Synthetic Actors:

development through, and contributing to, stage-play rehearsals

Literature Review

&

Research Proposal

Tony Meyer - Friday, 24 October 2003

1 Literature Review

1.1 A History of the Use of Computers in Theatrical Artistic Processes

The film and music industries have been extensively using computers in artistic processes for decades [1], while theatrical productions have rarely done so. Pinhanez provides a lengthy definition of the term 'computer theatre': "[it] is about providing means to enhance the artistic possibilities and experiences of professional and amateur actors, or of audiences clearly engaged in a representational role in a performance" [2]. This definition "restrict[s] the usage of the term computer theatre to performance situations (ruling out, for instance, user browsing and story-telling)" [2]. In brief, we are concerned with live, real-time, theatrical performances which use computers in the artistic process.

1.1.1 Virtual scenography

One use for computers in a live stage performance is generation of the scenery [3]; this is a logical progression from the use of lighting to set a scene, and also allows more rapid and dynamic scenery, well suited to generations that have grown up in front on the television, or watching movies, rather than at the theatre [4]. A group at the University of Kansas began testing virtual reality as a tool for traditional scenography [5], and then moved on to using virtual reality in productions; their first performance was the production of *The Adding Machine* in 1996 [6].

The Adding Machine [7] used elements of both computerised stages and electronic-puppets, but did not include autonomous computer generated characters, although all computer graphics were generated in real-time [8]. *The Adding Machine* made use of three-dimensional effects by requiring the audience to wear polarised glasses, although the aim of the play was to "explore the expressive and interpretive capabilities of VR without fouling the performance text with the scent of technological gimmickry" [8].

The group at the University of Kansas remain the primary group exploring virtual scenography. *The Adding Machine* was followed up by Samuel Beckett's *Play* [9], which utilised head-mounted displays, rather than polarised glasses. These two productions were followed by an adaptation of Arthur Kopit's *Wings* [10], in 1996, which made extensive use of technology, generating five different 'realities' (similar to [11]) that the audience could view (a rear-projection screen, a scrim screen, via the head-mounted displays, pre-recorded video, and the live actors). *Wings* was followed by David Gustav Fraser's *Tesla Electric* in 1998 [12], *Machinal* in 1999 [13], *A Midsummer Night's Dream* in 2000 [14] and *Dinosaurus* in 2001 [15].

1.1.2 Pre-generated sequences in live performances

Just as computer generated images are integrated into live action sequences in film, *Monsters of Grace*, from the Brooklyn Academy of Music in 1998 [16], integrated computer generated scenes generated prior to the performance into an opera. The music for the opera was conducted and performed live for each performance, but the imagery (projected onto a screen) was developed prior to the performances. Given the control over timing that the conductor possesses, synchronicity was not a concern.

1.1.3 Remote collaboration

The simplest form of remote collaboration is utilising computer technology to allow a number of authors to jointly develop a script for a performance, as Caridad Svich did, using email to collaborate with thirteen playwrights from around America in the creation of the script *Stations of Desire* [17]. Moving the use of technology into the same temporal space as the performance is 'cyberformance': a live performance in that uses online chat technology to bring together remote performers in real time [18, 19].

Adriene Jenik and Lisa Brenneis from the University of California, San Diago, performed Beckett's *Waiting for Godot* as 'Desktop Theatre' [20]. This performance didn't have a live component – audiences watched it on any desktop

computer with an Internet connection via chat software. One of the most interesting components was that the chat room in which the performance took place was open; anyone that happened to wander into the room during the performance was integrated into the story. *Waiting for Godot* was followed by *Santaman's Harvest* (1999), *Spectacled Society* (2000), *World of Park* (2000), and *Water Wars[2]* (2001), which was performed live in Denmark.

Plaintext Players create completely text-based performances using standard Internet software; from 1994 to 2003, they have produced eight series of productions. Under the direction of a director, the performers improvise complex dramas based on prepared but fairly loose scenarios. For each series, the director develops the themes, writes the scenarios (one for each performance), finds and casts the performers, and directs the performances in real time. The director's goal is to provide the combination of elements – a workable structure, a space of mutual trust, clear direction, compelling storylines – that will set the performers free to improvise at a high level of inventiveness [21].

Avatar Body Collision is a group that have presented a number of 'cyberformances', including *BodyBeat* in 2003, *s w i m* in 2002-3, and *Screen Save Her* and *the[abc]experiment* in 2002. Essentially, a live audience watches one of a group of performers live and the rest of the group on a projected screen via a graphical chat program (the remote participants are viewed as avatars, rather than through captured images). Although *Avatar Body Collision* has concentrated on this form of computer theatre, their aim is to explore how technology can be merged with theatre, and how this changes the definition of theatre. They are currently developing new software, *UpStage*, to provide a richer performance.

1.1.4 Synthetic actors

In 1995, Barbara Hayes-Roth suggested that artificial intelligence researches expand their purview to include characters and actors whose performances are designed to create particular kinds of impressions on their human audiences [22]. In *Play*, one of their experiments in virtual scenography, the University of Kansas group utilised a "piercing spotlight" as a "live, visible agent", although the

spotlight was still under direct human control. Synthetic actors are similar to digital characters in film, but are (to varying extents) autonomous. These characters range from those that only interact with other synthetic characters, to those that interact with humans as well; from characters that interact with a single user via a computer to characters that entertain a large audience in a traditional theatre.

From 1996 to 2001 Claudio Pinhanez at the Media Lab at M.I.T. produced various experiments in computer theatre, concentrating on computer, or synthetic, actors. In 1996's *SingSong*, four disembodied heads interacted with a human conductor to perform a short musical item in front of a live audience [23]. All the characters were completely autonomous, working off a known script, and able to adapt to (limited) changes in the performance by the human actor. 1998's *It/I* told the story of a computer actor (*It*) who tries to keep the attention of a human actor (*I*) by presenting various images, generating sounds, and changing lighting. *It* is an example of a disembodied computer actor, similar to the spotlight in *Play* [1]. *It/I* was followed by *It*, an interactive exhibit, based on *It/I* [23]. Pinhanez's actors were able to adapt to temporal changes in the performance, and, to a certain extent, vary their performance each time it was presented [24].

The Virtual Theater project at Stanford University aimed to provide a multimedia environment in which users could fill all the roles associated with putting on a play in an improvisational theatre company, including producer, playwright, casting directory, set designer, music director, real-time directory, and actor. Those roles not filled by the user were filled by intelligent agents. Productions that formed a part of the Virtual Theater project included *the Woggles*, *CyberCafe*, *The Forest Sauvage*, *Tigrito*, and *Master/Servant Scenarios*. The focus of the Virtual Theater project was on improvisation in the classical mode of improvisation first practiced by the Commedia dell'Arte during the Italian Renaissance, rather than performances that were scripted [25]. Their synthetic actors improvised their behaviour according to an abstract scenario that specified a plot, roles and characters [26].

Ideally, synthetic actors are capable of following a script, but also reacting according to its own role. Reilly and Bates state,

> "We are not necessarily interested in lifelike or realistic agents. We don't really care if our characters only have four fingers or are talking animals. The arts have always abstracted away from reality and created characters that were 'larger than life' and that were much more interesting than any mere 'real' people or animals we might know. We [want] interactive versions of these abstracted characters that have been so successful in non-interactive media" [27]

Some synthetic characters are limited to interacting with users via the traditional human-computer interface, rather than in a theatrical setting. Mateas's group aims to develop interactive drama that builds "dramatically interesting virtual worlds inhabited by computer-controlled characters, within which the user … experiences a story from a first person perspective" [28]. A key point is that "it should not just be an interesting demo in a closed door lab, but be experienced by people in the real world. Ultimately, this is the only way to validate the ideas." [28].

Synthetic actors are also, of course, used outside the theatrical domain – for example in commentating soccer games [29], agent research [30], and education [31]. Computers have also been used in other 'creative' areas, such as generating music [32-34] and line drawings [32].

1.2 Improvisation

Improvisation is often used to refer to performance "with little or no preparation" [35], although it has an additional meaning – "to play … extemporaneously, especially by inventing variations" [35]. In fact, many improvised performances are often created by inventing variations, or created "with whatever materials are at hand" [35], more than they are truly with little or no preparation. The aim, however, is **to appear to** have carried out no preparation. Within the boundaries of a scripted play, both rehearsal and performance, it is the 'inventing variations' definition that is of most interest. Anderson and Evens define improvisation as "creating minor to extensive variations on a routine in a satisficing manner in real time" [36].

Synthetic characters are often strongly autonomous – in fact this is usually a stated goal of the project – but characters in a story are not meant to be strongly autonomous. Story characters have a certain amount of freedom of expression, but are required to meet certain goals and reach a specified end. While the characters should convey a strong personality, this personality should be guided by the script, and the characteristics of the personality should result in the actions required to move the story forward.

1.2.1 Improvisation as planning

With this definition in mind, rather than improvisation requiring little or no planning, improvisation can be seen as a form of planning [37]. Improvisational planning complements the long term planning that is already present in a play (the script) by providing plans for the immediate term. The actor uses improvised plans to achieve goals that move the actor towards the goals in the long term plan (script). An advantage of 'planned improvisation' is that the improviser can look ahead (anticipate) and develop improvisation that will more coherently lead towards a goal rather a simple random selection.

Laird's Quakebot [38] utilises anticipation in an attempt to outplay a human opponent in the game Quake [39]; by 'pretending' to be its opponent, the Quakebot improvises a plan to ambush the other player. When improvised plans

are to be used, a difficulty arises in that these plans must be integrated into the overall plan (i.e. the script). Many believable agents ("autonomous characters exhibiting rich personalities, emotions, and social interactions" [28]) are able to perform some sort of improvisation. Comparatively few are able to integrate this "local, reactive behaviour" [28] within the limits of a global story plot.

Blumberg [40] suggests that character-based artificial intelligence can be made easier by using character and situation constraints to prune the search space. In other words, other those regions of the search tree that seem relevant given the emotional state, perception, and attention of the character are expanded. Burke believes that there are twelve dimensions in which contexts may be classified: "time, type of time, geographic location, type of place, culture, sophistication, topic, granularity, modality/disposition, argument-preference, justification, and domain assumptions" [41].

1.2.2 Directed improvisation

Early efforts at 'directed improvisation' involved little more than macros. For example, Hayes-Roth's early system [42] allowed children to direct animated characters. The characters acted by themselves unless given a direction, in which case they would follow the direction. The actions of the characters, however, were simply pre-recorded behaviours (of which the children could build more). Mateas's more advanced 'drama manager' acts as an overseer for a group of synthetic characters, 'pushing' them towards certain goals in order to keep them in line with the script [28]. Similarly, Assanie describes a 'narrative manager' [43], where synthetic characters behave as improvisational actors, but receive instructions from an all-seeing director. The aim for the characters is to be able to cope with unexpected instructions from the director, as well as unexpected actions from the user.

The integration of improvisation and a script falls somewhere on a spectrum between completely autonomous, believable, characters, in a story setting, and characters under the control of an omniscient, God-like director [37]. The centre of the spectrum, the synthesis of the two ideas, involves a director making

observations on what should change and giving the characters a high level goal to carry out; it is then up to the characters as to how the goal is reached. Although this synthesis allows the characters greater freedom, when considered within the confines of a traditional theatre production, the actors do not normally have this freedom.

A difficulty arises in that "the direction can produce behavior that either conflicts with the author's vision for the character or is incongruent with the character's previous behaviour" [43]. In a fully autonomous system (where the director/drama manager/narrative manager is also software), the director would need to be aware of the vision for the character, and take into consideration the character's previous behaviour. Alternatively, the characters must be capable of reasoning over multiple goals and finding a method of removing the conflict in an inconspicuous way. Systems where each character is aware of the script, and does not rely on a 'god-like' director for story instruction, do not have this problem, although each character therefore has more constraints to satisfy.

1.2.3 Timing

When performing improvisation within the limits of a script, a constraint is added that certain events must follow others, and certain events must occur at a specific time [37]. Laborie summarizes [44] the main approaches to fitting five time relationships into an integrated planning and scheduling framework. His work is similar to that of Pinhanez's PNF propagation [24], although only five time relationships are possible, compared to Pinhanez's (exhaustive) thirteen. Like the work of Pinhanez, Laborie's algorithms are efficient even when the set of activities is not completely defined, and when the activities may occur over an extended period of time. Such flexibility is necessary in theatrical work, and especially so when improvisation is taken into account.

Reaction time is also important – the actor cannot take too long to react to an event in the performance. If the actor does not respond to an event within two seconds, it is likely to be labelled 'broken'; within thirty seconds, the actor should demonstrate understanding of the event (or it will be labelled 'confusing') [45].

1.2.4 Evolving improvisation

John Biles has successfully developed an agent, GenJam that improvises jazz music [33], playing along with a human artist. Like theatrical improvisation, improvising jazz is a creative process where the constraints are typically either based on time or the actions of the other agents in the system (here, other actors, or musicians). GenJam uses evolutionary computation by listening to what the human improvises and intelligently mutates and cross-overs this into an improvised response. As a result of the to-and-fro nature of the jazz improvisation, similar to that found in theatrical improvisation, GenJam is able to present its human partner with a reasonable challenge [34] – a desirable effect.

The obvious problem with using a genetic algorithm is that a 'fitness test' is required – and it is very difficult to algorithmically determine the 'fitness' of a creative work, as this derives from aesthetic judgements. These systems usually end up as interactive genetic algorithms, which depend on the feedback of a human mentor, who must experience all the individuals in a population, and indicate each one's merit. Such systems suffer from a bottleneck caused by the need for the mentor to evaluate each individual in the population [46]. This was initially the case with GenJam; however, Biles has recently modified GenJam to operate without fitness evaluation [47], after unsuccessfully attempting to use a neutral network based fitness function. GenJam now uses a more intelligent mutation and cross-over, such that any child will be musically 'valid'.

Another early example of using evolutionary techniques with autonomous creatures is the work of Sims [46, 48], in which both the minds and bodies of creatures are evolved with certain specified aims (such as swimming, walking, following, or capturing an object). Sims suggests that one method of incorporating aesthetic judgements into evolutionary algorithms is to have a human observe the final results of a number of evolutions, and use the human-selected individuals to seed a new evolutionary round. Sims, like others [49], also suggests that competitive evolution is a more effective technique to evolve desired behaviour.

1.2.5 A dramatic perspective

1.2.5.1 Improvisational 'rules'

The performing arts literature outlines various rules for good improvisation, which can be group into two underlying cognitive heuristics [50]. The first is to welcome possibilities; don't plan too far ahead, and just let the performance take you where it will while looking for relationships. The other is to pursue promising possibilities: relate present actions to past actions, keep the action on stage, make the natural response, and accept offers. While traditional agents are forward looking (goal directed), improvisers are backwards looking – trying to reincorporate themes and behavioural qualities that have been previously presented.

These heuristics form the basis of the 'Ten Commandments' of Theatresports:

1. Thou shalt not block.
2. Thou shalt always retain focus.
3. Thou shalt not shine about thy team-mates.
4. To gag is to commit a sin that will be paid for.
5. Thou shalt always be changed by what is said to you.
6. Thou shalt not waffle.
7. When in doubt, break the routine.
8. To wimp is to show thy true self.
9. S/he that tries to be clever, is not; while s/he that is clever, doesn't try.
10. When thy faith is low, thy spirit weak, thy good fortune strained, and thy team losing, be comforted and smile, because it just doesn't matter! [51]

For the most part, these extend past Theatresports into any improvisational theatre [52]. A synthetic actor has some advantages (as well as many disadvantages) in following these commandments; particularly, it is unlikely that a synthetic actor would waffle, lose focus, try to be clever (i.e. show off rather than develop the scene), gag (play for a laugh at the expense of the scene) or intentionally shine above others.

Good, simple, rules for synthetic actors can be extracted from this list, too – particularly, don't block (i.e. accept all offers), don't wimp (refuse to take their share of the responsibility for developing the scene), and (to a lesser extent) breaking the scene can be acceptable. Although it seems like the last rule would result in substandard performance, it, too, is useful, interpreted as "occasional poor performances are acceptable, as long as good performances are the norm"; this is a particular advantage of development through rehearsal over improvisation in a 'live' situation.

1.2.5.2 Status

A widely held belief, mostly traceable to the work of Johnstone [53], is that *status* is a key factor in successful improvisation. The idea is that every movement and nuance of movement implies a status; no action is due to chance; it can be a confusing term [53] unless it is understood as something one does – you may be low in social status, but play high, and vice-versa. Johnstone believes that it is not **what** is said that is important, but **how** it is said (i.e. in which status it is played); it is important that status is played to objects as well as people. Table 1 outlines examples of actions which are of high and low status [53].

High status	**Low status**
Holding a stare	Breaking eye contact and looking back
Long vocal pauses (errrr, ummmm, ...)	Short vocal pauses (er, um, ...)
Holding head still	Jittering about
Toes pointing outwards	Toes pointing inwards
Speaking in complete sentences	Smile with teeth covering bottom lip
Occupying more space	

Table 1: High and low status actions

1.3 Emotion and Affect

A large number of synthetic character systems utilise emotion ([54-60]) – either detecting emotion in order to more accurately interact with human users, or modelling emotion in order to make decisions (either action selection, or action modification [61]). Isla and Blumberg argue strongly that emotion is a key in producing a strong synthetic character [40], as do Prendinger and Ishizuka [62]; however, Knight suggests that the role of emotion in drama is often over-emphasized, and that most 'good' dramas have a plot driven by characters who make purposeful actions towards their goals [63], that is, what a hero wants – not what he or she feels – is what makes a story come to life.

1.3.1 Detecting emotion

While wearable systems are able to easily detect heart rate, skin temperature, muscle tension [57], and many other indicators, an affordable synthetic actors system needs to rely on simpler detection methods, such as face, voice, posture and gesture analysis. It is generally agreed that there are six different emotion families to detect [45, 60]: anger, fear, distress/sadness, enjoyment/happiness, disgust and surprise, which involve distinctive facial expressions – simpler detection methods should be capable of distinguishing between these families, although perhaps not finer analysis within each family.

Massey University's Next Generation Intelligent Tutoring System (NGITS) will include components both for generating affect in the tutoring and for detecting affect and integrating it into a model of the student. This is quite similar to the requirements of a synthetic actor (replacing the model of the student with a model of other actors). While some of the detection methods are not applicable to the theatrical domain (heart rate, for example), facial and voice analysis are; these are also two of the initial domains that are to be utilised by the project. Ideally the output of affect detection systems from the NGITS could be plugged into the input of the synthetic actor system as easily as the student model in the NGITS; unfortunately, at the moment the current (facial) system takes around three minutes per frame to determine emotional state, and the voice analysis system is not yet built [31].

1.3.2 Emotion-based decision making

The somatic marker hypothesis involves the user of a covert, non-conscious biasing mechanism to direct the character towards the 'correct' decision. The main idea is that "decisions that are made in circumstances similar to previous experience, and whose outcome could be potentially harmful, or potentially advantageous, induce a somatic response used to mark future outcomes that are important to us, and to signal their danger or advantage" [60]. In this way, biological systems conserve the memories that are most frequently used [64].

Similarly, Scalar Expectancy Theory [30] pertains to the onset of the conditioned response following a stimulus onset, revealing both "when" and "for how long" the response should occur. According to the theory, when salient stimulus is perceived, an internal timer is started that records the (subjective) interval between that stimulus and another salient stimulus, such as a reinforcing stimulus. Later, when the first stimulus is perceived again, the animal starts another timer, and decides when to respond by using the ratio of the elapsing interval to the remembered interval. When the ratio exceeds a threshold, the subject responds [30].

The 'emotional system' of [60] includes two thresholds: the first is used to determine when an 'emotional episode' occurs – once the intensity goes above this threshold, the system releases its output signal to other emotional systems and to the behaviour system (which selects an appropriate behaviour according to that state). The second threshold level specifies a level of saturation for that emotion, which is based on biological emotional systems in which levels of arousal will not exceed certain limits. Each emotional system also has a function for the temporal decay of its intensity and all inner obstacles have an equilibrium value that they decay back toward at a scaleable rate. By specifying the equilibrium values and decay rates for a character's inner obstacles, the character's personality (and therefore behaviour) is defined.

Similarly, the 'actiontuples' of [64, 65] are vectors that define how rewarding it is to perform an action in a given context; by creating new actiontuples with more specific contexts, the actor gains a more concise understanding about which aspects of the context, followed by the action, produce a reward or punishment. The system uses back-propagation to determine the value of the actiontuples. Each action may itself be parameterised, so that the actor can learn to modify aspects of an action to make it more rewarding/less punishing; also similar are the 'drivevectors' of [30]. The variables here 'drift' towards certain values in the absence of any other input (i.e. an equilibrium value), and also have a point at which the drive is considered satisfied. The strength of the drive is proportional to the magnitude of the difference between these two points.

1.3.3 Expressing emotion

There are many ways for an actor to express emotion. The simplest is through action selection (as outlined in the section above), where only actions that suit the current emotion are executed. Additionally, actions may be modified to suit the current emotion (walking, for example, may suit any of the emotions, but may be modified into a 'happy' walk, or 'angry' walk) [45]. The reactions of other characters also contribute to the conveyance of emotion [45]; this includes both human actors reacting appropriately to synthetic actors and synthetic actors reacting appropriately to human actors (and also human-human and synthetic-synthetic reactions).

The addition of affect to synthesised speech is useful in presenting dramatic material [66]; it has been possible for over a decade to take a description of affect in speech and reproduce its significant acoustical features in synthesized speech. The primary parameters that govern the emotional content of speech are loudness, pitch (level, variation, range), and prosody. Cahn describes a system for mapping emotional quality (e.g., sorrow) onto speech synthesizer settings, including articulation, pitch, and voice quality [29]; for example, happiness is usually expressed with faster speech and a higher pitch, and sad speech is typically slow and low pitched [62].

1.3.4 Mood, personality and attitudes

Like emotion, modelling mood, personality and attitude can improve an actor's performance – Rousseau and Hayes-Roth consider that personality traits, moods, and interpersonal relationships are essential for a synthetic actor that has to give a believable performance [26, 67]. The personality of a character is fixed – it cannot be modified during a performance [26] (and probably not during the rehearsal period, either); personality determines a person's adjustment to the environment [25], and is what makes one person act differently to others. Comparatively, moods and interpersonal relationships are changed by actions and events during a performance [26]. Moods correspond to emotions (e.g., happiness or anger) and sensations from physical needs (e.g., fatigue, hunger, or thirst). Interpersonal relationships, or attitudes, tend to be more stable than moods, although they change more rapidly than personality; examples include status, degree of sympathy and trust [25].

2 Proposed Research

2.1 Synthetic Actors

Believable synthetic characters are simple, but complete, three-dimensional situated agents who can do and express the right things in a particular situation or scenario [61]. Examples of synthetic characters include non-player characters in computer games, digital 'extras' in films, computer-based artificial pets, and those outlined in section 1.1.4.

Synthetic actors are agents that work collaboratively with other agents [68] – the other (human and not) actors in the performance. The aim of the agent is to maximise the quality of the performance given; measuring this quality is difficult, though, as it involves an aesthetic judgement. However, by the time the performance reaches the 'end-users' (i.e. the audience), the performance is, for the most part, fixed. It is during the rehearsal stage that the quality (in subsections) of the performance is evaluated and, as necessary, altered. During the rehearsal, a much easier performance measure is available – and is supplied to human actors in a 'normal' rehearsal – feedback from the director. This follows the typical format for a rehearsal – that both the cast and director make suggestions, but that only the director makes decisions [69].

2.2 Aims

2.2.1 Overall aim

As such, the aim of the synthetic actor can simply be to perform in such a way that maximises the positive feedback and minimises the negative feedback from the director. Ideally, of course, the actor should be able to infer this from comments that the director makes, rather than having the director 'score' each performance explicitly. The aim of this project is the development of a model that, when implemented, provides control of a synthetic character in such a way that it rapidly learns the most appropriate way to improvise, such that feedback is good, and the constraints of the script and character are not violated. The aim is not to be creative as per the definition of [70] - 'to do something which they *could not* have done before', but rather 'to do something which they *would not* have done before'.

2.2.2 Character design aims

As in [65], the characters should be robust (work with imperfect knowledge), reactive (react appropriately), adaptable (able to learn from experience), expressive, sensible, and scalable (the ability to have a whole cast of synthetic actors); unlike [65], the actors do not need to be honest, as they have knowledge about what the future will bring, but do need to be true to the script (consistent [71]), and to their character, instead. The characters should also achieve the design goals of [43] (see figure 1).

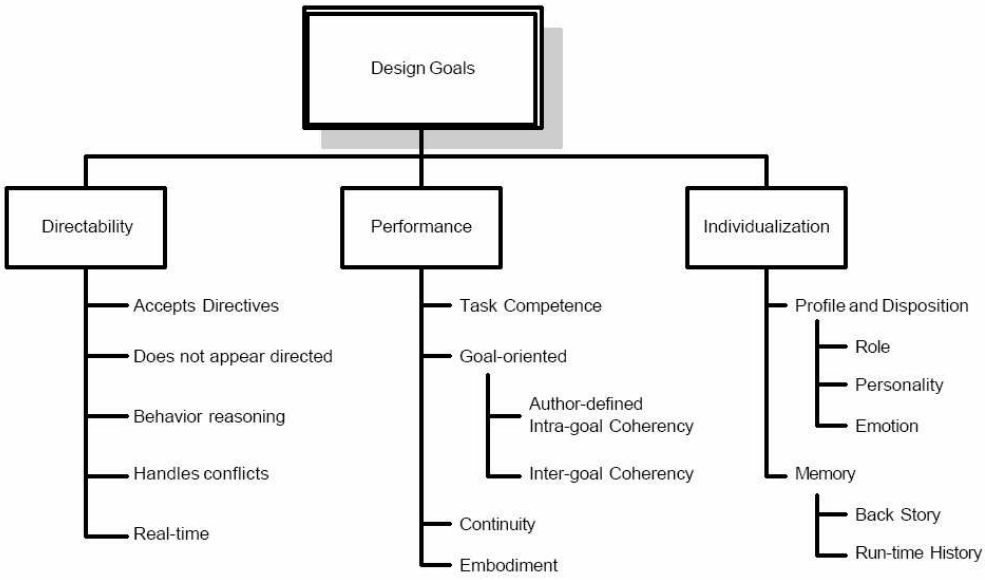

Figure 1: Design Goals

2.3 Justification

Dramatic characters represent a kind of 'middle ground' between the agents of today and human beings themselves. The dramatic character has been honed, over centuries, to be as believable and life-like as possible. Understanding the nuts and bolts of drama may shed light on how to construct believable agents of all kinds, regardless of application; surprisingly, the nuts and bolts of drama look more like classic agent architecture than might be guessed. Dramas, like agents, are driven by objectives, goals, actions, and tactics [63]. Synthetic characters can produce performances that are motivated, believable and theatrically interesting [26]. Synthetic characters have much to offer live stage performances (augmenting existing theatre tools, not replacing existing tools [72] – "who wants to play 'man in background #4 anyway?" [65]), but the techniques used to create and teach these characters should also be applicable to education [73], film [74], games [65] and other entertainment [40, 75], and interface development [76], among other areas.

2.4 Current experimental system

I have developed the basis of an experimental system which will allow synthetic character models to be tested in a theatrical setting. The system does not currently include any facilities to interact with a director in a rehearsal setting, or any form of intelligent core, but does include a number of actuators and sensors (see figure 2).

2.4.1 Underground Theatre

The intention is for the experimental system to be used in Massey University's Underground Theatre in Auckland (see figure 3), and integrate tightly into the theatre's existing systems. The theatre includes a stage 10 meters wide and 5 meters deep, 0.9 meters above the ground, with black curtains covering the upstage wall, except for the upstage-centre wall, which has a cyclorama cloth on

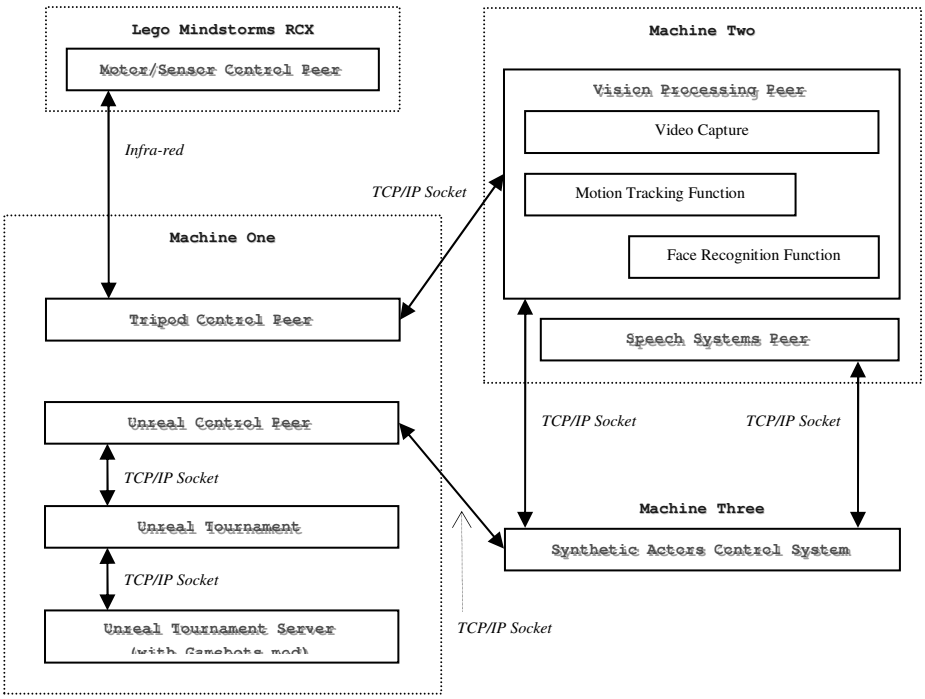

Figure 2: Experimental System Layout

which the synthetic characters can appear; it is possible that over the next six months a second screen on which the character can appear will be installed into the venue ('distributed rendering' [65] is easily handled by the graphics system, as in [77]). The theatre sound system allows input from the various computers that

form part of the synthetic actor system, and is able to output any mix of inputs to

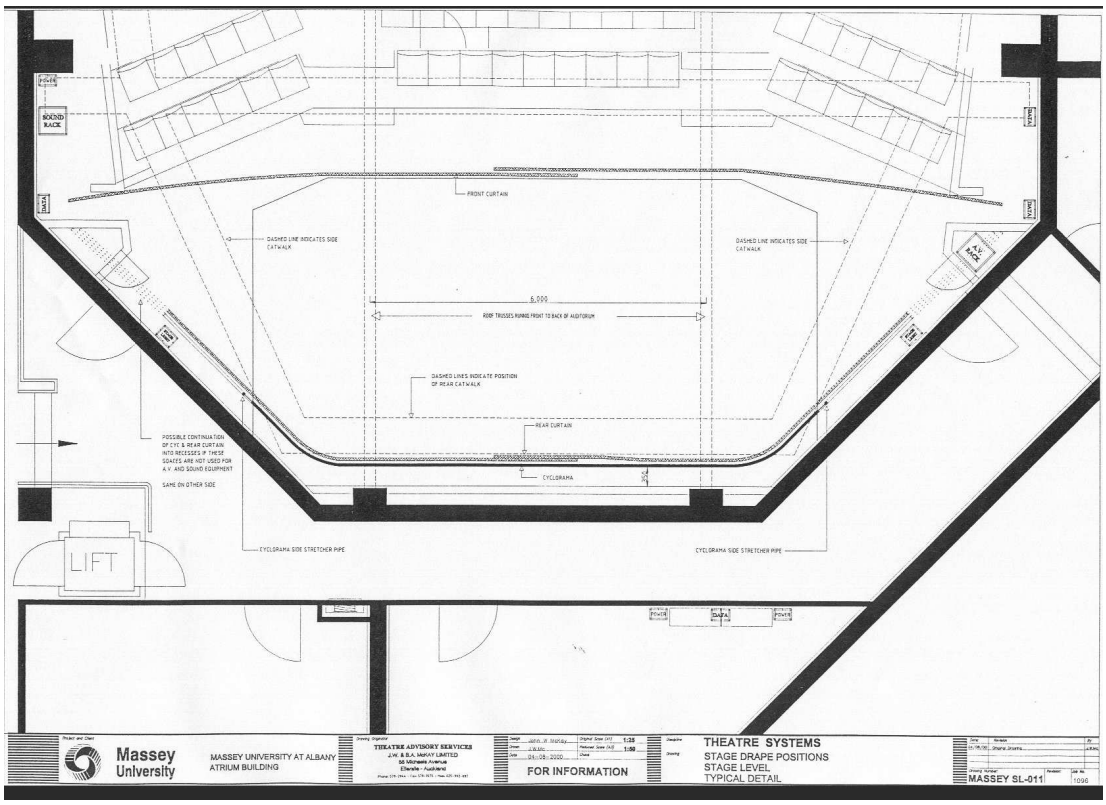

Figure 3: Underground Theatre Layout

the audio recognition module. The theatre is wired with a microphone in the ceiling which picks up all sound in the room at a suitable level and quality that the audio recognition module is able to make use of it (eliminating the need to have all actors wear a microphone).

2.4.2 Graphics component

The graphics component uses the graphics engine from the commercial game Unreal Tournament (as in [37, 78]). Commercial games typically include a complex run-time environment, including rendering of 3D graphics, internet-based client-server functionality, and real-time multi-user synchronisation [79, 80]. Another advantage is that there are usually many freely available and reusable 'mods' online, for example, *Esc Online* was developed to function as a forum for creating autonomous (or semi-autonomous) characters that interact with

real people. It combines music, dance and chat and uses the Unreal Tournament graphics engine and is open-source.

The characters from *Esc Online* are more suitable to a theatre setting than the warrior-type characters that are installed with Unreal Tournament, and so these are used in the current demonstration. These characters have a range of basic movements, including running, walking, turning, various dance moves, falling over, and waving. The graphics system is not limited to humanoid characters, however (bird and fish models are also available in the current system); Blumberg believes that as the character's mind cannot be realistically human, the body should not look human either [45].

UnrealScript has most of the features of Java. Like Java, it is byte compiled to a platform independent byte code for a virtual machine that is simulated at run time. UnrealScript simulates threads by calling each object's code in turn. This drastically simplifies inter-thread communication and eliminates the overhead of context switching, allowing thousands of independent threads. C (or any other language) can be used for computationally intensive tasks, or to extend UnrealScript's capabilities. Unreal Tournament's navigation and path planning, central components of First Person Shooters, are accessible through functions such as MoveTo(vector) and MoveToward(object) [78].

The GameBots module [81], which provides an interface between Unreal Tournament and any other program capable of using a standard TCP/IP socket, is written completely in UnrealScript. Code describing the low-level movement of characters (walking, waving, dancing, and so on) is written in UnrealScript, controlling standard 3D models. This code is called by other UnrealScript code, which handles network connections and commands given to the component, via a standard TCP/IP socket connection. A higher level interface to this component is coded in Python, which takes commands (via a socket) and sends the appropriate commands to Unreal Tournament.

2.4.3 Audio recognition and speech synthesis component

The audio recognition component currently uses the Microsoft Speech API [82] (included with Microsoft Office XP) for both speech recognition and speech synthesis. The free and open-source speech generation packages Festival [83] and Festival-Lite (Flite) have also been investigated, and may be used as an alternative – the Microsoft system has the advantage of simplicity and integration into the Windows operating system, but the Festival system has a wider variety of voices, more 'natural' sounding voices, and is cross-platform. The free and open-source speech recognition package Sphinx [84] (II and III) has also been investigated, although to a lesser extent, and is likely to be used if Festival is providing speech synthesis; Sphinx appears to have more of an ability to recognise non-speech sounds, and is cross-platform.

Whichever recognition and synthesis packages are used, these are wrapped in a Python module, which accepts standard TCP/IP socket connections. All incoming data is spoken, and all recognised data is transmitted to the connected peer. When the Microsoft Speech API is used, the Python module utilises COM to connect to the API; when Festival/Flite and Sphinx are used these are linked to as an extension module.

2.4.4 Vision and tripod components

The vision system [85] is made up of two Logitech Quickcam Express web cameras mounted on two tripods made of Lego that can pan left and right (in an arc of approximately ninety degrees). This component is the least ready to be used in a production, as the exact requirements for the component are likely to vary depending on the script. The system can currently detect (front-facing) faces, track these faces (panning the camera left and right), and detect motion; the system can also operate in a mode where one camera searches for motion, and when detected instructs the other camera to move towards that motion and start searching for faces.

2.4.4.1 Tripod

The two (effectively identical) tripods are built out of standard Lego Technics blocks; using Lego allows rapid modification of the tripod design, and is cost effective [86]. The tripods are basically a motor (geared down twice, first by a ratio of 7:20, and then by a ratio of 1:5, giving a total ratio of 7:100) turning the spherical camera. The high gear ratio is necessary to generate enough power to turn the heavy camera, especially as the camera's USB cable (connected to a computer) adds additional drag. The tripods also have a sensor which is activated when the camera reaches the apex of the panning arc; a wireless camera could be allowed to turn in a never-ending circle, but as the cameras are wired, the cable would end up wrapped around the tripod – only an arc of approximately ninety degrees is necessary to cover the entire stage area. One tripod also includes a Lego Mindstorms RCX [87], which includes an infra-red sensor and simple programmable computer.

The software controlling the tripods is made of two separate components. One component resides on the RCX and converts infra-red signals to movement of the two motors. The software currently accepts eight commands: move camera 1 or 2 slightly to the left or right, start camera 1 or 2 panning from left to right, start both cameras panning from left to right, and stop all movements; when panning, if the sensor at the apex of the panning arc is activated, the RCX software reverses the direction of the pan (independently of any infra-red commands). This program is written using the Lego Mindstorms graphical program builder, as it is not complex.

The second component of the tripod system resides on a standard computer and accepts high level commands (e.g. "pan camera 1 left") and sends the appropriate infra-red commands (via the Lego Mindstorms USB Infra-red Tower) to the RCX. The code which sends the commands to the tower is extracted from Dave Baum's NQC package [88]; although the NQC software is designed to be an alternative language for programming the RCX, it also includes code to send commands to the tower, which is written in C. There are variants of this code for Microsoft Windows, MacOS, and Linux. This code is wrapped into a Python module that performs the socket communication and other utility code.

2.4.4.2 Vision algorithms

Capturing data from the two cameras is done with code based on the Python VideoCapture module [89]. Once captured, the data is passed to an arbitrary set of functions, which yield a set of high level data to pass to the peer connected to the vision module (again, via a standard TCP/IP socket), for example "face detected at 104,21 on camera 1". These functions can also pass commands to the tripod component, for example to try and centre a camera on a detected face. The capturing system retrieves images at a resolution of 352 by 288 (640 by 480 is possible, but retrieving data at this resolution from two cameras over a single connection would exceed the available USB2 bandwidth) at around 150 frames per second (with no functions processing the data).

Two functions are currently implemented to process data captured from the cameras, both based on functions from Intel's Open Computer Vision (OpenCV) library [90]. One detects (front-facing) faces in the captured image, and the other detects motion over a series of captured images. Further image processing is possible, but has not been implemented yet as it is likely that the requirements will be semi-specific to the particular play that is performed (for example recognising a particular stage prop, recognising a particular person (out of the cast), or using the face detection (and possibly gesture recognition) to do some form of emotional analysis [31]). The image processing, particularly the face detection, introduces a significant bottleneck into the component, and the frame-rate drops to around two frames per second; this is, however, likely to be sufficient for the task [85].

The face detection algorithm performs rapid object detection with a cascade of boosted classifiers based on Haar-like features [91]; this algorithm is not limited to detection of frontal-facing faces, but a pre-trained face classifier is included in the OpenCV library package. If other objects were to be recognised, the same algorithm can be used; however, this requires a large time and data commitment to training, and faster methods may be possible. The underlying representation used for motion tracking is a Motion History Image (MHI) that temporally layers

consecutive image silhouettes into a single template form. The MHI system was designed to work in real-time and considered entertainment such as computer theatre as one of the potential applications, as Dr. Aaron Bobick worked on both the MHI [92] and Pinhanez's computer actors [93].

2.4.5 Interval engine

The interval engine code from [94] has been significantly tidied up and ported to Python (for reliability and ease of modification). I expect that automatic generation of an interval script that matches a given script (in 'traditional' form) will not be a difficult task. The interval engine can then be utilised to provide 'cues' to the central controlling module of the synthetic actor. The actor would then be able to evaluate whether to follow the timing cue, or not, as appropriate. It seems likely that only those events significant enough to be outlined in the 'traditional' form of the script (spoken lines and some stage movements) are significant enough to be included in the interval script. If this does not provide sufficient control over timing, the actor could have the ability to modify (or supplement) the existing interval script added.

2.4.6 Future enhancements

The most obvious future enhancements that will be required are: development of the core 'brain' of the system, development of additional vision data processing functions (such as facial 'action units' recognition [29]), and development of additional 3D models and movements. I intend to write the former as a Python module, which (like the other components) uses standard TCP/IP sockets to communicate with the other components.

Development of additional models and movements is fairly straightforward, and can be done with almost any 3D modelling software (such as 3D Studio Max or Poser) and converted into the Unreal Tournament format; code to control the new models and movements is also fairly straightforward and is not likely to be significantly different from the existing code. Should complex models be required (or desired), I may be able to find someone with 3D modelling expertise

who is willing to create models for me; alternatively, many models are freely available via the Internet. Unreal Tournament 2003, the sequel to Unreal Tournament, has recently dropped in price significantly; as the components that work with the game are all meant to be upwards compatible, it is likely that this will replace Unreal Tournament as the graphics engine, providing superior graphics at little cost.

If, in the test production, speech forms an important part of the synthetic character(s), then I will spend a small amount of time adding (limited) affect to the synthesised speech; this should be possible without a significant amount of effort, adapting pitch and the rate of speech [66]. It is likely that Festival can be easily adapted to do this, as it supports Sable (a SGML for speech [29]). Cahn et al. argue that linguistic style is a key aspect of character [95].

It is likely that various planning, learning, and improvising techniques will be experimented with in the central core of the system (a primary reason for using Python is the ease of rapid development and testing). Some of the techniques have existing software implementations that are likely to be able to be used with little (or no) modification. These include the techniques outlined in [68] (available from [96]), the Bayesian classifier from SpamBayes[1] [97], and PyGP (an implementation of Koza's genetic programming techniques in Python) [98]. All these implementations are open source, which allows simple modification as required for the project (and any generic improvements can be contributed back to the implementation projects).

[1] Although the primary purpose of the SpamBayes project is classification of email, only the tokeniser is email-specific; the classification code will work with any set of tokens, and has been used for non-email purposes.

2.5 Methodology

The main method of testing I intend to use is testing various implementations of synthetic actors in a specially-written performance. During each rehearsal new models or techniques can be tested and evaluated, using the director's feedback and my own evaluation as results. The following section outlines a rough plan for the next twelve months and then a series of hypotheses that I would like to test, including a brief description of the technique that I intend to use to test them. I expect that other hypotheses will be formulated over the rehearsal period; for the most part, I hope to be able to independently test each hypothesis, and test them in combination (within the limits of the rehearsal time).

2.5.1 Twelve month plan outline

November 2003	Either find a playwright, or give up on finding a playwright and create a suitable stage-play based on a short story.
December 2003	Find a director and cast for the play, most likely drawing on the performing arts group on campus (Massey @ Albany Drama, Singing, Operatic and Dance Society) as well as any other interested parties. Create the basic 'brain' of the synthetic actor, and tools to interact with the director, such as the ability to start at any point in the script, the ability to understand basic (oral, through a microphone) commands (and possibly more complex typed commands), and a tool to load a traditional script into the system.

February 2004	Begin rehearsals for the play, holding a one to two hour rehearsal once per week until the end of August. Each rehearsal is basically a test of the current abilities of the actor, which will begin without any improvisational ability at all, and develop through the rehearsal period. Record (on videotape) each rehearsal for further analysis, as well as storing extensive logs from the various components (the existing components already have logging capabilities). During each week, enhance the abilities of the synthetic actor system. This is likely to involve creation of a system which should result in good performance, rehearsing without any other actors, testing at a rehearsal, then modification of the system. Via this method, test each of the hypotheses laid out below. As stated in [61], testing several different systems should prove beneficial – the existing experimental system should easily allow a new component to be 'plugged into' the system.
June 2004	Update literature review as necessary.
August 2004	Finish rehearsals, and prepare for a public performance of both the final product, and a rehearsal (which is where the capabilities of the system would really be demonstrated).
September 2004	Hold public performance and rehearsal.
October 2004	Wrap up development of the system apart from minor experiments. Alternatively, depending on how the previous year has done, develop one or two small plays that can be used as test arenas for new theories.

2.5.2 Range of improvisation

2.5.2.1 Hypothesis

A wide range of improvisation is possible with a small number of variables.

2.5.2.2 Test

One of the first tasks is enabling forms of improvisation. Given that an almost limitless number of variables can be introduced into the actions that the synthetic actor takes, a suitable set must be chosen. I suspect that enabling only a small amount of variation in each action (e.g. pitch and rate in speech, speed and 'smoothness' in movement, and reaction time) will result in a wide range of improvisational ability, as a result of the combinations of the various variables, and the level of change given a small change in one of the variables. To test this, I will compare the results of rehearsing with a various numbers of variables enabled.

2.5.3 The role of status

2.5.3.1 Hypothesis

Including a model of status (as in [53]) into the synthetic actor model, and the ability to deduce the status of other actors will result in improved improvisation.

2.5.3.2 Test

Given the number of synthetic/believable character projects that reference Johnstone's work, it seems likely that this hypothesis will prove true. I will add an indicator of status to the synthetic actor (perhaps a separate component responsible for maintaining the status of both the actor itself and other actors). This variable can then become another which may be manipulated to alter the performance (as in section 2.5.2 above), although it is likely that the status of a character remains fairly constant. I hope to be able to establish a method of automatically determining an approximate level of status, perhaps from a character description, utilising the arts literature.

Estimating the status of other actors within the performance is a much more difficult task; however, I hope to be able to create an algorithm that will be reasonably successful at this, too. I suspect that the most relevant information will be the actor's own status and the relationship between the two actors; again the arts literature will be of use here.

2.5.4 Self-rehearsal

2.5.4.1 Hypothesis

An actor that rehearses on its own (using its own judgement of fitness) will more rapidly produce highly fit improvisation during a rehearsal.

2.5.4.2 Test

I will enable synthetic actors to receive feedback either from the director, or from some other source. I will create some sort of 'automatic' feedback (perhaps based on prior experience, degree of difference from the original, and/or generally accepted theatrical processes) and try rehearsals where the actor has already established a set of improvisations to attempt, and rehearsals where all improvisation is in real-time, and compare the results.

2.5.5 Use of 'smart' evolutionary techniques

2.5.5.1 Hypothesis

Use of techniques for evolutionary computation (such as genetic programming), combined with human knowledge of good improvisation techniques (as in [47]), will result in rapid generation of highly fit improvised scenes.

2.5.5.2 Test

Using a simple genetic programming framework, the techniques outlined in [47], and the generally accepted rules of improvisation [51], I will evolve a number of improvised scenes and evaluate the results from including these in a rehearsal. Assuming that the computation is not overly expensive, evolution of new

improvised scenes (assimilating the feedback from the director) can be carried out during rehearsals.

2.5.6 Competitive evolution

2.5.6.1 Hypothesis

If evolutionary computation techniques prove effective, then using competitive evolution (various actors, perhaps differing in personalities), will prove more effective than otherwise; this is perhaps similar to auditioning different human actors for a role.

2.5.6.2 Test

This should be a fairly straight-forward addition to the previously discussed genetic programming code.

2.5.7 Emotion

2.5.7.1 Hypothesis

Integrating basic emotion modelling into improvisation will allow the actor's personality to be more apparent than straight improvisation.

2.5.7.2 Test

Section 1.3 outlines several techniques for modelling emotion, moods and personality – many are in fact very similar. Adding a relatively simple implementation of one of these models to the synthetic actor system should not be overly difficult. The actor can then be tested utilising emotional information, and the results compared to rehearsal without this information.

2.5.8 Learning from human actors

2.5.8.1 Hypothesis

Observing and then imitating human techniques at improvisation (as in [47]) will allow actors to produce fitter improvised scenes.

2.5.8.2 Test

Creating improvisation rules from the data recorded is likely to be a difficult task. One observation that should not be overly difficult is timing – the vision and speech systems should be able to match actual events to expected events and note that differences in timing between different runs. This information can be matched to feedback from the director (keeping the performance of the synthetic actor constant) and used to create timing rules that may assist with improvisation. I suspect that this will not result in any noticeable improvement, because it will need to have additional context information added (e.g., 'this is the right timing rule, **when** …'). I hope to be able to also try extracting some context from the situation (perhaps from the script, or from a model the synthetic actor could generate of the human actors' emotions), and see if this makes a difference. If this is successful, then observing more than timing could be attempted.

2.5.9 Bayesian learning

2.5.9.1 Hypothesis

Using Bayesian techniques (as in [99-101]), an actor will be able to learn which patterns of improvisation are best with only positive examples and no (or few) negative examples (this is also similar to the ideas outlined in [102], although that does not utilise Bayesian techniques).

2.5.9.2 Test

It should not be difficult to create a tokenising function that converts the parameters of an improvised performance into a series of tokens. Once this is created, it can form the input into a Bayesian classification engine and give an estimation of how close the improvised performance would be to 'good' or 'bad'. The actor would then be able to evaluate a set of possible improvised behaviours for a scene to be rehearsed, and select the behaviour closest to 'good', based on results learnt from previous scenes.

2.6 Summary

My aim is to develop a system that allows a synthetic character to develop good improvisational techniques through the paradigm of stage-play rehearsals. I hope to explore a variety of different artificial intelligence techniques (for both learning and planning) to do this, from traditional planning algorithms through to evolutionary computation methods. I also hope to develop a good model for a synthetic actor, based on both the work in the literature, and on my own requirements. This model should include knowledge (particularly about the script and characters) and affect data (not unlike that of [31]), and should be generic enough that it can be utilised outside the theatrical domain, if the appropriate techniques are replaced.

References

[1] C. Pinhanez and A. Bobick, ""It/I": A Theater Play Featuring an Autonomous Computer Graphics Character," M.I.T. Media Lab, Perceptual Computing Section Technical Report No. 455, January 1998.

[2] C. Pinhanez, "Computer Theatre," M.I.T. Media Lab, Perceptual Computing Section Technical Report 378, 1996.

[3] M. Reaney, "The Theatre of Virtual Reality: Designing Scenery in an Imaginary World," *Theatre Design and Technology*, vol. 24, pp. 29-32, 1992.

[4] M. Reaney, "Digital Scenography: Bringing the theatre into the information age.," presented at L'Art et le numerique, 2000.

[5] M. Reaney, "Art in Real-Time: Theatre and Virtual Reality," presented at cirEN: Art - Science - Technologie, Paris University, France, 2000.

[6] M. Reaney, "Virtual Scenography: The Actor/Audience/Computer Interface," *Theatre Design and Technology*, vol. 32, pp. 36-43, 1996.

[7] M. Reaney, "The Adding Machine: A Virtual Reality Project", [online] 1995, http://www.ukans.edu/~mreaney/machine/ (Accessed: May 12 2002)

[8] L. Gharavi, "Theatre and Virtual Reality," *Profile*, vol. 58, pp. 10-14, 1997.

[9] M. Reaney, "Play", [online] 1996, http://www.ukans.edu/~mreaney/play/ (Accessed: May 12 2002)

[10] M. Reaney, "Virtual Reality Sprouts Wings", [online] 1996, http://www.ukans.edu/~mreaney/wings/ (Accessed: May 12 2002)

[11] W. Piekarski and B. Thomas, "ARQuake: The Outdoor Augmented Reality Gaming System," *Communications of the A.C.M.*, vol. 45, pp. 36-38, 2002.

[12] M. Reaney, "Tesla Electric", [online] 1998, http://www.ukans.edu/~mreaney/tesla/ (Accessed: May 12 2002)

[13] M. Reaney, "Machinal", [online] 1999, http://www.ukans.edu/~mreaney/machinal/ (Accessed: May 12 2002)

[14] M. Reaney, "A Midsummer Night's Dream", [online] 2000, http://www.ukans.edu/~mreaney/midsummer/ (Accessed: May 12 200)

[15] M. Reaney, "Dinosaurus", [online] 2001, http://www.ukans.edu/~mreaney/dinosaurus/ (Accessed: May 12 2002)

[16] P. Griffiths, "Shock of the Surreal In a Digital World," in *New York Times*, Late ed. New York, 1998, pp. 1.

[17] H. V. Jamieson, "From paper and ink to pixels and links," *Open Page*, 2001.

[18] H. V. Jamieson, "Cyberformance", (Cyberformance), [online] 2002, http://www.creative-catalyst.com/cyberformance/ (Accessed: 19th August 2003)

[19] S. J. Norman, "Dramatis Personae: Casting Cyberselves," presented at Fifth International Conference on Cyberspace, 1996.

[20] S. Rosenberg, "Clicking for Godot," in *Salon*, 1997.

[21] P. Players, "FAQ", (Plaintext Players), [online] 2002, http://yin.arts.uci.edu/~players/ (Accessed: 19th August 2003)

[22] B. Hayes-Roth, "Agents on Stage: Advancing the State of the Art of AI," Department of Computer Science, Stanford University, Knowledge Systems Laboriatory Report 95-50, May 1995.

[23] C. Pinhanez, "Representation and Recognition of Action in Interactive Spaces," Massachusetts Institute of Technology, 1999.

[24] C. Pinhanez and A. Bobick, "Human Action Detection Using PNF Propagation of Temporal Constraints," M.I.T. Media Lab, Perceptual Computing Section Technical Report 423, 1997.

[25] D. Rousseau and B. Hayes-Roth, "Improvisational Synthetic Actors with Flexible Personalities," Department of Computer Science, Stanford University, Knowledge Systems Laboratory Report 97-10, December 1997.

[26] D. Rousseau and B. Hayes-Roth, "Personality in Synthetic Agents," Stanford, Knowledge Systems Laboratory Report KSL-96-21, 1996.

[27] W. S. Reilly and J. Bates, *Natural negotiation for believable agents*. Pittsburgh, Pa.: School of Computer Science Carnegie Mellon University, 1995.

[28] M. Mateas and A. Stern, "Towards Integrating Plot and Character for Interactive Drama," presented at AAAI Fall Symposium on Social Intelligent Agents: The Human in the Loop Symposium, 2000.

[29] K. Binsted and S. Luke, "Character Design for Soccer Commentary," *Lecture Notes in Computer Science*, vol. 1604, pp. 22-25, 1999.

[30] R. C. Burke and B. M. Blumberg, "Using an ethologically-inspired model to learn apparent temporal causality for planning in synthetic creatures," presented at AAMAS, 2002.

[31] NGITS, "The Next Generation Intelligent Tutoring System", [online] 2003, http://iims/research/ngits/index.html (Accessed: 20 October 2003)

[32] M. A. Boden, "Creativity," in *Artificial Intelligence*, *Handbook of Perception and Cognition*, a. A. Boden, Ed., 2nd ed. London: Academic Press, 1996, pp. 267-289.

[33] J. Biles, "GenJam: A Genetic Algorithm for Generating Jazz Solos," presented at ICMC, 1994.

[34] J. A. Biles, "GenJam: Evolutionary Computation Gets a Gig," presented at 3rd Conference on Information Technology Curriculum, Rochester, NY, 2002.

[35] Houghton Mifflin Company, *The American Heritage® Dictionary of the English Language*, 4th Ed. ed: Houghton Mifflin Company, 2000.

[36] J. Anderson and M. Evens, "Constraint-Directed Improvisation," presented at Eleventh Biennial Canadian Society for Computational Studies of Intelligence Conference, Canada, 1996.

[37] B. Magerko, "A Proposal for an Interactive Drama Architecture," 2002.

[38] J. E. Laird, "Research in Human-Level AI Using Computer Games," *Communications of the A.C.M.*, vol. 42, pp. 32-35, 2002.

[39] J. E. Laird, "It knows what you're going to do: adding anticipation to a Quakebot," presented at Fifth International Conference on Autonomous Agents, Montreal, Canada, 2001.

[40] D. Isla and B. Blumberg, "New Challenges for Character-Based AI for Games," presented at AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment, Stanford University, 2002.

[41] R. C. Burke, "Learning “The Hard Way”: Negative Thinking in Context Spaces", [online] 2000, http://web.media.mit.edu/~solan/society_of_mind/essay2.htm (Accessed: July 21 2003)

[42] B. Hayes-Roth, L. Brownston, and E. Sincoff, "Directed Improvisation by Computer Characters," Stanford Univeristy Knowledge Systems Laboratory, Palo Alto, California, Technical Report KSL95 -04, 1995.

[43] M. Assanie, "Directable Synthetic Characters," 2002.

[44] P. Laborie, "Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results," *Artificial Intelligence*, vol. 143, pp. 151-188, 2003.

[45] B. Blumberg, B. Tomlinson, and M. Downie, "Multiple Conceptions of Character-Based Interactive Installations," *Computer Graphics International*, pp. 5-11, 2001.

[46] K. Sims, "Evolving Virtual Creatures," *Computer Graphics*, vol. Annual Conference Series, pp. 15-22, 1994.

[47] J. A. Biles, "Autonomous GenJam: Eliminating the Fitness Bottleneck by Eliminating Fitness," presented at 2001 Genetic and Evolutionary Computation Conference Workshop on Non-routine Design with Evolutionary Systems, San Francisco, 2001.

[48] K. Sims, "Evolving 3D Morphology and Behaviour y Competition," presented at Artificial Life IV, 1994.

[49] P. J. Angeline and J. B. Pollack, "Competitive Environments Evolve Better Solutions for Complex Tasks," presented at 5th International Conference on Genetic Algorithms, 1993.

[50] B. Hayes-Roth and L. Brownston, "Multi-Agent Collaboration in Directed Improvisation," presented at First International Conference on Multi-Agent Systems, San Francisco, 1995.

[51] L. Pierse, *Theatresports Down Under*, 2nd ed. Sydney: Improcorp Australia, 1995.

[52] B. Way, *Development through Drama*. London: Logman, 1967.

[53] K. Johnstone, *Impro: Improvisation and the Theatre*, 2nd ed. London: Methuen Publishing Ltd, 1989.

[54] J. Bates, *The role of emotion in believable agents*. Pittsburgh, Pa.: School of Computer Science Carnegie Mellon University, 1994.

[55] J. Bates, A. B. Loyall, and W. S. Reilly, *An architecture for action, emotion, and social behavior*. Pittsburgh, Pa.: School of Computer Science Carnegie Mellon University, 1992.

[56] J. Bates, A. B. Loyall, and W. S. Reilly, *Integrating reactivity, goals, and emotion in a broad agent*. Pittsburgh, Pa.: School of Computer Science Carnegie Mellon University, 1992.

[57] R. W. Picard, "Toward Agents that Recognize Emotion," presented at Actes Procedings IMAGINA, 1998.

[58] W. S. Reilly, "Believable social and emotional agents." Pittsburgh, Pa.: School of Computer Science Carnegie Mellon University, 1996, pp. x, 288 p. :.

[59] W. S. Reilly and J. Bates, *Building emotional agents*. Pittsburgh, Pa.: School of Computer Science Carnegie Mellon University, 1992.

[60] J. D. Velásquez, "When robots weep: Emotional memories and decision-making," presented at National Conference on Artificial Intelligence (AAAI), 1998.

[61] C. Kline and B. Blumberg, "The Art and Science of Synthetic Character Design," presented at AISB1999 Symposium on AI and Creativity in Entertainment and Visual Art, Edinburgh, Scotland, 1999.

[62] H. Prendinger and M. Ishizuka, "Simulating Affective Communication with Animated Agents," presented at Proceedings Eighth IFIP TC.13 Conference on Human-Computer Interaction, Tokyo, Japan, 2001.

[63] A. Bruce, J. Knight, S. Listopad, B. Magerko, and I. Nourbakhsh, "Robot Improv: Using drama to create believable agents," presented at ICRA, 2000.

[64] R. C. Burke, "Time, For a Change", [online] 2000, http://web.media.mit.edu/~solan/society_of_mind/essay3.htm (Accessed: July 21 2003)

[65] R. C. Burke, D. Isla, M. Downie, Y. A. Ivanov, and B. Blumberg, "Creature Smarts: The Art and Architecture of a Virtual Brain," presented at Game Developers Conference, 2001.

[66] J. E. Cahn, "The Generation of Affect in Synthesized Speech," *Journal of the American Voice I/O Society*, 1990.

[67] D. Rousseau, "Personality in Computer Characters," presented at Working Notes of the AAAI-96 Workshop on AI/ALife, Menlo Park, CA, 1996.

[68] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Sydney: Prentice Hall, 2003.

[69] J. W. Rodgers and W. C. Rodgers, *Play Director's Survival Kit*: The Center for Applied Research in Education, 1995.

[70] M. A. Boden, "Autonomy and Artificiality," in *Cognitive Architectures in Artificial Intelligence*, vol. 2, *Artificial Intelligence and Cognitive Science: Conceptual Issues*, A. Clark and J. Toribio, Eds. London: Garland Publishing Inc., 1998, pp. 300-306.

[71] N. Szilas, "A New Approach to Interactive Drama: From Intelligent Characters to an Intelligent Virtual Narrator," presented at AAAI Spring Symposium on Artificial Intelligence & Interactive Entertainment, Stanford University, USA, 2002.

[72] K. T. Simsarian, "Developing collaborative storytelling tools with children in a nascent medium," presented at IZI Bavarian Broadcasting Association Symposium on "The Future of Storytelling", Munich, Germany, 1999.

[73] B. Hayes-Roth, "Adaptive Learning Guides," presented at IASTED Conference on Computers and Advanced Technology in Education, Baniff, Canada, 2001.

[74] A. Hill, "Digital characters give solid performances in 'Fantasy'", (Ahwatukee Foothills News), [online] 2003, http://www.ahwatukee.com/afn/entertainment/articles/010713c.html (Accessed: 20 October 2003)

[75] Academy of Machinima Arts & Sciences, "What is Machinima", [online] 2003, http://www.machinima.org/whatis.html (Accessed: 2 April 2003)

[76] E. André and T. Rist, "Presenting through performing: on the use of multiple lifelike characters in knowledge-based presentation systems," presented at Int. Conf. on Intelligent User Interfaces, New Orleans, Louisiana, 2000.

[77] J. Jacobson and Z. Hwang, "Unreal Tournament for Immersive Interactive Theatre," *Communications of the A.C.M.*, vol. 42, pp. 39-42, 2002.

[78] M. C. Martin, "Esc Online: A Venue for Believable Agents," in *Robotics Institute*: Carnegie Mellon University, 2002.

[79] R. St. Amant and R. M. Young, "Artificial intelligence and interactive entertainment," *Intelligence*, vol. 12, pp. 17-19, 2001.

[80] M. Lewis and J. Jacobson, "Game engines in scientific research," *Communications of the Acm*, vol. 45, pp. 27-31, 2002.

[81] G. A. Kaminka, M. M. Veloso, S. Schaffer, C. Sollitto, R. Adobbati, A. N. Marshall, A. Scholer, and S. Tejada, "GameBots: A flexible test bed for multiagent team research," *Communications of the Acm*, vol. 45, pp. 43-45, 2002.

[82] A. Ghobakhlou and R. Seesink, "An interactive multi modal system for mobile robotic control," presented at Fifth Biannual Conference on Artificial Neural Networks and Expert Systems (ANNES2001), 2001.

[83] P. Taylor, A. W. Black, and R. Caley, "Architecture of the Festival Speech Synthesis System," presented at Third ESCA/COCOSDA Workshop on Speech Synthesis, Australia, 1998.

[84] X. Huang, F. Alleva, W. Hon, M. Hwang, and R. Rosenfeld, "The SPHINX-II Speech Recognition System: An Overview," Carnegie Mellon University, CMU Technical Report CMU-CS-92-112, 1992.

[85] T. A. Meyer, "Using Computer Vision to Enhance Existing Knowledge of an Environment," presented at CIRAS, Singapore, 2003.

[86] A. Kumar, "Using Robots in an Undergraduate Artificial Intelligence Course: An Experience Report," presented at 31st ASEE/IEEE Frontiers in Education Conference, 2000.

[87] H. J. Levesque and M. Pagnucco, "Legolog: Inexpensive experiments in cognitive robotics," presented at Second International Cognitive Robotics Workshop, Berlin, Germany, 2000.

[88] H. H. Lund and L. Pagliarini, "Robot Soccor with LEGO Mindstorms," 1999.

[89] M. Gritsch, "VideoCapture", [online] 2003, http://videocapture.sourceforge.net (Accessed: October 9 2003)

[90] Intel Corporation, "Intel Research", (Intel), [online] 2003, http://www.intel.com/research/mrl/research/opencv/ (Accessed: 8 April 2003)

[91] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *IEEE ICIP*, vol. 1, pp. 900-903, 2002.

[92] A. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," presented at 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), Sarasoto, FL, 1996.

[93] C. S. Pinhanez and A. F. Bobick, ""It/I": A theater play featuring an autonomous computer character," *Presence-Teleoperators and Virtual Environments*, vol. 11, pp. 536-548, 2002.

[94] T. A. Meyer, "Development of Computer-Actors within the Interval Script Paradigm," in *Institute of Information and Mathematical Sciences*. Auckland: Massey University, 2002.

[95] M. A. Walker, J. E. Cahn, and S. J. Whittaker, "Improvising Linguistic Style: Social and Affective Bases of Agent Personality," presented at First International Conference on Autonomous Agents, 1997.

[96] P. Norvig, "AIMA Python Code", [online] 2003, http://aima.eecs.berkeley.edu/python/readme.html (Accessed: 10 October 2003)

[97] SpamBayes-Development-Team, "SpamBayes: Bayesian anti-spam classifier written in Python", [online] 2003, http://spambayes.org (Accessed: 10 October 2003)

[98] A. Wilding and C. Harrison, "PyGP", [online] 2003, http://pygp.sourceforge.net/index.php3 (Accessed: 10 October 2003)

[99] J. B. Tenenbaum, "Bayesian modeling of human concept learning," in *Advances in Neural Information Processing Systems*, vol. 11, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1999.

[100] S.-Y. Yoon, B. M. Blumberg, and G. E. Schneider, "Motivation Driven Learning for Interactive Synthetic Characters," presented at Characters AGENTS, Barcelona, Spain, 1999.

[101] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman, "AutoClass: A Bayesian Classification System," in *Readings in Machine Learning*, *The Morgan Kaufman Series in Machine Learning*, J. W. Shavlik and T. G. Dietterich, Eds. San Mateo, California: Mogan Kaufmann Publishers, Inc., pp. 296-306.

[102] W. F. Finzer and L. Gould, "Rehearsal World: Programming by Rehearsal," in *Watch WhatI Do: Programming by Demonstration*, A. Cypher, Ed. London: The MIT Press, 1984, pp. 79-102.