hide-and-seek" simulator based on intelligent agents [4] and a military training system for checkpoint control featuring virtual soldiers, the latter also accepting natural language instructions to update the knowledge of virtual actors (Figure 4).

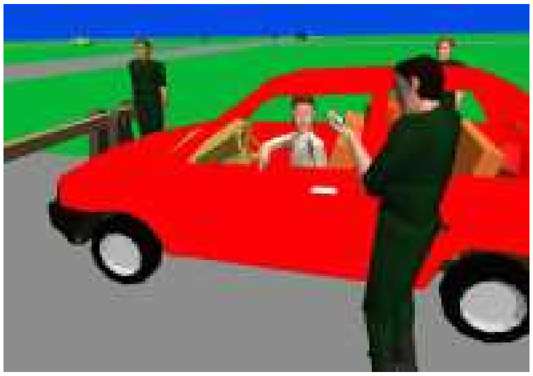


FIGURE 4 Checkpoint control appl.

At the other end, plans can also interface with abstract specifications of behaviours through intentions and high-level goals: this, in particular, makes possible to accept behavioural instruction in natural language. The determination of an agent's behaviour through natural language will be discussed as part of the next section.

An example that uses both planning and a rule-based inference engine is the STEVE system already mentioned in 3.2 [54,55]. The cognitive component was implemented in Soar [35] a mature generic AI architecture. A STEVE agent acts as a tutor or demonstrator in a domain where the student is trying to learn correct procedures, that is, sequences of actions, for operating machinery.

STEVE thus requires a representation of a plan, a correct sequence of actions. A standard AI planning formalism is used [68] for this, with actions represented as nodes in a partial ordering over the causal links between them. A causal link occurs when the effect of one action (the goal that it achieves) is the precondition for the execution of the next. Actions themselves can be primitive, that is executable by the agent, or expandable, that is, reducible by more planning to a set of causally-linked actions.

Primitive actions are passed to a sensorimotor component, which also monitors execution, although as commented above this is some way from true virtual sensing. STEVE is also very simple graphically, represented either by a floating torso or by a simple detached hand, so that the motor actions required are relatively straightforward.

This is less true of a second example, the Merman of Funge [24]. Here, a different formalism, the situation calculus [42], is used as a cognitive layer over the sensing and motor systems developed for the artificial fish already discussed [61]. The situation calculus is a first-order state-based logic using *sorts* (snapshots of a current state) and *fluents* (a world property that can change, taking a situation as its argument).

In the Merman system, a fast-swimming shark whose behaviour is driven by a few simple rules, tries to catch a slower-swimming merman, whose cognitive layer allows him to reason about possible future situations. In an underwater scene containing rocks behind which the merman can hide. This cognitive layer allows the merman to predict the shark's behaviour and pick a rock which conceals him. In a second scenario, the Merman has a pet, and the cognitive layer also allows him to predict whether an attempt to distract the shark from eating the pet will prove fatal and should not be attempted.

Intelligent actors based on other AI techniques than planning have been described: for instance rule-based systems like the Soar system have been used to develop intelligent “Quakebots” [36]. The main advantage of rule-base systems is that they can benefit from efficient implementations and support fast reaction times for virtual humans. Rule-based system can implement hierarchical behaviour, though some explicit representational properties are often lost in the process. Plans, on the other hand, tend to provide a readily visible representation that can serve as a resource for various types of actions. And, as we have seen, their hierarchical nature facilitates their integration with action patterns and physical simulations.

Both STEVE and the Merman are examples of agents with goal-driven cognitive components which allow them to select their own actions on the basis of a future projected state. However ALIFE has been more concerned with insects and animals than with huimans, and concepts such as *drives* which are more relevant than goals for these creatures.

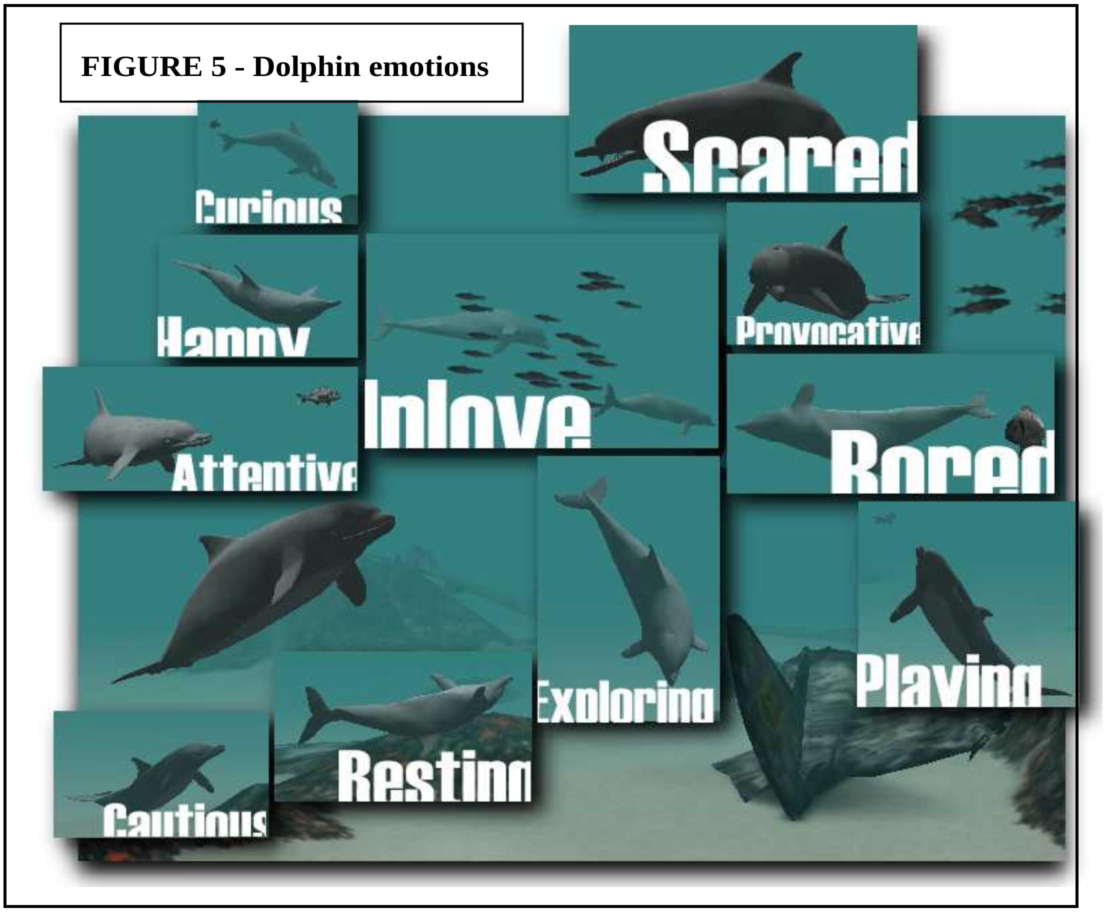
Behavioural architectures are subsymbolic, and a behaviour can be thought of as a coupling - or functional mapping - between sensory stimulus and motor response. It is the motor response which then animates the body. However, identical sensory inputs may produce different behaviours according to internal drives which have the effect of activating different parts of the behavioural repertoire. Activation and inhibition relationships between behaviours will also come into play. The architecture used on Silas the dog [8] applied these ideas along with the ability to learn new sensorimotor couplings.

Similar ideas were explored in the Virtual Teletubbies [2]. Drives included hunger, fatigue, and curiousity. The problem of producing coherent sequences of behaviour was tackled by linking different repertoires of behaviours together with the level of internal drives allowing such a sequence to be activated.

3.4 Putting in Emotion

The discussion of the use of *drives* as a means of motivating behaviour brings us to the related topic of a*ffect*, an area which has become much more active on the recent period. This may be partly because an embodied agent has many more channels for expressing it - facial

FIGURE 5 - Dolphin emotions



expression, bodily posture, gesture - and partly because the richness of a virtual world provides much more scope for interaction between IVAs and between an IVA and a user. One should remember that in AI, where affect has been investigated for some time, the standard experimental environment had consisted entirely of typed text interaction.

Just as the control architectures discussed above seem to fall into behavioural systems using internal drives, and reasoning systems using symbolic planning and inferencing, so work in affect tends to split along similar dimensions. This division mirrors a long-standing debate within psychology [51] and can be traced right back to the cartesian mind-body split. In behavioural architectures, affect is a fundamental part of the control process [14]; in the more cognitive architectures it is the interaction between affect and reasoning that is of interest.

The most popular model of affect in the more cognitive architectures is based on the work of Ortony, Clore and Collins - the OCC theory [47]. They based their model on the three concepts of *appraisal*, *valence*, and *arousal*. Here appraisal is the process of assessing events, objects or actions; valence is the nature of the resulting emotion - positive or negative - and arousal is the degree of physiological response. Emotions can then be grouped on these dimensions.

For example, the *well-being* group simply contains *joy* and *distress*, where the former is a positive emotion arising from the appraisal of a situation as an event, and the latter is a negative emotion arising from the same appraisal. Given an agent with goals, it is easy to translate this into positive emotion where an event supports the agent's current goal and negative where it runs counter to a current goal. The *fortunes-of-others* group starts from a different appraisal - of a situation as an event affecting another. It then includes *happy-for*, *sorry-for*, *gloating* and *resentment*, where, for example, *happy-fo*r is a positive emotion for an event causing a positive emotion in the other, and *resentment* is a negative emotion for an event causing a positive emotion in another. The definitional system can be expanded to compound various emotions also, so that *shame* + *distress* = *remorse*.

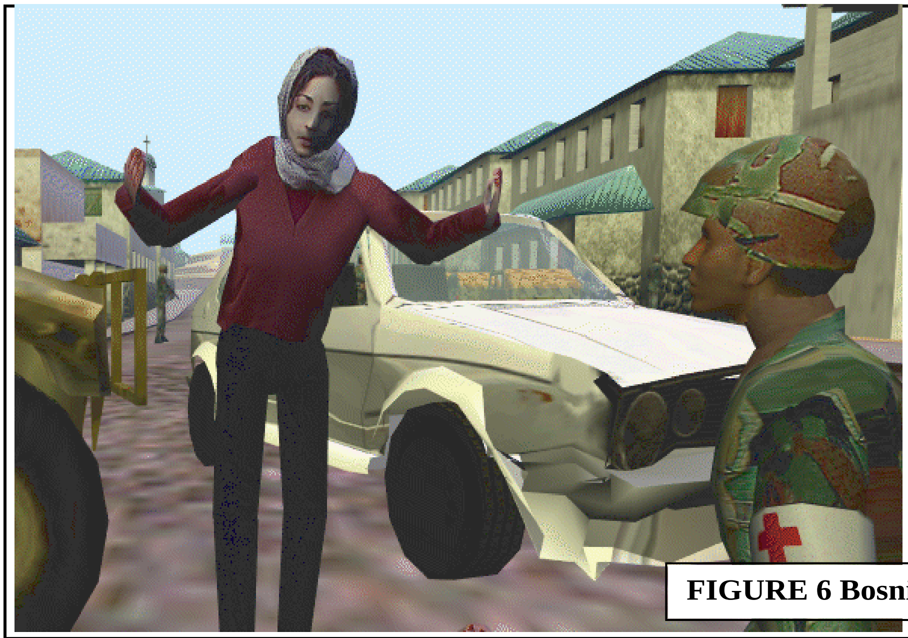


FIGURE 6 Bosnia scenario

This produces a taxonomy containing a total of 22 related emotions.

While this has been characterised by some as a 'folk-theory' of emotion, it has been widely implemented, for example in the Oz project 'Edge of Intention' where graphically simple creatures called Woggles reflected their emotional responses on a very simple set of features [7]. While classically this approach was employed with natural language interaction, so that *appraisal* was a simple assessment of what had been said, it is quite possible to apply it to a VE in which sensed events are appraised and applied to behaviour.

This was seen in the virtual dolphins [40] referred to above. In order to map the 22 emotions onto the behaviour of the dolphins, the emotions derived from the appraisal process were then aggregated along four dimensions. These were the total positive (*pleased*) and negative (*displeased*) valence involved, and two dimensions called *passionate* and *frighten* - the former the sum of all love emotions in the taxonomy, and the latter of all fear emotions.

The creation of a characteristic set of emotional responses was used to produce personality. For example, the male dolphin, of the two produced, reacted with hate and fear to a crashed aircraft inserted in the scene as this was a human-constructed object, while the female dolphin reacted to the same object with curiosity. Figure 5 gives some indication of the way each aggregation of emotions changed the behaviour exhibited.

A further interesting use of this type of emotional/cognitive response system can be seen in a development of the STEVE system already discussed. This is an immersive training system which models a US soldier carrying out a peace-keeping mission in Bosnia [28].. The scenario tries to model the conflicting situations such a soldier might face through an example where an army truck runs over a local child and at the same time reinforcements are needed elsewhere. Here the emotional system embodied in the virtual character playing the mother of the child is being used communicate strong emotion to the trainee who has in turn to manage the combination of emotion and reason so as to arrive at a choice of action.

The mother's emotional state changes from fear and shock, expressed as crouched, arms around herself, little language behaviour; to anger, if it appears that surrounding troops are being moved off as reinforcements rather than helping to evacuate the injured child. Anger is expressed as standing, arms actively out (see Figure 6) and angry language behaviour. Thus this type of emotional theory can be integrated well into planning as well as expressive behaviour.

4. Language Technologies in Virtual Reality

There has been a sustained interest in the use of natural language technologies for virtual environments applications. The rationale for the use of NL initially came from an interface perspective: speech does not disrupt visualisation and presence and is compatible with other I/O devices in VR. It is thus a convenient way of accessing information and controlling VE systems [32]. However, language is also a convenient way to describe spatial configurations. As such, it can be used to provide a high-level description of scenes to be created in VE.

This specific approach has been described in several military applications [16,65]. In these applications, situation assessment can be assisted by inputting natural language descriptions of the battlefield [16] or querying the corresponding database. NL can also serve as a transparent interface to virtual agents in a distributed simulation [38]. This makes possible for the user to address in a transparent fashion other entities in the simulation without being able tell the synthetic forces from the manned simulators, which would otherwise influence his behaviour.

Clay and Wilhelms [19] have described a NL interface for the design of 3D scenes in virtual environments. In this system, NL descriptions replace traditional interfaces to construct a scene from 3D objects, or at least makes it possible to quickly explore design ideas, to be subsequently refined with more traditional methods. Not surprisingly, there is a strong emphasis on the processing of spatial expressions and reference to cognitive linguistics for the specific semantics of spatial expressions (not only prepositions but also Lakoff's semantic theory of space).

The incorporation of a NLU layer in a virtual environment system is based on well-described principles, which are common to most of the systems described so far [67,70,44]. One aspect consists in the linguistic analysis step (parsing) and the other in the identification of discourse objects in relation with the virtual world contents.

The implementation of the parser and its linguistic resources (semantic lexicon and grammar) is dictated by the way users express themselves and by the target system functions of the VE system. The parsers used in IVE tend to specifically generally target sub-language applications, though in some cases large-coverage parsers have been customised to the requirements of the application. Linguistic input is strongly biased towards spatial expressions, as it tends to describe object configurations to be reproduced in the VE, or spatial instructions for navigation in the environment. Another very common linguistic phenomenon observed is the use of definite descriptions to verbally designate objects in the virtual world having salient perceptual properties, for instance "the ship with the admiral flag", "the door at the end of the room with an 'exit' sign on it".

These sorts of expressions are known to carry specific syntactic and semantic problems. At the syntactic level, they are organised around spatial prepositions, which generate syntactic attachment ambiguities. Consider for instance the expression "open the red door with the key". From a purely syntactic perspective, the prepositional phrase "with the key" could relate both to "door" or to "open". Semantic information stating that a key is an instrument for an opening action can be used to disambiguate parsing.

Semantic processing is mostly concerned with identifying entities in the VE: this particular step is known as reference resolution. It consists in identifying objects in the virtual world from their linguistic expressions. Objects can be directly referred to or designated through definite descriptions.

The above applications are mostly concerned with scene descriptions, and navigation within the environment. However, NL is also a convenient way to instruct virtual actors. The principles behind NL control of artificial actors are somehow different than those governing NL interfaces to VR system. A way of illustrating the difference between the use of NL input for VR systems control and its use to instruct intelligent characters is to remark that in the former case emphasis is more on the identification of objects, as the NL input will trigger the creation of these objects in the virtual world, while in the latter it is more on the detailed semantics of action, as the ultimate system interpretation will consist in producing a parametrised action representation.

Further, communication with virtual actors essentially falls under two different paradigms: the instruction of virtual actors to carry out autonomous tasks in the VE [67], and the communication with virtual actors for information exchange, assistance and explanations [55], [44].

The use of NL to instruct artificial actors has been described for instance in [Webber et al., 1995] and [70]. Most of the fundamental principles have been laid out in the "AnimNL" project [67], in particular the classification of instruction according to their interpretations in terms of plans to be generated. For instance, doctrine statements (e.g. "remain in a safe position at all times while checking the driver's ID"), convey generic statements about an agent behaviour that are to be valid throughout the simulation. This has been illustrated in a checkpoint training simulation **(Figure 4)**. In this system, virtual actors control a checkpoint passed by various vehicles, some of which might contain a hostile individual. Whenever they fail in the procedure, their behaviour can be improved by high-level NL instructions, such as "watch the driver at all times".

Cavazza and Palmer [70] have described a NL interface to a semi-autonomous actor in a "Doom-like" computer game **(Figure 7)**. In this system, agent control takes place at a lower level, the agent only generating scripted action sequences in response to NL instructions, such as "run for the plasma gun near the stairs on the left".

The target actions correspond to a small set of possible actions within the game. Emphasis is more on the processing of complex spatial expressions, fast response times and the possibility to receive new instructions while previous actions are still being executed (e.g. "go to the blue door at the end of the room", "turn right after the door").

The "Lokutor" system [Milde, 2000] is an intelligent presentation agent embodied in a 3D environment, communicating with the user through dialogue. Its current task is to present a car to a user, as some kind of virtual sales assistant. The NL input is analysed by a parser and then passed to a deliberative system in charge of the high-level goals of the agent. These in turn control a behavioural module, which determines the next action to be taken by the agent. Simple directives can however directly determine actions to be taken by Lokutor. Reference resolution takes advantage of the agent's embodiement to solve context-dependent reference and indexical expressions. The dialogue aspects of the system are strongly task-dependent, as language generation by Lokutor is essentially based on the car's user manual, rather than on a full dialogue model.

There is more emphasis on the dialogue aspects in the Steve system [Rickel and Johnson, 2000], as Steve is developed as a tutoring agent. Several scenarios have been implemented, including the maintenance procedure for a high-pressure air compressor aboard a ship. Steve presents the procedure on a step-by-step basis, pointing at the corresponding parts of the 3D device and explaining the various maintenance steps through spoken output (e.g. "open cut-out valve three"). Steve can shift between various modes in which it either demonstrates the procedure or monitors the trainee performing the maintenance task herself. The system always remains interactive through its dialogue capabilities. The demonstration mode can be interrupted by the user for explanations or for requesting permission to complete the current task ("let me finish").

In the monitoring mode, Steve is available to assist the student, who can turn

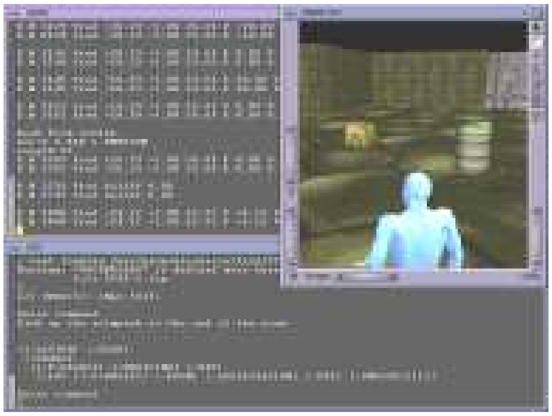


Figure 7: *A NL Interface to a Computer Game (courtesy of Ian Palmer).*

Figure 8: *An Interactive Storytelling Environment*

to him for advice while operating on the compressor: if she is unsure about the procedure, she can ask "what should I do next?". Steve will, in response, provide advice such as "I suggest that you press the function test button" [55]. Human-computer dialogue in this context also requires rule for the display of multimodal information (e.g., gestures pointing at the relevant parts of the compressor). Finally, the actions to be taken by Steve are associated with relevant communicative acts, and are represented with Augmented Transition Networks (though these are implemented with Soar rules).

To summarise, it can be said that the integration between NL instructions and the spatial properties of VE is a major characteristic of the inclusion of language technologies into IVE. It relates to active research areas of NLP, which deal with the processing of spatial expressions. The other specific aspect is the semantics of action. This is of specific relevance to the control of virtual actors, where the semantics of action has to be related to sophisticated action representation, in which actions can be parametrised according to the detailed meaning of NL instructions.

5. A Case Study in IVE: Virtual Interactive Storytelling

The development of intelligent characters naturally leads to a new kind of entertainment systems, in which artificial actors would drive the emergence of a story. This should make possible real-time story generation and henceforth user intervention in the storyline. User involvement can take various forms: the user can participate in the story, playing the role of an actor, or interferes with the course of action from a spectator perspective.

Interactive Storytelling is strongly AI-based, and integrates many of the techniques discussed above for virtual actors, such as planning, emotional modelling and natural language processing. For these reasons, it constitutes a good illustration of many of the concepts introduced so far. We will restrict the discussion to those storytelling applications which make reference to some kind of storyline. Other forms of storytelling in which the user is the principal source of interpretation, through processes of "storyfication" or emotional communication have been discussed in previous sections.

Interactive storytelling is also faced with the same kind of alternative approaches as other IVE systems: intelligence can be primarily embedded in the system or in the virtual actors that populate it. In the former case, explicit plot representations govern the dynamic behaviour, while in the latter case it is agents' plans represent their roles, from which the actual story will emerge.

Explicit plot structures can dictate virtual actors' behaviours from the strict perspective of the storyline. Maintaining such an explicit plot representation supports the causality of narrative actions, but might require alternative choices and decision points to be made explicit to the user. In an approach called user-centred resolution, Sgouros et al. [58] have based their narrative representation on an Assumption-based Truth Maintenance System (ATMS).

Considering the duality between character and plot, an alternative option is to follow a character-based approach. In this case, planning is once again the main technique to implement actors' behaviour [69,41,17]. There is indeed some continuity between generic behavioural techniques based on planning and the storytelling approach. However, and even though formalisms might be similar, there is a significant difference in perspective. The planning ontology for the narrative approach differs from the one used in a more traditional "cognitive" approach: plan-based representations for storytelling are based on narrative concepts instead of generic beliefs and desires.

The various sub-goals in the plan do correspond to a set of narrative functions such as “gaining affection”, “betraying”, etc. This is more than just a difference in ontology, it is a difference in granularity as well. Though artificial actors’ behaviours could in theory be derived from generic high-level beliefs and intentions, there is no guarantee that the action derived would be narratively relevant. Nor would long-term narrative goals or multiple storylines be derived from abstract principles only. In that respect, artificial actors can be said to be playing a role rather than improvising on the basis of very generic concepts. The relation between cognitive approaches and narrative approaches can be seen as one of specification and is essentially dictated by representational and relevance issues. This point is not always covered in the technical literature due to the relative immaturity of the field and the absence of large scale narrative implementations based on a storyline. Other justifications for the use of narrative representations have been discussed by Szilas [60].

We have implemented such an interactive storytelling system **(Figure 8)**, in which a generic storyline played by artificial actors can be influenced by user intervention. The final applications we are addressing consist in being able to alter the ending of stories that have an otherwise well-defined narrative structure. Ideally, it would make possible to alter the otherwise dramatic ending of a classical play towards a merrier conclusion, but doing so within the same generic genre of the story. Our own solution to this problem consists (in accordance with our final objectives stated above) in limiting the user involvement in the story, though interaction should be allowed at anytime. This is achieved by driving the plot with autonomous characters’ behaviours, and allowing the user to interfere with the characters’ plans. The user can interact either by physical intervention on the set or by passing information to the actors (e.g., through speech input).

The storyline for our experiments is based on a simple sitcom-like scenario, where the main character (“Ross”) wants to invite the female character (“Rachel”) out on a date. This scenario tests a narrative element (will he succeed?) as well as situational elements (the actual episodes of this overall plan that can have dramatic significance, e.g., how he will manage to talk to her in private if she is busy, etc.). Our system is driven by characters’ behaviours. However, rather than being based on generic “cognitive” principles, these actually “compile” narrative content into characters’ behaviours, by defining a superset of all possible behaviours. Dynamic choice of an actual course of action within this superset is the basis for plot instantiation [69]. However, further to the definition of actor roles, no explicit plot representation is maintained in the system. In this regard, there is no external narrative control on the actors’ behaviour, as this is made unnecessary by the approach chosen. Each character is controlled by an autonomous planners that generates behavioural plans, whose terminal actions are low-level animation in the graphic environment we’ve been using (the game engine Unreal Tournament).

We have adopted this framework to define the respective behaviours of our two leading characters. We started with the overall narrative properties imposed by the story genre; sitcoms offer a light perspective on the difficulties of romance: the female character is often not aware of the feelings of the male character. In terms of behaviour definition, this amount to defining an “active” plan for the Ross character (oriented towards inviting Rachel) and a generic pattern of behaviour for Rachel (her day-to-day activities). To illustrate Ross’ plan: in order to invite Rachel, he must for instance acquire information on her preferences, find a way to talk to her, gain her affection and finally formulate his request (or having someone acting on his behalf, etc.). These goals can be broken into many different sub-goals, corresponding to potential courses of action, each having a specific narrative significance.

An essential aspect is that planning and execution should be interleaved: this is actually a pre-requisite for interactivity and user intervention. The character does not plan a complete solution: rather it executes actions as part of a partial plan and these constitute the on-going story in the eye of the user. From his understanding, he might then decide to interfere with the plan. When (for whatever reason, internal causes or user intervention) the character's actions fail, it re-plans a solution. We have achieved this by implementing a search-based planner. This planner is based on a real-time variant of the AO* algorithm, which search the task network corresponding to an agent's role. Within a narrative framework, sub-goals for an agent corresponding to different scenes tend to be independent: this makes possible to compute a solution by directly searching the task network [64]. We have extended the AO* algorithm to interleave planning and execution: we use a depth-first real-time variant that selects and executes the best action rather than computing an entire solution plan, which will be unlikely to remain valid in a dynamic environment where the agents interact with one another and with the user.

This system has been fully-implemented and is able to generate simple but relevant episodes, the outcome of which is not predictable from the initial conditions. A more detailed description can be found in [Cavazza et al., 2001].

6. Conclusion

It should be clear from the range of work discussed in this report that the combination of ideas and technologies from VEs and AI is a very active field indeed, with many different research groups concerned with different aspects. We have necessarily given a partial account, with perhaps less emphasis on ALife virtual ecologies and the incorporation of AI technologies into multi-user shared environments than is deserved. The increase in believability as well as functionality that AI technologies are able to contribute seems to have a positive impact for users insofar as this has been evaluated [46]. Education, training and entertainment seem to provide the most active application areas, including in the latter the somewhat piecemeal uptake of AI technology into computer games.

The amount of activity and energy revealed in this field is a plus: however the lack of agreement on the approaches taken, the absence of widely used common tools or methodologies and the inconsistent use of terms such as *avatar*, *autonomy*, *behaviour* are negatives. While mpeg4 may offer a way out of the political and technical pitfalls of 3D interactive graphics, its facilities for 3D and facial animation are extremely low level. In the same way the H-Anim proposals for hominoid animation lie entirely at the basic geometrical manipulation of the virtual skeleton. If this report contributes towards a greater understanding of both the potential of AI technolgies in VEs, and the facilities and integration needed to make them widely exploitable, then it will have achieved its objective.

Acknolwedgements

We express our thanks to Norman Badler Demtri Terzopolous and Carlos Martinho for the use of some of the illustrations.

References

1. Axling, T; S. Haridi & L. Fahlen, Virtual reality programming in Oz. In: *Proceedings of the 3rd EUROGRAPHICS Workshop on Virtual Environments*, Monte Carlo, 1996
2. Aylett, R; Horrobin, A; O'Hare, J.J; Osman, A & Polshaw, M. Virtual teletubbies: reapplying robot architecture to virtual agents. Proceedings, *3rd International Conference Autonomous Agents*, ACM press pp 338-9 1999
3. Badler, N; C. Phillips, and B. Webber. Simulating Humans: Computer Graphics Animation and Control. Oxford University Press, New York, NY, 1993.
4. N. Badler, B. Webber, W. Becket, C. Geib, M. Moore, C. Pelachaud, B. Reich, and M. Stone, Planning for animation. In N. Magnenat-Thalmann and D. Thalmann (eds), *Interactive Computer Animation*, Prentice-Hall, pp. 235-262, 1996.
5. Badler,N; M. Palmer, R. Bindiganavale. Animation control for real-time virtual humans, *Communications of the ACM* 42(8), August 1999, pp. 64-73. 1999
6. Bandi, S & D. Thalmann. Space Discretization for Efficient Human Navigation. *Computer Graphics Forum*, 17(3), pp.195-206, 1998.
7. Bates, J. The role of emotion in believable agents. *Communications of the ACM* 37(7):122-125 1994
8. Blumberg, B. & Galyean, T. Multi-level control for animated autonomous agents: Do the right thing…oh no, not that. In: Trappl. R. & Petta, P. (eds)

Creating personalities for synthetic actors pp74-82 Springer-Verlag 1997

9. Blumberg, B. Go with the Flow: Synthetic Vision for Autonomous Animated Creatures. *Proceedings, 1st International Conference Autonomous Agents*, ACM press pp 538-9 1998
10. Bouvier, E; Cohen, E. & Najmann, L. From crowd simulation to airbag deployment: Particle Systems, a new paradigm of animation. *J. Electronic Imaging*, 6(1) 94-107 1997
11. Brooks, R. Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, pp. 14-23, 1986
12. Brooks, R. Intelligence Without Representation. *Artificial Intelligence* 47, pp139-159, 1991
13. Calderon, C & M. Cavazza, Intelligent Virtual Environments for Spatial Configuration Tasks. *Proceedings of the Virtual Reality International Conference 2001 (VRIC 2001)*, Laval, France, 2001.
14. Canamero, D. Modelling motivations and emotions as a basis for intelligent behaviour. *1st International Conference Autonomous Agents*, ACM press pp 148-155 1998
15. Cassell, J. Not just another pretty face: Embodied conversational interface agents, *Communications of the ACM*, 2000
16. Cavazza, M; J.-B. Bonne, D. Pernel, X. Pouteau & C. Prunet, Virtual Environments for Control and Command Applications. *Proceedings of the FIVE'95 Conference*, London, 1995.Cavazza, M. & I. Palmer. Natural Language Control of Interactive 3D Animation and Computer Games. *Virtual Reality*, 4:85-102; 1999
17. Cavazza,M. F. Charles, S.J. Mead & Alexander I. Strachan. Virtual Actors' Behaviour for 3D Interactive Storytelling. *Proceedings of the Eurographics 2001 Conference (Short Paper)*, to appear, 2001
18. Chi, D; M. Costa, L. Zhao, and N. Badler: The EMOTE model for Effort and Shape, *ACM SIGGRAPH '00*, New Orleans, LA, July, 2000, pp. 173-182 2000
19. Clay, S.R & J. Wilhelms. Put: Language-Based Interactive Manipulation of Objects, *IEEE Computer Graphics and Applications*, vol. 6, n.2.
20. Codognet, P. Animating Autonomous Agents in Shared Virtual Worlds, *Proceedings of DMS'99, IEEE International Conference on Distributed Multimedia Systems*, Aizu, Japan, IEEE Press, 1999.
21. Codognet, P; Behaviours for virtual creatures by Constraint-based Adaptive Search. In: Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford, USA, pp. 25-30, 2001
22. Damasio, A. Descartes Error. Avon Books. 1994
23. Delgado, C & Aylett,R.S. Do virtual sheep smell emotion? Proceedings, Intelligent Virtual Agents 2001, Madrid, Springer-Verlag, to be published 2001
24. Funge, D.E. Making them Behave: Cognitive Models for Computer Animation. PhD thesis, University of Toronto 1998
25. Geib,C. & B. Webber. A consequence of incorporating intentions in means-end planning.. *Working Notes – AAAI Spring Symposium Series: Foundations of Automatic Planning: The Classical Approach and Beyond*. AAAI Press, 1993.
26. C. Geib, The Intentional Planning System: ItPlans, *Proceedings of the 2nd Artificial Intelligence Planning Conference (AIPS-94)*, pp. 55-64, 1994.
27. Grand, S. & Cliff, D. Creatures: Entertainment software agents with artificial life. *Autonomous Agnets and Multi-agent Systems*. 1(1) 39-57 1998
28. Gratch, J; Rickel, J; & Marsalla, S. Tears and Fears, *5Th International Conference on Autonomous Agents*, pp113-118 2001
29. Grzeszczuk, R, D. Terzopoulos, G. Hinton. NeuroAnimator: Fast neural network emulation and control of physics-based models, Proc. ACM SIGGRAPH 98 Conference, Orlando, FL, July, 1998, in Computer Graphics Proceedings, Annual Conference Series, 1998, 9-20.
30. Jordan, M. & Rumelhart, D. Supervised learning with a distal teacher. *Cognitive Science* 16:307-354 1992
31. Kallmann, M. & Thalmann, D. A behavioural interface to simulate agent-

object interactions in real-time. Proceedings, Computer Animation 99. IEEE Computer Society Press 1999

32. Karlgren, J; I. Bretan, N. Frost, & L. Jonsson, Interaction Models, Reference, and Interactivity for Speech Interfaces to Virtual Environments. In: *Proceedings of 2nd Eurographics Workshop on Virtual Environments, Realism and Real Time*, Monte Carlo, 1995.
33. Koga, Y; Kondo, K; Kuffner, J & Latombe, J. Planning motions with intentions. Proceedings, *SIGGRAPH '94*, pp395-408 1994
34. Laban, R. The Mastery of Movement. Plays Inc Boston 1971
35. Laird, J; Newell, E. & Rosenbloom.P. Soar: An architecture for general intelligence. *Artificial Intelligence* 33(1) 1-64 1987
36. Laird. J. It Knows What You're Going To Do: Adding Anticipation to a Quakebot, *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, Technical Report SS-00-02, AAAI Press, 2000.
37. Loyall, A, & Bates, J. Real-time control of animated broad agents. Proc. 15th *Conference of the Cognitive Science Society*, Boulder Co pp664-9 1993
38. Luperfoy, S. Tutoring versus Training: A Mediating Dialogue Manager for Spoken Language Systems, *Proceedings Twente Workshop on Language Technology 11 (TWLT 11) Dialogue Management in Natural Language Systems*, Twente, The Netherlands, 1996.
39. Magnenat-Thalmann, N & D. Thalmann. Digital Actors for Interactive Television. *Proc. IEEE*, Special Issue on Digital Television, 1995.
40. Martinho, C; Paiva, A. & Gomes, M. Emotions for a Motion: Rapid Development of believable Panthematic Agents in Intelligent Virtual Environments. Applied Artificial intelligence, 14-33-68 2000
41. Mateas, M. *An Oz-Centric Review of Interactive Drama and Believable Agents*. Technical Report CMU-CS-97-156, Department of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 1997.
42. McCarthy, J. & Hayes, P. Some philosophical problems from the standpoint of artificial intelligence. In: B.Meltzer & D.Michie (eds) Machine Intelligence 4, 463-502 Edinburgh University Press, 1969
43. McNeil, D. Hand and Mind: What Gestures Reveal about Thought University of Chicago 1992
44. Milde, J-T The Instructable Agent Lokutor. *Working Notes – Autonomous Agents 2000 Workshop on Communicative Agents in Intelligent Virtual Environments*, Barcelona, Spain, 2000.
45. Monsieurs, P; K. Coninx, & E. Flerackers, Collision Avoidance and Map Construction Using Synthetic Vision. *Proceedings of Virtual Agents 1999*, Salford, United Kingdom, 1999
46. Nass, C; Moon, Y; Fogg, B.J; Reeves, B. & Cryer D.C. Can computer personalities be human personalities? International *Journal of Human-Computer Studies* 43:223-239 1995
47. Ortony, A; Clore, G. & Collins, A. The cognitive structure of emotions. 1998
48. Pelachaud, C. & Poggi, I. Performative facial expressions in animated faces. In: Cassell et al (eds) Embodied Conversational Characters, MIT Press 1999
49. Perlin, K & Goldberg, A. Improv: A system for scripting interactive actors in virtual worlds. In *ACM Computer Graphics Annual Conf.*, pages 205—-216, 1996.
50. Petta, P. & Trappl, R. Why to create personalities for synthetic actors. In: R.Trappl & P.Petta (eds) Creating Personalities for Synthetic Actors, Springer-Verlag, pp1-8 1997
51. Picard, R. Affective Computing. MIT Press 1997
52. Prophet, J. Technosphere. Interpretation, 2(1) 1996
53. Reynolds, C. Flocks Herds and Schools: A Distributed behavioural model. *Proc. SIGGRAPH '87*, Computer Graphics 21(4) 25-34 1987
54. Rickel, J. & Johnson, W.L Integrating pedagogical capabilities in a virtual environment agent. Proceedings, 1st International Conference on Autonomous Agents, ACM Press pp30-38 1997
55. Rickel, J & W.L. Johnson, Task-oriented Collaboration with Embodied

Agents in Virtual Worlds. In: J. Cassell, J. Sullivan and S. Prevost (Eds.), *Embodied Conversational Agents*, MIT Press, Boston, 2000.

56. Schank, R.C. & R.P. Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*, Hillsdale, New Jersey, Lawrence Erlbaum, 1977
57. Schweiss, E; Musse, R; Garat, F. & Thalmann, D. An architecture to guide crowds based on rule-based systems. Proceedings, 1st International Conference Autonomous Agents, ACM press pp334-5 1999
58. Sgouros, N.M; G. Papakonstantinou, & P. Tsanakas, A Framework for Plot Control in Interactive Story Systems, *Proceedings AAAI'96*, Portland, AAAI Press, 1996
59. Sims, K. Evolving 3D morphology and behaviour by competition. Artificial Life 1:353-372 1995
60. Szilas, N. Interactive Drama on Computer: Beyond Linear Narrative. *AAAI Fall Symposium on Narrative Intelligence*, Technical Report FS-99-01, AAAI Press, 1999
61. Terzopolous, D; Tu, X. & Grzeszczuk, R. Artificial fishes: Autonomous, Locomotion, Perception, Behavior, and Learning in a simulated physical world, Artificial Life 1(4) 327-351, 1994
62. Terzopolous, D; Rabie, T. & Grzeszczuk, R. Perception and learning in artificial animals. Proceedings, Artificial Life V pp313-20 1996
63. Thalmann, N.M. & Thalmann, D. The Virtual Humans Story. IEEE Annals of the History of Computing 20(2) 50-1 1998
64. Tsuneto, R.; D. Nau, & J. Hendler. Plan-Refinement Strategies and Search-Space Size. *Proceedings of the European Conference on Planning*, pp. 414-426, 1997.
65. Wauchoppe, K; S. Everett, D. Perzanovski, and E. Marsh. Natural Language in Four Spatial Interfaces. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 8-11, 1997.
66. Webber, B & N. Badler, Animation through Reactions, Transition Nets and Plans, *Proceedings of the International Workshop on Human Interface Technology*, Aizu, Japan, 1995.
67. Webber, B; N. Badler, B. Di Eugenio, C. Geib, L. Levison and M. Moore, Instructions, Intentions and Expectations, *Artificial Intelligence Journal*, 73, pp. 253-269, 1995
68. Weld, D. An introduction to least-commitment planning. AI Magazine 15(4) 27-61 1994
69. Young, R.M. Creating Interactive Narrative Structures: The Potential for AI Approaches. *AAAI Spring Symposium in Artificial Intelligence and Computer Games*, AAAI Press, 2000.
70. Cavazza, M & I. Palmer. Natural Language Control of Interactive 3D Animation and Computer Games. *Virtual Reality*, 4:85-102; 1999.