Universidade do Minho
Escola de Engenharia

Rogério Eduardo da Silva

Creating Partly Autonomous Expressive Virtual Actors for Computer Animation

Junho de 2013

Universidade do Minho
Escola de Engenharia

Rogério Eduardo da Silva

Creating Partly Autonomous Expressive Virtual Actors for Computer Animation

Tese de Doutoramento em Tecnologias e Sistemas de Informação

Trabalho efetuado sob a orientação do
Professor Doutor Manuel Filipe Vieira Torres dos Santos
e do
Professor Doutor Ido Aharon Iurgel

Junho de 2013

“Science may set limits to knowledge, but should not set limits to imagination.”

Bertrand Russell
British author, mathematician, & philosopher (1872 – 1970)

Acknowledgements

This thesis would not be possible without the valuable ideas and comments of professors Ido Aharon Iurgel and Manuel Filipe V. T. dos Santos; for those, I thank them a lot. I also would like to thank all other professors of the Information System Department of the University of Minho.

Another valuable collaborator that has dedicated much of his time to encourage me and make precious suggestions on my work was my friend Irapuan Noce. Most definately this thesis would not be possible without your help.

To my colleagues and students in Brazil that also give important contributions to this work. And yet, I am thankfull to the vital financial support provided by both Santa Catarina State University (UDESC) and Fundação para Ciência e Tecnologia (FCT) through the VirtualActor project at the Centro de Computação Gráfica (CCG) of the University of Minho, where this research has originated.

Of course, a very special thanks to my family: wife, parents and brothers for their patience and incentive during the whole time. My achievements are also yours!

Abstract

Autonomous digital actors represent the next stage in the animation industry in its search for novel processes for authoring character-based animations. In this research, we have conducted a literature review on the art of acting, to obtain an understanding of how apprentice actors learn their skills; this has enabled us to draw up a list of requirements for a proposed autonomous agent architecture for digital actors. The purpose of this was to suggest an improvement in the current technology on digital actors and the way "believable" characters are used by the game and animation industries. Our solution considers three main layers in terms of what skills autonomous actors should display: first, they should be able to interpret script representations autonomously; second, there is a deliberation phase which aims at implementing an agent architecture to work out suitable ways of enacting the previously interpreted script and third, these enactments are translated into animation commands that are suitable for a given animation engine. We have outlined four versions for this virtual actors' framework, the third of which resulted in a prototype built using the Python language, for evaluation. The final solution is a prototype that meets the list of requirements that were listed at the outset of the research. Although determining the best process for creating autonomous digital actors remains an open question, we believe that this thesis provides a better understanding of some of its components, and can lead towards the development of the first fully functional autonomous digital actor.

Keywords

Virtual Actors, Authoring Process, Deliberative Agents

Resumo

Atores Digitais Autônomos representam o próximo avanço para a indústria da animação, em sua busca por novos processos de autoria de animações baseadas em personagens. Nesta investigação, foi realizada uma revisão de literatura relativamente à arte da atuação cênica, a fim de se obter uma melhor compreensão acerca de como atores aprendizes aprendem suas competências; isto nos permitiu produzir uma lista de requisitos para uma arquitetura para agentes autônomos que atuem como atores digitais. O objetivo disto era sugerir melhorias na tecnologia atual de atores digitais e na maneira como personagens "credíveis" são utilizados pelas indústrias de jogos e animações. Nossa solução considera três camadas principais em termos de quais habilidades os atores autônomos deveriam demonstrar: primeiramente, eles deveriam ser capazes de interpretar uma representação abstrata de um roteiro de forma autônoma; a seguir, existe uma etapa de deliberação cujo objetivo é implementar uma arquitetura de agentes para determinar maneiras adequadas de atuação para o roteiro previamente interpretado; e por último, tais atuações são então traduzidas em comandos de animação reconhecíveis por uma dada ferramenta de animação. Foram desenvolvidas quatro versões para este modelo de atores virtuais, sendo que a terceira delas resultou em um protótipo construído na linguagem Python, para avaliação. A solução final é um protótipo que atende a todos os critérios previstos pela lista de requisitos inicialmente proposta por esta investigação. Apesar do fato de que encontrar as melhores práticas de construção de atores digitais autônomos permanecer como uma questão em aberto, acredita-se que esta tese fornece uma melhor compreensão sobre alguns de seus componentes, e com isso aponta caminhos em direção ao desenvolvimento do primeiro ator digital autônomo, plenamente funcional.

Palavras-chave

Atores Virtuais, Processo de Autoria, Agentes Deliberativos

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer animation is the art of telling stories through films with the help of a computer. One of the most important features of storytelling is characters. Computer animation has been studying how to create compelling characters for many decades now. The technology to produce synthetic characters started nearly three decades ago with the first fully computer-generated short film: *John Lasseter's The Adventures of André & Wally B.* in 1984 (Figure 1.1). At that time, all animation processes (especially those concerning the characters' acting performance) were handled by animators in every minor detail.

Figure 1.1: First fully computer-generated synthetic character

Source: `http://www.pixar.com/`

Since then, technology has evolved and produced a range of computer-based assistant software for a variety of tasks concerned with animation: from modeling the character's body to physically simulating muscles, skin, hair, clothes, etc. They are carried out in such a realistic manner that the audience feels uneasy when trying to distinguish them from real beings.

However, despite all these advances, there is at least one aspect of these, so-called, "believable characters" that still leaves significant room for improvement: the quality of the acting. Currently, the acting performances carried out by these characters are renditions of real actors' performances (captured by special devices) transposed to animated characters (digital actors) so they seem to behave just like us (flesh-and-blood human beings). However, it should be noted that, even though these virtual actors are capable of giving breathtaking performances, they are in fact, an illusion, because virtual actors cannot decide for themselves how to perform since they lack *agency*.

The word agency can be defined as the ability of characters to make their own decisions without (or at least with minimal) human intervention. Virtual actors endowed with some level of agency are called *autonomous characters*. The computer game industry has taken the lead in using this technology. In games, non-player characters (NPC) are autonomous characters capable of performing specific tasks within the game environment, and interacting with player in a natural way as if they were just another player. In contrast, the animation industry, currently tends to rely on autonomous characters as digital extras, that are kept in the background and have very little effect on the storytelling, like the armies in the battles scenes in Peter Jackson's The Lord of the Rings trilogy.

Digital extras are autonomous characters with very little agency. They have a limited acting repertoire. This simplicity is acceptable for digital extras as the audience does not usually pay much attention to their performance. This is not the case with leading actors (or even supporting actors).

A leading actor is the person who plays the central character of a story. Supporting actors are all the actors that have an influence on the story, other than the lead. In real life, both leading and supporting roles are given to experienced actors, because they have to be played convincingly. Most unsuccessful films, in terms of viewing figures, have been those whose leading performances did not appeal to the audience.

The same applies to animated films. All leading and supporting roles are acted out as careful and detailed renditions by skilled teams of animators, who spend a lot of time, effort (and money) on the final production.

The idea of using autonomous characters as leading (or supporting) actors to animate a story may well represent the next stage in terms of "authoring" character animations.

Iurgel and Marcos [Iurgel and Marcos, 2007] have first suggested employing the term 'virtual actor' as "*an analogy to a real actor, which autonomously, and by its independent interpretation of the situation, can perform its role according to a given script, as part of a story*".

Later, Perlin and Seidman stressed that:

> *"3D animation and gaming industry will soon be shifting to a new way to create and animate 3D characters, and that rather than being required to animate a character separately for each motion sequence, animators will be able to interact with software authoring tools that will let them train an Autonomous Digital Actor (ADA) how to employ various styles of movement, body language, techniques for conveying specific emotions, best acting choices, and other general performance skills"* [Perlin and Seidman, 2008].

This thesis aims at studying new ways to model deliberative agents as autonomous digital actors. Although there have been several other studies in the literature that focus on creating autonomous 'actors', we argue what is still missing is a systematic attempt to incorporate real actors' practices in formal models of virtual acting.

The identification of this research problem has been originated after two years working as a researcher for the `Project VirtualActor` at the Centro de Computação Gráfica (CCG)[1] of the University of Minho. The Project VirtualActor aimed at "*the creation of virtual actors that can be employed in analogy to real actors. The user*

[1] http://www.ccg.pt/

becomes the director of an animation film. He directs the virtual actors with simple commands, for instance 'be more polite, in this scene', or 'wait a bit until you show your indignation'. In this way, in a very short time, an animation film of high quality can be created, that complies with the intentions of the director".

The Project VirtualActor focused mainly on studying novel authoring processes for creating character-based animation films. Resulting from this, a prototype tool called *CREACTOR* has been developed as to test alternative hypotheses (more detail is presented in Section 3.3.3.8). Despite this thesis has followed several ideas developed for the project VirtualActor, I have decided not to focus on the authoring tool but on the development of the virtual actors themselves; how do they can be modeled inspired on real actors practice. Thus, in a sense, it is believed that the agent architecture proposed here can work as a complement to the findings discovered by the Project VirtualActors.

Virtual actors must display a kind of behavior that is somehow related to that of real actors, which means being able to play roles based on a script, but also to draw on their own previous experiences and adapt their performances to new contexts. Other than that, they must also be able to understand abstract commands from a director/animator who should guide them towards the desired performances in a particular situation.

For a long time, artificial intelligence techniques have been used to produce believable characters that display autonomous behavior and rely on tacit knowledge for choosing the right response to an event. This thesis aims at developing an agent architecture for autonomous digital actors that takes account of the practices of real actors to understand the basic requirements for a compelling acting performance from the interpretation of a script.

1.1 Structure of the Thesis

This thesis is divided into four chapters (in addition to this Introduction):

- Chapter 2 provides an outline of the methodology employed for this thesis. It examines the question of motivation, the intended goals of the research, the expected results and the adopted validation procedures.

- Chapter 3 conducts a literature review about the subject of virtual acting, and examines the different understandings described in the literature, several approaches that have been adopted to implement them and some possible applications. It also includes, an extensive study of real actors; how does an apprentice actor learn his/her art? What skills should be mastered by someone planning to act? How do each of these skills work? How do actors and the director work together while enacting a story?

- Chapter 4 explores our understanding of what virtual actors could be and compiles a literature review. This includes the first two versions of the suggested agent architecture and aims at providing a virtual actor who is capable of interpreting a given script, analysing the current context of the role the actor is enacting and finally, suggesting an acting performance for it that allows the animator (working as the director) to make any alterations when necessary. These suggestions have only been evaluated at a theoretical level.

- On the basis of these proposals, we have produced a third agent architecture that has been implemented as an experimental prototype, and which is described in Chapter 5.

- Chapter 6 sets out the final version of the autonomous digital actor agent architecture adopted in this research.

- Finally, Chapter 7 summarizes this thesis by discussing the conclusions that can be drawn from this work. It also makes recommendations for further studies and possible improvements to the suggested model.

Chapter 2

Description of the Research

This chapter aims at describing the factors that led to this research, the question being investigated, the initial hypothesis and the methodology adopted for the solution. It also discusses what results were expected and what these represent in terms of novelty for the improvement of the knowledge domain, including the limitations of the proposed solution.

2.1 Motivation

Authoring character animations has become a very demanding, costly and specialized task for large groups of experts. Despite this, there is a growing market that is eager for products that apply character-based animations to several different types of domains: from animated feature films to educational interactive narrative tools. In the light of this, it can be concluded that finding new, simpler and faster ways to author character-based animations, is a task that is essential for these applications. This statement is in line with the standpoint of Perlin & Seidman quoted in the last chapter.

Autonomous Digital Actors represent one possible solution to the problem of developing simpler-to-use authoring tools for character-based animations. The main problem is the fact that the process, and even the requirements, involved in creating a fully computer-generated actor, are still not completely understood.

In Arts, 'believable characters' are animated characters that "*provide the illusion of life, and thus permit the audience's suspension of disbelief*" [Bates, 1994]. Bates proposed the creation of 'believable agents' that are Artificial Intelligence-based agents, especially trained to mimic certain aspects of human behavior (or that of another living creature) in order to produce self-animated believable characters. Although these agents are capable of being autonomous and acting believably (i.e. rationally) within their surrounding environment, they have not been designed for plot-based animations. This means that they lack acting skills, and can only act in terms of their own character; as a result, they disregard acting techniques and any analysis of the story being played, much as human actors do while acting out a play or script.

Thus, in this investigation it is worth exploring the possibility of endowing believable characters[1] with the ability to understand a script and thus produce proper acting performances by tackling the problem of creating the first fully autonomous digital actor.

2.2 Research Question

The research question can be postulated as follows: can the acting skills of believable characters be improved by, somehow, mimicking the practice of real actor? And if so, how can one develop an autonomous digital actor's agent, similar to those imagined by Perlin & Seidman?

2.2.1 Research Hypothesis

Preliminary studies indicate that four factors are often present when actors are preparing to play a particular character in a story:

Interpretation of the Script - an actor should understand the script of the story, and also the roles played by the characters, and any other relevant information that can help him/her to give a convincing acting performance;

[1] In this thesis, we use the terms believable characters and believable agents indistinguishably.

Making Dramatic Inferences - an actor must be able to make inferences from a dramatic situation and understand the implications of what he is being asked to perform, such as, the meaning (causes and consequences) of an utterance it is expected to him to say;

Acting - by relying on their specialized knowledge, autonomous agents should suggest proper acting performances and in doing this take account of their current dramatic situation and their interpretation of the script;

Guidance - to end with, the actors must adapt these suggestions to the guidances offered by the animator regarding what is a good and not so good performance (as in an actor/director metaphor).

2.3 Objectives

To design an agent architecture that can serve as an Autonomous Digital Actor, this should at least include the four main factors described earlier.

A number of several secondary goals must be achieved to attain this goal:

1. giving an intelligible interpretation of the script that is employed;
2. providing an autonomous dramatic context that can be inferred from the environment (stage set) during a performance;
3. determining a suitable knowledge representation for traditional acting techniques, that allows training agents to adopt specific acting styles;
4. drawing up criteria that allow agents to suggest suitable acting performances even in unforeseen situations; and
5. offering alternatives to how the animator (acting as a director) can convey 'criticisms' to the agents. This can allow them to adapt their suggestions in an attempt to meet the animator's expectations and, by doing so, learn how to make better suggestions in the future.

2.4 Methodology

According to Carvalho (2012) [Carvalho, 2012], *knowledge-for-understanding* corresponds to an "*existing understanding of human, group, organizational or social phenomena*", which are classifications, descriptions, explanations or causal relationships; while *knowledge-for-a-purpose* corresponds to "*knowledge about human-designed artifacts*". These two types of scientific knowledge lead to different types of research paradigms: behavioral science and design science (or just design research as suggested by the author).

This thesis adheres to the knowledge-for-a-purpose approach as it seeks to design an artifact (an agent architecture in our case) and according to Carvalho, this artifact should "*aim at responding to some human necessity, either existing or foreseen*".

Barab and Squire [Barab and Squire, 2004] suggest another approach called 'design-based research' that involves a number of features that distinguish it from other approaches:

- It focuses on understanding real-world practice;
- It involves a revision of the flexible design, multiple dependent variables, and an ability to capture social interaction;
- The participants are not "subjects" assigned to processing but instead are treated as co-participants in both the design and analysis;
- It aims at outlining a profile or theory that characterizes the design in practice (as opposed to simply testing a hypothesis);
- in terms of evaluation, this approach resembles formative evaluation methodologies: this include articulating goals, operationalizing measures, examining a phenomena and understanding the implications of its use (both intended and unintended). It also involves producing demonstrable changes at a local level and regarding changes in these contexts as evidence that a theory is feasible.

It is our understanding that these two approaches (design research and design-based research) are, in fact, similar and, for this reason, we are adopting a methodology that combines elements from both.

The aim is to prove that the use of autonomous digital actors provides a viable approach to simplifying the creation of character-based animation films.

Several steps must be taken to achieve this:

1. A study of the art of acting and how actors learn their craft;
2. A study of the digital actors that represent the current state of the art in terms of synthetic character endowed with performative skills;
3. A study of believable characters (and virtual actors in particular);
4. Compiling a list of requirements that virtual actors must possess to behave in the expected way (to give an autonomous acting performance);
5. A study of artificial intelligence techniques that can enable these requirements to be implemented;
6. Designing artifact software (e.g. agent architecture, library, plug-in) that can be used to implement all the main components of a virtual actor;
7. The implementation of a software prototype for experimentations;
8. Tests and validation of both the suggested architecture and the software;
9. Evaluation of the outcomes; and
10. Recommendations for improvements (if necessary).

2.5 Expected Outcomes

Carvalho (2012) [Carvalho, 2012] argues that, despite the fact that the design research aims at producing designed artifacts, these should not be regarded as its main outcome. In fact, what it seeks is knowledge about artifacts such as its structural representation (architecture), possible applications, or descriptions about the way they work and are operated.

In view of this, two important outcomes from the design research are expected: (1) they will lead to an advance in the authoring of plot-based animations by proposing an ontological improvement in the existing knowledge about autonomous digital actors; and (2) they will assist in designing and producing a (prototype) software tool for an existing animation engine, that implements an agent architecture which acts as an experimentation to provide evidence that the model suggested in this thesis, really meets the requirements for constructing an autonomous digital actor (proof of concept).

2.6 Validation

Four validity criteria are being employed in this thesis [Carvalho, 2012]:

1. **Artifact Success :** this can be measured in terms of:
    - *Usefulness* - this means the degree to which the artifact (resulting tool) contributes to the achievement of a result.
    - *Efficacy* - means the degree to which the artifact achieve the expected results;
    - *Efficiency* - means the ability to obtain efficacy with the paucity of resources involved.

2. **Generalization**: scientific knowledge is general if its applicability is not restricted to specific situations. Depending on the nature of the science it can be universal or restricted in time, space and circumstances. Knowledge-for-a-purpose should be applicable to different classes of situations.

3. **Novelty :** new knowledge in this context means new classes of artifacts or represent a significant improvement to those that already exist.

4. **Explanation Capability :** the reasons for the success of the designed artifacts should be explained.

Chapter 3

Literature Review

3.1 Actors

According to an ancient Greek legend, a man called Thespis created the art of acting when he decided to step out of a chorus of performers and utter a few solo lines. Before Thespis, the world had never seen a character performance but, only groups of performers singing in unison. Thespis narrated a tale from the perspective of the character instead of the standpoint of a third person, and thus changed theatre and the dramatic arts forever. Until today, actors are still referred to as *thespians*.

According to [Kundert-Gibbs and Kundert-Gibbs, 2009], "acting is defined as the art or practice of representing a character on a stage or before cameras and derives from the Latin word *agere*, meaning 'to do' ".

In conclusion, actors are individuals specially trained to act out a stage representation of a story. They practice how to mimic human behavior so that they can impersonate a character. There are two important lessons that any apprentice actor must learn:

- **Performance Skills**: how to use his/her body (e.g. arms, legs, face, voice) to tell a story more effectively. There are several acting schools that suggest their own list of skills;

- **Script Analysis**: apart from studying performances, an apprentice actor must learn how to study his/her role by learning what kind of information should be drawn from the script.

3.1.1 The Art of Acting

Before one can understand how actors learn their trade, it is necessary to look at how acting schools teach acting. Although there are many acting schools from which one can learn the art of acting, the two most prominent are based on the teachings of Stanislavski and Strasberg as discussed in the next section.

3.1.1.1 Constantin Stanislavski

Constantin Stanislavski, a Russian actor and director, was a legendary acting teacher whose lessons are still being followed by acting students throughout the world today. He developed a method of acting that takes account of the following statements [Stanislavski, 1937, Stanislavski, 1950]:

- **Imagination and Emotions**: imagination is a powerful source of emotions. A performance without emotions looks artificial;
- **Concentration and Relaxation**: an actor should focus his/her attention exclusively on the performance and, an actor should learn how to relax his/her muscles to ensure his movements are natural while he is on stage;
- **Objectives, obstacles and actions**: a performance should be explained in terms of objectives (what motivates the characters to act?), obstacles (what prevents the actor from attaining his/her objectives?) and actions (what does the character do to achieve his/her goals?). Stanislavski warns against examining a performance in too much detail. Only the most central features should be described (the general idea);

- **Emotion Memory**: an actor should recall his/her own previous experiences and the emotions they aroused (and not try to mimic others); this should allow him to recall specific memories if needed;
- **Sense of Truth**: truth on the stage is anything that the audience can believe in, which is not the same as replicating the real world. An actor should set out by deciding how to perform 'from within' (from the standpoint of the character) and at all costs avoid overacting (including too much detail in the acting);
- **Adaptation**: this requires an actor to be able to adjust his performance to changes made by other actors or the surrounding environment;
- **Unbroken Line**: is a continuous line of action that unifies the events throughout the entire play.

3.1.1.2 Lee Strasberg

Lee Strasberg, a former student of Stanislavski, and co-founder of Actor's Studio (one of the most prestigious American acting schools), developed "The Method", which is (according to Strasberg himself) a continuation of the Stanislavski method. 'The Method' relies on the following basic principles:

- **Relaxation and Concentration**: following Stanislavski, Strasberg also stresses the importance of the preparation phase before acting, which includes being relaxed and focusing on "believability";
- **Sense Memory**: an actor should recall objects and sensory experiences, for instance, what was the sensation like when drinking a cup of coffee (he should take account of all the five senses);
- **Emotion Memory**: unlike sense memory, emotion memory involves recalling emotional experiences from the actor's past, rather than trying to force a reaction or emotion;

- **Characterization**: an actor should know how to create a character, both physically and psychologically, and employ the sense memory and emotion memory to achieve this;
- **Character's Driving Forces**: when preparing for a role, actors should fully understand their characters so that they can make them believable. Thus they should be prepared to 'answer' questions like "who is the character"? "What does he want"? "Why does he want it"? "How is he planning to achieve his/her goals"?

Other important acting methods include: Meisner and Stella Adler.

3.1.1.3 The Meisner Technique

Sanford Meisner was trained as a "method" actor; however, later he rejected several of Strasberg's ideas and proposed his own technique. In particular, he rejected Strasberg's reliance on inner states (like sense memory) and used instincts instead. Meisner's basic premises are as follows:

1. **The reality of doing:** an actor should be commited to the action being played by doing what the character would do if he was in the real world;
2. **Moment-to-Moment Acting:** any acting performance should be understood as a series of 'moments' where an actor should only care about a particular moment that is being acted out and how this moment connects to the next;
3. **Imagination and Preparation:** Meisner believed that "the power of imagination" is a much more reliable source of feeling than the sense memory and the emotion memory from our own past that are drawn on by Strasberg. The idea is that, instead of recollecting a situation that triggered a specific emotion, the actor should imagine one.

3.1.1.4 Stella Adler System

This was another Stanislavski student who developed her own method during her years of practice in the theatre. The main points of Stella's method are:

1. **Acting is Doing:** acting means *always do something* on stage and always ensuring that it has a justification;
2. **Developing the Imagination:** by observing the world in detail, an actor can create images of everyday life situations and use them to determine his performance on stage;
3. **Training the Mind:** an actor must fully understand the character being played. To do so, he has to make thorough study of the script and even research real social situations if necessary;
4. **Size:** actors need to bring 'size' to the characters by having a strong physical presence and a resonant voice on the stage.

3.1.2 The Art of Directing

According to Weston [Weston, 1996], good direction means suggesting actions that are playable, since this affects the way the actor behaves. The following sections outline two types of directing (one that is on the whole considered to be unsuitable and the other to be, more objective).

3.1.2.1 Result-Oriented Direction

The author claims that the main complaint actors make about directors is that they do not know what they want (or cannot clearly explain it). She explains that usually, while reading the script, directors automatically conjure up a clear image of how they imagine that the script should be performed and they 'demand' actors should match their preconcieved image, which leads to conflict between the director and actors.

Weston has called this *result-oriented directing*, and she defined it as "an attempt to shape the actor's performance by describing the result you are after, i.e., how you want it to end up looking or sounding". She outlined at least ten situations to illustrate what could lead to result-oriented directing, so they could be avoided:

1. **Telling actors what effect the director wants** as in "can you make it more quirky?" does not clarify the director's intentions and only leads to a 'guess what I want game'. Ultimately, it leads to situations where actors are more concerned with the effect of their performance than on the performance itself. It can have very undesirable side-effects too, such as when an actor 'forces' himself to look serious and ends up with a comical effect;

2. **Asking general or vague instructions** like "can you give it more energy?", which can lead to actors laying emphasis on uninteresting parts of their performances;

3. **Giving the actor a line reading**, which means telling him the expected way to act (like voice inflection, for instance) usually leads to mechanical performances (those lacking a deeper reflection on the part of the actor);

4. **Telling the actor what feeling the character should be having** or any other mental state (e.g., "I think he is disappointed"), usually leads to acting that is not believable. This is because, since people usually do not think about their feelings, they should not have to struggle to show them. Actors should be free to feel the way they think their characters are likely to feel;

5. An extension of the previous statement is when the director tries to **tell the actor what reaction to have** as in "when she tells you she doesn't have the money, you must get angry". Again, the situation is where the director is making all the decisions for the actor;

6. Even worse is the situation when the **director gives the actor a loaded emotional "roadmap"**, for example: "when the scene starts he is worried because she is late. He is relieved when she arrives, but then disappointed because she hasn't got the money, and then he becomes suspicious that she might be holding out on him". This kind of directing usually raises several problems like being tedious, long-winded, superficial and lacking a 'through-line' (sense of objective). This can produce *indicative performances* that can be characterised as when the audience can tell that an actor is trying to show a particular state instead of experiencing it;

7. **When the director states how he understands the character** this is unproductive (just as we should not tell the cast how to feel or react). Instead of complaining that the character *is* bad, it is better to say that he *acted* badly. The first is simply plain criticism, the second, an opportunity to change;

8. "Can you make him aggressive, but pleasant?". Directors quite often give **confusing/unplayable directions**. Actors cannot express two things at once, however they can rapidly alternate between states (which is completely different from playing two things at the same time);

9. **Negative judgments:** it is not up to the director (or the actors for that matter) to make snap judgements about the character and actions as in "he is a punk". Pre-judging a character in negative terms can seriously handicap an actor's ability to decide what the character should do in a situation;

10. **The director should not be the person who decides what attitude the character should adopt** like "it has a hostile attitude toward his father". This is another problem that can lead to indicative performances.

3.1.2.2 Quick Fixes

Good direction should be: active, objective, sensory, dynamic and kinetic. These "quick fixes" represent five tools that aim at responding to each of these qualities:

Verbs: one common mistake directors make is to suggest actions by using adjectives (e.g. "be defensive") when they should use verbs instead, because verbs denote experiences rather than the conclusions of experiences (results). But not every verb is appropriate, only transitive verbs that have an object, because they have an emotional and a physical component. A suggested list of such verbs can be found in [Weston, 1996]. Thus, in the example given, instead of "be defensive", a director could suggest actions like to *complain*, to *belittle* or to *warn*, that are 'defensive actions';

Facts: there are two kinds of facts: factual backstory and imaginative backstory. Facts are always self-explanatory and should never need further explanations. In fact, trying to explain facts usually makes them less clear. For example, the fact that "she wrote a letter to her mother every day during her honeymoon" explains the character's relation to her mother much more clearly than "she is very attached to her mother";

Images: images relate to what we perceive in the world when we use all our five senses. Images can trigger emotions more effectively than explanations. For instance, the image of your father closing the door behind him, is in itself an explanation that can evoke an expected "feeling of loneliness" in a scene;

Events: events are the main feature of every scene. They are used to indicate to actors the nature of the story being told or the theme of the movie being made. An understanding of this is vital for actors to enable them to imagine what their appropriate reactions should be while performing the scene and why;

Physical Tasks: sometimes actors can enter a state called 'self-consciousness', which is thinking too much about their own performance. Giving them an extra physical task (like trying to open a stuck door while speaking his lines) can be a very effective way of preventing actors from experiencing this condition and enabling them to give compelling performances.

3.1.3 Plot and Script

A plot is the narrative description of the main events of a story. Although, there are several possible structures that can be adopted to represent a plot, one common approach (proposed by Gustav Freytag [Freytag, 1990]) is to divide the narrative into five parts, as shown in Figure 3.1.

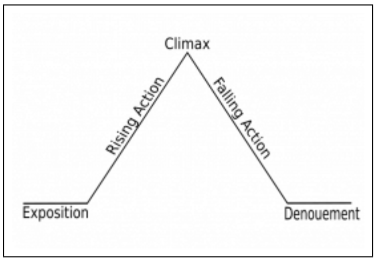


Figure 3.1: Gustav Freytag's Five Parts Narrative

Adapted from [Freytag, 1990]

Exposition: introduces all the characters (especially the protagonist), their goals and motivations;

Rising Action: is the phase where the main conflict is presented. The protagonist understands what his/her goals are;

Climax: represents the turning-point where the main character makes the big decision on which hinges the outcome of the story;

Falling Action: indicates the phase in which the protagonist is further from accomplishing his goals but also beginning to overcome his difficulties;

Resolution: is when the story determines who is the winner in the main conflict and describes the long-term consequences of any decisions that have been made.

There are two types of plots that should be considered when dramatising a story: *linear* stories are those where every event within the narrative only can be followed by the next event or *non-linear* stories where each event can result in alternative outcomes, meaning that a choice is made to determine the actual course of events that takes place.

A script is a record of events that is designed to describe a plot as a series of 'actions' (mostly dialogues) intended for theatrical performances. Actors rely on scripts to determine how and when to carry out their acting so they can tell a story. It is the script that encompasses much of the information actors draw on to 'create' their characters. Two important factors that an actor should understand when studying a script are: script analysis and the script subdivisions. Section 3.1.3.1 summarizes techniques used for script analysis. According to [McKee, 1997], a script can be subdivided into five components:

Beats are the smallest dramatic unit of any script. A beat is defined by any change in the character's inner or outer states like, for instance, emotional displays, mood, intentions or speech;

Units represents groups of beats that can be linked to a topic;

Scenes are groups of units that take place in any given scenario. A new scene is established for every change of scenario;

Sequences are groups of scenes combined in response to specific dramatic situations;

Acts are a series of sequences that can be defined according to the turning-points of the plot.

3.1.3.1 Script Analysis

There is a consensus among acting teachers that apart from just learning proper acting skills, an actor should also study his/her character in depth to prepare for his/her part. The following factors summarize important aspects of studying a script [Thomas, 2009, Waxberg, 1998]:

- **Action Analysis**: this is the first analysis of the script. It aims at a less complex analysis than formalist analysis (which results from combining all the features of script analysis analysed below), and is responsible for detecting the following elements in the text:

- *Sequence of External Events*: events that change characters;
- *Review of the Facts*: this describes who the main characters in each event are, what they are doing, where the action takes place, why it happens, when it happens;
- *Seed*: this describes a moral commandment that has been violated, and then justifies the action;
- *Sequence of Internal Events*: this describes the connections between the seed and the external events;
- *Three Major Climaxes*: this defines the three major turning-points of the story (the beginning, middle and end);
- *Theme*: what was the main character's response to the seed?
- *Super-Objective*: this is the goal that the main character is striving to achieve;
- *Through-Action*: this state the main conflict of the play in one concise sentence;
- *Counter Through-Action*: what factors interfere with the through-action of the main character;

• **Given Circumstances**: this describes the unique combination of past and present situations that mark the beginning of the play. In [Thomas, 2009], a series of given current circumstances is described:

- *Time*: this describes when the play takes place, how the audience is made aware of the passage of time;
- *Place*: this describes where the story takes place, and the main features of the setting;
- *Society*: this describes the relationships between people living in this 'place';
- *Economics*: this describe, in general terms, the economic system of the play, and how the 'society' lives in this system. How does money influence their lives?

  - *Politics and Law*: what is the government system? And how does this influence people's lives?
  - *Learning and the Arts*: what is the general level of culture and artistic taste of the characters?
  - *Spirituality*: this describes the characters' spiritual and religious beliefs;
  - *The World of the Play*: how do all previous given circumstances work together? What is 'the world' that the character lives in?
  - *After Action Analysis*: how does the 'Seed/Theme' relate to these given circumstances?

- **Background Story**: indicates everything that happened before the beginning of the play;

- **External and Internal Actions**: "external actions" represent all the entrances (the moment characters enter into a dialog) and exits (the moment their involvement ends). "Internal actions" this describes the assertions made about people, places, things and events during the action;

- **Progressions and Structure**: progressions are subdivisions of the play into units (or beats), scenes and acts. The structure defines the driving-force behind the story, the main conflicts, and the range of intensity of the emotional scenes;

- **Character**: is defined in terms of:

  - *Objectives* - what is his/her super-objective? And his/her minor objectives in each scene, unit and beat?
  - *Conflicts* - this describes the conflicts of the main characters (in terms of role and objectives)? When do they occur? What complications arise from them?
  - *Will Power* - how does the character attempt to carry out his/her objectives? Does he follow a steady course? Does he vacillate?
  - *Values* - what does each character stand for and what is he against? What does he believe to be right and wrong/good and bad?

  - *Personality Traits* - describe each character in terms of: energy level, consistency of behavior, impulses and inhibitions;
  - *Complexity* - how self-aware is the character? Who are the type characters, intermediate characters and complex characters?
  - *Relationships* - what are the character's primary and secondary relationships?

- **Idea**: is represented by:
  - *Words* - these describe the meanings of the play in terms of the title of the play, speeches, illustrations, allusions, etc;
  - *Characters* - is there a narrator or chorus? What other types of characters are there in the story: skeptical, confident, and so on;
  - *Plot* - are there any repetitive situations? Intellectual conflicts? Symbolism?
  - *Statement of the Main Idea* - this explains the main idea of the story in a single concise sentence;
  - *Mise-en-scène* - how does the main idea influence and contribute to the mise-en-scène?

- **Dialogue**: is described as:
  - *Words* - what kind of words are used in the dialogs? Concrete? Abstract? Formal? Informal? Professional jargon? Slang?
  - *Sentences* - what is the average length of the sentences? What types of sentences are employed? Any special kind?
  - *Speeches* - What were the speeches written for? How are they connected? By words? By thoughts? By meanings?
  - *Special Qualities* - is the dialog written in verse? Or prose?
  - *Theatricality* - in what ways an actor express internal and external actions? How emotional is it? Is there any subtext? How is this expressed?

- **Tempo, Rhythm and Mood**: with regard to the tempo, how often is information conveyed about the plot or characters? And about rhythm, how do the emotional tensions work on each scene, act or the entire play? How does this affect each character? And finally, on the question of the mood, are there any atmospheric feelings within a given situation? Or character? Or idea? What controls the atmosphere of the world of the play?

3.2 Digital Actors

For centuries, the only way of enacting stories was through stage performances (performed by actors). It was only in the 20^{th} Century, with the advent of synthetic characters that this hegemony started to fade. Synthetic characters are handmade animated characters that exist only in the story world (they do not have a physical representation in the real world). The first synthetic character ("Gertie – The Dinosaur") dates from 1914 (see Figure 3.2 on the left). Later on, with the advent of computers, this technology evolved, not in terms of concepts, but in terms of form, into digital actors.

Digital actors can be found in the literature, and have been referred to by several different terms like synthespians (as in synthetic thespians), virtual humans or believable characters, all relating to similar definitions: a computer-based virtual character capable of delivering acting performances in digital space. Digital actors are renditions of real actors' performances (captured by special devices) transposed to animated characters, so they seem to behave just like us (flesh-and-blood human beings).

In 1984, the world watched the first fully computer-generated short film: John Lasseter's "*The Adventures of André & Wally B.*" (see Figure 3.2 in the middle). Back then, all the characters were manually animated and relied on special authoring software. Later, in 2001, the first photo-realistic character (Dr. Aki Ross appeared in the movie "*Final Fantasy: The Spirits Within*", as shown in Figure 3.2 on the right). It was the first time that a character had been conceived to appear in multiple movies in different roles.

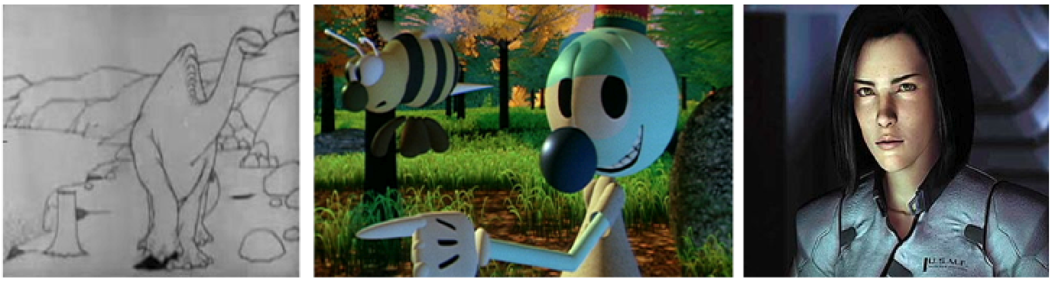


Figure 3.2: (left) First animated character (center) First fully computer-generated character (right) First photorealistic reusable character

Currently, technology has evolved to the point that it allows human actors to give performances alongside digital actors. There are plenty of examples of digital actors interacting with humans in the movie industry: "Jar Jar Binks" in Star Wars, "Dobby" in Harry Potter, "Smeagol" in Lord of the Rings, and many others. In all these examples, the process of producing these apparently interactive performances, involves all the individual performances being recorded separately and then combined by means of special authoring software. In this process, one key factor is how to construct digital actors for the performances. A description of this process follows in the next section.

3.2.1 The Art of Digital Acting

The technology used to develop digital actors is becoming very sophisticated and constantly evolving, to the point that at present it is possible to produce simulations that are nearly indistinguishable from real human performances. According to Thalmann & Thalmann [Thalmann and Daniel Thalmann, 2005] there are three levels that should be taken into account when modeling a virtual human, each one representing a challenge for engineers in the field:

1. **Appearance Modeling**: the visual representation of the actor's body should be as close as possible to that of a real human. This means studying novel techniques in computer graphics and geometric modeling, to model features like skin, hair, eyes, clothes and so on (see Figure 3.3);

Figure 3.3: (left) Synthespian Project (center) Kaya Project (right) Hair simulation

2. **Realistic, Smooth and Flexible Motion**: ideally, digital actors should be able to move in a way that makes them indistinguishable from humans. Traditionally, computer animators have solved this problem by *motion capture*. Motion capture is a technique that employs special devices that are capable of recording human performances, and transposing them to animated figures so they can perfectly mimic these performances. Figure 3.4 illustrates the actor Tom Hanks wearing a motion capture suit during a recording session for the production of the movie "Polar Express" produced by Robert Zemeckis, and also the final animated character mimicking his performance;

Figure 3.4: Motion capture session vs. the final product

3. **Realistic High-Level Behavior**: digital actors must be able to mimic human behavior convincingly (e.g. in decision-making, emotional situations, problem-solving and improvisation). This is perhaps the most difficult feature to bestow

on a computer-generated being. Digital actors work as digital puppets, which means that they have no intelligence skills and are unable to decide on their own acting performances by themselves during a scene or movie. It is up to the animators to decide each and every action via an authoring system.

3.3 Autonomous Digital Actors

It should be noted that, despite some of the breathtaking performances these virtual actors can give, they are in fact, an illusion, because virtual actors cannot decide for themselves how to perform. They lack *agency*.

Agency means that an artificial character has the ability to make its own decisions without (or at least with minimal) human intervention. Digital actors that are endowed with some level of agency are called autonomous characters. The computer games industry is the leader in using this kind of technology. In its games, non-player characters (NPC) are autonomous characters that are capable of carrying out specific tasks within the game environment. They have been common for decades now and have become very sophisticated in terms their ability to emulate complex behavior. The players interact with them as if they were any other player. The animation industry, on the other hand, mainly relies on autonomous characters as digital extras. These are background characters with no direct importance to the story being told, like soldiers in the armies of the battles scenes in Peter Jackson's The Lord of the Rings trilogy.

Iurgel & Marcos [Iurgel and Marcos, 2007] have suggested employing the term 'virtual actor' as "*an analogy to a real actor, which autonomously, and by its independent interpretation of the situation, can perform its role according to a given script, as part of a story*".

Later, Perlin & Seidman [Perlin and Seidman, 2008] predicted that "*3D animation and gaming industry will soon be shifting to a new way to create and animate 3D characters, and that rather than being required to animate a character separately for* *each motion sequence, animators will be able to interact with software authoring tools that will let them train an Autonomous Digital Actor (ADA) how to employ various styles of movement, body language, techniques for conveying specific emotions, best acting choices, and other general performance skills*".

3.3.1 The Art of Autonomous Digital Acting

In this thesis we use the terms virtual actors and autonomous digital actors indistinguishably. In addition, this research has been largely restricted to "talking heads", which are characters that are only capable of expressing themselves through their heads (i.e. through facial expressions, gaze, head movements and blinking, along with the speech and lip sync). Plantec [Plantec, 2004] presents a list of seven essential concepts in facial acting:

1. **The face expresses thoughts beneath:** a character's display of emotions is dictated by its own inner emotional state. Autonomous digital actors should find a way to convincingly emulate human cognition by inferring appropriate emotions in every dramatic situation;
2. **Acting is reacting:** every facial expression is a reaction to something, which means that an emotional reaction is determined after sensing the environment (inferring the current dramatic situation). Ortony et al [Ortony et al., 1988] argue that this 'something' which one should react to is, in fact, the perception of events, objects in the environment and the actions of another agent;
3. **Know your character's objective:** knowing a character's objectives involves determining the appropriate (plausible) reactions it can have in every situation. Without this, choosing an action might be left to chance, which is not believable;
4. **Your character moves continuously from action to action:** an actor should be carrying out an activity the whole time to maintain the illusion of life and the illusion of self-consciousness, even if she is simply waiting for something to happen. People are never still. They may be only making very subtle movements

like shifting their gaze or blinking, but they will never be completely frozen. In character animation, a character can/should achieve this, according to Disney's 12 principles of animation, by exaggerating these movements without being concerned about losing believability;

5. **All action begins with movement:** as established earlier, people can never be completely still. This remains true even when one is only carrying out mental activities like calculating or imagining. Everything should be reflected in the character's body. The audience should always be able to infer that something is going on in the character's mind from their attitude and movements;

6. **Empathy is audience glue:** empathy represents how the audience relates to the character. If it is handled properly, it can encourage the audience to form an emotional bond with the virtual character;

7. **Interaction requires negotiation:** being able to display compelling emotional reactions is the most important feature of an autonomous character. If there is a failure to display emotional reactions properly (with the body, voice and choices), the interaction with humans is doomed to result in boredom and lack of interest.

3.3.2 Requirements for Autonomous Digital Actors

In the light of all previous studies (regarding actors and digital actors), we propose a list of requirements that any agent should be able to fulfil to act as an autonomous digital actor [da Silva et al., 2010]:

1. **Autonomous Script Interpretation:** the most basic of any actor's skills must be the ability to obtain information regarding his course of action when interpreting the script. This can be divided into two periods: first, the actor reads the script to learn his lines and understand his activities from clear descriptions of the scene; second, using script interpretation techniques the actor constructs his/her character by containing a detailed description of the scene and combining it with a common sense knowledge of the real world to 'prepare' for the given role;

2. **Acting Knowledge:** actors need knowledge before undertaking any role. They must understand what it means to act a particular character, or what it means to experience a particular situation, etc. As discussed earlier, actors practice this experience of memory retrieval, especially in terms of emotion and sensing; and

3. **Dramatic Performances:** another important skill that actors constantly practice is their expressivity, which means physical, muscular, facial and vocal control etc. when giving performances that have an emotional effect on the audience.

All these requirements relate to individual acting performances. In Chapter 5, we set out our proposed solutions for each of them. Along with individual performances, another important factor that should be considered is how the agents relate to the animator via the authoring tool. In addition, in [Iurgel et al., 2010], we discuss the requirements of the authoring aspects of ADA.

3.3.3 Related Work

In the past few years, several works have been published that, to some extent, have assisted in establishing the concept of the virtual actor. In the next sections we examine some of these studies, and discuss their strengths and weaknesses with regard to this concept.

3.3.3.1 Virtual Humans

According to [Thalmann and Thalmann, 1993], a virtual human is "*a visualization of the simulation of the behavior of realistic human beings*". There are three levels of modeling concerning virtual humans [Thalmann and Daniel Thalmann, 2005]: (1) realistic appearance modeling, (2) realistic, smooth and flexible motion modeling and (3) realistic high-level modeling of behavior.

1. Realistic appearance means studying how to create characters that look like real people. This entails studying features like hair, clothes and body modeling;

2. Realistic motion modeling means studying ways of representing both path planning and physical locomotion like walking, jumping, etc; and

3. Realistic modeling of behavior means establishing types of behavior based on models that reflect real human behavior, that are likely to have been inspired by psychology and other human sciences.

With regard to this research, the only relevant level is behavior models. There are several studies that aim at describing how to model and implement virtual humans (for instance, [Sevin and Thalmann, 2005, Kshirsagar and Thalmann, 2002, Egges et al., 2003]). These studies also discuss possible applications of virtual humans, in terms of behavioral simulation, which, in general, rely on psychological models that aim at mimicking real humans in given contexts. One feature that most of these works have in common is to use personality traits and emotion models as a way of producing autonomous believable characters; these are capable of showing individuality as if they have been governed by these characteristics just like any human.

Another interesting study of expressive virtual humans that are capable of giving compelling performances can be found in [de Melo and Paiva, 2006]. In this study, the authors outline a multimodal expression model for virtual humans based on six layers:

Deterministic Expression : is responsible for animating virtual humans based on subsets of keyframe animations and combination mechanisms like weighted combined animations, body group animations and pose animations;

Non-Deterministic Expression : consists of 'human-free' procedural animations like inverse kinematics, joint interpolation, function based interpolation, frame interpolation and Jacobian-based animations;

Vocal Expression : represents the subsystem responsible for implementing the notion of speech and its components like voice synthesis (via a text-to-speech tool), communication protocols, utterance structure and voice intonation controls;

Facial Expression : the component responsible for determination of facial expression for emotional displays by a muscle control system;

Gesticulation Expression : represents a subsystem for determining constraints for hand properties such as shape, position and orientation;

Environment Expression : responsible for creating the environment in which a story is told. This environment is based on dramatic expressions (of characters), text, illumination, make-up, sounds and music, that work together as a setting with 'points of interest', that competes for the audience's attention.

3.3.3.2 Master-Servant

In [Hayes-roth et al., 1997], the authors describe a research study on "*agents that function as improvisational actors who spontaneously and cooperatively generate their stories at performance time*".

Their synthetic characters are improvisational actors that are defined by three structures: a *plot* which is a constrained sequence of actions that involves a set of individuals, a *role* which is a class of characters whose prototypical behavior, relationships, and interactions are known to both actors and audience, and a *character* that is a personality defined as a set of psychological traits.

Improvisational actors must follow a series of predefined heuristics (directions) to produce attractive performances like, for instance: accepting all offers, not blocking your partner, doing the natural thing, trying not to be clever, and incorporating previously generated elements.

In this situation, directing improvisational actors means that constraints can be defined for any one of these three elements:

Role-Constrained Improvisation: establishes which sets of actions are considered to be 'appropriate' and 'neutral' for each given role. Any actor should respect these constraints and only choose a kind of behavior within these sets. For instance, a servant should open the door for his master.

Plot-Structured Improvisation: defines particular plot outlines that actors use within their joint performances, like, for instance: in a plot the servant acts 'as usual' and behaves deferentially while in another plot the servant could defy master's authority by refusing to obey an order.

Character-Constrained Improvisation: determines the different kinds of behavior an actor could perform based on a series of variables that stresses the current status of the character. Three status variables were suggested by the authors:

1. *Demeanor* refers to a character's innate behavior like body posture, jerky movements, hand gestures, and so on.
2. The *relationship* refers to the character's social behavior like authority or subordination, avoiding eye contact, pointing fingers, etc.
3. The character's status in *space* denotes its relationship with the surrounding environment and objects. Is the character willing to interact with them or not?

3.3.3.3 Teatrix

Teatrix [Prada et al., 2000, Paiva et al., 2001] is a game-like interactive learning environment populated with 3D virtual characters that act autonomously and play specific roles. The project aims at providing such an interactive collaborative environment in which children can create their own stories by choosing scenes and characters and by acting and writing.

In Teatrix, the creative process for the stories was divided into three phases: set-up, creation and writing. In the story set-up, children can describe the main features of the story like the scenes, characters and props. The scenes provide the spatial setting for the characters. The characters are specified by their name, type and role (villain, hero, helper, magician, beloved one and beloved relative). The props can be used to illustrate the characters' behavior (like a magic wand, for instance).

In the second phase (creation of the story), children can select (from a predefined set) all the actions they want the characters to perform. This means that they can create several different stories from a single story set-up (as established earlier). The third phase is the writing of the story and consists of a recorded movie of the entire usage of the software, which allows the children to store and share their stories, and thus enhance their learning experience.

3.3.3.4 BEAT

BEAT (or Behavior Expression Animation Toolkit) "*allows one to animate a human-like body using just text as input*" [Cassell et al., 2001].

The system is composed of three main modules:

1. The **Language Tagging** module is responsible for "*annotating the input text with linguistic and contextual information*". This is carried out by converting the text into a tree representation of the most important units (utterance, clause, theme/rheme, action and/or object);
2. The **Behavior Generation** module (as the name suggests) is responsible for generating appropriate behavior after analyzing a previously annotated sentence by the language tagging module. This behavior generation is divided into two sub-modules:
   - *Behavior Suggestion:* this takes the resulting tree from tagging a sentence and produces a list of possible (coherent) types of behavior for each of its nodes;
   - *Behavior Selection:* this consists of a series of filters that are designed to remove from the behavior's list of nodes, those have not been regarded as 'playable'. This is due to the fact that two different types of behavior are trying to manipulate the same DOF at the same time;
3. Finally, there is the **Behavior Scheduling** module which takes the resulting tree and converts each of its nodes into abstract animation commands for the characters. As well as this, it is responsible for converting these abstract commands into intelligible commands for a selected animation tool.

3.3.3.5 Façade

Mateas and Stern define Façade as "*an attempt to create a real-time 3D animated experience akin to being on stage with two live actors who are motivated to make a dramatic situation happen*" [Mateas and Stern, 2003].

In Façade, the player acts like a friend trying to intervene in a quarrel between a married couple. He/she can type sentences that are autonomously interpreted by the characters, and change their 'opinions' and course of actions. The system includes several languages with specific purposes and is designed to assist in the authoring of the characters and the dramatic situation:

A Behavior Language allows specification of the behavioral responses for the actors;

Natural Language Understanding is a forward-chaining template language that aims at mapping the typed sentences of users into speech patterns;

Reaction Decider describes all the possible reactions to the speech patterns; and

Beat Sequencing assists the management of dramatic situations.

3.3.3.6 FearNot!

FearNot! (Fun with Empathic Agents Reaching Novel outcomes in Teaching) [Aylett et al., 2005] describes a simulated environment populated with autonomous virtual characters that aims at fostering an empathic reaction among children playing the characters of the victims of bullying.

The inspiration for this approach came from the theater. A Brazilian dramatist has proposed a role-playing "enactment exercise" of political activism. FearNot! then, adopted an approach where children were asked to play the part of an 'invisible friend', and give advice to the victim character, and thus influence the agent's decisions. The purpose was to attempt to achieve a sense of responsibility among children, while also analyzing what just had taken place.

3.3.3.7 Puppet & CrossTalk

The Puppet project [Klesen, 2005] was a system designed to improve learning through improvisational theater. Improvisational theater means creating interesting scenarios for the improvisers to act in, and is based on a set of explicit or implicit rules like, for instance, the status (the relationships with other characters and with the surrounding environment) and attitudes of the characters (a manner of feeling and behaving while pursuing one's goals).

CrossTalk [Baldes et al., 2002, Gebhard et al., 2003] is a system that incorporates animated presentation agents in a TV commercial-like situation, to advertise products for sale. It combines two sub-systems: cyberella (a conversational agent) and an inhabited marketplace (IMP), which works as a scenario for two virtual actors (Tina and Ritchie). These actors play the roles of a salesman and a customer, respectively, and the IMP works as a virtual showroom, the difference being that it is not a mere enumeration of facts about the product nor does it have a fixed discourse for presentation.

The following factors can be determined in advance for the user: characters' roles, attitudes toward a product, personalities and interests. The IMP sub-module can work in two different modes: presentation (each character plays the role attributed to it) and off-duty (a character will step out of its role, acting as a professional actor), based on a concept called meta-theater. Meta-theater creates a mixture of fiction and reality by providing actors with roles and meta-roles, which means that, during the performance, the actor can decide to jump out of his role and start 'acting as himself'.

In this project, the autonomous agents were designed to autonomously react by correlating actions with specific contexts and previously annotated sentences. Table 3.1 shows the dialogue act tags list suggested by the authors.

3.3.3.8 CREACTOR

CREACTOR (see Figure 3.5) is a project aiming at "*devising an authoring framework for interactive, adaptable, situation aware, partly autonomous virtual actors for the*

Table 3.1: Dialog Act tags for CrossTalk

Dialogue act	Description
admire	Speaker expresses admiration for the addressee.
attack	Speaker attacks addressee by a verbal argument.
bad_joke	Speaker makes a bad joke that targets the addressee.
boast	Speaker praises him-/herself in front of the addressee.
calm	Speaker calms addressee.
chide	Speaker seriously chides addressee either for doing something wrong or doing something morally bad.
command	Speaker commands the addressee to perform some action.
congratulate	Speaker sincerely congratulates the addressee (if meant ironically, use mock).
console	Speaker consoles addressee who faces some bad event.
correct	Speaker corrects a mistake the addressee has made.
criticize	Speaker criticizes the addressee for performing an action not well enough.
doubt	Speaker expresses doubt concerning something the addressee said or did.
defend	Speaker defends his/her argument against the addressee (reaction to “attack” or “correct”).
encourage	Speaker encourages the addressee to do or believe in something.
excuse	Speaker excuses her-/himself in front of the addressee.
good_joke	Speaker makes a good joke with addressee as an audience.
insult	Speaker directly or indirectly insults the addressee.
mock	Speaker mocks addressee. Weaker than an insult.
praise	Speaker praises addressee for some action or attitude.
regret	Speaker regrets some action or attitude in front of addressee.
reproach	Speaker reproaches addressee for some action or opinion.
sulk	Speaker expresses her/his being filled with indignation by the addressee’s action.
tease	Speaker teases addressee. This is milder than an insult or mocking.

creation of linear computer animation with 3D-virtual characters” [Iurgel et al., 2009]. This project focused on investigating several requirements towards the development of a novel authoring tool for character-based animations. The proposed system should allow the user to author animations by defining acting performances at several different levels of abstractions, from abstract command of a director to changes on the level of the body of the characters.

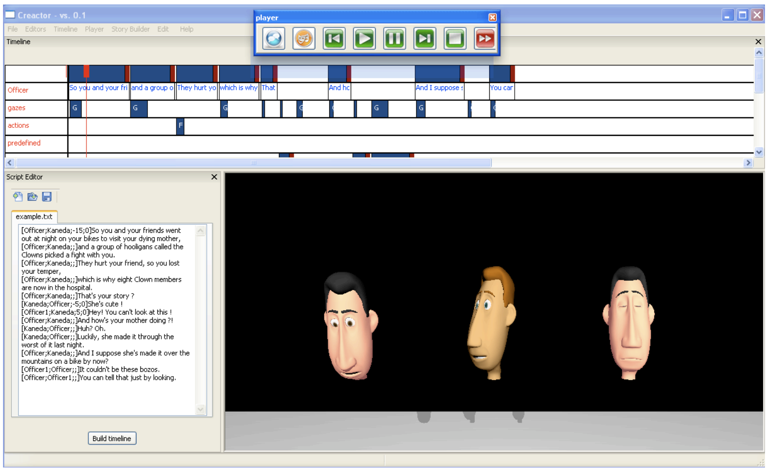


Figure 3.5: View from the CREACTOR interface

In [Iurgel et al., 2010] it has been identified, among others, the following non-trivial, specific properties of a Virtual Actor: AI-Kinematics, AI-Tweening, Continuous-Handling of Underspecification, Horizontal Coherence Maintenance, Vertical Coherence Maintenance, Multi-Level Control, Narration Understanding, Dialogic Authoring.

AI-Kinematics: is a concept that refers to a second Virtual Actor adapting autonomously to manual or generated changes of another actor (as in figure 3.6 (Left)). For instance, a manual gaze change of actor A requires automatic adaptation of the gaze of actor B, if there was gaze interplay between both. This important example also clarifies why visualizing adaptation dependencies is important, since the author will want to know what is affected by his/her changes, and will want to control, block and correct adaptations.

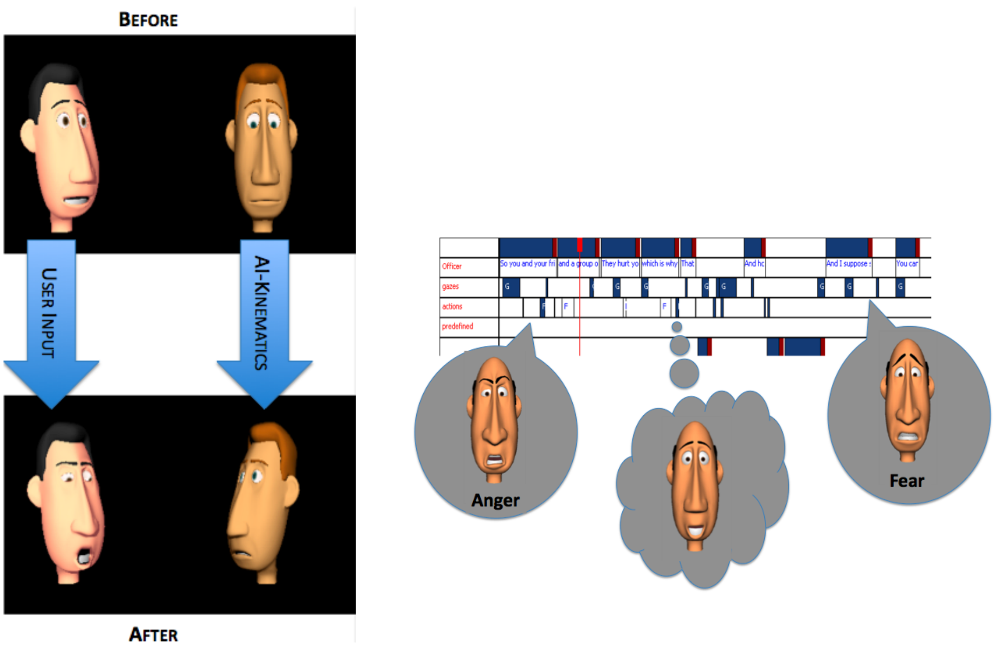


Figure 3.6: On the left, AI-Kinematics suggests coordinated actions between actors. On the right, AI-Tweening filling the gap between suggested behaviours

AI-Tweening: this concept is an analogy to the tweening between key-frames with interpolation functions. But AI-Tweening occurs at the level of abstract directing of expressive behaviour (see figure 3.6 (Right)). The system autonomously fills-in missing commands related to expressions and to acting by estimating coherent implications of surrounding commands of the author, and on contextual parameters. For instance, a command of the author to “play a fierce anger” followed later (leaving an underspecified gap in time), by a command to “play a calm person”, would lead the ADA-system to fill-in autonomously between both commands of the author a command “to cool down”, since it normally there is a cool down phase between anger and calm.

AI-Tweening can be correlated to a guided authoring process that follows leads from script-writing. The authoring process can be guided to follow these steps: (a) detect the main plot-points (“beats”) of the story; (b) define in detail the behaviour of the Virtual Actors at these plot-points; (c) let the system employ *AI-Tweening* to fill-in the gaps (thus generating *AI-Tweens*); (d) correct the generated behaviour, focusing on secondary plot-points, and letting AI-Tweening fill in remaining gaps; (e) correct generated AI-Tweens where necessary, which enables further autonomous improvement of existing AI-Tweens.

Continuous-Handling of Underspecification: the author can issue commands at various levels, but can leave parts of the scene underspecified, or specify for example only at a lower, but not at a higher command level (cf. below). This means that the system has to handle this underspecification autonomously, by filling in the non-specified elements. This auto-completion of gaps (figure 3.7) means that, if the author does not animate something (e.g. not the blinking, or not whole parts of a scene, or not a complete scene, or even the whole play of an individual actor) then the system will animate the missing parts autonomously. There is never any *dead* moment of the animation where actors are not coherently animated. This shall be possible based on context information, including information about how the others Virtual Actors of the scene are behaving.

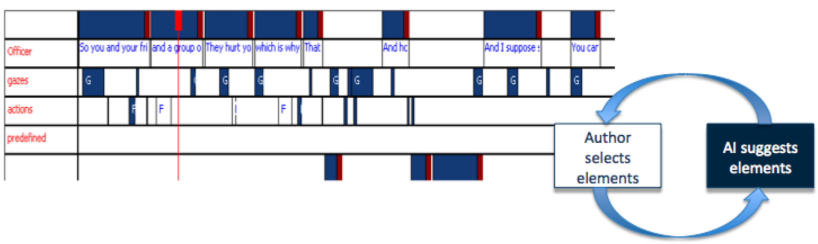


Figure 3.7: AI-assisted authoring process

Horizontal Coherence Maintenance: this concept means that the user can define behaviours at several moments of a scene. The transitions between behaviours and the mutual influence of the behaviours must be assured by the system. This includes the maintenance of the coherence of behaviours that are not subsumed by a common command (e.g. the commands “play a marriage quarrel” followed by the command “play a remorse”), and commands that are both controlled by a common higher-level command (e.g. the command to “look at someone”, followed by the command to smile in a certain way, while both commands are under the control of the command “play a marriage quarrel”). AI-Tweening is an example of this kind of coherence maintenance.

Vertical Coherence Maintenance: this coherence maintenance is required when an upper level command that generates behaviours that can be described with lower levels is overwritten by the user at (certain aspects of) the lower levels. The system must be able to maintain the coherence of the performance and to integrate and complete the interference of the author.

Multi-Level Control: relevant command levels construe a hierarchy. This hierarchy has precedents for instance in the work of [Stefan et al., 2007], but especially in the efforts of the SAIBA working group, cf. also [Vilhjlmsson et al., 2007]. Possible control levels are:

Virtual Body Level: represents the control of the actor's body, explicitly defining at a time t the position of the head, eyes, eyebrows, mouth, etc.

Moving Level: moving level commands can be composed of a series of “virtual body” level commands. The moving level differs from the virtual body level in being temporally extended, e.g. “yawn”, “look at”.

Expression Behaviour Level: it contains intended expressions such as “friendly turn-taking”, which can be realized in several manners, for instance with a change of eye-gaze, with a smile, with a bending of the head, etc. Expression behaviour level elements may contain elements that translate one-to-one to moving level behaviours.

Director Behaviour Level: director behaviour is a behaviour that is hierarchically composed of expression level commands. Here, the author defines specific acting directions for the Virtual Actors. The director behaviour contains specific acting autonomy in the sense that the system chooses which expression level behaviours to employ to fulfill the directions.

Narration Understanding: the Virtual Actors must be able to perform according to their interpretation of a story. Script analysis of different points of view on setting a scene (point of view of the actor, the director, the audience)[Thomas, 2009, Weston, 2003] can be a guidance to developing such faculties. Finding ways of allowing Virtual Actors to autonomously interpret a story, and from that, to enact it, will be determinant for what can be achieved with Virtual Actors. Currently, human-made semantic annotations of the script seem to be the most reliable path to follow, though we are also investigating the possibility of interactive assistance of this process of annotation.

Dialogic Authoring: the system shall be able to "ask", i.e. to collect further information about the reasons why the user authors in such and such a way. This is important because of the expected vast occurrence of underspecification. For example, the system could ask whether a certain Virtual Actor is not displaying anger at a certain moment because it has calmed down, or because it is dissimulating, if this difference is important for the system to autonomously act out other parts of the play.

3.4 Understanding the Authoring Process

To animate literally means to bring to life. Computationally, this means to set up a series of actions over time which are displayed by computer systems.

Authoring describes the process used to define the structured actions that will be animated. Basically, there are two approaches to author animations: interactive animations and batch animations.

In interactive animations, also called *character-based animations* [Mullen, 2007], the actions are displayed as soon as the acting decisions have been made. The animator (via sending inputs to the animation program), can only influence the characters' behavior but not determine it. This approach mainly relies on reactive agents (see Section 3.5.1) whose main goal is to properly come up with plausible reactions to all the user inputs, and take account of its own goals (regarding the plot) in carrying this out. Since there are no explicit representations for the story being enacted, the agents cannot reason about the story and their role in it 'holistically'. This approach is suitable for authoring events quickly, but handicaps the author when seeking to control explicit events.

Batch animations (or *plot-based animations*), on the other hand, consider 'batching' all the actions that need to be animated in time, as a data structure (e.g. a list, a tree or a graph) called a `timeline`. After this, the animation actions are scheduled for rendition. The main drawback of this approach is that there is a temporal gap between the decisions about the agents' enactment and the visualization of the final animation, which results in a significantly lengthy authoring process. The advantanges are that, as the rendering process is carried out separately, more sophisticated rendering, sounding and editing, can be considered during the authoring process. For instance, by using a timeline, an animator can design the whole plot (character's actions, camera shots, music, lighting, etc.) iteratively in several stages, and only schedule it for rendering when satisfied with the description.

3.4.1 Animation of Acting

According to Mullen [Mullen, 2007], the process for creating and modeling a character for an animation film can be divided into the following steps:

1. Modeling the 'body' of the character as a triangles mesh. This stage also includes all the details regarding the appearance of the character like coloring, texturing and particles (e.g. hair);

2. Creating the skeleton and rigging for the body. The skeleton is a simplified hierarchical model representation for the character's bones and the rigging combines each bone with a series of vertices of the body;

3. Creating shape keys (predefined facial expressions) and facial rigging;
4. Determining a series of poses for the character by modifying the bone structure to a specific configuration;
5. In an analogous way to the modeling of the body, determining facial posing and lip sync poses;
6. Animating the character by creating 'timelines' that describe the acting for each scene of a film. This stage also includes the character's interaction with props and other elements of the scene.

Further issues involving the animation process are: lighting, rendering and editing. This research will not deal with these stages, so that the process can be simplified and will delegate these areas to the animator.

All the first five stages are related to creating the physical components for the characters and are the responsability of the graphics artists. Moreover, this research assumes these issues have already been solved and only focuses on the animation stage (step #6). Owing to the need for simplification, the characters are constructed by three main components: the body, face and head. The next sections describe ways of animating each of them.

3.4.1.1 Body Postures

According to Maestri [Maestri, 2006], two basic components need to be borne in mind when animating a character's body naturally: balance and anatomical correctness.

Mullen [Mullen, 2007] argue that the body reflects the person's inner states better than the face (which can be contrived). He suggests four basic generic forms of body postures (Figure 3.8): open, closed, forward and back.

These generic body postures can be combined to form four modes: responsive, reflective, combative and fugitive.

Other gestures that characters use when trying to communicate with their bodies are:

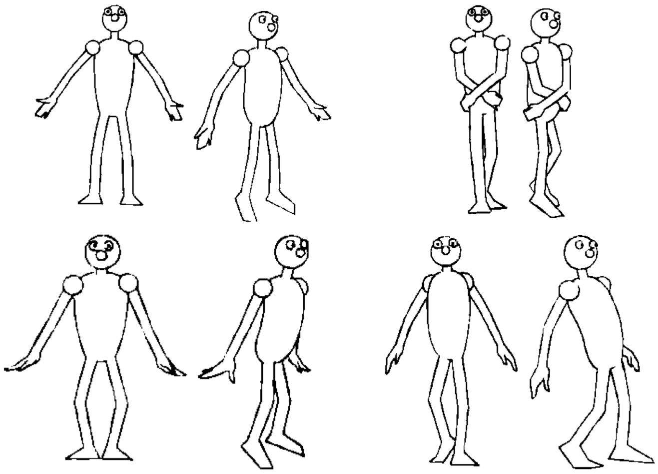


Figure 3.8: Four Generic Forms of Body Postures

Adapted from [Roberts, 2007]

Table 3.2: Four Body Postures Modes

Mode	Combined postures	Some associated feelings
Responsive	Open + Forward	happy, interested, engaged, in love, wanting something, liking something
Reflective	Open + Back	considering something, thinking, evaluating or feeling perplexed
Fugitive	Closed + Back	rejected, bored, sad, miserable, in denial, reluctant
Combative	Closed + Forward	angry, defiant, having an argument

Hips and Torso: the hips support the weight of the upper body and spread it through the legs to the ground. Almost every pose starts with hip placement followed by counterbalancing the hips with the torso (the spine and shoulders).

Palm Gestures: palms upward and open suggest honesty and submission, while open and turned downward suggests dishonesty and dominance. Closed palms suggest anger and aggression.

Hand Gestures: rubbing hands together suggest eagerness and excitement; clasping hands with interlocked fingers can be interpreted as meaning that the person is attempting to suppress negative feelings like frustration; if both thumbs and all the fingertips are pressed together, this indicates superiority.

Crossing Arms: signals discomfort with the situation and if accompanied with clenched fists, indicates a hostile intention. Partially crossed arms is associated with a lack of confidence and crossing them at the wrists shows humility.

Legs and Feet: the legs usually bend slightly to support the weight of the body, and the feet will usually point outwards with the knees following the same pattern.

Crossing Legs: can indicate defensiveness, displeasure, nervousness or lack of confidence. It can also simply be a sign that the person is trying to make themselves comfortable.

3.4.1.2 Facial Expressions

Human beings use facial expressions to convey their feelings to other people. Our brains have evolved to quickly recognize the significance of facial expressions, so this communication channel has become one of the most important in human interaction.

Psychologists have been studying facial expressions for several decades, although there is still no consensus about whether there are universal facial expressions and, if so, how many. Ekman and Friesen [Ekman and Friesen, 1978] initially argued that there must be at least six inherited basic emotions that are widely recognizable by every human being (happiness, surprise, fear, sadness, anger, disgust / contempt). Several years later, they adjusted their theory to distinguish disgust from contempt, thus making it seven basic emotions [Ekman, 1992].

With regard to character animation, Roberts [Roberts, 2007] supports the use of no less than eight basic emotions (happiness, sadness, surprise, fear, anger, disgust / contempt, interest and pain), which can be combined in around 5000 expressions.

True expressions of emotion tend to last for short periods of time, as opposed to false ones, that contradict a character's inner states (like smiling when someone feels like crying). The main expressions the face can display are as follows:

Gazing: one key element of any facial expression is the eyes. The eyes lead all the movements of the body: they move first, then the head, then the shoulders and then the waist. If someone does not move his eyes, he will look groggy. Another important feature of the gaze is pupil dilation. When a person is looking at something that he likes, his pupils dilate; when he feels frightened, the pupils contract and when surprised his eyes will bulge.

Blinking: the author suggests using at least three different types of blinking:

1. the 'standard' blink, when the eyes will close slowly and open quickly.
2. the 'in love' blink, when the eyes close and open slowly.
3. and the 'sleepy' or 'stupid' person who will close his eyes slowly and open them even more slowly.

Happiness: there are several different ways to show happiness, that vary in intensity. Usually, the eyebrows are raised evenly, with the corners of the mouth rising symmetrically. The mouth may be closed or open showing the teeth. Crow's-feet only appear around each eye when the smile is sincere.

Sadness: can vary from disappointment to despair. The inner corners of the eyebrows are raised and the outer part of the eyebrows lowered, the upper eyelids droop, the corners of the mouth are turned down with the jaw dropping to form a long face. When crying, the eyes will be closed or partially closed. A pretended sadness is shown by a slight curling of the mouth, crow's-feet, puffy bags under the eyes and irregular eyebrows.

Surprise: the eyebrows are raised high and the brow is furrowed. The mouth is usually open with the jaw dropped. The anticipation of surprise causes a rapid movement of the head and a blink. It can vary from slight surprise to amazement.

Fear: varies from apprehension to terror. The main characteristics of fear are: a dropped jaw, the mouth open with the corners pulled outward and downward. The eyebrows raised and knitted. The eyes wide open with tense eyelids and dilated pupils. Pretended fear usually means having an exaggerated expression that lasts too long, with regular pupils moving erratically.

Anger: is characterized by the inner corners of the eyebrows being pulled downwards and inwards. The pupils contract. The nostrils swell and flare out. The lips might be either closed and pressed together or open with the teeth clenched, or even wide open. A pretended display of anger will involve forcing this expression thus causing it to last a short period of time as the person is unable to keep it up for long. Anger can vary from irritation to fury.

Disgust/Contempt: the head recoils from the object causing the emotion, the eyebrows are lowered and pulled together horizontally. The eyes are narrowed and the pupils contracted. One or both corners of the upper lips are raised. In a pretended look of contempt, the person is unable to sustain the expression. It ranges from dislike to nausea.

Interest: this is in fact a combination of other emotions. It can assume several forms such as alertness, attentiveness, expectancy and anticipation. It is revealed by one or both eyebrows being raised, and the pupils might be dilated. The person always stares without blinking. The head might tilt to one side. The opposite of interest is boredom, which is characterized by a relaxed unanimated face; the eyes will droop and then look up or straight ahead. The person may also yawn.

Pain/Distress: this expression can range from discomfort to extreme pain. The mouth might be open and gaping or closed with clenched teeth. The eyebrows are together with the inner ends up and the outer ends down. The eyes are closed most of the time. Real pain usually lasts much longer than when it is assumed.

3.4.1.3 Head Positions

Following one of the 12 principles of animation [Johnston and Thomas, 1995], the head moves in arcs relatively to the neck and shoulders. Table 3.3 summarizes a list of head positions and possible ways they can be interpretated.

3.4.1.4 Acting in Beats

As discussed earlier, beats are separations of sections of a scene. They represent "*changes in dynamic, emotion, or purpose*" for a character within the action [Roberts, 2007]. A character beat is divided into three parts:

Table 3.3: Possible Interpretations of Head Positions

Head Position	**Interpretations**
Forward + Upward	Willingness to engage (interest)
Forward + Downward	Argumentative, determination, tiredness, stupidity
Backward + Upward	Arrogant or snobbish appearance
Backward + Downward	Frightened or nervous
At an angle	Interest if eyes stare the direction or shyness if the head faces the opposite direction from the eyes
Tilting the head	Showing doubt or Asking a question

1. **Objective** is the specific, limited thing the character wants at a particular moment.

2. **Obstacle** is the specific, limited thing that prevents the character from attaining his objective.

3. **Action** represents the tactic or strategy the character choose to overcome the obstacle to attain an objective.

3.4.2 Representations of Actions

Animation of acting can be described as constituting a series of actions. There are several different types of actions that need to be synchronized over a period of time: body postures, facial expressions, gazing and head movements and lip sync. As animations are a particular case of general multimedia presentations, one can take advantage of these authoring paradigms. According to Bulterman and Hardman [Bulterman and Hardman, 2005], authoring in multimedia presentations involves a series of concerns:

Media assets: the media objects used in a presentation.

Synchronized composition: describes the relations between the different media assets.

Spatial layout: the mechanism that controls how the assets are displayed.

Asynchronous events: indicate the conditions that influence the synchronized composition.

Adjunct/replacement content: describes alternative approaches to giving presentations.

Performance analysis: feedback regarding the performance of the presentation.

Publishing formats: describe the format adopted for encoding presentation.

In addition, authoring multimedia presentations can be carried out by taking account of four paradigms:

Structured-based Paradigm: is an abstract representation that defines media assets as a hierarchy with intermediate nodes containing compositions, and root nodes containing media assets.

Timeline-based Paradigm: is a representation with a clear temporal relationship between the assets.

Graph-based Paradigm: is a representation based on a directed graph that is used to characterize the synchronization of the media assets.

Script-based Paradigm: uses a procedural representation to specify types of behavior, during the graphical representation used by all the three previous paradigms. This allows for a more precise description of the behavior that is being represented, while also being more abstract.

With regard to character animation authoring, it seems that the second paradigm (timeline-based) is the most suitable, since it is the only one concerned with the representation of temporal relations. However, there is another factor to consider: non-linear animations. In non-linear animations, actions are represented at different levels of abstraction (from physical actions like ‘open mouth’ or ‘blinking’ to abstract behavior like 'walking furtively' are represented in the same data structure. Usually, these representations are not concerned with explicit temporal relations, and prefer working with other kinds of associations.

The behavior markup language (or BML) [Kopp et al., 2006] is an example of a script language that aims at specifying animation commands in a structured manner. Figure 3.4 summarizes the tags used in the BML language.

Table 3.4: BML specification tags

BML Element	Description
<head>	Movement of the head independent of eyes. Types include nodding, shaking, tossing and orienting to a given angle.
<torso>	Movement of the orientation and shape of the spine and shoulder.
<face>	Movement of facial muscles to form certain expressions. Types include eyebrow, eyelid and larger expressive mouth movements.
<gaze>	Coordinated movement of the eyes, neck and head direction, indicating where the character is looking.
<body>	Full body movement, generally independent of the other behaviors. Types include overall orientation, position and posture.
<legs>	Movements of the body elements downward from the hip: pelvis, hip, legs including knee, toes and ankle.
<gesture>	Coordinated movement with arms and hands, including pointing, reaching, emphasizing (beating), depicting and signaling.
<speech>	Verbal and paraverbal behavior, including the words to be spoken (for example by a speech synthesizer), prosody information and special paralinguistic behaviors (for example filled pauses).
<lips>	This element is used for controlling lip shapes including the visualization of phonemes.

Source: [Kopp et al., 2006]

A suitable approach to apply on autonomous digital actors, combines the BML structured abstract commands with the temporal relations stressed by a timeline-based paradigm.

3.5 Artificial Intelligence

This section aims at summarizing some Artificial Intelligence techniques that we consider relevant for the development and understanding of this thesis.

3.5.1 Affective Agents

Maes [Maes, 1995] defines these as follows "*autonomous agents are computational systems that inhabit some complex, dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed*".

There are basically three types of agents: reactive, deliberative and hybrid. Reactive agents act by reflex, which means that they establish a direct correlation between input and output; deliberative agents act as if, for each input, the agent would be capable of some sort of analysis and planning before acting; hybrid agents possess the characteristics of both the reactive and deliberative agents.

In Macedo [Macedo, 2007], the author outlines a taxonomy for agents based on several works. This study claims that, an agent should display the following characteristics: autonomy, temporal continuity, character (personality), communication, adaptability and mobility. These characteristics, in fact, depend on the domain of application which the agents are designed to inhabit.

The agent architectures discussed here are defined on the basis of four key modules [Byl, 2004, Nareyek, 2002]: `perception` which is responsible for sensing the environment and converting stimuli into percepts, `inference` which carries out a percept-based form of decision making, `selection` which determines an order (priority) for the actions that have been produced by the inference module previously and `behavior` which executes the set of selected actions.

Kline and Blumberg [Kline and Blumberg, 1999] subdivide the perception module into external and internal influences. External influences are the perception of the world which can be provided by real-world sensing (in robots), synthetic vision (extracting information from a world model) or direct sensing (asking the surrounding objects for guidelines). Internal influences are agent's resulting interpretations of the world like, for instance, its emotions.

Reactive Agents: The simplest agent architecture, also called *reflex agents*, works on the principle of stimulus-response. This model attempts to represent quick thinking instinctual behavior through a predefined set of rules. For example:

if hear(noise) then action(activate_alarm)

The main features of this architecture are:

- The perception module acts as a fact receiver, which means that it looks at the surrounding environment in an attempt to recognize facts. A fact is an entity that can assume two possible boolean states: *true* or *false*. In the example above, `hear(noise)` is a fact;
- As perception can only provide one of two possible responses, the inference module can only make a single decision: *to carry out a given action or not?*. In the example above, the action `activate_alarm` can only be selected if the fact `hear(noise)` is considered to be true;
- The selection module is unnecessary in this architecture because there will at most be only one possible action to be performed; and
- The behavior module in this case will always perform one action at a time. In the example above a sound could be set off whenever the agent hears a noise.

This kind of agent can be implemented by being mainly based on Rule Based Systems (RBS) and Finite State Machines (FSM). They show the advantage of performance (it was accompanied by a quick response that is ideal for real time applications such as digital entertainment); however the programmer is responsible for foreseeing all the possible environmental states of the agent, which could be very a difficult task due to the complexity of some environments.

Triggering Agents: A triggering agent is basically the same as a reactive one, and also has the same drawbacks, except for the fact that its perception module can detect both internal and external stimuli, which means that the agent knows its own current state.

A triggering agent rule could be:

if hear(noise) and state(awake) then action(activate_alarm)

In this case, agents have a memory for past events and are able to use this information to make future decisions. In the example above, the agent might fall asleep, and thus fail to activate the alarm.

Goal-based Agents: This introduces the concept of goals. A goal is a relationship between actions where a specific stimulus can lead to a set of sub-actions (called a *plan*), unlike the previous architectures where they have a direct association stimuli-response.

Byl [Byl, 2004] offers the following example:

Rule 1: *if hear(noise) then goal(alert_others)*
Rule 2: *if goal(alert_others) then goal(activate_alarm)*
Rule 3: *if goal(alert_others) then goal(use_radio)*
Rule 4: *if goal(activate_alarm) then plan(locate_alarm, move_to_alarm, sound_alarm)*
Rule 5: *if goal(use_radio) then plan(pick_up_radio, set_frequency, call_for_help)*
Rule 6: *if plan(x,y,z) then action(x),action(y),action(z)*
Rule 7: *if hear(alarm) then ¬goal(activate_alarm)*

In the previous example, the current agent state `alert_others` will trigger two sub-goals: `activate_alarm` and `use_radio`, each with an associated plan of action. A goal-based agent is unable to determine which plan is better and usually tries each one sequentially.

Utility-based Agents: As discussed earlier, a goal-based agent has no preference over which plan to execute. In view of this, the concept of **utility** is introduced, which is a value that represents any quantified information of a goal for the decision making process and can change over time, such as the amount of ammo or health remaining [Collins et al., 2001, Russell and Norvig, 2002]. The selection module of an agent can use this utility value to choose a plan for stating a preference, for instance, if the agent's current state is "`alert_others`", he can `activate_alarm` and/or `use_radio`. However, he might choose one or the other regarding to the distance from the alarm and radio.

Anytime Agents: The term anytime agents is employed for those that always have a plan in hand, no matter what the circumstances may be. To achieve this goal, restriction are imposed on the agent's internal processes (particularly in the inference and selection modules) that are defined by the time that has elapsed since the last update of the current plan.

This approach has several advantages when compared with the previous one, like being able to ensure that the agent always has a plan. In addition, it may not have to compute an entire plan because it can update it after a slack-time.

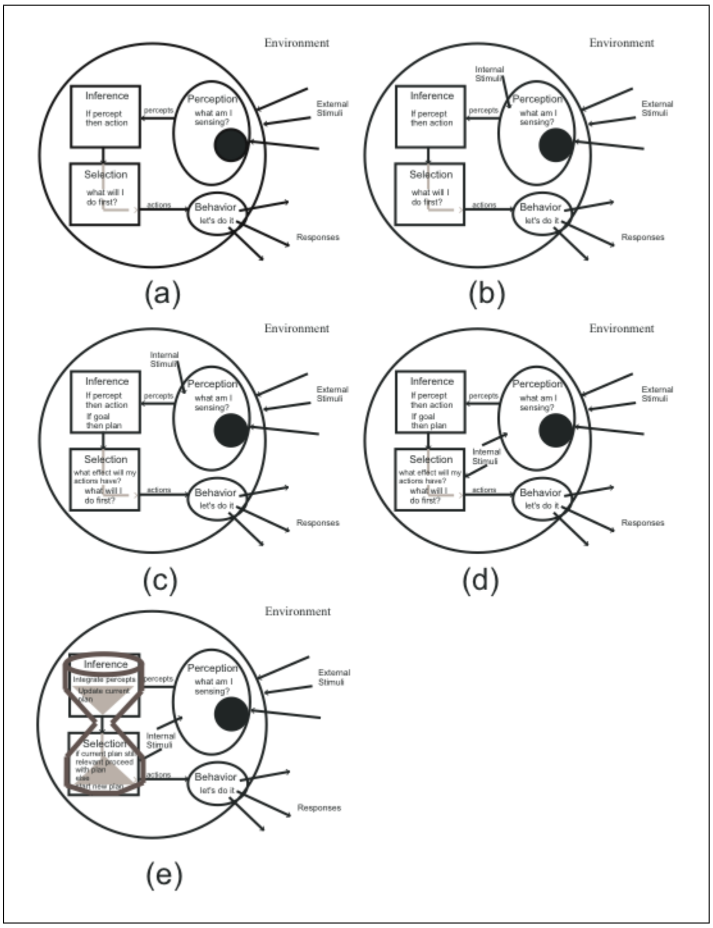


Figure 3.9: Agent Architectures: (a) reactive agent (b) triggering agent (c) goal-based agent (d) utility-based agent (e) anytime agent

Adapted from [Byl, 2004]

3.5.2 Psychologically-Aware Computing

In the same way as most autonomous character, virtual actors are very likely to be expected to interact with human users. Before they can be enabled to do this in a wholly believable and appealing manner, these characters must be able to understand human psychology in some sense.

Ball and Breese [Ball and Breese, 2000] argue that to produce psychologically aware computers, "a system must be able to (1) evaluate the personality and emotional state of the user and then (2) synthesize and communicate an appropriate social response from the computer".

Thus, emotion and personality are two key features that need to be understood by our autonomous digital actors so that they can behave as if they have their own psyche, and hence become more believable.

3.5.2.1 Emotions

Human decision-making is strongly influenced by a person's mood (or emotional state) and can not be disregarded when planning the specifications of a computational model for the human decision-making process. Picard [Picard, 1997] states that there are three principal groups where *Affective Computing* takes place, but she warns that these categories are not mutually exclusive, which means that research projects can make use of more than one at the same time:

Recognition of Human Emotions: the effect known as sentic modulation, which is our physical movements and expressions when under the influence of emotions, such as facial expressions, pupillary dilation, heart beat rate, blood pressure and so on, could provide essential information to increase the number of human-machine interactions;

Modeling and Simulation of Human Behavior: the purpose of this is to create computer representations of human behavior in a realistic manner through models of human emotions;

Expression of Emotional-like Behavior: the idea is to create synthetic characters that are capable of expressing emotive behavior as if they are actually feeling emotions.

Autonomous digital actors are (at least) expected to be able to understand the emotional implications involved in the abstract scenarios in which they are embedded, and also cope with these implications by reflecting the appropriate behavior through their communication channels. Thus, emotional expression and simulation are important skills that any actor agent should be endowed with.

Emotion Expressions: Autonomous characters (just like real people) have several emotional channels, like the face, body and voice, to display emotions. It is through emotional responses (expressions + interpretation) that people understand if what they are saying is being understood by their interlocutors and, if what they are saying is producing the expected reaction in others.

The most common channels that people use while expressing emotional states are the face, body and voice. How to control specific muscles in our body and to match a given display, is an ability that we all learn from the time we are born, and we keep practicing and improving these skills throughout our lives. As well as this, we need to learn how to recognize these same displays on others, so that we can interpret (infer) what they are trying to communicate to us. Paul Ekman [Ekman, 1999, Ekman, 2003] argues that there must be a set of emotional facial expressions that are inborn and universal.

By facial expressions we mean any physical expression of emotion that involves a combination of the muscles of the human face. Ekman & Friesen [Ekman and Friesen, 1978] have proposed the, now commonly used, Facial Action Coding System (FACS) to measure and describe all the possible facial expressions. There are 46 AUs defined by the FACS like, for instance, brow lowerer, inner brow raiser, wink, cheek raiser, upper lip raiser, and jaw dropping.

Emotional Simulation: believable characters have to be able to infer proper emotional responses from any known situation. Emotion models aim at just that. There are plenty of emotion models in the literature. The OCC model is probably one of the most popular ones:

OCC Model: the OCC model [Ortony et al., 1988], proposed by Ortony, Clore and Collins, defines emotions as valenced responses to events in the environment and is evaluated (through subjective criteria) by the agent (the reason why this model became popular for computation is probably because it is easily implemented by a computer program [Aylett et al., 2005]).

This model proposes that 22 emotions are valenced responses to events, agents or objects, as illustrated in Figure 3.10.

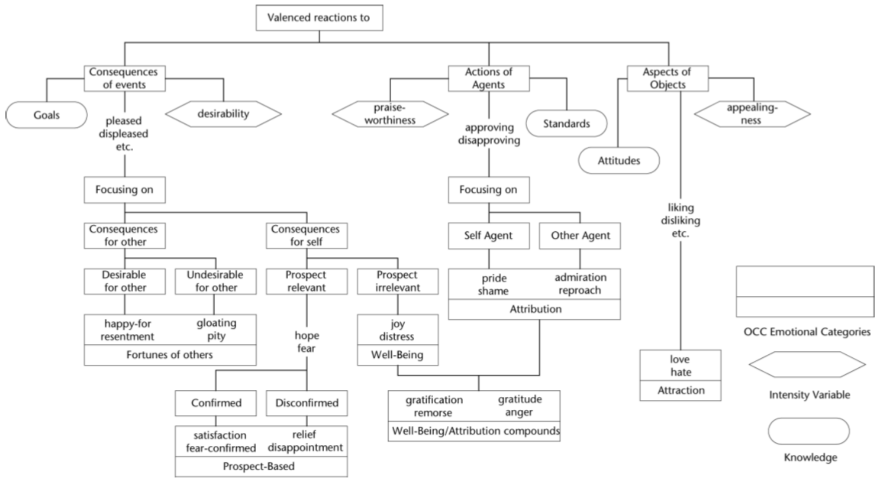


Figure 3.10: OCC Model

Source: [Bartneck, 2002]

3.5.2.2 Personality

Prendinger et al [Prendinger and Ishizuka, 2001] define personality as comprising patterns of thought, attitudes, and behavior that are permanent or at least change very slowly.

OCEAN Model: this popular theory argues that human personality can be described as consisting of five universal traits [Tupes and Christal, 1992, Busato et al., 1999]. One way to determine the values of these personality traits is to carry out online tests that are specially designed to assess them. An example of such a test can be found at `http://www.outofservice/bigfive/`.

Openness – inventive/curious (trait value = 100) vs. consistent/cautious (trait value = 0). Appreciation of art, emotion, adventure, unusual ideas, curiosity, and range of experience.

Conscientiousness – efficient/organized (trait value = 100) vs. easy-going/careless (trait value = 0). A tendency to show self-discipline, act responsibly, and aim for achievement; planned rather than spontaneous behaviour.

Extraversion – outgoing/energetic (trait value = 100) vs. solitary/reserved (trait value = 0). Energy, positive emotions, drive, and the tendency to seek stimulation in the company of others.

Agreeableness – friendly/compassionate (trait value = 100) vs. cold/unkind (trait value = 0). A tendency to be compassionate and cooperative rather than suspicious and antagonistic towards others.

Neuroticism – sensitive/nervous (trait value = 100) vs. secure/confident (trait value = 0). A common tendency to experience unpleasant emotions, such as anger, anxiety, depression, or vulnerability.

16 Basic Desires: Reiss [Reiss, 2004, Reiss, 2008] argues that "sixteen psychological needs (called basic desires) drive the human psyche and potentially explain a wide range of human experiences, everything from relationships to values". A basic desire can be understood as comprising five qualities: universal motivation, psychological needs, intrinsic motivation, intrinsic values and psychological significance. The sixteen basic desires are:

Power – desire for achievement, competence, leadership, influence

Independence – desire for freedom and self-reliance

Curiosity – desire for knowledge and truth

Acceptance – desire for a positive self-image and self-esteem

Order – desire for cleanliness, stability and organization

Saving – desire to collect and amass property

Honor – desire to uphold the values of morality, loyalty and having a good character

Idealism – desire for fairness and justice

Social Contact – desire for friendship and fun

Family – desire to have children

Status – desire for wealth, titles, attention, awards

Vengeance – desire to win, to compete

Romance – desire for sex and affection

Eating – desire for food

Physical Exercise – desire to be fit

Tranquility – desire for relaxation and safety

Cognitive Styles: Plantec (2004) [Plantec, 2004] (following the works of Gregorc) suggests that "every human being is endowed with a uniquely proportioned set of mental qualities for interacting with the world. These endowments are manifested as specific behaviors, characteristics, mannerisms and products known collectively as style". There are four dominant cognitive styles:

Concrete Sequential (CS) – punctilious, hard working, and a bit rigid in his ways.

Abstract Sequential (AS) – the absent-minded professor, with an active internal life.

Abstract Random (AR) – passionate social butterfly with cosmic connections.

Concrete Random (CR) – creative, self-centered, highly intuitive, fun.

3.5.3 Learning Classifier Systems

Before autonomous digital actors can be in a position to give compelling performances, they must be able to learn from previous experiences, improve their suggestions skills and comply with some sort of fitness criteria like, for instance, the animator's preferences.

Learning classifier systems (LCS) [Bull and Kovacs, 2005] "are a machine learning technique which combines evolutionary computing, reinforcement learning, supervised learning or unsupervised learning, and heuristics to produce adaptive systems". According to these authors, at least three variants of this algorithm have been proposed so far: LCS, ZCS and XCS. A fourth variation is suggested by [Santos et al., 2009].

Holland's LCS encodes traditional condition-action rules as binary strings (known as "classifiers") where each character comes from an alphabet {0, 1, #}, where # indicates a wildcard that can assume any value, so the rule `001#-001` matches both the following strings 0010 and 0011.

The algorithm works in accordance with the following stages:

1. create a match set ([M]) composed of all rules (within the knowledge base) that match the current external message under analysis
2. select a desired rule (from [M]) through a bidding mechanism that considers three factors with regard to time t: the *fitness* value of each rule, the *specificity* of the rule given by the number of non-# characters and a constant factor $\beta \in [0, 1]$:

$$bid(C, t) = \beta.specificity(C).fitness(C, t)$$

3. reinforcement maintains that all the winners' rules are rewarded with a scalar number that will improve its fitness function by making them more 'desirable' and thus chosen in the next instant of time.

Wilson's ZCS is a simplified version of the LCS model described earlier. Wilson disregarded the bid mechanism when selecting the desired rules and only based them on their fitness:

1. create the match set [M]
2. select the most suitable rule within [M] (highest fitness value)
3. create an action set [A], composed of all the rules (in [M]) that result in the same action as the previously selected fittest rule

4. all the members in the previous action set $[A]_{-1}$ are considered by means of a discount factor γ (temporal learning factor)
5. all the members within [A] receive a fraction (β) of the reward from the environment

$$fitness([A]) \leftarrow fitness([A]) + \beta[Reward + \gamma.fitness([A]_{-1}) - fitness([A])]$$

Wilson's XCS in this variant, instead of rewarding desired selected rules by a payoff (P), it rewards the most accurate predictions (p) of the payoff:

1. create a match set [M]
2. a system prediction (for each action in [M]) is formed on the basis of a fitness-weighted average of predictions in [A]
3. reinforcement consists of updating the following parameters:
   (a) the rule's error: $\varepsilon_j = \varepsilon_j + \beta(|P - p_j| - \varepsilon_j)$, where $0 \leqslant \beta \leqslant 1$.
   (b) the rule's prediction: $p_j = p_j + \beta(P - p_j)$
   (c) the rule's accuracy: $K_j = \begin{cases} \varepsilon < \varepsilon_0 & 1 \\ & \alpha(\frac{\varepsilon_0}{\varepsilon})^\nu \end{cases}$ where α, ε, ν are constants that control the accuracy function.
   (d) A relative accuracy is determined by $K'_j = \frac{K_j}{\sum allK_j}$
   (e) The fitness function (F_j) is adjusted by (K'_j) as follow:

$$F_j = \begin{cases} adjusted\ \frac{1}{\beta}\ times & F_j + \beta(K'_j - F_j) \\ & average\ of\ previous\ K'_j \end{cases}$$

   XCS fitness is an inverse function of the error in reward prediction, with errors below ε_0 not reducing fitness.

Santos' UCS: derived from previous XCS approach, the sUpervised Classifier System (UCS) is designed for supervised learning scheme. This variant relies on two states: the training and the testing and on a set of parameters as follows:

Accuracy: is a measure of the classifier performance. It is classified as correct (those with the highest payoff) or incorrect (those with the lowest payoff);

Number of Match: represents how many times the condition part of each rule in the testing stage matches the condition part of the training stage;

Number of Correct: indicates the number of times both the condition and actions parts of the testing and training stages match;

Correct Set Size: is the size of the population of the list of correct matches;

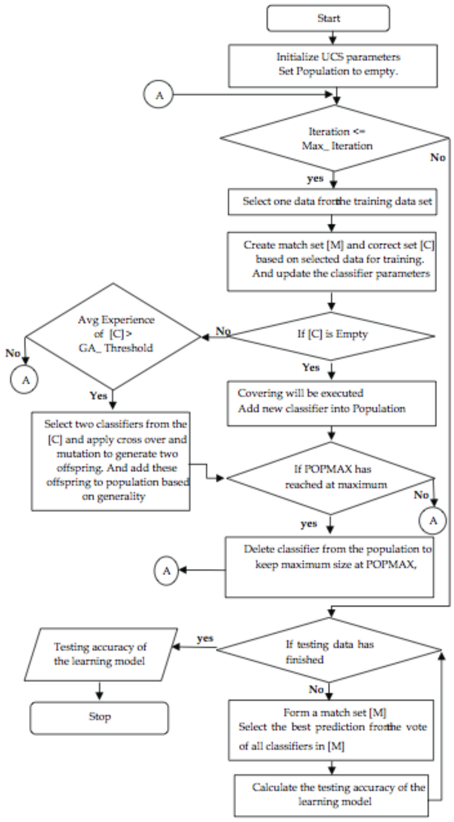


Figure 3.11: Flow chart of the UCS system

Source: [Santos et al., 2009]

Chapter 4

Road to an Agent Architecture

On the basis of literature review, some ideas have been investigated about how traditional acting techniques can help inspire the design of an autonomous digital actor:

1. Not every human acting technique is applicable to virtual actors. For instance, it does not make much sense to talk about relaxation for characters that only exist in a virtual environment and have no physical body to relax. However, a character can gain much in terms of believability by understanding how to fake conditions such as tiredness, boredom, sloth, and physical injury. With a view to maintaining a degree of simplicity on the implementation, we have opted to suppress these features in this version of the system, although they might be considered again in the future;

2. Improving characters' knowledge about the plot by teaching them about action analysis and the given circumstances, is important to allow actors to contextualize their performances and make suitable adjustments. The background story helps virtual actors to decide on a starting-point for their acting;

3. Emotion and sense memories have the potential to simplify the process of selecting coherent performances;

4. External and internal actions can be related to the character's perception of the environment (what is going on in the surroundings of the scene);

5. Being capable of establishing temporal relations between different activities would allow virtual actors to reason about their progression and structure;

6. Obtaining extra information from the dialogues could help virtual actors to infer alternative interpretations for the meaning of the sentences being spoken. This could help them contextualize their performances; and

7. In the same way as with traditional acting, creating compelling character animations (especially for autonomous characters) depends on choosing an appropriate timing for the scenes being played.

On the basis of these notes, we followed the guidelines for design-based research (see Section 2.4) by repeatedly designing several versions of an agent architecture for autonomous digital actors, until we found one that met all the requirements mentioned above. Each of these alternative ideas is addressed in the next sections that follows.

4.1 Preliminary Ideas

The literature review has shown that the computer games industry has become the leader in creating autonomous characters. In view of this, our first attempt to design an autonomous digital actor was to attempt to mimic the approaches adopted for classic non-player characters (NPC). De-Byl (2004) [Byl, 2004] suggests that there are four generic stages that every NPC should follow (see Figure 3.9 on page 58):

Perception: sensing the surrounding environment for perceivable new events and environmental changes (called *percepts*);

Inference: elicitation of alternative plausible actions (that take account of current percepts) regarding a character's goals and preferences;

Selection: to select, from among the inferred plausible actions, those that should be scheduled for execution first (called a *plan*) in accordance with established fitness criteria;

Behavior: execution of the selected plan of actions.

The drawback of these architectures is that they were designed for 'general purposes', which means they had to operate as simplified models that are applicable to any kind of situation autonomous characters might face. As a result, the characters are conceived in a way that allows them to reply convincingly, but not necessarily coherently, to all events. These NPC must be more specialized so that they can operate in more complex domains and applications such as an autonomous digital actor (which is the aim of this thesis). Thus, a direct application of NPC classic agent architecture was not a suitable solution to our problem.

4.2 Designing ADA (I)

The purpose of this is to create a system that resembles real actors in their practice and their relations, with the director, while making a film (to use as an actor/director metaphor). This metaphor is in fact a loosely-coupled simulation for agents that can work by autonomously suggesting acting performances from scripts that describe the story to be enacted (similar to way actors work while playing roles in a play or film). Moreover, the animator should be able to request specific actions (and types of behavior) from actors and, by doing so, guide their performances (in an analogous way to a director directing the cast during a filmmaking session or rehearsal of a play).

4.2.1 System Architecture

Using the four stages of the NPC architecture mentioned above as a starting-point, an acting framework can be established that assumes that two professionals will be involved in authoring autonomous actors to produce a film (see Figure 4.1):

- the `acting teacher` is a specialist responsible for training actors by writing acting rules for knowledge bases

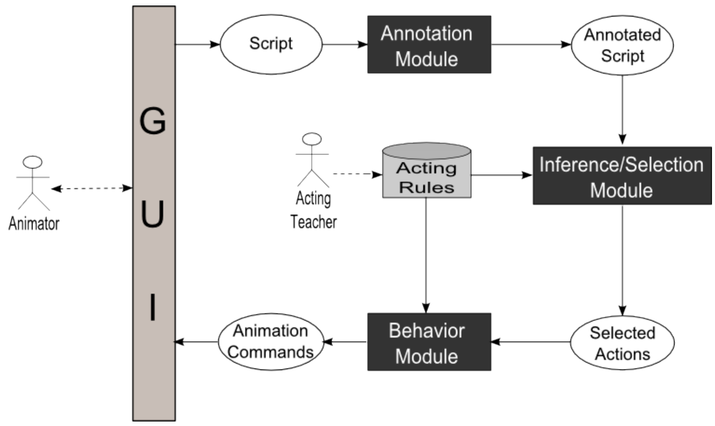


Figure 4.1: Actor/Director Metaphor Architecture

- the `animator` is the person responsible for 'directing' ADAs during the authoring process.

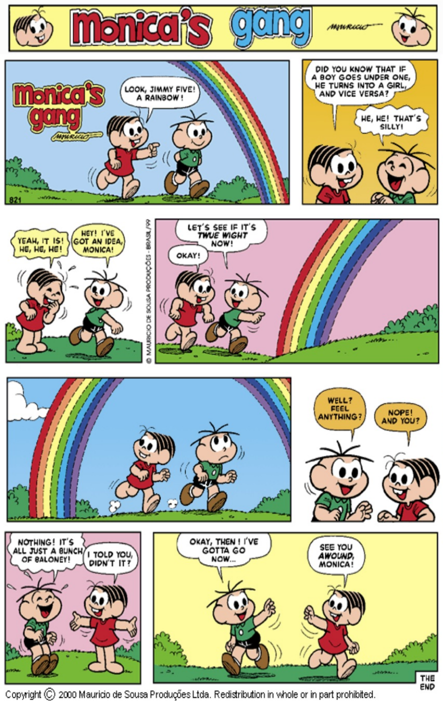
In the next sections, there is a discussion of each component in this architecture and how they should operate.

4.2.1.1 Script

As examined in Section 3.1.3, a script contains a detailed description of the scenes to be enacted. A script can be defined as a series of script units. A script unit is any performable action that an actor is supposed to enact while performing the story.

As a means of constructing the simpler scripts, we have decided to regard each actor as a "talking head", which means that the only expressive action they are able to perform involves the head (lip sync, head movements, gazing and blinking). This decision allowed us to work with one type of script unit: `speech`. Speech units are represented by the tuple `[Actor, Target, Text, Tag]`, where *actor* indicates the name of the actor speaking a *text* to another actor (the *target*); characters will use the *tag* as their semantic/pragmatic interpretations for the text (see next section for more detail).

As an example of a script, let us consider a Brazilian comic strip called Monica's Gang (Figure 4.2).



Agent	Target	Text	Tag
Monica	Jimmy Five	Look, Jimmy Five! A rainbow!	ADMIRE
Monica	Jimmy Five	Did you know that it a boy goes under one, he turns into a girl, and vice versa?	DOUBT
Jimmy Five	Monica	He, he! That's silly!	GOOD_JOKE
Monica	Jimmy Five	Yeah, it is! He, he, he!	ENCOURAGE
Jimmy Five	Monica	Hey! I've got an idea Monica!	TEASE
Jimmy Five	Monica	Let's see if it's *twue wight* now	COMMAND
Monica	Jimmy Five	Okay!	ENCOURAGE
Jimmy Five	Monica	Well? Feel anything?	DOUBT
Monica	Jimmy Five	Nope! And you?	DOUBT
Jimmy Five	Monica	Nothing! It's all just a bunch of baloney!	MOCK
Monica	Jimmy Five	I told you, didn't it?	DOUBT
Jimmy Five	Monica	Okay, then I've gotta go now...	CALM
Monica	Jimmy Five	See you *awound* Monica!	ENCOURAGE

Figure 4.2: Example of a script after Monica's Gang Comic Strip

Source: `http://www.monica.com.br/`

4.2.1.2 Annotation Module

In the light of the NPC architectures discussed earlier, the first module an ADA should process is 'perception'. Unlike NPCs, autonomous digital actors use script information as a percept to infer current context. Hence, it should be possible for an ADA, to interpret the semantics involved in each utterance of the script.

Ideally, an ADA should have the ability to interpret the script being enacted autonomously; however, even if we take account of current natural language processing technology [Jurafsky and Martin, 2000], this would be an unduly demanding task for agents at this point. Some simplifications had to be adopted instead.

The use of annotated tags with sentences is a very popular approach in the literature for agents that need to infer semantics (and pragmatics) from the text. Annotating a sentence means combining it with semantic/pragmatic interpretations (called dialogue act tags), so the character can use this information.

The inspiration for our annotation module came from the CrossTalk project [Baldes et al., 2002]. In our solution, we manually annotate sentences from the script being enacted. For instance, consider the first sentence in the Monica's Gang script: one possible annotation could be the tag "admire" as Monica is clearly admired for spotting a rainbow and then mention that to Jimmy Five. The complete annotated script for the sample script is given in Figure 4.2.

The annotated script is then passed to the inference/selection module responsible for inferring plausible actions, as discussed in the next section.

4.2.1.3 Inference/Selection Module

This module uses annotated scripts to select acting rules that the match current situation in the performance. An acting rule is written in the traditional condition-action fashion as in `if agent = MONICA and target = JIMMY FIVE and tag = ADMIRE then perform(SURPRISE_FACE, medium, fast)`, where the performable action is described by the action name, its intensity and duration.

A knowledge base enumerates all the known situations by the actor including a default alternative situation that is used for all other unforeseen events (see Table 4.1).

Table 4.1: Example of a knowledge base

Actor	Target	Tag	Action
Monica	Jimmy Five	Admire	display_surprise_face, high, slow
Jimmy Five	Monica	Excuse	display_anger_face, low, slow
Monica	Jimmy Five	Doubt	display_surprise_face, low, fast
Helen	Bob	Mock	display_happy_face, low, fast
Jimmy Five	Monica	Sulk	display_sad_face, low, fast
Any	Any	Any	display_neutral_face, high, slow

If more than one rule matches the current situation, this module randomly selects one for execution.

4.2.1.4 Behavior Module

After the action has been selected, this module translates it into animation commands for the animation engine of choice. However, since we have decided not to implement this design (for a reason that will be given in the next section), this module has not yet been studied in depth.

4.2.2 Validation of the Design

This design was validated in accordance with the four elements of the criteria described in Section 2.6. As it has not yet been implemented only the resulting agent architecture was analyzed.

1. **Artifact for Success**: this version of the system allows reactive agents to suggest a single acting performance by using sentence annotations as the dramatic context;
2. **Generalization**: the only level of script analysis is script unit annotation, thus, the agents can only make suggestions within the scope of these annotations, which handicaps them in terms of their ability to contextualize dramatic situations;
3. **Novelty**: as this architecture was designed to follow the design of classical non-player characters, it does not include new components in terms of autonomous behavior, although the application is new;
4. **Explanation of Capabilities**: given the limitations of the suggested architecture, a number of possible improvements were recommended:
   (a) Different actors must be able to suggest different types of enactment for the same dramatic contexts. To achieve this, the agents should be able to express their individual inner states at some level;
   (b) By combining the inference and selection modules, the actors can only suggest one single resulting performance; and

(c) By considering one possible solution per context at most, the approach runs the risk that no solution will be selected, and this will leave a gap in the final suggested timelines.

4.3 Designing ADA (II)

They should be able to show inner states to allow the ADAs to give expressive individual performances. Inner states are sets of values that describe special states relative to the agents living in the surrounding environment. An example of an inner state is the character's emotional state. To be able to describe each agent as a unique individual, the architecture should at least take account of basic personality traits during the decision process of searching for appropriate acting rules. We have decided to adopt the popular five traits of the OCEAN model (described in Section 3.5.2.2).

The next sections describe the improvements made in the system architecture and conducts an analysis to determine if it would meet all the requirements raised by this research.

4.3.1 System Architecture

The architecture has been updated with the addition of an `emotion elicition` module (see Figure 4.3), and the inference and selection modules have been separated. It also included personality traits as attributes for the rules in the actor's knowledge bases.

The script representation, annotation module and resulting annotated script have not been modified since the last version. We adopted the OCEAN model for our characters' personality model (for more details refer to Section 3.5.2.2).

After taking the personality test for the characters, the following results were obtained: Monica: (O = 65, C = 58, E = 91, A = 44, N = 94) and Jimmy Five: (O = 53, C = 52, E = 89, A = 17, N = 27). These personality values were then used to match the rules both for emotional states and actions, as discussed in the next sections.

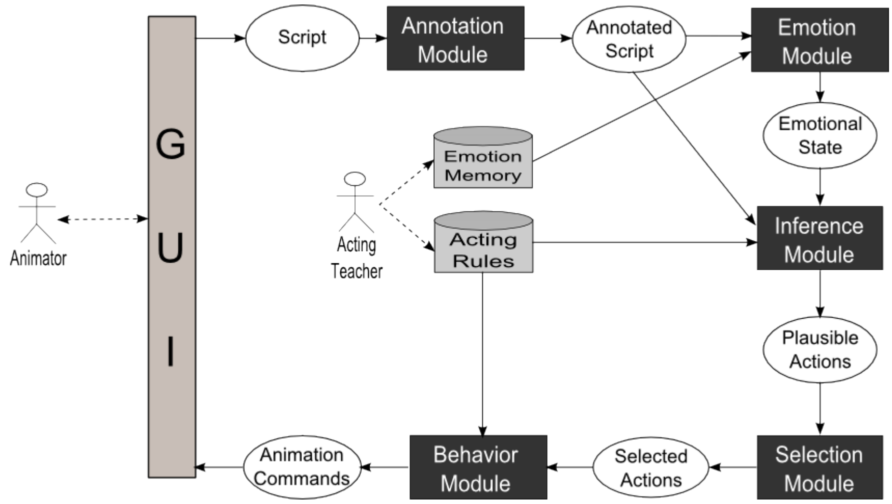


Figure 4.3: Improved Actor/Director Metaphor Architecture

4.3.1.1 Emotion Module

Each ADA maintains a series of special states called *emotions* that are used to more accurately infer proper responses to current situations being performed. According to the findings of Paul Ekman [Ekman, 2003, Ekman, 1999], everybody is able to display (and recognize in others) at least seven 'basic emotions': anger, confusion, disgust, fear, happiness, sadness and surprise. Our characters can 'feel' them all.

The emotion elicitation module is, in fact, a function that maps annotated sentences and character's personality into proper emotional responses.

$$emotion\_elicitation(annotation, personality) \Rightarrow emotion_{name}$$

4.3.2 Validation of the Design

This second version of a proposed agent architecture for designing ADA was validated by applying the same four criteria as before (section 4.2.2) as follows:

Table 4.2: Example of Rules for Emotion Elicitation Inference

Curr Emotion	Tag	Personality	New Emotion
anger	Criticize	O=65, C=58, E=91, A=44, N=94	anger
happiness	Excuse	O=65, C=58, E=91, A=44, N=94	sadness
anger	Doubt	O=53, C=52, E=89, A=17, N=27	confusion
anger	Mock	O=65, C=58, E=91, A=44, N=94	anger
happiness	Sulk	O=53, C=52, E=89, A=17, N=27	sadness

1. **Artifact Success**: this version considered using inner states in the suggestions for the two levels of the enactment performance: emotion and action;

2. **Generalization**: the addition of a personality description allowed consideration to be given to some level of individuality; however, its possible for many different roles to share the same personality description. Thus, a more detailed characterization is necessary;

3. **Novelty**: the emotion memory and acting rules databases represent are adapted to traditional NPC solutions;

4. **Explanation of Capabilities**: the following issues were identified in this suggested version:

   (a) this system does not allow the description of individual roles to be specified; thus, the characterization needs to be improved so that it does not just represent personality traits;

   (b) at this point, the system selects an appropriate dramatic context based on perfect match criteria, which means that the representation of the current dramatic context must match perfectly with one of the selected rules in the database, otherwise, none will be selected and this will leave a gap in the resulting timeline. More flexible approaches, in terms of matching, need to be considered.

Chapter 5

Designing ADA (III): Experimentations

This chapter aims at describing our adopted solution for an ADA agent architecture. It considers three major changes to the previous versions:

1. To incorporate imperfect match criteria that can be applied to select rules in the database and allow actors to make suggestions (even for unforeseen situations), by considering a similarity function;

2. To include an analysis of the script as part of the context of the rules, and allow ADAs to carry out an in-depth analysis to define the current context while suggesting enacment performances; and

3. To be able to assess which rules are more suitable for the animator's preferences in terms of authoring.

The next sections discuss each of these problems and the suggested solutions we have adopted.

5.1 Imperfect Matching

Traditionally, classifying an attribute means determining which class (among a list of possible classes) the given attribute belongs to and can be compared with. Like determining the model of a car or the color of a shirt. This approach is simple to implement because it is only a matter of establishing a criterion that relates attributes with classes. For instance, when determining if an object is a car, one could check if the given object is shaped like a car, if it has four wheels, two doors (at least), an engine, etc. Before the object can be classified as really belonging to the 'car' class, it must respect all the criteria used during the classification test. This is called a 'perfect' match criterion. If just one criterion has not been satisfied, the classification fails.

The problem with this approach is that, it is usually quite difficult (or, in some cases, even impossible) to tell which class a given attribute fits best, as it might belong to more than one class at the same time. This happens due to the fact that the classification criteria are not precise enough ("what is the color of a soccer ball?") or because subjective criteria are being used for classification ("who is the prettiest?").

An attempt to solve the "perfect match" problem, adds uncertainty to the selection criteria. This can be carried out by including a degree of confidence in each possible class within the domain. In this way, instead of determining which class fits a given attribute best, the criteria will determine how likely the attribute will be suited to each class, thus resulting in a *classification vector* of the form:

$$\vec{Attribute} = ([class_1, value_1], [class_2, value_2], \ldots, [class_N, value_N]) \tag{5.1}$$

where each $value_i$ indicates the degree of probability that the attribute will belong to a particular $class_i$. This representation allows us to indicate uncertainty when classifying attributes if we are not sure which class the attribute really belongs to. For instance, let us say we want to classify a person's gender (male or female), but we have no clue about what it is. In this case, both classes have an equal likelihood of being true and we can represent it by the vector $gender = [0.5, 0.5]$, where the first position indicates the probability for the subject being a *male* and the second a *female*.

Once we have represented each attribute as a classification vector, the next stage is to establish a similarity function for calculating how similar two vectors are (this can be used to reflect current contexts with rules in the database). We chose to use Euclidian distance for this, as shown in equation 5.2 below.

$$dist(\vec{A}, \vec{B}) = \vec{A} - \vec{B} = \sum_{1}^{N} \vec{A}_i - \vec{B}_i \tag{5.2}$$

Two vectors will be considered as matching if the Euclidian distance between them is lower than a given threshold ($dist(\vec{A}, \vec{B}) < threshold$).

5.2 System Architecture

Two new components have been added (Figure 5.1) to the architecture (compared with previous versions): the `script analysis` record and the `behavior correction` module. In addition, as the knowledge representation for the rules has changed to consider the imperfect match criteria, the implementation for all the other modules have changed as well, as described below.

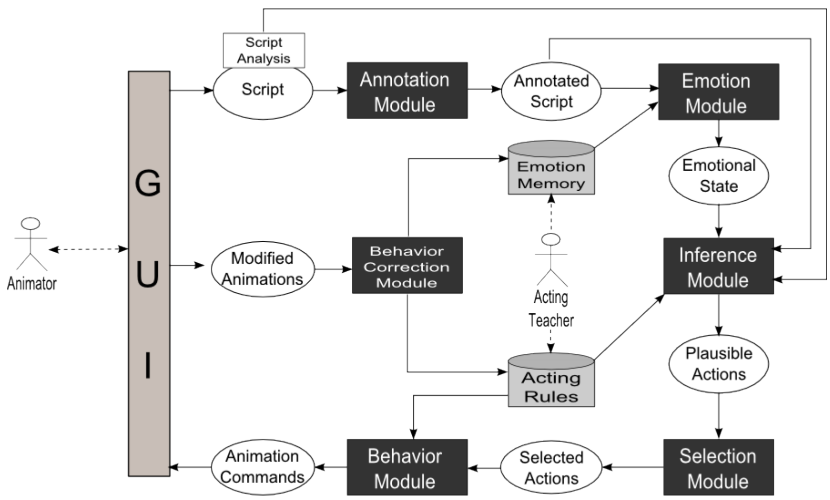


Figure 5.1: Expressive Trainable Actor/Director Metaphor Architecture

A script is still represented as a series of script units, each describing the actions to be played by the actors in a scene. Furthermore, the idea of adding a *script analysis record* has been put forward to describe the main aspects of the plot being enacted. At this point, this record must be defined by the script writer while writing the script that is to be enacted.

Script Analysis Record: the script analysis record contains a range of information to help actors obtain a deeper understanding of the script being interpreted. This information is used for reflecting on the performances themselves. This first trial only involves a subset of all the elements studied, and that are usually found in a common script analysis document (as shown in Section 3.1.3.1) with a limited number of possible values for each element.

The features that are currently supported by the script analysis record and respective possible attributes are summarized in Table 5.1.

Table 5.1: Script Analysis Record

Element	***Attribute***	***Possible values***						
Action Analysis	Theme	Quarreling	Interrogation	Conversation				
Given Circumstances	Time	Day	Night	Past	Present	Future		
	Place	Home	Work	Other				
Character	Name							
	Age	Child	Teenager	Adult	Elder			
	Gender	Man	Woman					
	Personality	Openness	Conscientiouness	Extraversion	Agreeableness	Neuroticism		
	Emotional State	Anger	Confusion	Disgust	Fear	Happiness	Sadness	Surprise

An example of a script description is given in Appendix A.

Script Unit: as described earlier in Section 4.2.1.1, script units are represented by a list of tuples in the form of [actor, target, text, tags].

The new annotation module now includes two classification vectors instead of a single label for annotating a unit: the annotation and the role.

`Annotation` represents the manual interpretation of each unit being enacted. It has been adopted the same list of tags used on the CrossTalk project, but representing them as a classification vector, where each position in the vector stress the interpretation of one given tag of the dramatic context being represented. Each value in this vector represents the degree of certainty (between 0 and 1) the actor has regarding a given attribute. In Figure 5.2, it is illustrated an example where a script unit has been annotated as having three possible interpretations: admire, doubt or regret (indicated by the '1's in the annotation vector). The `role` vector describes how the actor is performing the script unit, whether as an actuator, effector or bystander in a similar way to the annotation. In the example below, the actor plays the role of the 'actuator'.

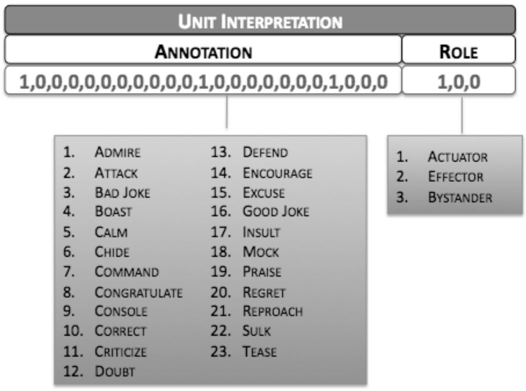


Figure 5.2: Script Unit Annotation Vectors

Not many changes to the previous version of the system were proposed for the emotional response module. The emotion module matches dramatic situations with emotional responses. A dramatic situation is a condition-action fashion rule that comprises the unit interpretation, the script analysis record and the description of the character.

The conditional statement is subdivided into:

Unit Interpretation: script unit annotation + role (see Figure 5.2)

Script Analysis: action analysis + given circumstances (see Figure 5.3)

Character: description of the character: personality, current emotional state, gender and age (see Figure 5.3)

The resulting action can represent two possible outcomes: emotional reactions (inner states) or an acting performance. At this point, agents are limited to 'feeling' or 'expressing' seven basic emotion, as illustrated in Figure 5.4.

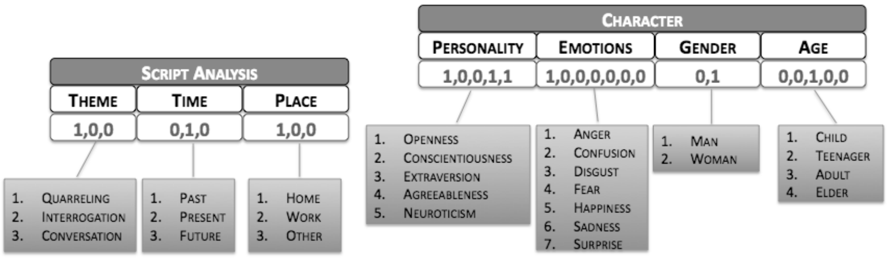


Figure 5.3: Script Analysis and Description of Characters

Figure 5.4: Description of Emotional State

I decided to adopt a reinforcement learning approach to allow agents to match their suggestions with the animator's preferences. Each rule keeps track of two auxiliary variables of the applications of the rule: the `support level` represents the number of times a rule has been selected as plausible (possible usage) and the `confidence level` is the number of times the animator provided positive feedback after the rule has been applied successfully. The degree of confidence of the rule is obtained by the ratio $degree\_of\_confidence = \frac{confidence\ level}{support\ level}$.

Later, the selection module is responsible for determining (from the list of plausible rules), those that will be scheduled for execution in the behavior module. The selection criteria for the rules can be set out as follows:

1. The rules must be chosen to allow a shorter distance to the current situation vector;

2. In case more than one rule is selected, the rule with the greatest degree_of_confidence should be chosen;

3. If still more than one rule is selected, one of the remaining rules should be selected at random.

One key element when creating an ADA, is to define its repertoire of animations; these describe all the resources the 'actor' has at its disposal to display specific intentions as animations. Although, in theory, any animation technique could be used to produce the ADA's repertoire, in this study, we have confined ourselves to two methods: poses and animation clips.

Poses involve specific setups of a character's skeletal structure to stress facial and physical expressions like, for instance, displaying an angry face. Animation clips are timeline snippets composed of a series of poses over a period of time, that allow the definition of more expressive kinds of behavior like *blinking* or *nodding*.

It should be noted that this repertoire of animations does not necessarily mean the same as the action clause of an acting rule, discussed earlier. Actions concern the 'intentions' of the actor in terms of acting performance, while animations are the realization of these intentions through facial and bodily displays. For instance, an intention to express happiness may be displayed by means of a smile or a laugh.

The behavior module is responsible for transposing the list of suggested actions into animations (either as poses or clips) by producing a timeline. Two steps need to be taken to undertake this:

1. To determine which animation setups are most suitable for an action (intention) that has to be displayed with all available ADA poses and clips. A dictionary that maps one into the other is the simplest way to accomplish this.
2. To suggest timing parameters for each animation so that a timeline can be scheduled for an animation engine. The simplest approach is to synchronize animations with each script unit length, although this produces unrealistic movements. In future other approaches should be studied to improve the timing actions, such as per word or even per visemes [Osipa, 2003].

This module produces a timeline that can, later, be rendered and visualized as an animation. Taking this rendition into account, the animator can consider correcting the ADAs. This is carried out by using the behavior correction module: the system keeps track of two reinforcement learning variables (the confidence factor and support factor) for each rule within the database. Whenever a rule is scheduled for animation, its support factor value is updated. After the animator obtains the resulting animated performance that is required, he/she can still send back correction messages like flags, for each action in the performance.

A *like* message causes the actor to increment the confidence factor for the given rule by signaling it as preferable, otherwise the rules will simply not be updated.

5.3 Experimenting with ADA

A training tool for authoring animations with ADA (see Figure 5.5) has been implemented to produce a proof of concept. This tool has been implemented in Python (`www.python.org/`), using wxWidgets (`www.wxpython.org/`) for the interface. The interface is subdivided into three parts: ADA, Script and Performance.

5.3.1 ADA

This represents the training phase of any new ADA. The process for creating a new ADA follows these stages:

1. creating a new entry in the list of ADA. An entry is defined by the tuple [name ID, Emotion Memory, Acting Rules, Repertoire], where the `name ID` is the only identifier name of the actor, `Emotion Memory` is the knowledge base name that contains all the previously-trained emotional response rules for the given actor; similarly, the `Acting Rules` is the knowledge base containing the performances acting rules. Finally, the `repertoire` is the list of actions (in fact, a set of predefined animations) modeled by this actor;

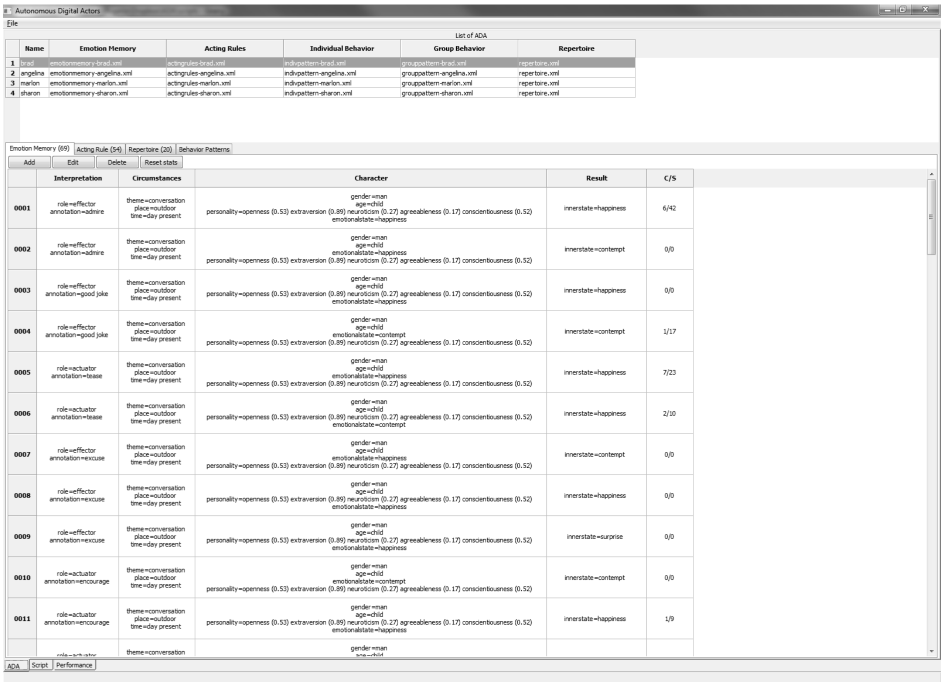


Figure 5.5: ADA Tool Interface

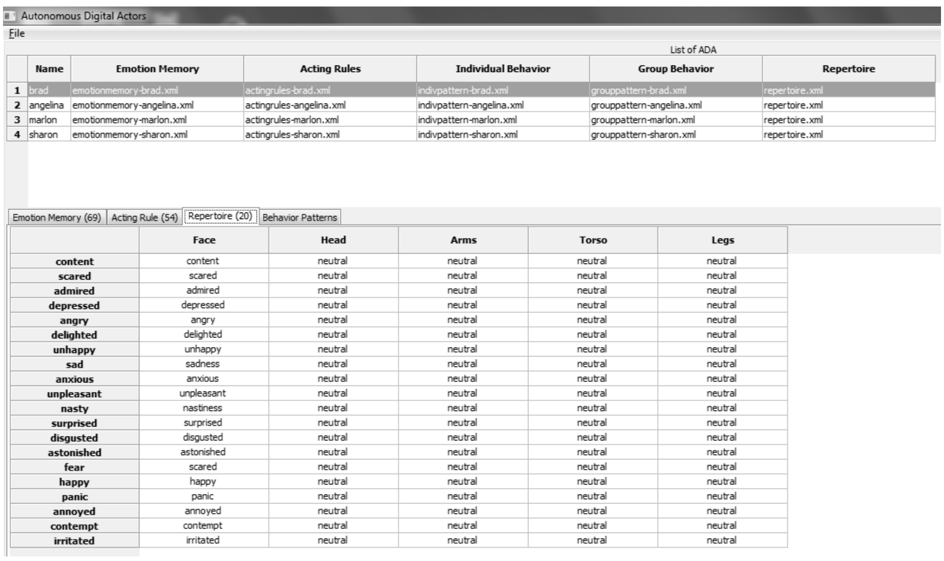
2. training in making appropriate emotional inferences after dramatic contextualized evaluations. An emotional state is an inner state that denotes how the actor is 'feeling' while enacting a particular unit within the script;

3. the third phase describes the acting rules, that govern the way an actor should perform a particular dramatic situation during the enactment of a script unit. A dramatic situation includes all the inferred context (script unit interpretation, script analysis record and character description) plus the character's inner state which has been inferred in the previous emotional inference phase. Section 5.4 describes how to lay down rules for both emotion memory and acting rules databases;

4. the fourth phase specifies the actor's repertoire. The selection of the repertoire depends on an actor's acting skills and how these have developed during the preparation of a character part. In summary, it represents the sum of all the predefined animations designed for it. The repertoire editor (see Figure 5.6) describes this list of animations that are later used when an ADA is trained in

the acting rules. This animation list is defined during the modeling phase of the character, using the animation engine.

Although our actors are regarded as “talking heads”, since the only supported animation commands are those emanating from the face (emotion expression), the repertoire system was designed to work for the whole body. Animation commands work on a multi-layer basis, where each layer issues commands for a specific part of the ADA’s body. Thus, an action is described by the tuple [action ID, face, head, arms, torso, legs], where the “`action ID`” is the label that identifies a given action, and “`face`” indicates the facial expression that must be made. Supported values are `anger, contempt, doubt, fear, happiness, sadness, and surprise`. “`Head`” indicates the commands of head movements. Supported values are e.g. `neutral, lean left, lean right, look down, look up, look left, look right, look at something`. Supported values for the “`arms`” are e.g. `neutral, open, down, up`. For the “`torso`” the animator can choose from the following states – `neutral, lean forward, lean backward, lean to left, lean to right` and for the “`legs`” the only supported value is `neutral`.



	Name	Emotion Memory	Acting Rules	Individual Behavior	Group Behavior	Repertoire
1	brad	emotionmemory-brad.xml	actingrules-brad.xml	indivpattern-brad.xml	grouppattern-brad.xml	repertoire.xml
2	angelina	emotionmemory-angelina.xml	actingrules-angelina.xml	indivpattern-angelina.xml	grouppattern-angelina.xml	repertoire.xml
3	marlon	emotionmemory-marlon.xml	actingrules-marlon.xml	indivpattern-marlon.xml	grouppattern-marlon.xml	repertoire.xml
4	sharon	emotionmemory-sharon.xml	actingrules-sharon.xml	indivpattern-sharon.xml	grouppattern-sharon.xml	repertoire.xml

	Face	Head	Arms	Torso	Legs
content	content	neutral	neutral	neutral	neutral
scared	scared	neutral	neutral	neutral	neutral
admired	admired	neutral	neutral	neutral	neutral
depressed	depressed	neutral	neutral	neutral	neutral
angry	angry	neutral	neutral	neutral	neutral
delighted	delighted	neutral	neutral	neutral	neutral
unhappy	unhappy	neutral	neutral	neutral	neutral
sad	sadness	neutral	neutral	neutral	neutral
anxious	anxious	neutral	neutral	neutral	neutral
unpleasant	unpleasant	neutral	neutral	neutral	neutral
nasty	nastiness	neutral	neutral	neutral	neutral
surprised	surprised	neutral	neutral	neutral	neutral
disgusted	disgusted	neutral	neutral	neutral	neutral
astonished	astonished	neutral	neutral	neutral	neutral
fear	scared	neutral	neutral	neutral	neutral
happy	happy	neutral	neutral	neutral	neutral
panic	panic	neutral	neutral	neutral	neutral
annoyed	annoyed	neutral	neutral	neutral	neutral
contempt	contempt	neutral	neutral	neutral	neutral
irritated	irritated	neutral	neutral	neutral	neutral

Figure 5.6: ADA Tool - Repertoire Editor

5.3.2 Script

With the aid of the script editor (see Figure 5.7), the animator can perform the following tasks: (1) creating a new script or load an existing one; (2) maintaining the list of characters; (3) editing the script analysis record (given circumstances) and (4) maintaining the list of script units.

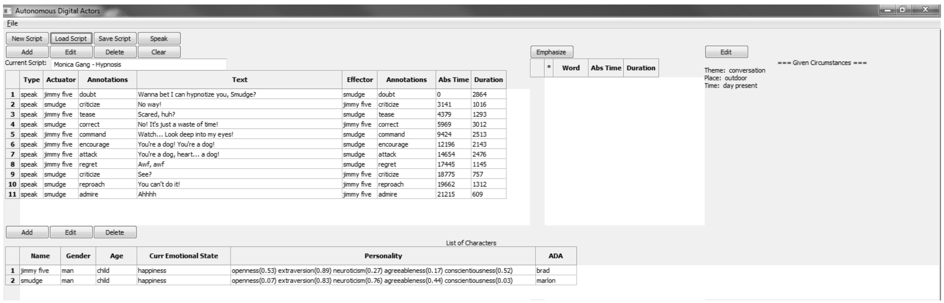


Figure 5.7: ADA Tool - Script Editor

1. to create a new script, the first step is to complete the script title field;

2. following this, in the given circumstances, the animator describes the script analysis record of the plot in terms of theme, place and time, as illustrated in Figure 5.8;

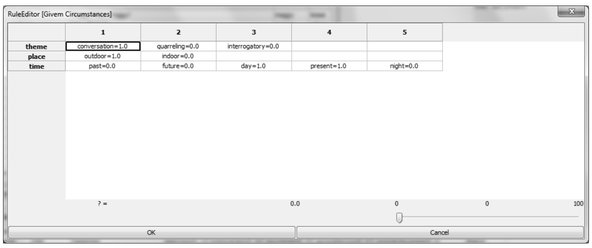


Figure 5.8: Script Editor - Given Circumstances

3. in the following stage, the animator creates/edits the cast by selecting a list of characters that need to be performed by the previously established ADA and described in Section 5.3.1. A character description comprises three main components: its name, list of attributes and the name of the actor that will perform

it in the script. The currently supported list of attributes is: `gender` meaning (male or female). The `age` includes the following values: child, teenager, adult or elderly person. `Emotional state` denotes the character's inner states before the plot begins and `personality` indicates the personality traits in accordance with the Big Five personality model. Figure 5.9 shows the "character editor" in its three stages.

Figure 5.9: Script Editor - Characters

5.3.3 Performance

Once the group of ADA have been created and trained, and a script has been written, the animator can use these to produce an animation. The procedures involved for this are described next:

1. establishing a threshold value for the rule matching criteria. By lowering this value, makes the matching more precise (a value of zero means perfect matching);
2. producing a timeline for the script, where actions are suggested for each script unit. These suggestions can be made in two ways: using the previously trained

specialized knowledge (button '`Suggest`' in Figure 5.10) or simply using random suggestions (the button labelled '`Random Suggest`'). The next section (5.3.3.1) explains how these processes work;

3. once a timeline has been suggested (either via ADA or randomly), the animator exports it, using the button '`Animate`', and it is loaded in Autodesk™ Maya™. More details are provided in Section 5.3.4. By exporting a selected timeline, the system updates each selected rule's confidence level attribute;

4. after the animation has been produced, the animator can accept the suggestions made by the system, and this causes the selected support values of the rules to be updated. These updates enable the system to increase each rule's degree of confidence, thus improving the chances of being selected on future usage.

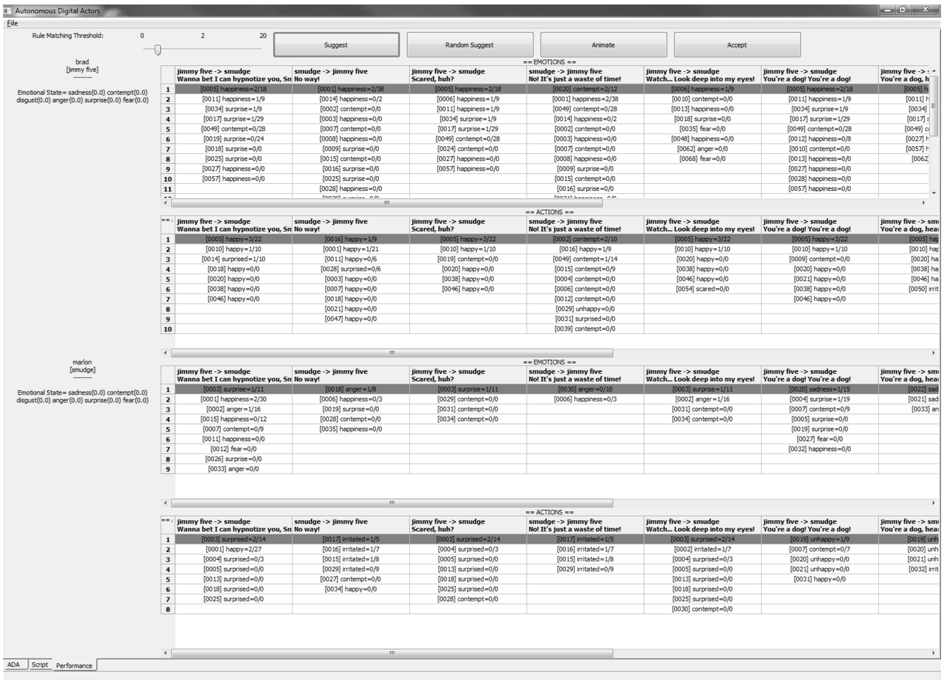


Figure 5.10: ADA Tool - Performance Suggestions

5.3.3.1 Performance Suggestions

In this context, making performance suggestions for acting means determining which rule (in the actor's knowledge bases) are most closely applicable to the current dramatic situation. A dramatic situation is composed of three elements: the script interpretation (the script unit annotation and the role played by the character), the script analysis (theme, time and place) and the character description (age, gender, personality and current emotional state). The whole dramatic situation is represented as a classification vector (see Section 3.5.3) where (for purposes of simplification) each attribute is represented as a binary vector, apart from the personality vector.

$$
Context = \left\{
\begin{array}{rcl}
\text{Unit Interpretation} & = & \left\{ \begin{array}{rcl} \text{annotation} & = & 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ \text{role} & = & 0\ 1\ 0 \end{array} \right. \\
\text{Circumstances} & = & \left\{ \begin{array}{rcl} \text{theme} & = & 1\ 0\ 0 \\ \text{time} & = & 1\ 0\ 0\ 1\ 0 \\ \text{place} & = & 1\ 0 \end{array} \right. \\
\text{Character} & = & \left\{ \begin{array}{rcl} \text{gender} & = & 1\ 0 \\ \text{age} & = & 0\ 0\ 1\ 0 \\ \text{personality} & = & 25\ 88\ 16\ 75\ 4 \\ \text{emotional state} & = & 1\ 0\ 0\ 0\ 0\ 0\ 0 \end{array} \right.
\end{array}
\right.
$$

The process for suggesting acting performances is described in Algorithm 5.1.

Algorithm 5.1: Performance Suggestion Pseudo-Code

```
1  for unit in ListOfUnits
2      currentSituation[ annotation ] = unit[ annotation ]
3
4      if unit[ actor ] == currentADA
5          currentSituation[ role ] =  actuator
6      else if unit[ target ] == currentADA
7          currentSituation[ role ] = effector
8      else
9          currentSituation[ role ] = bystander
10
11     currentSituation[ character ] = currentADA[ character ]
12
13     currentSituation[ circumstances ] = currentScript[ circumstances ]
14
15     suggestedNewEmotionalState = selectRule( EMOTIONALRULES, currentSituation )
16     currentSituation[ character ][ emotionalstate ] = suggestedNewEmotionalState
17
18     newActionRule = selectRule( ACTINGRULES, currentSituation )
```

First, the current situation is established, by combining the script unit annotation, the role being performed, the character's description, the given circumstances and current emotional state. After this, a new emotional state can be selected from the emotional rules database (see Algorithm 5.1, line 15). This new emotional state is used to determine suggestions for action performances (line 18). The rules selection process is described in Algorithm 5.2 below.

Algorithm 5.2: Rules Selection Process

```
proc selectRule( databasename, context )
    for rule in database[ databasename ]
        distance = EuclidianDistance( rule, context )

        if distance <= threshold
            suggestionsList.append( rule )

    SortByConfidence( suggestionsList )
```

5.3.4 Animation Engine

We decided to use Autodesk™Maya™ as our animation engine. Maya is a commercial authoring tool that is largely used by both the animation and games industries. It implements animations such as non-linear timelines of ‘character sets’ based on ‘clips’. A character set is a group of animatable channels like a head position in each axis. A setup of values in any given instant of time is called a keyframe or pose. A clip is a labeled series of poses describing a performative action (e.g. displaying anger or sorrow).

In our solution, only one character set has been proposed (the head). In view of this, eight poses were modeled for each ADA for the seven basic emotions (anger, contempt, disgust, fear, happiness, sadness and surprise), plus the neutral face. In addition, eight clips were modeled as the facial displays for the seven basic emotions plus a “blink of the eyes” clip. This group of clips represents the ADA’s repertoire. Four ADA have been modeled: Marlon, Angelina, Brad and Sharon and each is intended to be used to represent one of the four main characters in Monica’s Gang stories.

Once, the modeling is completed, animating a character means instantiating (over a period of time) a series of clips that determine the initial and duration frames for each. It is also possible to create tracks that are parallel timelines for the same character set; this allows overlapping actions that will later be combined during the rendering process. Figure 5.11 illustrates an example of a non-linear timeline produced by the Trax editor in Maya.

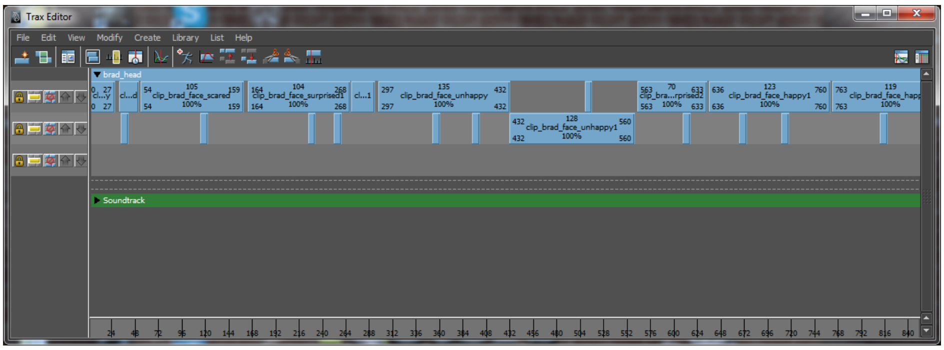


Figure 5.11: Non-Linear Animation Constructed in Trax Editor

A Python plugin for Maya has been developed and incorporated into its interface to translate all suggested acting performances into animations, as demonstrated in Figure 5.12. This plugin is responsible for loading the suggested timeline (previously exported at the performance suggestion phase), and allowing its rendering and visualization.

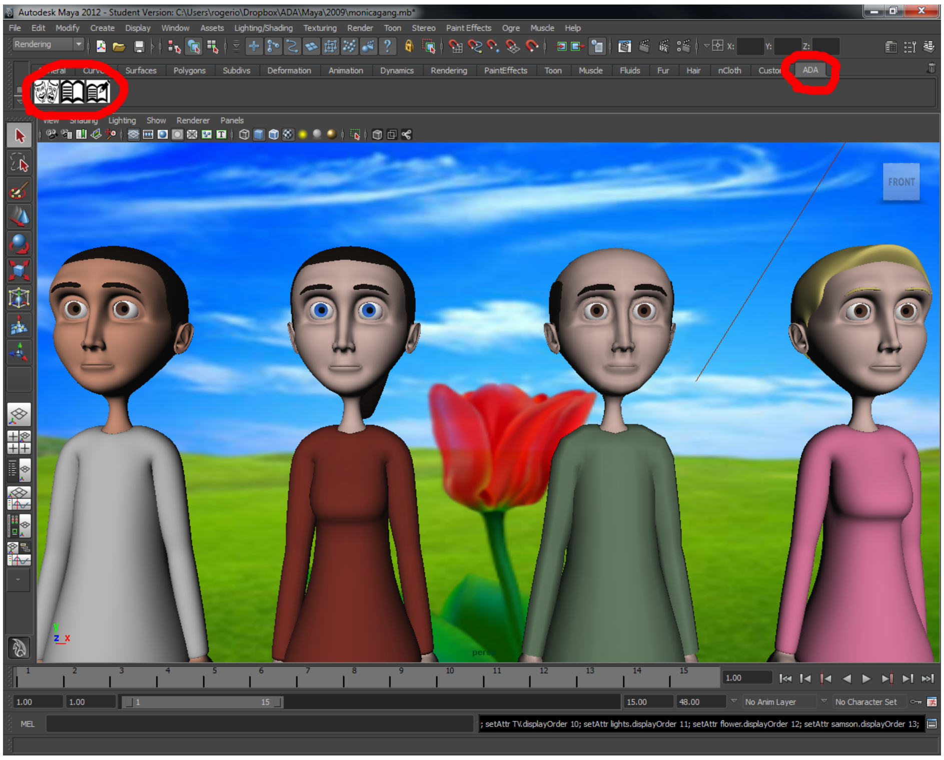


Figure 5.12: Plugin for Autodesk Maya

5.4 Training ADA

At this point, all the actors had been implemented as reactive agents. The procedures for training an actor consisted of manually including rules for both the emotion memory and acting rules databases. The methodology for creating each actor's knowledge bases was as follows:

First, a series of the popular Brazilian comic book stories Monica's Gang was selected (9 stories in total as shown in Appendix B). The choice was made on the basis of my familiarity with the characters, since that made it simpler to understand the characters' behavior, and the production of the acting rules. There are four main characters (each one with its distinct personality) which made it possible to experiment with different kinds of behavior in different contexts.

Second, we use these stories to infer emotional reactions and on the basis of this, write emotional. This process always involves the three levels of the description of context (script analysis, unit interpretation and character), and applies them to the new emotional state the character has to experience if in this kind of situation. Since each context is represented as a classification vector, the imperfect matching criteria can be employed to select the most similar situation for the performance. Figure 5.13 illustrates the construction of one of these rules.

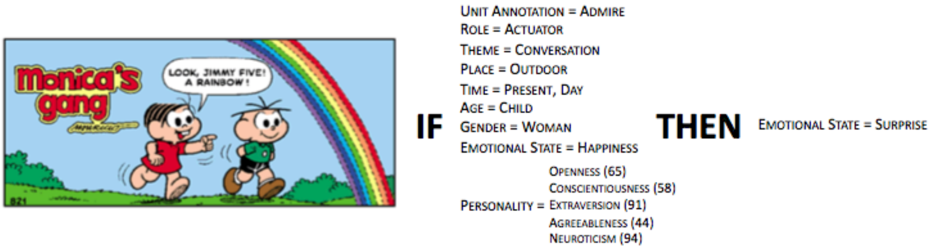


Figure 5.13: Example of an emotional rule produced out of a comic story book

In our test case scenario, these rules were implemented as in a finite state machine to make it as simple as possible to carry out a rapid prototyping. While selecting a new states that is applicable to a current state and dramatic situation, the main feature of finite state machines is determinism. Appendix C shows the resulting FSM for each of the characters, and describes when they are performing the role of both the actuator and the effector.

Once all the emotional responses rules have been written, the next step is to determine their acting performance rules. In this first version of the system, there are only seven actions available: irritated, contemptuous, disgusted, scared, happy, unhappy and surprised facial displays. Our approach simply duplicates the emotional rule context and links it to one of these actions (an example is given in Figure 5.14).

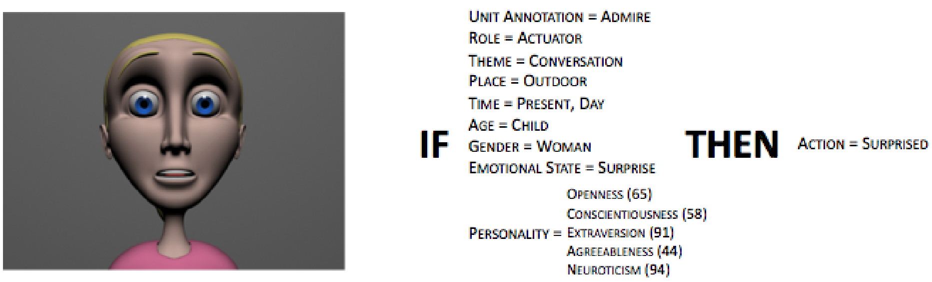


Figure 5.14: Example of an acting rule

Table 5.2: Size of the resulting training databases

ADA	Emotion Memory	Acting Rules
Angelina	98	77
Brad	69	54
Marlon	37	35
Sharon	36	35
Total	240	201

After 9 selected training stories have been examined, the training phase has laid down, 240 emotional rules and 201 acting performance rules[1] as summarized in Table 5.2.

5.5 Validation of the Design

On the basis of the validation process outlined at Section 4.3.2, we concluded that some improvements have been achieved with this third version of the proposed system. Although a few problems remain unanswered, these improvements can be set out as follows:

1. **Artifact Success**: the resulting implemented prototype allows the following:
    (a) the creation of new reactive agents;
    (b) new agents to be manually trained in terms of emotion memory and acting rules;
    (c) the creation of new scripts that contain the plot, the script analysis and the whole description of characters;
    (d) a suggestion for an enactment timeline that can later be edited/updated if necessary;

[1] The smaller amount of acting rules is due to the association of the same rule to different emotional contexts

   (e) the resulting timeline to be exported to a plugin in Maya that allows its rendering as the final animation.

2. **Generalization**: the detailed ontology used for specifying behavioral rules allow them to be used in a great variety of domains concerned with the representation of stories. Its main drawback is that is is based on the assumption that all ADA are implemented as "talking heads", which means that only variations of conversations, quarrels or interrogation can be explored with the tool;

3. **Novelty**: the emotion memory and acting rules databases, and now also the script analysis record represent a novel approach with regard to the state of the art for traditional believable characters;

4. **The Explanation Capability**: the prototype implemented after this third suggested agent architecture allowed the creation of enactment performances for ADA and its subsequent rendering in the animation engine of choice. This alone, is evidence that the original goals of the thesis have been achieved successfully. However, after using the tool in a few experiments, we noticed some other unforeseen requirements (with regard to acting performances) could improve this architecture even more. These can be listed as follows:

   (a) if only reactive agents are taken into account, it is more difficult for virtual actors to explore the dramatic context it is involved in depth, which means they will only carry out a "shallow deliberation". "Shallow deliberations" might lead to repetitive suggestions;

   (b) The current proposed architecture did not take into consideration other aspects of the art of acting, especially those regarding the actor's body and voice or those requiring deeper agency skills like: autonomous script interpretation and more sophisticated mental attitudes.

Chapter 6

Designing ADA (IV): Resulting Architecture

On the basis of the analysis carried out in the previous version of the agent architecture (outlined in Section 5.5), significant improvements have been proposed to the solution in an attempt to address most of the problematic issues. These changes are as follows:

- the perception module of the architecture was subdivided into two submodules: environmental sensing and script unit interpretation;
- the deliberation module was redesigned as a deliberative affective agent (EBDI architecture) to endow ADA with the ability to engage in more in-depth reasoning about the dramatic context in the area of emotions, beliefs, desires and intentions. A novel approach was also put forward for the personality model that regards basic desires as the driving-forces behind the decisions of an individual when taking action in a given context;
- the acting module represents the way an actor gives an expressive acting performance to convey the mental attitudes, which have been previously determined by the deliberation module. It comprises three steps: progression analysis, voice intonation and timing; and

- finally, the rehearsal module is an external module responsible for combining several different individual acting suggestions into one final renderable timeline suggestion, that is then passed on to the animation engine for rendition.

It is my understanding that controlling autonomous digital actors can now be seen as a multi-layered architecture (as presented in Figure 6.1), where each layer represents a different level of abstraction in terms of authoring agent's behaviors. I am proposing in this thesis a four layer architecture as presented follow:

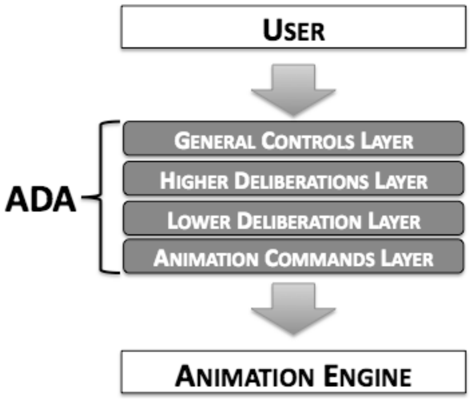


Figure 6.1: Multi-Layers View of the Proposed Agent Architecture

Animation Commands Layer: the simplest control level is direct animation commands. At this level, and according to a given animation engine, a series of animation commands like blinking, moving lips, waving, nodding, etc. are selected. It represents the means how an ADA deploys a suggested enactment as an animation. A successful implementation for this aspect should free the animator from having to specify the animation commands by himself, allowing him to work on higher levels of control. Nevertheless, it should be possible to him to make changes at this level if needed.

Lower Deliberation Layer: the agent's deliberation phase have been divided into two components: the psychological aspects and the acting performance aspects. This lower deliberation layer aims at taking a story specification and inferring proper inner states (emotions, beliefs, and desires) according to selected psychological models of deliberation. Next section presents these selected models in more details.

Higher Deliberation Layer: in this layer the agent manages intentionality, which means controlling aspects like how to convey believable performances from the inner states inferred earlier.

General Controls Layer: this is the highest (last) layer of the system, where the animator controls general aspects of virtual actors' behaviors. For instance, how actors should assemble the resulting timeline after processing all previous layers to match animator's preferences. In [Iurgel et al., 2010], we suggest a series of aspects for authoring virtual actors, including what we called a 'multi-level control' similar to this multi-layer architecture. Differently from the ideas presented in the article, in this architecture I am extending that notion a little further by taking real actors' practice and Psychology-inspired models into consideration.

These changes allowed us to propose an architecture that, I believe, met all the previously suggested requirements for the development of an autonomous digital actor. Even though the proposal of the agent architecture would, in itself, be enough to provide a validation of the research hypothesis, since it is in accordance with the design-based research methodology (see Section 2), I have decided to explore how the implementation of its modules can allow a deeper understanding of its usage. The next sections examine each of these elements in detail.

6.1 System Architecture

A general overview of this version of the system is outlined in Figure 6.2, and includes the following stages:

1. First of all, there is a script that describes the scene to be played, the set, the characters (including their personality traits) and the plot which is written by a script writer (possibly the animator itself);

2. After this, the script is submitted to a series of ADA (known as the 'cast') that must interpret it, and then make decisions about individual acting performances and suggestions for each script unit, by relying on their previously trained acting knowledge;

3. The third step is the rehearsal. This is when all individual acting performances are combined to produce a suggested timeline for the plot;

4. Finally, the fourth step is to translate the resulting timeline into animation commands for an animation engine of choice, and thus, an animation film is produced;

5. Eventually, the animator can send criticisms back to the cast, and point out the parts of the timeline that are not in line with his/her expectations. Agents can learn from this feedback, and make their suggestions more precise in the future (the current version only envisages using a reinforcement learning approach, and thus, criticism only takes on a like/dislike form).

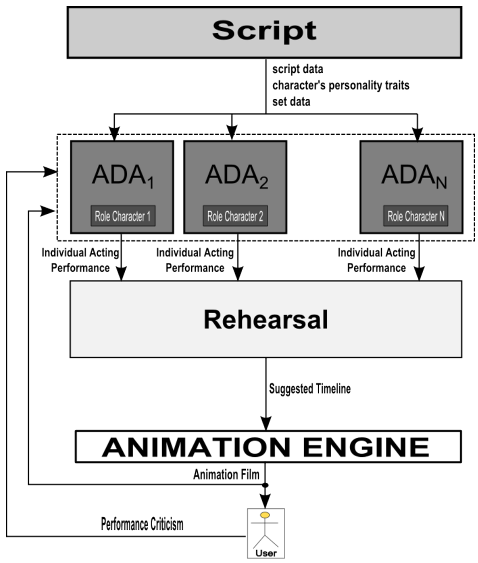


Figure 6.2: General Overview of the System

With regard to the construction of the agents, a detailed architecture has been proposed (see Figure 6.3). In the next sections each of its components will be explained in detail.

6.1.1 Script

The script is a structure that describes the following elements: the plot, the script analysis and the set.

Plot: This contains all the descriptions of the actions taken by the cast while enacting a scene. It is subdivided into *script units*. A script unit represents any elementary enactment a character is asked to perform, like speaking, laughing, running, falling over, throwing something, and so on. They are represented by a tuple of the form (Type, Actor, Target, Text) where: *type* indicates the nature of the script unit; *actor* is the name of the ADA performing the unit; *target* represents the person/object the intended action is directed at and *text* is the utterance being spoken. However, the only supported script unit is `speech`, when a character makes an utterance.

An excerpt from a Brazilian comic strip called Monica's Gang (Figure 4.2) is used as test case to illustrate how a plot is represented. A script unit that summarizes the first frame of this story could be: (speak, Monica, Jimmy Five, "Look, Jimmy Five! A rainbow").

Script Analysis: Before starting to act their roles, actors prepare for their characters by studying the script and trying to understand the plot. They rely on script analysis techniques to do this [Waxberg, 1998, Thomas, 2009]. One of these techniques is *action analysis*, where key aspects are extracted from the script to summarize it, like the theme of the plot, or the time and place where the story takes place. In our system, script analysis is represented by a record describing a few of the means that agents use to infer the current dramatic situation during the deliberation phase. At this stage, it is up to the script writer to write this script analysis record manually.

In the example mentioned earlier (Monica's Gang comic book), one possible script analysis record could be: theme (conversation), time (during the day at the present time) and place (outdoors).

Set: This contains the entire descriptions of the setting or environment in which the actors perform their scenes. In terms of implementation, a set is a data structure that enumerates all the components within the scene, such as buildings, objects, lights, cameras, and so on, with which agents can interact. Level design techniques can be employed that are similar to those used by computer games designers. One particular technique that is being investigated in this domain is *smart objects*. Smart objects are a special kind of object that store information about their functionality and physical attributes so that they can assist agents to interact with them [Fernandes et al., 2012].

In our example, the set is only composed of two objects: a grass field and a rainbow. A list of possible performable actions for the grass field object could be: stand, walk or sit; and regarding the rainbow could be: stare or point at.

6.1.2 Agent Architecture

The core element of this proposed system is its agent architecture, which describes the agent's deliberation process. Our proposed solution follows the traditional perception-deliberation-action process that is commonly used in the computer game industry [Byl, 2004], although several adaptations have been proposed, as described next.

Perception: The first step in every agent architecture, perception involves sensing an agent's surrounding environment, and transforming recognized inputs into *percepts*. Percepts are sensed inputs combined with semantic information, so the agent can use them to deliberate. I propose to subdivide perception into two submodules:

1. **Script Data Annotation Module:** it is expected that virtual actors will be capable of autonomously interpreting the plot, and inferring proper enactments for each script unit. One possible approach for this interpretation is sentence annotations. Sentence annotation is a common approach used to reduce the inherent linguistic complexity of natural language processing. It entails associating meaningful pre-processed tags to sentences, so that

agents can use these tags instead of having to cope with the complexity of the natural language. In [Baldes et al., 2002], the authors suggest a list of dialog act tags (more details of this can be found in [Gebhard et al., 2003]), to annotate sentences for their characters. I have adopted their list in our system, as illustrated in our running example:

Monica says "Look, Jimmy Five! A rainbow!"
`admire(rainbow)`

2. **Environment Sensing Module:** this emulates all the actor's senses to allow it to perceive changes in the surrounding set where it is acting. These sensors traverse the set looking for three kinds of percepts: actions of agents, consequences of events and aspects of objects. All these percepts are then passed on to four belief revision functions (BRF) and to the personality submodule, so the agent can infer a list of plausible actions used in the deliberation phase.

   In our example, both the agents (Monica and Jimmy Five) should be able to sense all the objects in the scene. Jimmy Five must also listen to what Monica is saying.

Personality: Our proposed personality model follows the Reiss Motivational Profile [Reiss, 2008] and is the part of the system that is responsible for mapping specific character traits (from the script analysis record) with basic desires, to determine a list of goals and respective satiating level. The integrator function integrates a range of information gathered from inner states (emotions and goals), relationships with other characters and actions inferred from the smart object repository to produce a list of actions that would be considered as coherent for the given personality. This, will later be filtered by the filter function (as explained next).

In our example, the agent Monica may regard staring at the rainbow as a coherent action because it shows that curiosity is a personality trait.

Deliberation: Once the incoming script has been interpreted in both the script annotation and environment sensing modules, the deliberation phase takes place. This is responsible for inferring plausible actions, and taking note of the personality aspects of the character being played. The suggested solution is based on a modified version of the EBDI architecture [Jiang et al., 2007]. EBDI agents (emotion-belief-desire-intention) are deliberative agents that reason on the basis of a series of *beliefs* about the current status of the surrounding environment, and are thus able to produce special inner states out of their interpretations of the inputs called *emotions*. In addition, they have their own list of *desires*, in the sense of goals to be achieved, and by combining all the three other components, the agents are able to infer *intentions* that are commitments to achieve specific goals.

The deliberation process starts with three belief revision functions (msg-, story- and sensorial-BRF) and receives these percepts produced at the perception phase. The msg-BRF is responsible for determining beliefs from messages (communications) between characters, the story-BRF updates the belief set with those detected from the script analysis record and the sensorial-BRF determines all the new beliefs learned from sensing the environment. These beliefs are then passed to the reactive-EUF (emotion update function) responsible for eliciting primary reactive emotions. The resulting beliefs and emotions are then combined into new beliefs in the input-BRF function. The cognitive-EUF processes secondary emotions, that are emotions we experience after thoughtful deliberation of the situation being experienced.

The core function of the deliberation phase is filter. This function obtains as input, current values for beliefs (B), emotions (E), desires (D) and intentions (I) sets, and then combines them (according to previously trained sets of rules) to produce new intentions:

$$I_{new} = E_{current} \times B_{current} \times D_{current} \times I_{current}$$

. Following this, these new intentions are used by the planning submodule to assemble 'plans' in a series of plausible actions (that come from the list of coherent actions determined by the personality module).

The deliberation phase results in a list of preliminary plans of actions, that will be adjusted at the rehearsal phase, to match other ADA suggestions, and combine them into a suggested timeline, as described in Section 6.1.3.

Continuing our example, the deliberation phase performs the following steps:

1. the msg-BRF takes previously annotated script unit (speak, Monica, Jimmy Five, “Look, Jimmy Five! A rainbow!”, admire) and translates it into a belief. I.e., `speak[Monica, Jimmy Five, admire]`;
2. the story-BRF generates the beliefs list for the script unit record. For instance: `[theme, conversation]`, `[place, outdoor]` and `[time, present]`;
3. the sensorial-BRF updates the agents beliefs, like for instance: `[object, rainbow]`, `[agent, Monica, speak]`;
4. reactive-EUF may suggest that Monica is experiencing “joy” after she has spotted some object she admires;
5. the input-BRF may suggest a new belief like `[rainbow, curious]` to denote that she regards seeing the rainbow as curious;
6. after this, the filter function collects all these inner states to determine that (for instance) the agent’s next intention might be “to demonstrate happiness”;
7. finally, planning may involve selecting the following sequence of actions: “perform(stare, rainbow)”, “perform(point at, rainbow)”, “display(happy face)” as a suitable plan to satisfy a current intention. This plan is the result that will be passed on so that it can be acted in the next phase.

Action: After the agent has deliberated over a list of preliminary plans of actions, it is time to translate each action into an individual acting performance. A performance corresponds to a list of actions that include temporal relations and dramatic elements. The Progression Line is a script analysis component that describe temporal relations between actions in the plot. This description is responsible for expressing which sequences of actions correspond to the story being told. The progression module is responsible for ordering plans of actions that will fit the progression line best. Following this, the voice intonation module includes commands for TTS (Text-to-Speech) tools to elicit emotional voice intonation features regarding the dramatic situation being enacted. Finally, the timing module provides appropriate timing parameters for each action in the suggested progression line. This module is responsible for determining factors like: when to start a given action, how long it should last and pauses, all regarding the dramatic techniques concerned.

The final result of the deliberation phase is the individual acting performance, which is a non-linear animation data structure that describes a series of actions and respective timing attributes. Following this, the rehearsal phase combines several of these individual performances (that originate from a different ADA), by translating actions into animation commands in a structure called 'timeline'.

6.1.3 Rehearsal

In dramaturgy, a rehearsal is the phase when actors practice their enactments together, and plan the best acting choices for each action. To do this, they first propose their own performances and then, combine them in accordance with the script and the director's guidelines.

The rehearsal phase in this architecture, uses a series of individual acting performances as input and combines them into a suggested timeline. This is accomplished in two stages: first, the structure module is responsible for searching for behavioral patterns in pairs of actors' performances. Whenever these patterns are detected, the system checks if any adjustment in the parameters of the actions (like voice intonation, facial expression and timing) is necessary. For instance, Jimmy Five would react differently to a criticism by Monica than it would if it was by Smudge. After this, an animation translator works with a given animation engine and translates each suggested action into animation commands to produce the final timeline.

The resulting timeline is ready to be loaded in the selected engine for rendering, as described next.

6.1.4 Animation Engine

The same approach as proposed (and implemented) for the previous version of the system, that relies on Autodesk™Maya™ to render the resulting animation can be used as the solution suggests using a general purpose XML-based file comprising the produced timeline that is later translated (by my own plugin) into the final render using the Trax editor in Maya, as illustrated in Figures 5.11 and 5.12 in the previous section.

6.2 Validation of the Design

This final suggested architecture has addressed all the previously detected issues in a satisfactory way, and produced an agent architecture that should help to provide a better understanding of the requirements for the development of a fully functional autonomous digital actor. Our analysis of this design is outlined below:

1. **Artifact Success**: this architecture allows the following actions to occur:
   (a) a description of cognitive agents;
   (b) manual training of new agents in areas in terms of their beliefs, basic desires, intentions, emotion memory and acting rules;
   (c) creation of scripts that contains the plot and the script analysis, and also includes detailed descriptions of characters and personality traits. There is also a description of the set of the play, in terms of smart objects (like those used in game designs);
   (d) allows agents to infer new beliefs from several sources (e.g. communicating with other ADA, interpreting the story and sensing changes in the set);
   (e) considers two levels of emotional reactions are possible (reactive and cognitive);

(f) the personality model to work as a filter that infers a list of coherent actions that correspond to the character's goals and dramatic context;

(g) to make improvements in the acting rules module to suggest plans of actions instead of single actions, and thus bring a more expressive performance;

(h) the voice intonation module of ADA to deliver compelling emotional voice intonation. This was not take into account in previous attempts;

(i) agents to reason about dramatic performances simply by modulating the speed of the suggested animation (through the timing module);

(j) a combination of several individual acting performances, thus allowing different ADA to actuate and plan the scene to be played as a whole (in a holistic view).

2. **Generalization**: by combining EBDI rules with previous imperfect match criteria (based on classification vectors), this approach suggests that it could be applicable to any domain of interest.

3. **Novelty**: although most of the components found in this architecture were inspired by previous studies, we believe that a combination of them in the way proposed in this study, represents an advance in terms of modeling expressive autonomous characters.

4. **Explanation Capability**: the list of requirements previously envisaged for an autonomous digital actor agent architecture seems to have been met with this architecture because all of its features have been covered as is shown in the following points:

   (a) it is possible to give a formal representation of the script, in terms of plot, the interpretation of the story and a description of the set in which the story takes place;

   (b) the proposed solution suggested alternatives to the agent's perception of its dramatic context, both in terms of an autonomous interpretation of the script and "sensing" the surrounding environment for classical game design techniques;

(c) the use of a hybrid agents architecture (with a reactive and a deliberative layer), allows the creation of a more flexible way of training agents for new domains in which knowledge can be reused and combined in different domains;

(d) in the literature, personality models are usually implemented as behavior modulation functions (those that force changes on how selected actions are performed) to allow the emulation of individualities[1]. The approach suggested in this work significantly differs from this by proposing a coherent action selection filter that selects (for each dramatic context) only those actions that a particular individual would be able to select;

(e) each agent is capable of suggesting individual acting performances by referring to several acting skills like emotional expression and voice intonation and also relating them to time constraints, which should (in theory) lead to a more precise performance;

(f) the rehearsal module allows several ADA individual performances to be integrated into one final renderable timeline, that is suitable for visualization in the animation engine of choice;

(g) machine learning techniques (like reinforcement learning) can be regarded as improving the agents' suggestions and enable them to learn the animators preferences. This will allow them to make better suggestions in the future, without the need for further training.

[1] Individuality in this context means that each agent behaves differently from each other

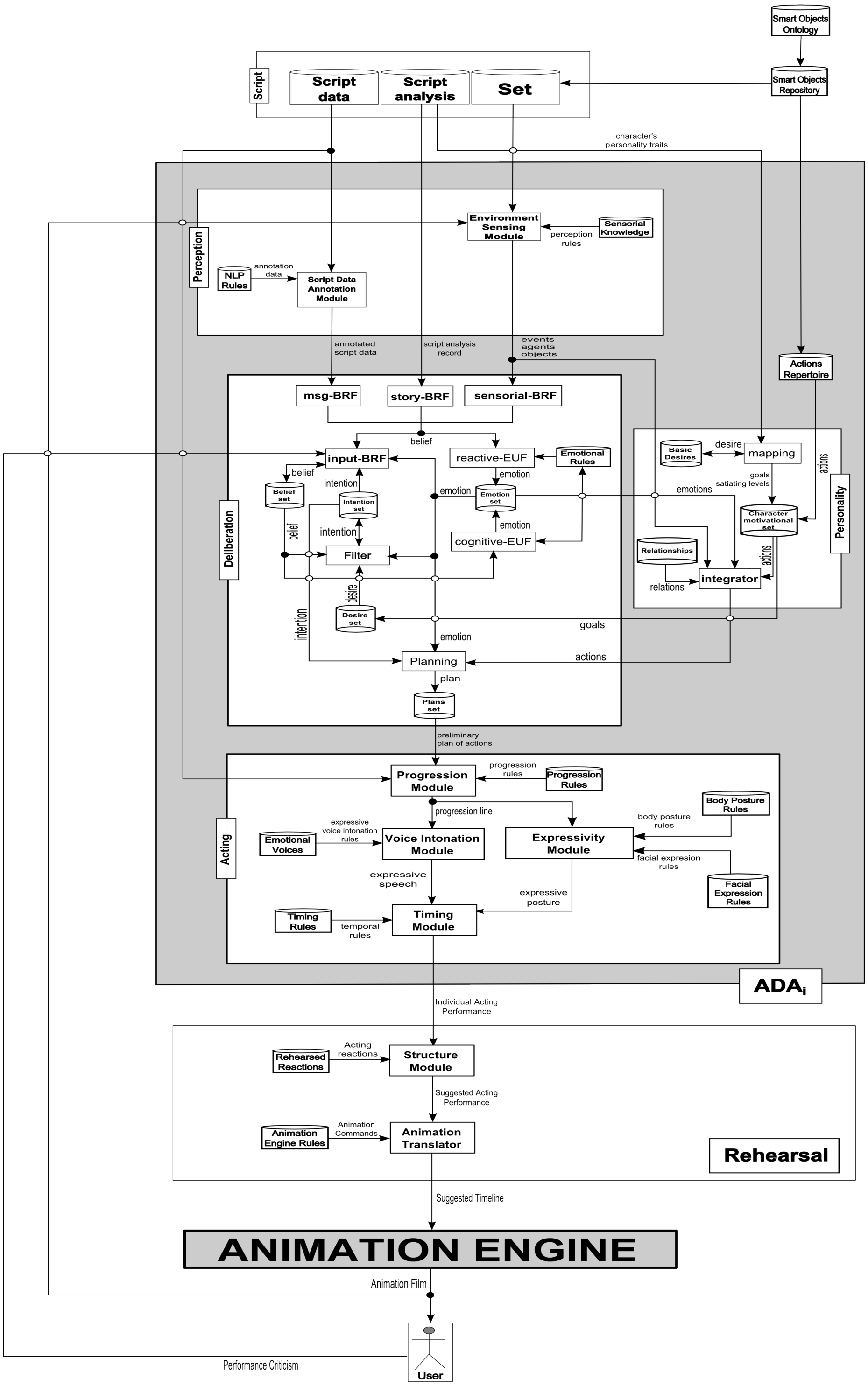


Figure 6.3: Detailed Final Agent Architecture

Chapter 7

Conclusions

It is well known that the animation industry has made great progress in its ability to give outstanding character performances. Due to the advances in technology that have affected the production of feature films, it is currently possible to produce characters whose appearance and motion are nearly indistinguishable from real-life human performances. However, these technologies are expensive and not accessible to everyone. Moreover, even in the case of those who have access to them, it is not an easy task to construct appealing movies, because of the complexity involved. Thus, new ways to allow faster, cheaper and simpler authoring techniques for character animations are necessary.

Almost two decades ago, a new concept was proposed which aimed at simplifying the authoring process: the use of believable characters. Believable characters are autonomous agents endowed with Artificial Intelligence and capable of demonstrating (at least some level of) *agency*. This means that they are capable of self-animation by coherently responding to changes sensed in the environment or following the plot of a script.

Several agent architectures, with a large range of applications, have been proposed in the literature, with the aim of producing believable characters that are suitable for either computer games or animations. However, despite all the efforts and advances made to understand the process of creating such believable characters, most of the initiatives taken in this area rely on reactive agents which are only capable of a stimulus-response type of decision-making (although there have been some attempts to use other kinds of agents, e.g. cognitive).

In 2007, Iurgel and Marcos [Iurgel and Marcos, 2007] have proposed a new idea about the authoring process for character animation: virtual actors. Virtual actors are "*an analogy to a real actor, which autonomously, and by its independent interpretation of the situation, can perform its role according to a given script, as part of a story*". Later, Perlin and Seidman [Perlin and Seidman, 2008] suggested a similar approach which they called 'Autonomous Digital Actors' or simply ADA.

The underlying concept of both works is that, instead of 'animating' characters, one should 'train' believable characters to perform acting skills autonomously. This research is intended to provide further investigation after the Project VirtualActor of the Centro de Computação Gráfica of the University of Minho. As the process for creating autonomous digital actors remains not fully comprehended, this thesis seeks to provide a better understanding of the requirements necessary to create the first fully functional ADA. Originally the Project VirtualActor has focused on studying the relations between virtual actors and animators by proposing a series of properties virtual actors should present during character-based animations authoring processes; while in this thesis, I have complemented this research by investigating novel agent architectures to design the autonomous digital actors themselves.

In carrying this out, I have conducted an extensive literature review on three major topics that, in my view, can assist in the development of an ADA:

The Art of Acting : In creating a believable character capable of understanding acting skills, a logical starting-point should be to understand how real actors learn their practice, and attempt to extrapolate from their techniques in a way that is applicable to autonomous agents. Our review of this subject taught us two important lessons:

1. there are several well-known traditional acting schools throughout the world which apprentice actors can rely on while learning how to act. The most

traditional standard acting methods are those of: Stanislavski, Strasberg, Meisner and Adler. These lay stress on the most elementary skills any actor must master: the use of imagination, emotion memory, sense memory and characterization.

2. apart from learning how to act, actors also have to study how to prepare for a role. One key feature in this preparation is script analysis. Script analysis comprises a wide range of information that the actor should be aware while studying his/her role. The most important factors are: action analysis, the given circumstances, background story, progression and structure, the character itself, the idea behind the story and dialogue.

The Art of Digital Acting : Digital actors represent the state of the art regarding creating virtual actors. They are special renditions of digitally captured human acting performances transposed to 'virtual humans' via specialized software. According to Thalmann & Thalmann [Thalmann and Daniel Thalmann, 2005], there are three main goals to produce compelling virtual humans (and the same applies to digital actors): realistic appearance modeling, realistic smooth and flexible motion and realistic high-level behavior.

Computer graphics has solved the problem of appearance by employing photo-realistic rendering techniques; the motion capture system has provided realistic motion and finally, a team of skilled animators have the ability to decide how each digital actor should behave in every scene, and thus produce a compelling performance for the audience.

The Art of Autonomous Digital Acting : We are now experiencing the dawn of autonomous digital acting. Believable characters and virtual humans are a few examples of recent attempts to create self-animated characters capable of delivering expressive performances. Moreover, one prerequisite of virtual actors is their reusability, which means that the same ADA should be usable in different roles for different stories; this does not occur in the technology employed with current digital actors, which requires new modeling for every new character.

I decided to adopt design research as the research methodology, because the main purpose of this thesis was to design an artifact (agent architecture or system) for an autonomous digital actor. In support of this approach, and because the list of requirements for constructing an ADA is not fully understood, we proposed our own list:

1. it should be possible for an ADA to autonomously interpret the script being enacted, which means inferring its role, the script analysis and the set in which the story takes place;
2. the ADAs should be able to keep a consistent mental attitudes to the whole dramatic context (which involves the imagination and acting skills as mentioned earlier);
3. each virtual actor should perform as a single individual, which suggests there will be different enactments for every dramatic context based on its own personality, along with the acting skills it has learnt;
4. a "suggested performance" should comprise actions that involve facial expressions, body postures and voice intonation. It should also be possible to combine several individual performances in such a manner that it meets the animator's expectations regarding the plot;
5. once an animation has been rendered, the system could also allow the animator to act like a director and send criticisms back to the cast, so that they can adjust their mental attitudes and learn how to improve their performance in the future. We call this the 'actor/director metaphor'.

The path that has led to the final proposal for the agent architecture outlined in this work, set out by defining the traditional four stages of the NPC design (perception-inference-selection-behavior) used by games industry. By taking this design as a starting-point, four architectures have been incrementally proposed:

1. the basic premise of the first attempt, was to determine the minimal requirements of a system that was capable of giving performances from a script. Thus, a reactive acting rules database that considered annotated script units was proposed. This architecture combined the inference and selection modules as we believed this would simplify the process further;

2. the second attempt separated the inference and selection modules and also incorporated a new module (emotion) and an emotion memory database (inspired by the actors' practice);

3. following this, the third version of the system, proposed a formal representation for the script that included the script analysis record. It also incorporated a behavior correction module because we realized that there was a need for the agents to keep improving their suggestions by learning the animator's preferences. In addition, it remodeled the way knowledge bases are trained by treating classification vectors as the rule representation which could allow an imperfect match inference to occur. This version has yielded promising results, which is why I have decided to validate it by implementing a prototype to work as a proof of concept. The implementation left a few remaining issues to be dealt with which were presumably solved by the definitive version;

4. the fourth (and definitive) version of the system relied on an extension of the traditional EBDI agent architecture (beliefs, desires, intentions and emotions). According to my judgement, this has met all the requirements that where analyzed earlier, and allowed me to conclude that:

   (a) the specifications of a formal representation of the script, allowed agents to infer what could be expected to be performed and how. The use of game design techniques (like Smart Objects) to describe the set, simplifies the task of sensing the surrounding environment, in terms of what can feasibly be animated in any given domain. It also, enables the resulting performance to be improved by avoiding the need for retraining agents whenever new objects are added to the scene;

   (b) by segmenting the rules and specifications into several components, the approach allows the creation of clusters of behavior, that can ensure a more flexible characterization. For instance, training knowledge bases for different cultures, genders, ages, ethnicities or any other desired feature, only entails rewriting their set of beliefs combining common beliefs with their specificities.

(c) we envisage the rise of new professionals for the character animation authoring process: these will include the ADA acting teacher, that is a professional specialized in training new knowledge for autonomous digital actors;

(d) although this thesis has only been concerned with the performances of human actors, there is no apparent reason for not adopting approaches for all the other elements of an animation production pipeline: for instance, training an ADA agent to work as the autonomous camera that decides what are the most suitable shots, plans and camera angles; or lighting, sound tracks and/or sound effects and so on.

For the reasons outlined here, I conclude that the initial goals for this research have been successfully achieved and an agent architecture has been produced for autonomous digital actors. In addition, the fact that a prototype has been implemented to work as a proof of concept for one of the designs, is further evidence that is feasible to adopt this approach. However, this research by no means represents a definitive solution of the problem and some of its limitations are discussed in the next section.

7.1 Limitations

The proposed agent architecture resulting from this study represents a preliminary step towards the creation of a fully functional autonomous digital actor, capable of autonomously interpreting its role in a story, and with a knowledge of the specialized acting skills required for compelling enactments and producing plot-based computer animations.

This section aims at outlining some of the drawbacks that have been detected in the solution. First of all, and most striking, is the lack of a prototype for the proposed agent architecture. Without a prototype, it is very difficult to validate several factors effectively, like how difficult it is to train an ADA for a particular role. How reusable the resulting agents are to different contexts and plots and so on.

Even after just have evaluated the system at a theoretical level, some important considerations could be raised:

Script Formalism : The script will have to be 'written' in some kind of formal language, to enable agents to interpret it. Interpreting a natural language is a very difficult task for autonomous agents to accomplish with current natural language processing technology. Thus, some sort of pre-processing is required by means of formalisms like conceptual graphs, semantic annotations, etc. while writing the plot. As a result, it follows that: (1) the script writer must be familiar with this formalism, and thus proper training is necessary; this means that the task is unsuitable for unskilled script writers and (2) language formalism often involves semantic simplification, which imposes constraints on the representation of plots. Thus making the proposed system not suitable for any kind of story. The rule of thumb is that, only those contexts covered by the initial language ontology can be taken into account while writing a story. Different kinds of story require different ontologies (or a broader one).

Limited Repertoire : In our solution, the ADAs rely on previously modeled smart objects to determine which actions they can perform in every scene, and in what way. Hence, whenever new actions (or skills) are necessary, new (or updated) smart objects must be incorporated into the set design. However, this approach seems to be in accordance with current game technology and should not be regarded as a limitation of any kind. The same factors apply to the voice intonation of the ADA performance: in the most likely scenario, specialized software tools like Text-To-Speak tools are needed to allow autonomous actors to speak in a believable way and thus, require specialized training in their use. Again, specialized professionals, who are trained to construct these components, would have to emerge, from the industry, to fill these positions.

Knowledge Base Training : Training an ADA involves adding new rules to their knowledge bases. I have proposed a rigid type of formalism to show how one can write these rules; thus the same limitations apply to the ontology of the domain. Even if we disregard this, writing a rule still involves constructing a reasoning template for a specific dramatic context. However, producing a description requires a mental exercise that is too complex for anyone, especially for broader domains, as it is the responsibility of the acting trainer not to teach contradictory instructions. Dedicated training editors might help in this task but, in the end, it is the responsibility of the teacher to write coherent acting rules. Further investigation is needed in the area of knowledge representation (and perhaps also technology) to find simpler ways of training the actors.

Experts Demanded : Apart from the fact that it might not be simple to translate tacit knowledge into acting rules (for the reasons explained earlier), another important problem is understanding the actual acting skills that have to be translated. Fully understanding the art of acting is essential to be able to train autonomous digital actors. In other words, training compelling ADA is not a task for the unskilled person.

7.2 Future Studies

Possible continuations to this thesis include the following:

1. First and foremost, it is necessary to complete the implementation of a fully functional prototype for the proposed architecture set out in this thesis. With this functional prototype, it would be possible to conduct a further series of experiments to validate the use of autonomous digital actors in real scenarios, and attempt to understand the validation criteria as [Carvalho, 2012] points out. Despite the fact that a fully functional system is not yet being operated, a few preliminary partial experiments have been conducted by students of mine in their final graduate degree course. This can be listed as follows:
    (a) Development of an ontology for the creation of smart objects [Fernandes and da Silva, 2011, Fernandes et al., 2012].
    (b) Experiments with autonomous script unit annotation based on speech acts [Magri and da Silva, 2011, Magri et al., 2012].
    (c) Proposal of an extension of the classical EBDI agent architecture for autonomous digital actors [Thumé and da Silva, 2012].
    (d) A suggested personality model based on Reiss Motivational Profile [Ramos et al., 2012].

   (e) An introductory study regarding technologies available for emotional voice intonation Text-to-Speech tools [Crivellaro and da Silva, 2012].

2. Apart from the implementation of the system itself, I envisage that further studies are necessary on the question of training in all the other elements in the agent architecture:

   (a) autonomous script interpretation relies on natural language processing techniques. The current technology only deals with certain aspects of human language, and these include the limiting agent's skills in inferring knowledge from the script.

   (b) good characterization, in my solution, largely depends on using the ADA trainer skills to achieve consistent knowledge representation and compile behavioral rules. If more formal new ways of describing character roles can be found, it should make ADA training simpler, faster, and suitable for the unskilled animator to create its own agents without the need of experts.

   (c) As well as having acting skills, autonomous agents will also need general semantic, episodic and procedural knowledge and be able to understand the meaning of ordinary everyday actions like, for instance, what a chair or bike is used for. In my view, these kinds of knowledge could be shared by all the agents, and would simplify their training by operating a "train one teach all" principle.

   (d) It is certainly a real challenge to combine different individual performances and at the same time take account of each dramatic context, individual personalities, the director's expectations and any other abstract criteria that need to be applied. The social sciences could be studied to find out how people live together, and respond to each other's behavior. Factors such as culture, religion and ethnicity, should be taken into consideration during the training of these *socially-aware autonomous digital actors.*

7.3 Publications

This section lists the publications that have appeared as a result of the writing of this thesis. Some of these publications were produced by members of the VirtualActor project at CCG – Centro de Computação Gráfica (`http://www.ccg.pt/`) of University of Minho, others as a member of CoCA – Computação Cognitiva Aplicada group (`http://www2.joinville.udesc.br/~coca`) at Santa Catarina State University (UDESC):

2009 Demo presentations:

1. IURGEL, Ido A., SILVA, Rogério E. da, RIBEIRO, Pedro R., SOARES, Abel B., SANTOS, Manuel F. dos *CREACTOR An Authoring Framework for Virtual Actors* In: Intelligent Virtual Agents, 2009, Amsterdam. Proceedings of the Intelligent Virtual Agents (IVA), LNAI. Springer, 2009. v.5773. p.562 - 563

Short papers:

1. SILVA, Rogério E. da, IURGEL, Ido A., SANTOS, Manuel F. dos *Towards Virtual Actors - The Next Step for the Entertainment Industry* In: Simpósio Brasileiro de Jogos e Entretenimento Digital, 2009, Rio de Janeiro.

2010 Full papers:

1. SILVA, Rogério E. da, IURGEL, Ido A., SANTOS, Manuel F. dos, ZAGALO, Nelson, BRANCO, Pedro *Understanding Virtual Actors* In: IX SBGAMES, 2010, Florianópolis, SC, Brasil. Simpósio Brasileiro de Jogos e Entretenimento Digital, 2010.
2. IURGEL, Ido A., SILVA, Rogério E. da, SANTOS, Manuel F. dos *Towards Virtual Actors for Acting out Stories* In: Edutainment, 2010, Changchun, China. 5^{th} International Conference on E-Learning and Games (Edutainment). Springer, 2010.

2011 Full papers:

1. SILVA, Rogério E. da, SANTOS, Manuel F. dos, IURGEL, Ido A. *Developing Virtual Actors Inspired by Real Actors' Practice.* In: Proceeding DMDCM '11, 2011, Hangzhou, Zhejiang China. Proceedings of the 2011 Workshop on Digital Media and Digital Content Management. Hangzhou, Zhejiang China: IEEE Computer Society, 2011.

2012 Full papers:

1. SILVA, Rogério E. da *D.R.A.M.A. - Developing Rational Agents to Mimic Actors.* In: XI SBGAMES, 2012, Brasília, DF, Brasil. Simpósio Brasileiro de Jogos e Entretenimento Digital, 2012.
2. MAGRI, Dino R. C., SILVA, Rogério E. da, SÁ, Claudio C. e HEINEN, Milton R. *Uma Revisão Teórica sobre Classificação Textual de Atos de Fala para Atores Virtuais Utilizando Aprendizagem de Máquina*, In: Computer on the Beach, Florianópolis, SC, Brasil, 2012.

Short papers:

1. THUMÉ, Gabriela e SILVA, Rogério E. da *An extended EBDI Model Applied to Autonomous Digital Actors.* In: XI SBGAMES, 2012, Brasília, DF, Brasil. Simpósio Brasileiro de Jogos e Entretenimento Digital, 2012. (Honored as one of the three best short paper of the Computing Track of the event).
2. RAMOS, Ricardo P., SILVA, Rogério E. da e REIS, Juliane C.K. *Personality Model based on Reiss Motivational Prole for Autonomous Digital Actors.* In: XI SBGAMES, 2012, Brasília, DF, Brasil. Simpósio Brasileiro de Jogos e Entretenimento Digital, 2012.
3. FERNANDES, Lucas G.A., SILVA, Rogério E. da e SÁ, Claudio C. *An Ontology for 3D Environment Modeling with Smart Objects for Autonomous Digital Actors*, In: Computer on the Beach, Florianópolis, SC, Brasil, 2012.
4. CRIVELLARO, Marcos E. e SILVA, Rogério E. da *Um Estudo sobre Expressão de Emoção na Voz Gerada por Conversão Texto-Fala*, In: Computer on the Beach, Florianópolis, SC, Brasil, 2012.

References

[Aylett et al., 2005] Aylett, R. S., Louchart, S., Dias, J., Paiva, A., and Vala, M. (2005). FearNot! - an experiment in emergent narrative. In *Intelligent Virtual Agents*, pages 305–316. Springer.

[Baldes et al., 2002] Baldes, S., Gebhard, P., Kipp, M., Klesen, M., Rist, P., Rist, T., and Schmitt, M. (2002). The interactive CrossTalk installation: Meta-theater with animated presentation agents. In *Proceedings of the Workshop on "Lifelike Animated Agents" held in conjunction with the Seventh Pacific Rim International Conference on Artificial Intelligence*, pages 9–15.

[Ball and Breese, 2000] Ball, G. and Breese, J. (2000). Emotion and personality in a conversational agent. In Cassell, J., Sullivan, J., Prevost, S., and Churchill, E., editors, *Embodied Conversational Agents*, pages 189–219. MIT Press.

[Barab and Squire, 2004] Barab, S. and Squire, K. (2004). Design-based research: Putting a stake in the ground. *Journal of the Learning Sciences*, 13(1):1–14.

[Bartneck, 2002] Bartneck, C. (2002). Integrating the OCC model of emotions in embodied characters. In *Proceedings of the Workshop on Virtual Conversational Characters: Applications, Methods, and Research Challenges*, Melbourne.

[Bates, 1994] Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125.

[Bull and Kovacs, 2005] Bull, L. and Kovacs, T. (2005). Foundations of learning classifier systems: An introduction. In Bull, L. and Kovacs, T., editors, *Foundations of*

Learning Classifier Systems, volume 183 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag, Berlin/Heidelberg.

[Bulterman and Hardman, 2005] Bulterman, D. C. A. and Hardman, L. (2005). Structured multimedia authoring. *ACM Trans. Multimedia Comput. Commun. Appl.*, 1(1):89–109.

[Busato et al., 1999] Busato, V. V., Prins, F. J., Elshout, J. J., and Hamaker, C. (1999). The relation between learning styles, the big five personality traits and achievement motivation in higher education. *Personality and Individual Differences*, 26:129–140.

[Byl, 2004] Byl, P. B.-d. (2004). *Programming Believable Characters for Computer Games*. Charles River Media.

[Carvalho, 2012] Carvalho, J. A. (2012). Validation criteria for the outcomes of design research. In *International Workshop on IT Artefact Design & Workplace Intervation*, Barcelona, Spain.

[Cassell et al., 2001] Cassell, J., Vilhjalmsson, H. H., and Bickmore, T. (2001). BEAT: the behavior expression animation toolkit. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 477–486. ACM.

[Collins et al., 2001] Collins, J., Gini, M., and Mobasher, B. (2001). Decision processes in agent-based automated contracting. *IEEE Internet Computing*, 5(2):61–72.

[Crivellaro and da Silva, 2012] Crivellaro, M. E. and da Silva, R. E. (2012). Um estudo sobre expressão de emoção na voz gerada por conversão texto-fala. In *Computer on the Beach*, Florianópolis, SC, Brazil.

[da Silva et al., 2010] da Silva, R. E., Iurgel, I. A., dos Santos, M. F., Branco, P., and Zagalo, N. (2010). Understanding virtual actors. In *2010 Brazilian Symposium on Games and Digital Entertainment (SBGAMES)*, pages 220–229. IEEE.

[de Melo and Paiva, 2006] de Melo, C. and Paiva, A. (2006). Multimodal expression in virtual humans: Research articles. *Comput. Animat. Virtual Worlds*, 17(3-4):239248.

[Egges et al., 2003] Egges, A., Kshirsagar, S., and Thalmann, N. M. (2003). A model for personality and emotion simulation. In *Knowledge-Based Intelligent Information & Engineering Systems*.

[Ekman, 1992] Ekman, P. (1992). Are there basic emotions? *Psychological Review*, 99(3):550–553. PMID: 1344638.

[Ekman, 1999] Ekman, P. (1999). Facial expressions. In Dalgleish, T. and Power, T., editors, *Handbook of Cognition and Emotion*. John Wiley & Sons Ltd., New York, NY, USA.

[Ekman, 2003] Ekman, P. (2003). *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Holt Paperbacks, 2nd edition.

[Ekman and Friesen, 1978] Ekman, P. and Friesen, W. V. (1978). *The Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto.

[Fernandes and da Silva, 2011] Fernandes, L. G. A. and da Silva, R. E. (2011). Uma ontologia para a modelagem de ambientes 3D com smart objects para atores digitais autônomos. Undergraduation final work, Santa Catarina State University (UDESC), Joinville, SC, Brazil.

[Fernandes et al., 2012] Fernandes, L. G. A., da Silva, R. E., and Sá, C. C. (2012). An ontology for 3D environment modeling with smart objects for autonomous digital actors. In *Computer on the Beach*, Florianópolis, SC, Brazil.

[Freytag, 1990] Freytag, G. (1990). *Technique of the Drama: an Exposition of Dramatic Composition and Art*. S.C. Griggs Company.

[Gebhard et al., 2003] Gebhard, P., Klesen, M., and Rist, T. (2003). Adding the emotional dimension to scripting character dialogues. *Proc. of the 4th International Working Conference on Intelligent Virtual Agents (IVA'03), Kloster Irsee*, 2003:48—56.

[Hayes-roth et al., 1997] Hayes-roth, B., Van Gent, R., and Huber, D. (1997). Acting in character. In *Autonomous Agents and Multi-Agent Systems*, pages 92—112.

[Iurgel et al., 2010] Iurgel, I. A., da Silva, R. E., and dos Santos, M. F. (2010). Towards virtual actors for acting out stories. In *Proceedings of the 5th International Conference on E-learning and Game (Edutainment)*, Changchun, China. Springer Verlag.

[Iurgel et al., 2009] Iurgel, I. A., da Silva, R. E., Ribeiro, P. R., Soares, A. B., and dos Santos, M. F. (2009). CREACTOR – an authoring framework for virtual actors. In *International Conference on Intelligent Virtual Agents (IVA)*, Amsterdam, The Netherlands. Springer Verlag.

[Iurgel and Marcos, 2007] Iurgel, I. A. and Marcos, A. F. (2007). Employing personality-rich virtual persons–new tools required. *Computers & Graphics*, 31(6):827–836.

[Jiang et al., 2007] Jiang, H., Vidal, J. M., and Huhns, M. N. (2007). EBDI: an architecture for emotional agents. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, Honolulu, Hawaii. ACM.

[Johnston and Thomas, 1995] Johnston, O. and Thomas, F. (1995). *The Illusion of LIfe: Disney Animation*. Disney Editions, rev sub edition.

[Jurafsky and Martin, 2000] Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, 1 edition.

[Klesen, 2005] Klesen, M. (2005). Using theatrical concepts for role-plays with educational agents. *Applied Artificial Intelligence: An International Journal*, 19(3):413.

[Kline and Blumberg, 1999] Kline, C. and Blumberg, B. (1999). The art and science of synthetic character design. In *Proceedings of Symposium on AI and Creativity in Entertainment and Visual Art (AISB)*, Edinburgh, Scotland.

[Kopp et al., 2006] Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., Thrisson, K., and Vilhjlmsson, H. (2006). Towards a common framework for multimodal generation: The behavior markup language. In *Intelligent Virtual Agents*, volume 4133, pages 205–217. Springer-Verlag.

[Kshirsagar and Thalmann, 2002] Kshirsagar, S. and Thalmann, N. M. (2002). Virtual humans personified. In *Proceedings of Autonomous Agents Conference (AAMAS)*, pages 356–359.

[Kundert-Gibbs and Kundert-Gibbs, 2009] Kundert-Gibbs, J. and Kundert-Gibbs, K. (2009). *Action! Acting Lessons for CG Animators*. Wiley Publishing, Inc., USA.

[Macedo, 2007] Macedo, L. M. M. L. (2007). *The Exploration of Unknown Environments by Affective Agents*. PhD thesis, University of Coimbra.

[Maes, 1995] Maes, P. (1995). Artificial life meets entertainment: Lifelike autonomous agents. *Communications of the ACM*, 38(11):108–114.

[Maestri, 2006] Maestri, G. (2006). *Digital Character Animation 3*. New Riders Press.

[Magri and da Silva, 2011] Magri, D. R. C. and da Silva, R. E. (2011). Classificação textual de atos de fala para atores virtuais utilizando aprendizagem de máquina. Undergraduation final work, Santa Catarina State University (UDESC), Joinville, SC, Brazil.

[Magri et al., 2012] Magri, D. R. C., da Silva, R. E., Sá, C. C., and Heinen, M. R. (2012). Uma revisão teórica sobre classificação textual de atos de fala para atores virtuais utilizando aprendizagem de máquina. In *Computer on the Beach*, Florianópolis, SC, Brazil.

[Mateas and Stern, 2003] Mateas, M. and Stern, A. (2003). Façade, an experiment in building a fully-realized interactive drama. In *Game Developers Conference*, San Jose, CA.

[McKee, 1997] McKee, R. (1997). *Story: Substance, Structure, Style, and the Principles of Screenwritting*. Harper Collins Publishers.

[Mullen, 2007] Mullen, T. (2007). *Introducing Character Animation with Blender*. Sybex.

[Nareyek, 2002] Nareyek, A. (2002). Intelligent agents for computer games. *Computer and Games, Second International Conference, CG 2002, Springer LNCS 2063*, 22(414).

[Ortony et al., 1988] Ortony, A., Clore, G. L., and Collins, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press.

[Osipa, 2003] Osipa, J. (2003). *Stop Staring: Facial Modeling and Animation Done Right*. Sybex, 1 edition.

[Paiva et al., 2001] Paiva, A., Machado, I., and Prada, R. (2001). Heroes, villains, magicians, ...: dramatis personae in a virtual story creation environment. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 129–136, Santa Fe, New Mexico, United States. ACM.

[Perlin and Seidman, 2008] Perlin, K. and Seidman, G. (2008). Autonomous digital actors. In *Motion in Games*, volume 5277/2008 of *Lecture Notes in Computer Science*, pages 246–255. Springer Verlag.

[Picard, 1997] Picard, R. W. (1997). *Affective Computing*. The MIT Press.

[Plantec, 2004] Plantec, P. (2004). *Virtual Humans*. Amacom, NY, USA, first edition.

[Prada et al., 2000] Prada, R., Machado, I., and Paiva, A. (2000). TEATRIX: virtual environment for story creation. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, pages 464–473. Springer-Verlag.

[Prendinger and Ishizuka, 2001] Prendinger, H. and Ishizuka, M. (2001). Affective and social behavior in animated agents. In *CD-Rom Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International*, pages 2918–2923, Vancouver, Canada.

[Ramos et al., 2012] Ramos, R. P., da Silva, R. E., and Reis, J. C. K. (2012). A personality model based on reiss motivational profile for autonomous digital actors. In *XI Brazilian Symposium on Games and Digital Entertainment*, Brasilia, DF, Brazil.

[Reiss, 2004] Reiss, S. (2004). Multifaceted nature of intrinsic motivation: The theory of 16 basic desires. *Review of General Psychology*, 8(3):179–193.

[Reiss, 2008] Reiss, S. (2008). *The Normal Personality*. Cambridge University Press.

[Roberts, 2007] Roberts, S. (2007). *Character Animation: 2D Skills for Better 3D*. Focal Press, second edition.

[Russell and Norvig, 2002] Russell, S. J. and Norvig, P. (2002). *Artificial Intelligence - A Modern Approach*. Prentice Hall, 2nd edition.

[Santos et al., 2009] Santos, M. F., Mathew, W., Kovacs, T., and Santos, H. (2009). Supervised learning classifier systems for grid data mining. In *Proceedings of the international conference on Computational and information science 2009*, CIS'09, page 416424, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society (WSEAS).

[Sevin and Thalmann, 2005] Sevin, E. d. and Thalmann, D. (2005). A motivational model of action selection for virtual humans. In *Computer Graphics International (CGI)*, New York, NY, USA. IEEE Computer SocietyPress.

[Stanislavski, 1937] Stanislavski, C. (1937). *An Actor Prepares*. Methuen Drama.

[Stanislavski, 1950] Stanislavski, C. (1950). *Building A Character*. Methuen Drama.

[Stefan et al., 2007] Stefan, G., Iurgel, I. A., Rossler, M., Hulsken, F., and Eckes, C. (2007). Design and narrative structure for the virtual human scenarios. *The International Journal of Virtual Reality*, 6(4):1–10.

[Thalmann and Daniel Thalmann, 2005] Thalmann, N. M. and Daniel Thalmann (2005). Virtual humans: thirty years of research, what next? *The Visual Computer*, 21(12):997–1015.

[Thalmann and Thalmann, 1993] Thalmann, N. M. and Thalmann, D. (1993). The world of virtual actors. *Virtual Worlds and Multimedia, John Wiley*, pages 113–126.

[Thomas, 2009] Thomas, J. (2009). *Script Analysis for Actors, Directors and Designers*. Focal Press, fourth edition.

[Thumé and da Silva, 2012] Thumé, G. S. and da Silva, R. E. (2012). An extended EBDI model applied to autonomous digital actors. In *XI Brazilian Symposium on Games and Digital Entertainment*, Brasilia, DF, Brazil.

[Tupes and Christal, 1992] Tupes, E. C. and Christal, R. E. (1992). Recurrent personality factors based on trait ratings. *Journal of Personality*, 60:225–251.

[Vilhjlmsson et al., 2007] Vilhjlmsson, H., Cantelmo, N., Cassell, J., E. Chafai, N., Kipp, M., Kopp, S., Mancini, M., Marsella, S., Marshall, A., Pelachaud, C., Ruttkay, Z., Thrisson, K., van Welbergen, H., and van der Werf, R. (2007). The behavior markup language: Recent developments and challenges. In *Intelligent Virtual Agents*, pages 99–111. Springer-Verlag.

[Waxberg, 1998] Waxberg, C. S. (1998). *The Actor's Script: Script Analysis for Performers*. Heinemann Trade, first edition.

[Weston, 1996] Weston, J. (1996). *Directing Actors: Creating Memorable Performances for Film and Television*. Michael Wiese Productions, USA.

[Weston, 2003] Weston, J. (2003). *The Film Director's Intuition: Script Analysis and Rehearsal Techniques*. Michael Wiese Productions, illustrated edition.

Appendix A

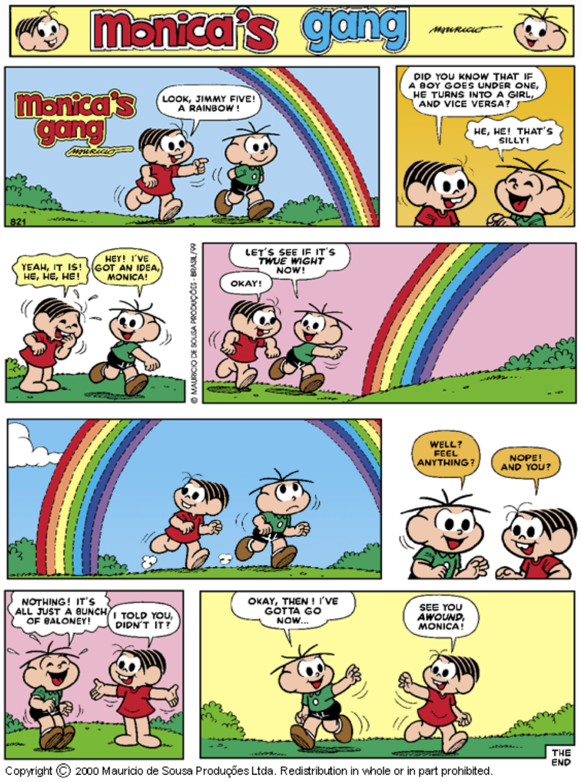
Script Example

```
<script title=’Monica’>
  <analysis>
    <attribute name=’theme’>
      <entry>conversation</entry>
    </attribute>
    <attribute name=’place’>
      <entry>other place</entry>
    </attribute>
    <attribute name=’time’>
      <entry>day</entry>
      <entry>present</entry>
    </attribute>
    <character name=’monica’ ADA=’angelina’>
      <attribute name=’personality’>
        <entry value=’0.6’>openness</entry>
        <entry value=’0.9’>extraversion</entry>
        <entry value=’0.9’>neuroticism</entry>
        <entry value=’0.4’>agreeableness</entry>
        <entry value=’0.6’>conscientiousness</entry>
      </attribute>
      <attribute name=’gender’>
        <entry value=’1.0’>woman</entry>
      </attribute>
      <attribute name=’emotional state’>
        <entry value=’0.3’>happiness</entry>
      </attribute>
      <attribute name=’age’>
        <entry value=’1.0’>teenager</entry>
      </attribute>
    </character>
    <character name=’jimmy five’ ADA=’brad’>
      <attribute name=’personality’>
        <entry value=’0.5’>openness</entry>
        <entry value=’0.9’>extraversion</entry>
        <entry value=’0.3’>neuroticism</entry>
        <entry value=’0.2’>agreeableness</entry>
        <entry value=’0.5’>conscientiousness</entry>
      </attribute>
      <attribute name=’gender’>
        <entry value=’1.0’>man</entry>
      </attribute>
      <attribute name=’emotional state’>
```

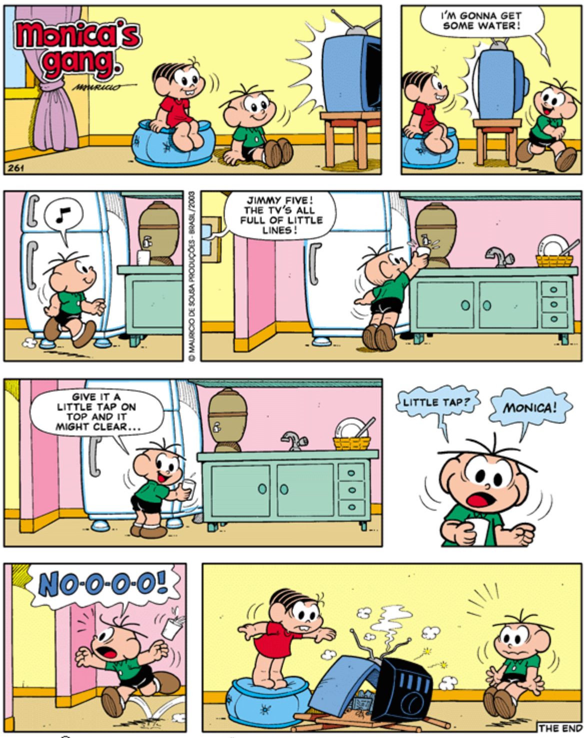
```
        <entry value='0.3'>happiness</entry>
      </attribute>
      <attribute name='age'>
        <entry value='1.0'>teenager</entry>
      </attribute>
    </character>
  </analysis>
  <scriptunit type='speak'>
    <target>jimmy five</target>
    <text>Look, Jimmy Five! A rainbow!</text>
    <actor>monica</actor>
    [...]
  </scriptunit>
  [...]
</script>
```

Appendix B

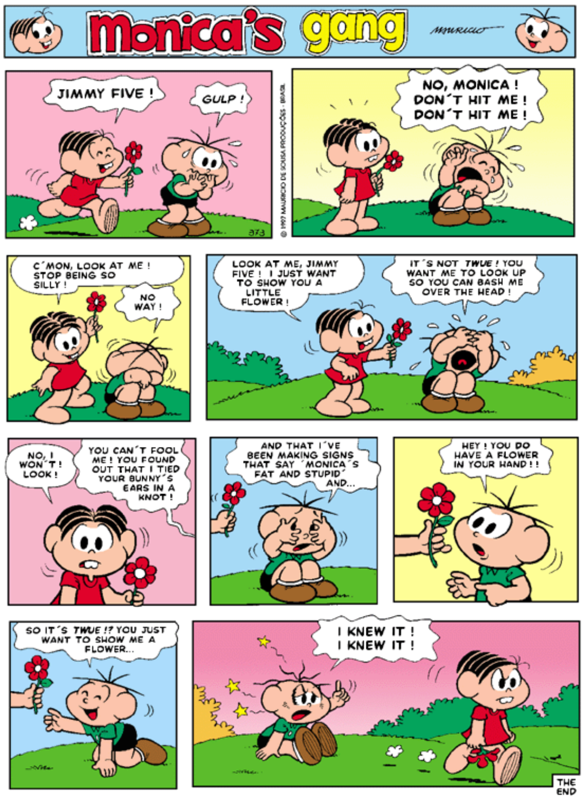
Comic Book Strips

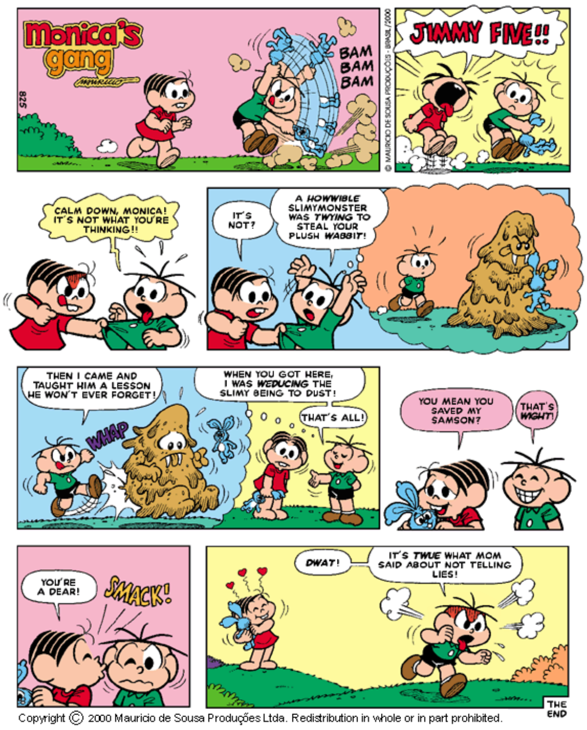


Copyright © 2000 Mauricio de Sousa Produções Ltda. Redistribution in whole or in part prohibited.

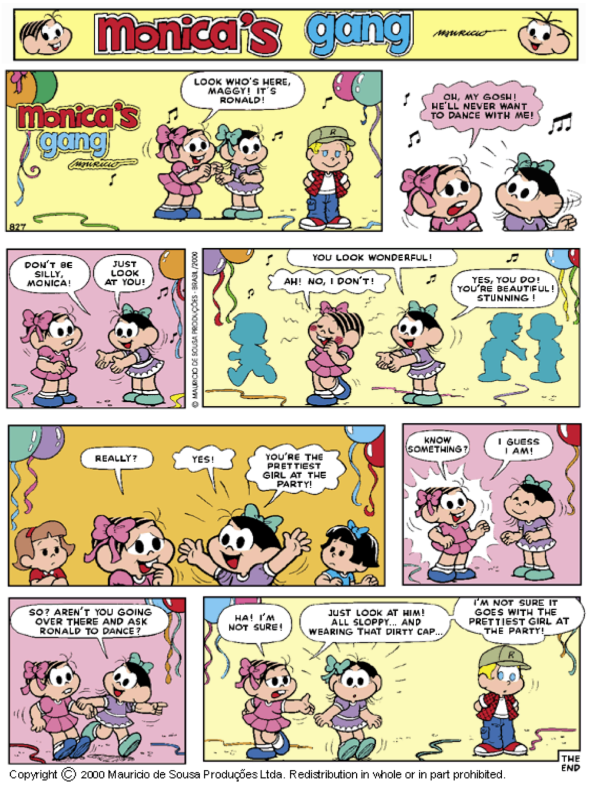


Copyright © 2003 Mauricio de Sousa Produções Ltda. Redistribution in whole or in part prohibited.

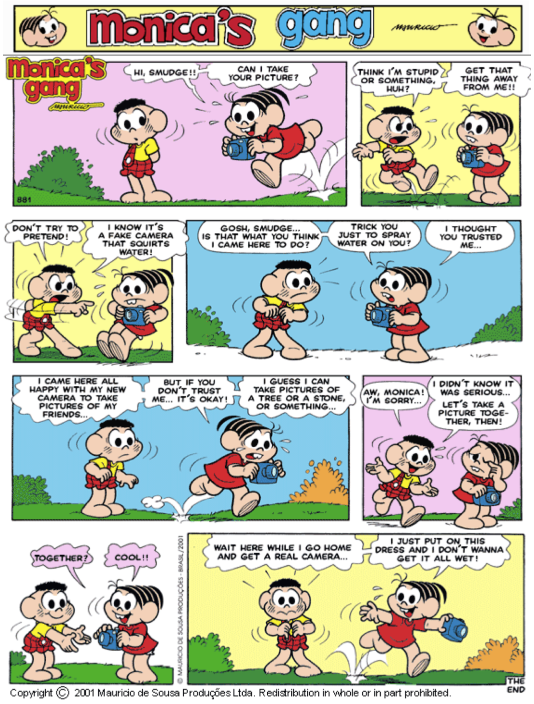




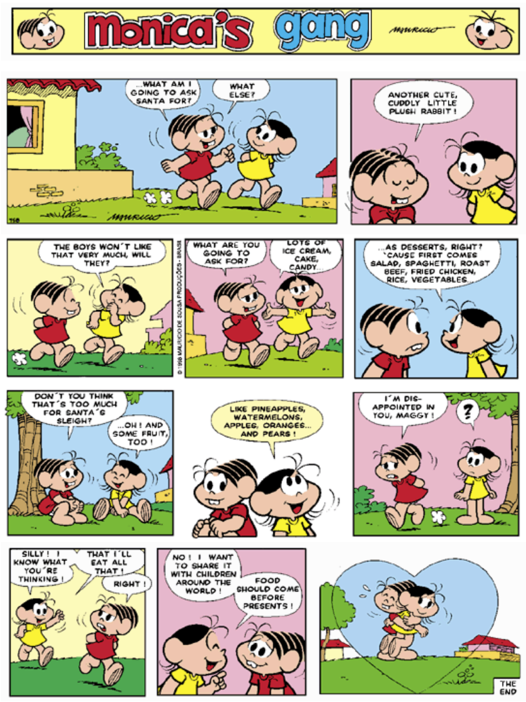
Copyright © 2000 Mauricio de Sousa Produções Ltda. Redistribution in whole or in part prohibited.

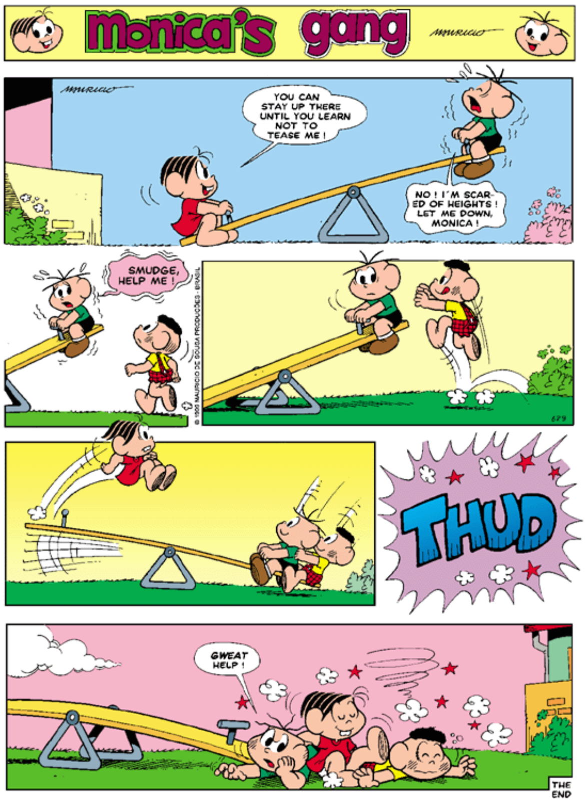


Copyright © 2000 Mauricio de Sousa Produções Ltda. Redistribution in whole or in part prohibited.



Copyright © 2001 Mauricio de Sousa Produções Ltda. Redistribution in whole or in part prohibited.




Monica's gang
MAURICIO
YOU CAN STAY UP THERE UNTIL YOU LEARN NOT TO TEASE ME !
NO ! I´M SCARED OF HEIGHTS ! LET ME DOWN, MONICA !
SMUDGE, HELP ME !
© 1990 MAURICIO DE SOUSA PRODUÇÕES - BRASIL
THUD
GWEAT HELP !
THE END

Appendix C

ADA Behavioral States

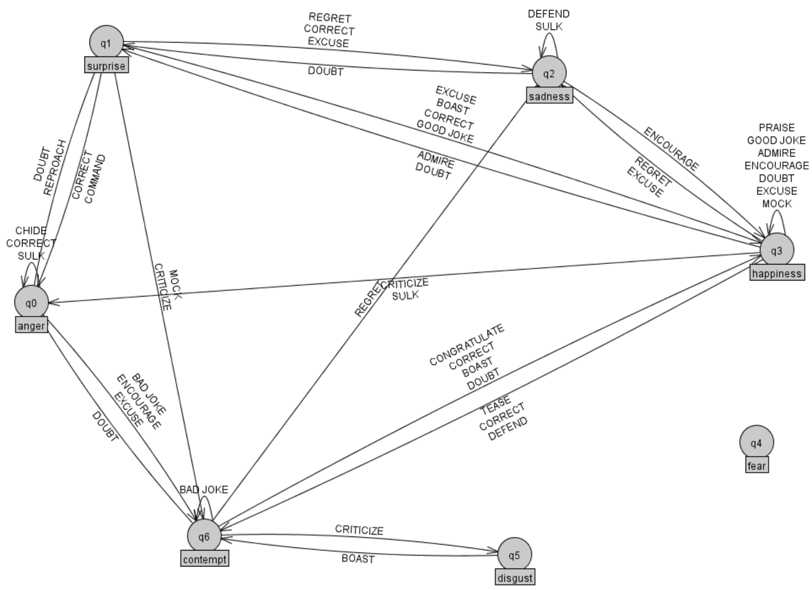


Figure C.1: Emotional reactions for Angelina acting as the actuator

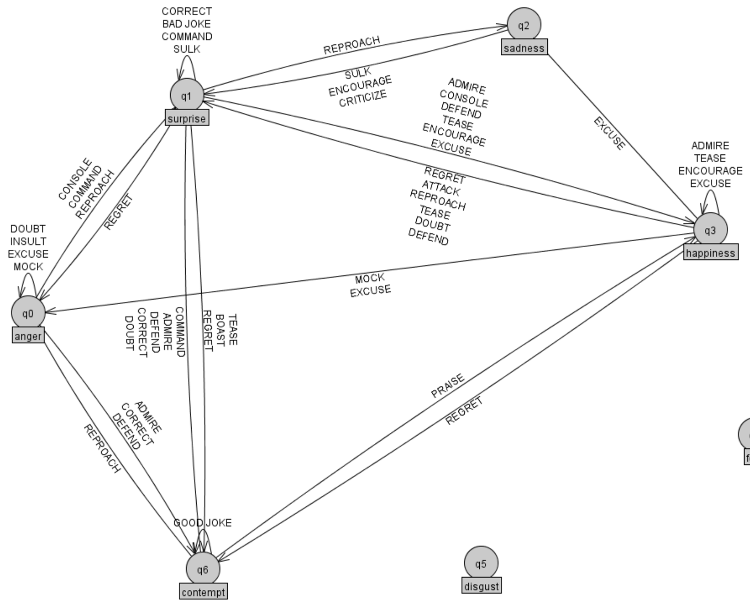


Figure C.2: Emotional reactions for Angelina acting as the effector

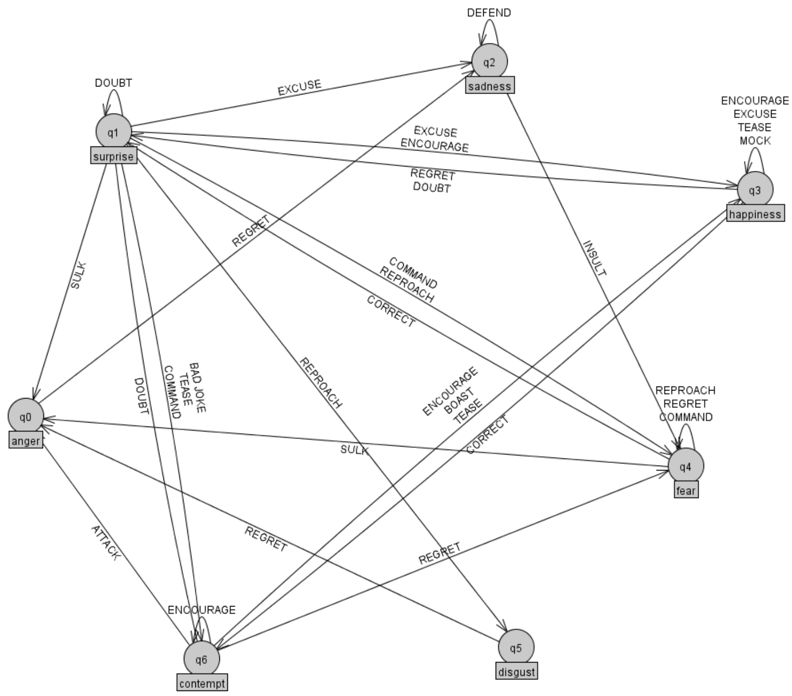


Figure C.3: Emotional reactions for Brad acting as the actuator

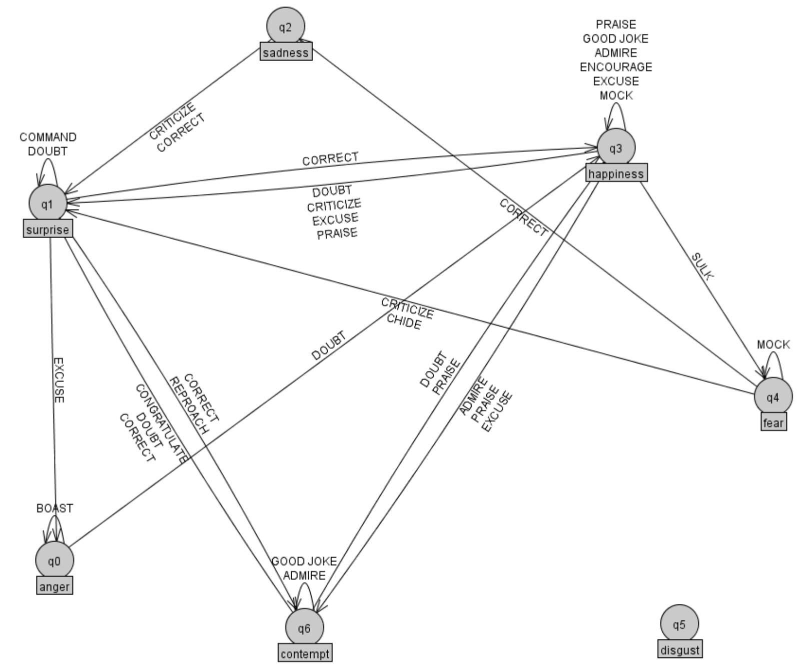


Figure C.4: Emotional reactions for Brad acting as the effector

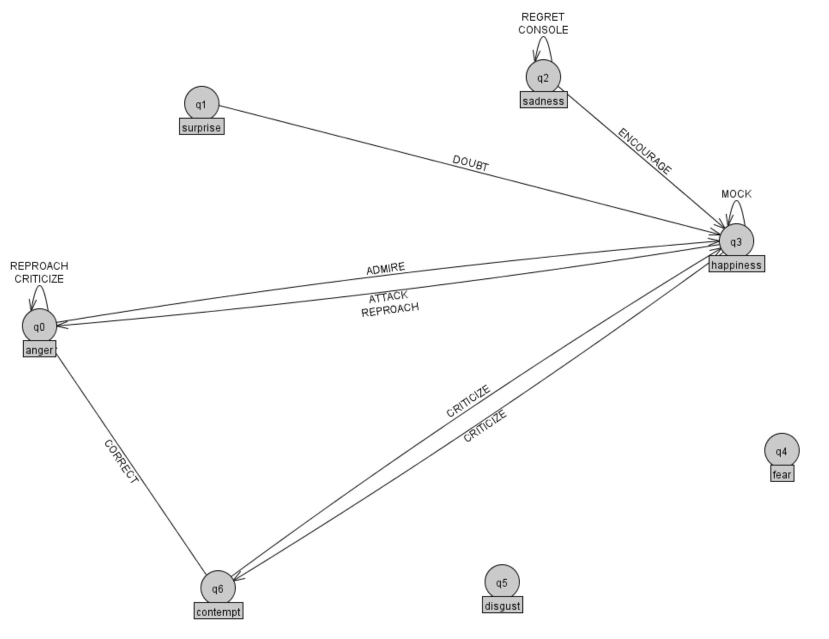


Figure C.5: Emotional reactions for Marlon acting as the actuator

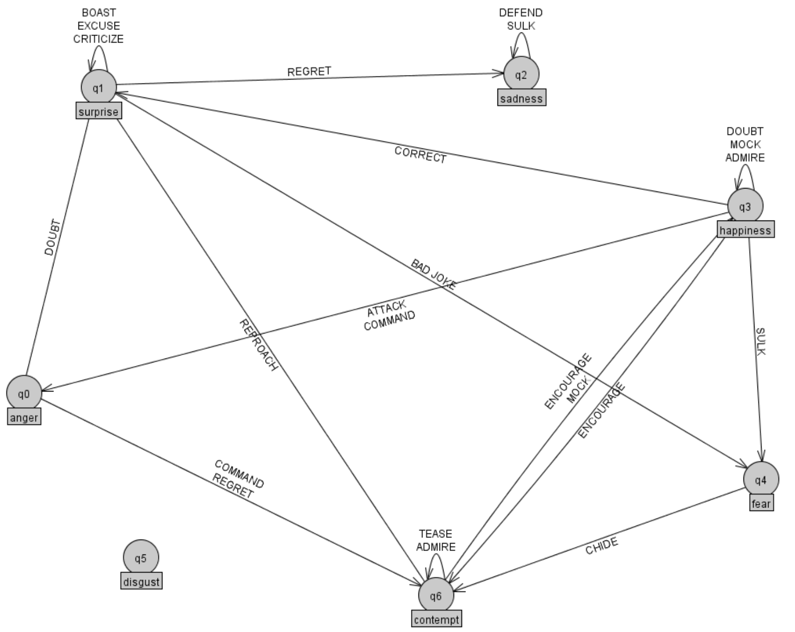


Figure C.6: Emotional reactions for Marlon acting as the effector

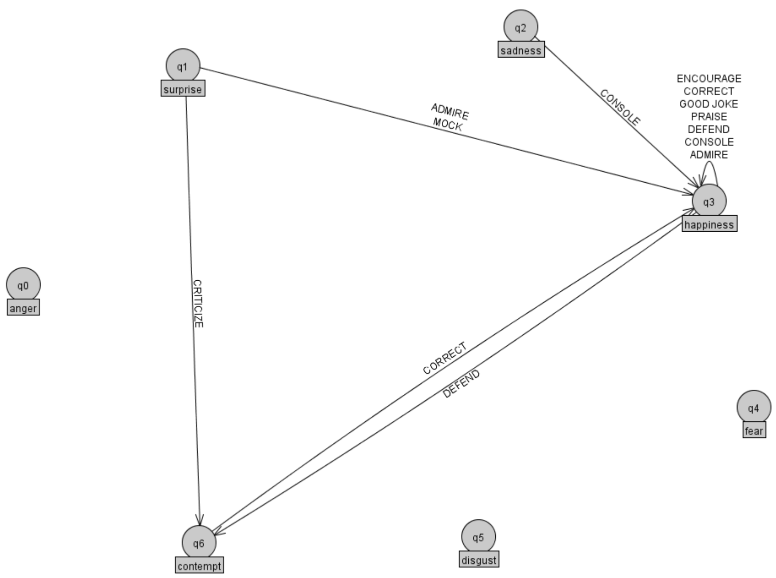


Figure C.7: Emotional reactions for Sharon acting as the actuator

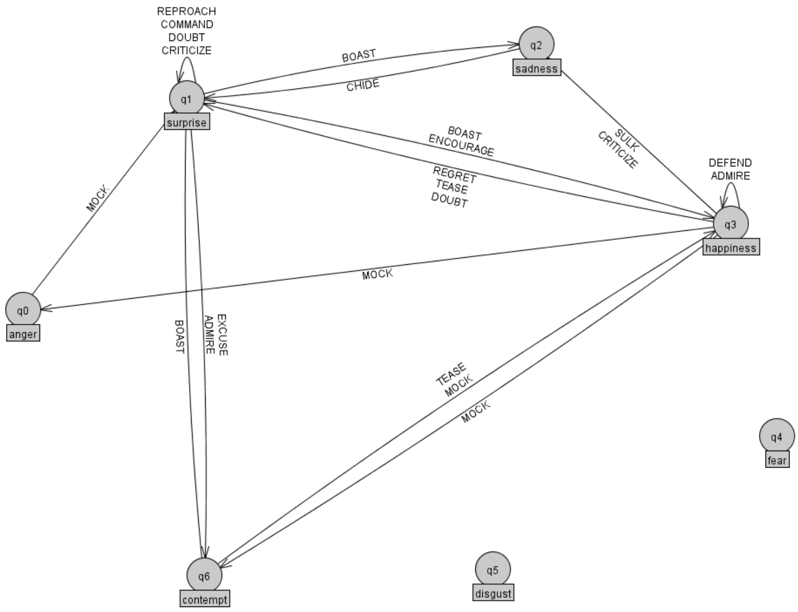


Figure C.8: Emotional reactions for Sharon acting as the effector

Index